



Force.com-Workbook: Salesforce Spring '10

# Force.com-Workbook



Zuletzt aktualisiert: 5. April 2010



# Inhaltsverzeichnis

Info zum Force.com-Workbook.....	3
Lernprogramm 1: Erstellen einer Warenlageranwendung.....	6
Schritt 1: Erstellen eines benutzerdefinierten Merchandise-Objekts.....	6
Schritt 2: Hinzufügen der Felder Beschreibung, Preis und Gesamtbestand.....	8
Schritt 3: Erstellen einer Registerkarte.....	10
Schritt 4: Erstellen einer Warenlageranwendung.....	11
Schritt 5: Erstellen eines Merchandise-Datensatzes.....	13
Zusammenfassung.....	13
Lernprogramm 2: Hinzufügen von Beziehungen.....	15
Schritt 1: Erstellen eines benutzerdefinierten Rechnungsanweisungsobjekts.....	15
Schritt 2: Erstellen eines Belegpostenobjekts.....	18
Schritt 3: Verknüpfen der Objekte.....	19
Schritt 4: Hinzufügen von Rechnungsanweisungen zur Anwendung.....	20
Schritt 5: Erstellen eines Rechnungsdatensatzes.....	20
Zusammenfassung.....	22
Lernprogramm 3: Verwenden von Formeln und Validierungsregeln.....	23
Schritt 1: Berechnen eines Werts für jeden Belegposten.....	24
Schritt 2: Berechnen der Werte für die Rechnungsanweisung mit einem Rollup-Zusammenfassungsfeld.....	25
Schritt 3: Prüfen des Bestands mit einer Validierungsregel.....	26
Schritt 4: Testen der Validierungsregel.....	27
Schritt 5: Optimieren der Validierungsregel.....	28
Zusammenfassung.....	29
Lernprogramm 4: Verwenden von Workflows und Genehmigungen.....	30
Schritt 1: Auffüllen des Belegpostenpreises mit einer Workflowregel.....	31
Schritt 2: Aktualisieren des Gesamtbestands bei der Erteilung von Aufträgen.....	33
Schritt 3: Testen der Workflowregeln.....	34
Schritt 4: Erstellen einer E-Mail-Vorlage.....	35
Schritt 5: Erstellen eines Genehmigungsprozesses.....	36
Schritt 6: Erstellen eines benutzerdefinierten Sicherheitsprofils.....	38
Schritt 7: Erstellen eines Benutzers.....	39
Schritt 8: Testen des Genehmigungsprozesses.....	40
Zusammenfassung.....	42
Lernprogramm 5: Hinzufügen von Geschäftslogik mit Apex.....	43
Schritt 1: Erstellen einer Apex-Auslöserdefinition.....	44
Schritt 2: Definieren einer Listenvariablen.....	45
Schritt 3: Durchgehen der Liste und Ändern des Preises.....	46
Schritt 4: Testen des Auslösers.....	48
Zusammenfassung.....	48
Lernprogramm 6: Hinzufügen von Tests zur Anwendung.....	50
Schritt 1: Erstellen einer Apex-Testklasse.....	51
Schritt 2: Hinzufügen von Testmethoden zur Klasse.....	51

Schritt 3: Schreiben von Code zum Ausführen des Auslösers.....	52
Schritt 4: Ausführen des Tests.....	53
Schritt 5: Anzeigen der Codeabdeckung und Verbessern der Tests.....	54
Zusammenfassung.....	56
Lernprogramm 7: Erstellen einer benutzerdefinierten Benutzeroberfläche mit Visualforce.....	57
Schritt 1: Aktivieren des Visualforce-Entwicklungsmodus.....	57
Schritt 2: Erstellen einer Visualforce-Seite.....	58
Schritt 3: Hinzufügen einer statischen Stylesheet-Ressource.....	59
Schritt 4: Hinzufügen eines Steuerelements zur Seite.....	61
Schritt 5: Anzeigen der Bestandszählerseite als Visualforce-Seite.....	62
Zusammenfassung.....	64
Lernprogramm 8: Erstellen einer öffentlichen Webseite mit Sites.....	65
Schritt 1: Erstellen einer Produktkatalogseite.....	65
Schritt 2: Registrieren eines Force.com-Domännennamens.....	67
Schritt 3: Erstellen einer Force.com-Site.....	68
Schritt 4: Konfigurieren und Testen der Site.....	69
Zusammenfassung.....	70
Lernprogramm 9: Erstellen einer Ladenzeile.....	71
Schritt 1: Erstellen eines Steuerelements.....	71
Schritt 2: Hinzufügen von Methoden zum Steuerelement.....	72
Schritt 3: Erstellen der Ladenzeile.....	73
Schritt 4: Zusätzlicher Schritt – Aktualisieren der Seite mit AJAX.....	75
Zusammenfassung.....	76
Lernprogramm 10: Erstellen einer Desktopanwendung mit Adobe Flash Builder für Force.com.....	78
Schritt 1: Konfigurieren Ihrer persönlichen Salesforce.com-Daten.....	79
Schritt 2: Erstellen einer Offline-Aktenkofferkonfiguration.....	80
Schritt 3: Erstellen Ihrer Unternehmens-WSDL und Herunterladen eines Client-Zertifikats.....	82
Schritt 4: Installieren und Starten von Adobe Flash Builder für Force.com.....	83
Schritt 5: Erstellen eines Force.com Stratus-Projekts.....	83
Schritt 6: Festlegen der Attribute für die Stratus-Anwendungskomponente.....	85
Schritt 7: Erstellen der Inventory Tracker-Fensterkomponente.....	86
Schritt 8: Testen der Inventory Tracker-Anwendung.....	90
Zusammenfassung.....	92
<b>Anhang.....</b>	<b>93</b>
Erstellen einer Sandbox und Anmelden bei dieser.....	93
Erstellen eines Projekts in der Force.com-IDE.....	93
Bereitstellen Ihrer Anwendung.....	94

# Info zum Force.com-Workbook

Im Force.com-Workbook erfahren Sie in einer Reihe von Lernprogrammen, wie man eine Anwendung erstellt. Die Force.com-Plattform kann dazu eingesetzt werden, nahezu alle Anwendungen zu erstellen, wovon die meisten bestimmte Eigenschaften wie die folgenden aufweisen:

- Eine Datenbank zum Modellieren der Informationen in der Anwendung
- Geschäftslogik und Workflow zum Ausführen bestimmter Aufgaben bei bestimmten Bedingungen
- Eine Benutzeroberfläche zum Anzeigen von Daten und Funktionalität für Benutzer, die sich bei der Anwendung angemeldet haben
- Eine öffentliche Website zum Anzeigen von Daten und Funktionalität im Internet



Die Lernprogramme beziehen sich auf das Erstellen eines einfachen Systems zum Verwalten eines Warenlagers. Die Anwendung wird von der Basis weg entwickelt, Sie erstellen also zuerst ein Datenbankmodell zum Nachverfolgen von Merchandise-Artikeln. Dann fahren Sie mit dem Hinzufügen von Geschäftslogik fort: Validierungsregeln zur Sicherung eines ausreichenden Bestands, Workflow zum Aktualisieren des Bestands bei Verkauf, Genehmigungen zum Senden von E-Mail-Benachrichtigungen bei großen Rechnungswerten und Auslöserlogik zum Aktualisieren der Preise in offenen Rechnungen. Nach dem Fertigstellen der Datenbank und Geschäftslogik erstellen Sie eine Benutzeroberfläche, um den Mitarbeitern den Produktbestand anzuzeigen, und eine öffentliche Website, um einen Produktkatalog anzuzeigen. Danach beginnen Sie mit der Entwicklung einer einfachen Ladenzeile. Wenn Sie offline entwickeln und Integrationen zur Anwendung hinzufügen möchten, haben wir ein abschließendes Lernprogramm zur Verwendung von Adobe Flash Builder für Force.com aufgenommen.

Jedes Lernprogramm fußt auf dem vorhergehenden, um die Entwicklung der Anwendung voranzutreiben und gleichzeitig eine bestimmte Funktion der Plattform herauszustreichen. Der Umfang klingt überwältigend, aber wie Sie in Kürze sehen werden, lässt sich dieser mühelos meistern.

## Auswählen einer Entwicklungsumgebung

Dieses Workbook ist darauf ausgelegt, mit einer Organisation in einer Developer Edition durchgeführt zu werden, in der alle erforderlichen Funktionen vorhanden sind. Um sich für eine kostenlose Developer Edition-Organisation zu registrieren, navigieren Sie zu [developer.force.com/join](http://developer.force.com/join).

Unter Berücksichtigung folgender Punkte können Sie auch die Free Edition verwenden.

- In manchen Lernprogrammen ist die Verwendung einer Sandbox erforderlich. Insofern ist es am einfachsten, wenn Sie alle Lernprogramme in Ihrer Sandbox-Organisation abschließen. Falls Sie noch keine Sandbox erstellt haben, finden Sie unter [Erstellen einer Sandbox und Anmelden bei dieser](#) auf Seite 93 weitere Informationen.
- Die Free Edition kann nicht dazu verwendet werden, Lernprogramm 10: Erstellen einer Desktopanwendung mit Adobe Flash Builder für Force.com abzuschließen. Adobe Flash Builder für Force.com ist nur in der Developer Edition als Entwicklervorschau verfügbar.
- Die Bilder im Workbook können etwas von der Anzeige auf Ihrem Bildschirm abweichen. So wird in der Developer Edition die Registerkarte "Hier starten" verwendet, in der Free Edition hingegen die Registerkarte "Erste Schritte". In den meisten Fällen kommt diesen Unterschieden keine weitere Bedeutung zu.

- Nach dem Erstellen der Anwendung können Sie die Änderungen von der Sandbox in eine Produktionsumgebung migrieren. Weitere Informationen finden Sie unter [Bereitstellen Ihrer Anwendung](#) auf Seite 94.

Um sich für eine Free Edition-Organisation zu registrieren, navigieren Sie zu [salesforce.com/form/signup/freeforce-platform.jsp](https://salesforce.com/form/signup/freeforce-platform.jsp).

## Aufbau des Workbooks

Die Lernprogramme in diesem Workbook bauen aufeinander auf und sind so konzipiert, dass sie in der vorgegebenen Reihenfolge durchgeführt werden.

Lernprogramm	Niveau	Geschätzte Bearbeitungszeit
<b>Erstellen der Anwendung</b>		
<a href="#">Lernprogramm 1: Erstellen einer Warenlageranwendung</a>	Anfänger	20-30 Minuten
<a href="#">Lernprogramm 2: Hinzufügen von Beziehungen</a>	Anfänger	20-30 Minuten
<b>Hinzufügen von Logik</b>		
<a href="#">Lernprogramm 3: Verwenden von Formeln und Validierungsregeln</a>	Anfänger	20-30 Minuten
<a href="#">Lernprogramm 4: Verwenden von Workflows und Genehmigungen</a>	Fortgeschrittene Anfänger	30-45 Minuten
<a href="#">Lernprogramm 5: Hinzufügen von Geschäftslogik mit Apex</a>	Fortgeschritten	20-30 Minuten
<b>Schreiben von Tests</b>		
<a href="#">Lernprogramm 6: Hinzufügen von Tests zur Anwendung</a>	Fortgeschritten	20-30 Minuten
<b>Erweitern der Benutzeroberfläche</b>		
<a href="#">Lernprogramm 7: Erstellen einer benutzerdefinierten Benutzeroberfläche mit Visualforce</a>	Fortgeschrittene Anfänger	30-45 Minuten
<a href="#">Lernprogramm 8: Erstellen einer öffentlichen Webseite mit Sites</a>	Fortgeschrittene Anfänger	20-30 Minuten
<a href="#">Lernprogramm 9: Erstellen einer Ladenzeile</a>	Fortgeschrittene Anfänger	30 Minuten
<b>Offline-Integration</b>		
<a href="#">Lernprogramm 10: Erstellen einer Desktopanwendung mit Adobe Flash Builder für Force.com</a>	Fortgeschritten	30 Minuten
<b>Anhang</b>		
<a href="#">Erstellen einer Sandbox und Anmelden bei dieser</a>		
<a href="#">Erstellen eines Projekts in der Force.com-IDE</a>		
<a href="#">Bereitstellen Ihrer Anwendung</a>		

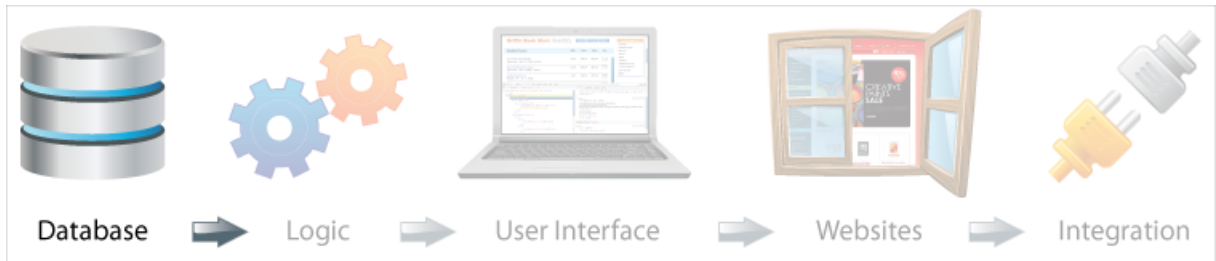
## Weitere Informationen

Am Ende eines jeden Schritts findet sich ein optionaler Abschnitt mit weiteren Informationen. Wenn Sie die Schritte rasch durcharbeiten möchten, fahren Sie direkt mit dem nächsten Schritt fort. Wenn Sie sich allerdings ein wenig mehr Zeit nehmen möchten, finden Sie dort viele hilfreiche Informationen und zusätzliche Schritte zum weiteren Ausarbeiten Ihrer Anwendung.

- Weitere Informationen zu Force.com und Zugriff auf viele Ressourcen erhalten Sie, wenn Sie Developer Force unter [developer.force.com](https://developer.force.com) besuchen.
- Die aktuelle Version dieses Workbooks und weitere Lernprogramme zum Erweitern der Anwendung finden Sie unter [developer.force.com/workbook](https://developer.force.com/workbook).

# Lernprogramm 1: Erstellen einer Warenlageranwendung

**Niveau:** Anfänger; **Dauer:** 20–30 Minuten



Diese Anwendung beruht vollständig darauf, was Sie verkaufen möchten: Merchandise. Sie beginnen also damit, ein Datenobjekt zu erstellen, das alle Elemente eines bestimmten Merchandise-Artikels wie Name, Beschreibung, Preis usw. nachverfolgt. Bei der Force.com-Plattform werden diese Datenobjekte *benutzerdefinierte Objekte* genannt. Wenn Sie mit Datenbanken vertraut sind, können Sie sich ein benutzerdefiniertes Objekt wie eine Tabelle vorstellen.

Ein Objekt weist Standardfelder und Fenster auf, mit denen Informationen zum Objekt aufgeführt, angezeigt und bearbeitet werden können. Sie können jedoch auch Ihre eigenen Felder hinzufügen, um alle nur erdenklichen Elemente nachzuverfolgen oder aufzuführen. Nach diesem Lernprogramm verfügen Sie über eine funktionstüchtige Anwendung mit eigenem Menü, einer Registerkarte und einem benutzerdefinierten Objekt, mit denen der Name, die Beschreibung und der Preis aller Merchandise-Artikel nachverfolgt werden können, sowie mit Fenstern, mit denen Sie alle diese Informationen anzeigen und bearbeiten können.

## Siehe auch:

[Schritt 1: Erstellen eines benutzerdefinierten Merchandise-Objekts](#)

[Schritt 2: Hinzufügen der Felder Beschreibung, Preis und Gesamtbestand](#)

[Schritt 3: Erstellen einer Registerkarte](#)

[Schritt 4: Erstellen einer Warenlageranwendung](#)

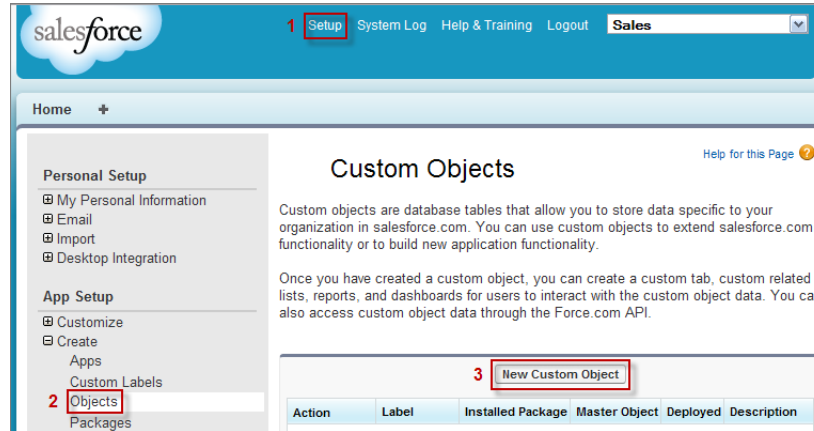
[Schritt 5: Erstellen eines Merchandise-Datensatzes](#)

[Zusammenfassung](#)

## Schritt 1: Erstellen eines benutzerdefinierten Merchandise-Objekts

Die Force.com-Plattform umfasst vordefinierte Standardobjekte. Sie können aber auch benutzerdefinierte Objekte erstellen, um jedes beliebige Element nachzuverfolgen. Als Erstes müssen Sie ein benutzerdefiniertes Objekt erstellen, das Informationen zu Merchandise umfasst.

1. Starten Sie Ihren Browser und wechseln Sie zu <https://login.salesforce.com>.
2. Geben Sie Ihren Benutzernamen (in Form einer E-Mail-Adresse) und Ihr Kennwort ein.
3. Klicken Sie in der oberen rechten Ecke auf **Setup** und dann in der Randleiste auf **Erstellen** ► **Objekte**, um die Seite "Benutzerdefinierte Objekte" anzuzeigen.



4. Klicken Sie auf **Neues benutzerdefiniertes Objekt**, um den Assistenten für neue benutzerdefinierte Objekte anzuzeigen.
5. Füllen Sie die Definition des benutzerdefinierten Objekts aus.
  - Geben Sie unter **Label** Merchandise ein.
  - Geben Sie unter **Plural Label** Label Merchandise ein.
  - Belassen Sie die anderen Werte, wie sie sind.

6. Klicken Sie auf **Speichern**, um das Erstellen des neuen Objekts abzuschließen.

### Weitere Informationen

Nach dem Speichern des neuen benutzerdefinierten Objekts wird die Detailseite für das benutzerdefinierte Objekt geöffnet. Betrachten Sie das folgende Bild, um sich mit dem benutzerdefinierten Objekt "Merchandise" vertraut zu machen.

**Custom Object**  
**Merchandise**

1 [Standard Fields \[4\]](#) | [Custom Fields & Relationships \[0\]](#) | [Validation Rules \[0\]](#) | [Standard Buttons and Links \[8\]](#) | [Custom Buttons and Links \[0\]](#) | [Page Layouts \[1\]](#) | [Search Layouts \[4\]](#) | [Record Types \[0\]](#) | [Apex Sharing Reasons \[0\]](#) | [Apex Sharing Recalculation \[0\]](#)

**Custom Object Definition Detail** [Edit](#) [Deletes](#)

Singular Label	Merchandise	Description	
Plural Label	Merchandise	Enable Reports	<input type="checkbox"/>
Object Name	Merchandise	Track Activities	<input type="checkbox"/>
2 API Name	Merchandise__c	Track Field History	<input type="checkbox"/>
		Deployment Status	Deployed
		Help Settings	Standard salesforce.com Help Window
Created By	Admin User, 10/6/2009 9:55 AM	Modified By	Admin User, 10/6/2009 9:55 AM

**Standard Fields** [Standard Fields Help](#)

Action	Field Label	Data Type
	Created By	Lookup(User)
	Last Modified By	Lookup(User)
3 Edit	Merchandise Name	Text(80)
Edit	Owner	Lookup(User,Queue)

**Custom Fields & Relationships** [New](#) [Field Dependencies](#) [Custom Fields & Relationships Help](#)

No custom fields defined

1. Merchandise-Detailseite: Auf dieser Seite werden alle erforderlichen Informationen zum benutzerdefinierten Objekt "Merchandise" angezeigt. Im Anschluss fügen Sie Felder, Beziehungen, Validierungsregeln und andere nützliche Funktionen zu diesem benutzerdefinierten Objekt hinzu.
2. API-Name: Beim Erstellen des Objekts "Merchandise" geben Sie keinen API-Namen an. Vielmehr wird dieser für Sie erstellt. Mit diesem Namen wird programmseitig auf das Objekt verwiesen. Alle benutzerdefinierten Objekte enden mit \_\_c. Auf diese Weise unterscheiden sie sich von Standardobjekten und sind leichter zu identifizieren.
3. Standardfelder: Einige Felder werden automatisch erstellt. Diese Felder werden als Standardfelder bezeichnet. Das Objekt "Merchandise" weist beispielsweise ein Standardfeld für den Inhaber auf. Dies bedeutet, dass automatisch nachverfolgt wird, welcher Benutzer den jeweiligen Datensatz erstellt.
4. Benutzerdefinierte Felder: Bislang haben Sie noch keine benutzerdefinierten Felder erstellt. Dies geschieht im nächsten Schritt. Wie Sie wohl richtig bemerkt haben, weisen benutzerdefinierte Felder Namen auf, die mit \_\_c enden.

### Siehe auch:

[Lernprogramm 1: Erstellen einer Warenlageranwendung](#)

[Schritt 2: Hinzufügen der Felder Beschreibung, Preis und Gesamtbestand](#)

## Schritt 2: Hinzufügen der Felder Beschreibung, Preis und Gesamtbestand

Ein Merchandise-Objekt sollte Felder aufweisen, die zum Nachverfolgen verschiedener Informationen verwendet werden, wie eine Beschreibung, wie viel der Artikel kostet und wie viele Artikel noch im Bestand sind. Sie können benutzerdefinierte Felder hinzufügen, um alle nur erdenklichen Elemente aufzuführen oder nachzuverfolgen.

Führen Sie die folgenden Schritte durch, um benutzerdefinierte Felder für die Beschreibung, den Preis und den Bestand zu erstellen.

1. Erstellen Sie das Feld "Description" (Beschreibung).
  - a. Blättern Sie nach unten zur Themenliste "Benutzerdefinierte Felder und Beziehungen".
  - b. Klicken Sie auf **Neu**, um den Assistenten für neue benutzerdefinierte Felder zu starten.
  - c. Wählen Sie für "Data Type" die Option **Text Area** aus und klicken Sie auf **Weiter**.

The screenshot shows a dialog box titled 'New Custom Field' with a 'Help for this Page' link. On the left, there are radio buttons for 'Picklist (Multi-Select)', 'Text', 'Text Area', 'Text Area (Long)', and 'URL'. The 'Text Area' option is selected and highlighted with a red box. On the right, there are descriptions for each option. At the bottom right, there are 'Next' and 'Cancel' buttons, with 'Next' highlighted by a red box.

d. Füllen Sie die Details des benutzerdefinierten Felds aus.

- Geben Sie für `Field Label` `Description` ein und drücken Sie die Tabulatortaste.
- Das Feld `Field Name` sollte bereits den Feldnamen aufweisen: `Description`.
- Aktivieren Sie das Kontrollkästchen `Required`. Eine Ihrer Geschäftsregeln besagt, dass keine neuen Merchandise-Artikel ohne Beschreibung erstellt werden können. Durch Festlegen des Felds "Description" als erforderliches Feld wird dieser Geschäftsregel Genüge getan.
- Optional können Sie die Felder `Description` und `Help Text` ausfüllen. Das Ausfüllen dieser Felder ist vorteilhaft, um den Rahmen dieses Handbuchs aber nicht zu sprengen, werden diese für den Rest des Workbook außer Acht gelassen.
- Füllen Sie das Feld `Default Value` nicht aus.

The screenshot shows the 'Merchandise New Custom Field' dialog box at 'Step 2 of 4: Enter the details'. The 'Field Label' and 'Field Name' are both set to 'Description'. The 'Description' field contains the text 'This is a description of the merchandise. This is a required field.' The 'Help Text' field contains 'Use up to 255 characters.' The 'Required' checkbox is checked with the label 'Always require a value in this field in order to save a record'. The 'Default Value' field is empty and has a 'Show Formula Editor' link below it. At the bottom, there are 'Previous', 'Next', and 'Cancel' buttons.

- e. Klicken Sie auf **Weiter**, nehmen Sie die Standardwerte an und klicken Sie nochmals auf **Weiter**.
- f. Klicken Sie auf **Speichern & Neu**, um das Feld `Description` zu speichern und zum ersten Schritt des Assistenten zurückzukehren.

2. Erstellen Sie das Feld "Price" (Preis).

- a. Wählen Sie für "Data Type" die Option `Currency` aus und klicken Sie auf **Weiter**.
- b. Füllen Sie die Details des benutzerdefinierten Felds aus:
  - Geben Sie für `Field Label` `Price` ein.
  - Geben Sie für `Length` `16` ein und für "Decimal Places" `2`.
  - Aktivieren Sie das Kontrollkästchen `Required`.
- c. Belassen Sie für die restlichen Felder die Standardwerte und klicken Sie auf **Weiter**.
- d. Nehmen Sie im nächsten Schritt die Standardwerte an und klicken Sie nochmals auf **Weiter**.

- e. Klicken Sie im nächsten Schritt auf **Speichern & Neu**, um das Feld `Price` zu speichern und zum ersten Schritt des Assistenten zurückzukehren.



**Anmerkung:** Diese Anwendung weist nur ein Feld "Preis" auf und wir verwenden dieses sowohl für den Lager- als auch den Verkaufspreis. Falls Sie möchten, können Sie jetzt jedoch ein weiteres Feld für den Lagerpreis erstellen. Befolgen Sie dazu einfach die obigen Schritte und verwenden Sie den Namen "Stock Price" (Lagerpreis).

3. Erstellen Sie das Feld "Total Inventory" (Gesamtbestand).
- Wählen Sie für `Data Type` die Option `Number` aus und klicken Sie auf **Weiter**.
  - Füllen Sie die Details des benutzerdefinierten Felds aus:
    - Geben Sie für `Field Label` `Total Inventory` ein.
    - Aktivieren Sie das Kontrollkästchen `Required`.
  - Belassen Sie für die restlichen Felder die Standardwerte und klicken Sie auf **Weiter**.
  - Nehmen Sie im nächsten Schritt die Standardwerte an und klicken Sie nochmals auf **Weiter**.
  - Klicken Sie auf **Speichern**, um das Feld `Inventory` zu erstellen und zur Seite des benutzerdefinierten Merchandise-Objekts zurückzukehren.

Action	Field Label	Data Type	Controlling Field	Modified By
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">Description</a>	Text Area(255)		Admin User: 9/22/2009 11:48 AM
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">Price</a>	Currency(12, 2)		Admin User: 9/22/2009 11:51 AM
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">Total Inventory</a>	Number(18, 0)		Admin User: 9/22/2009 11:53 AM

Nun verfügen wir über eine recht gute Darstellung der Artikel im Warenlager: Sie weisen alle einen Namen, eine Beschreibung und einen Preis auf. Des Weiteren wird die gesamte Anzahl an Artikeln ("Total Inventory") aufgezeichnet, die von jedem Merchandise-Artikel vorhanden ist.

### Weitere Informationen

Sie haben sicherlich bemerkt, dass das System bei Eingaben im Feld `Field Label` automatisch auch das Feld `Field Name` ausfüllt. Warum werden dann beide Felder benötigt? Das Feld `Field Label` wird dem Benutzer angezeigt, sodass es gut lesbar sein dürfte und Leerzeichen enthalten kann. Das Feld `Field Name` wird intern im Code verwendet und darf keine Leerzeichen oder nicht zulässige Zeichen enthalten. Das Feld "Customer ph#:" wird im Code z. B. als `Customer_ph` bezeichnet (das System ersetzt Leerzeichen mit einem Unterstrich und entfernt die Zeichen "#" und ":").

### Siehe auch:

- [Lernprogramm 1: Erstellen einer Warenlageranwendung](#)
- [Schritt 1: Erstellen eines benutzerdefinierten Merchandise-Objekts](#)
- [Schritt 3: Erstellen einer Registerkarte](#)

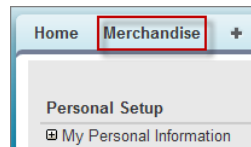
## Schritt 3: Erstellen einer Registerkarte

Registerkarten sind eine einfache Möglichkeit, Objekte und Datensätze zu finden und zu organisieren. In diesem Schritt erstellen Sie eine Registerkarte für das benutzerdefinierte Objekt "Merchandise". Wenn Benutzer auf diese Registerkarte klicken, können sie Datensätze erstellen, anzeigen und bearbeiten.

1. Klicken Sie im Setup-Bereich auf **Erstellen** ► **Registerkarten**.
2. Klicken Sie in der Themenliste "Registerkarten für benutzerdefinierte Objekte" auf **Neu**, um den Assistenten für neue benutzerdefinierte Registerkarten zu starten.
3. Wählen Sie in der Dropdown-Liste **Object** das Element "Merchandise" aus.
4. Klicken Sie bei der Option **Tab Style** auf das Nachschlagesymbol und wählen Sie das **Symbol Box** aus.

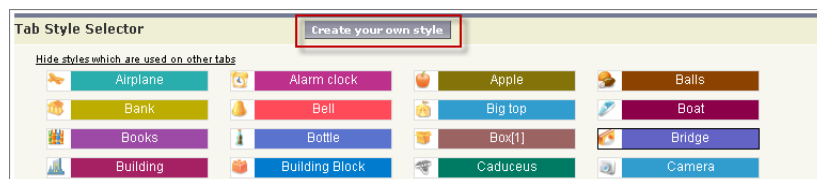
5. Übernehmen Sie die restlichen Standardwerte und klicken Sie auf **Weiter**.
6. Klicken Sie auf **Weiter** und dann auf **Speichern**, um das Erstellen der Registerkarte abzuschließen.

Direkt nach dem Erstellen der Registerkarte wird diese oben im Bildschirm angezeigt.



### Weitere Informationen

Beim Erstellen von Registerkarten müssen Sie nicht zwingend die Standardsymbole und -farben verwenden. Durch Klicken auf **Eigenen Stil erstellen** können Sie Ihre eigene Farbe und ein benutzerdefiniertes Bild auswählen.



### Siehe auch:

[Lernprogramm 1: Erstellen einer Warenlageranwendung](#)

[Schritt 2: Hinzufügen der Felder Beschreibung, Preis und Gesamtbestand](#)

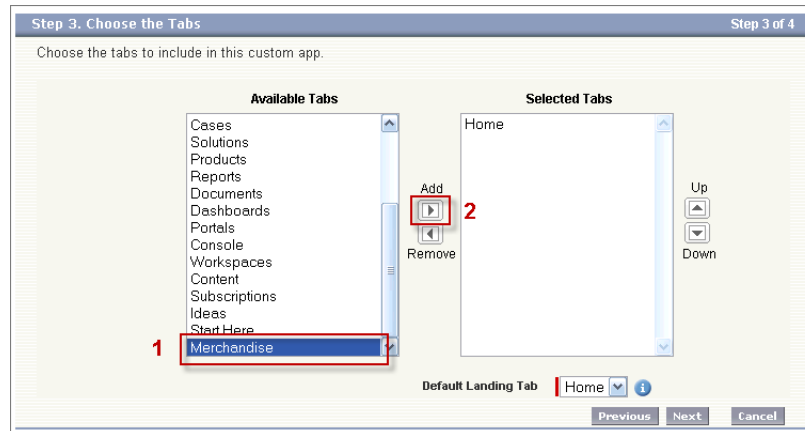
[Schritt 4: Erstellen einer Warenlageranwendung](#)

## Schritt 4: Erstellen einer Warenlageranwendung

Eine Anwendung ist eine Sammlung mit Registerkarten. In diesem Schritt erstellen Sie eine Warenlageranwendung und fügen eine Registerkarte hinzu. Später fügen Sie weitere Registerkarten zur Anwendung hinzu.

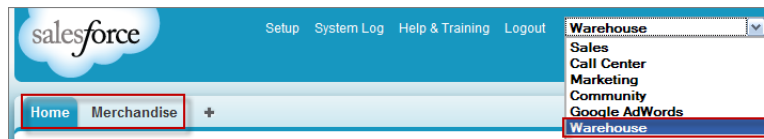
1. Navigieren Sie zu **Setup** ► **Erstellen** ► **Anwendungen**.

2. Klicken Sie auf **Neu**, um den Assistenten für neue benutzerdefinierte Anwendungen zu starten.
3. Füllen Sie die Details der benutzerdefinierten Anwendung aus.
  - Geben Sie unter Label Warehouse ein.
  - Geben Sie unter App Name Warehouse ein.
4. Klicken Sie auf **Weiter**.
5. Übernehmen Sie auf der nächsten Seite die Standardwerte und verwenden Sie dabei das Standardlogo der Anwendung. Klicken Sie auf **Weiter**.
6. Suchen Sie in der Liste "Verfügbare Registerkarten" nach der Registerkarte "Merchandise" und klicken Sie auf **Hinzufügen**, um sie zur Liste der ausgewählten Registerkarten hinzuzufügen.



7. Belassen Sie den Wert unter Default Landing Tab mit "Startseite" und klicken Sie auf **Weiter**.
8. Aktivieren Sie das Kontrollkästchen **Visible**, um die Anwendung in allen Benutzerprofilen anzuzeigen.
9. Klicken Sie auf **Speichern**, um die Warenlageranwendung zu erstellen.

Direkt nach dem Erstellen der Anwendung wird diese im Force.com-Anwendungsmenü oben rechts auf der Seite angezeigt.



### Weitere Informationen

Registerkarten in einer Anwendung müssen keinen Bezug zueinander haben. In der Tat können Sie benutzerdefinierte Anwendungen so bearbeiten, dass alle häufig verwendeten Registerkarten an einem Ort zusammengefasst sind. Wenn etwa die Registerkarte "Hier starten" häufig verwendet wird, können Sie diese zur Warenlageranwendung hinzufügen. Durch Auswählen über das Menü können Sie zwischen erstellten, gekauften oder installierten Anwendungen hin- und herwechseln.

### Siehe auch:

- [Lernprogramm 1: Erstellen einer Warenlageranwendung](#)
- [Schritt 3: Erstellen einer Registerkarte](#)
- [Schritt 5: Erstellen eines Merchandise-Datensatzes](#)

## Schritt 5: Erstellen eines Merchandise-Datensatzes

Zu diesem Zeitpunkt haben wir bereits eine funktionsfähige Anwendung erstellt. Beim Definieren von Objekten auf Force.com erstellt die Plattform automatisch eine Benutzeroberfläche, mit der Sie Datensätze erstellen, anzeigen, aktualisieren und löschen können. Nun prüfen wir, wie die neue Anwendung funktioniert.

1. Wählen Sie im Force.com-Anwendungsmenü die Warenlageranwendung aus.
2. Klicken Sie auf die Registerkarte "Merchandise" und dann auf **Neu**, um ein neues Produkt zu erstellen.
3. Füllen Sie alle Felder aus.
  - Geben Sie im Feld `Merchandise Name` `Wee Jet` ein.
  - Geben Sie im Feld `Description` `A small plane` ein.
  - Geben Sie im Feld `Price` `9.99` ein.
  - Geben Sie im Feld `Total Inventory` `2000` ein.

Siehe folgende Abbildung

4. Klicken Sie auf **Speichern**.

### Weitere Informationen

Die Warenlageranwendung ist auch mit nur einem Produkt voll funktionsfähig. Realistischer wäre es jedoch, wenn sie mehrere Merchandise-Artikel aufweist. Fahren Sie also fort und erstellen Sie weitere Artikel. Tipp: Verwenden Sie die Option **Speichern und neu**, um rasch neue Datensätze zu erstellen.

### Siehe auch:

- [Lernprogramm 1: Erstellen einer Warenlageranwendung](#)
- [Schritt 4: Erstellen einer Warenlageranwendung](#)
- [Zusammenfassung](#)

## Zusammenfassung

Herzlichen Glückwunsch! Sie haben soeben eine Warenlageranwendung erstellt, die die Produkte in Ihrem Bestand nachverfolgt. Wie Sie erfahren haben, können mit der Onlineoberfläche und den Assistenten mühelos ein benutzerdefiniertes Objekt, Felder und eine Registerkarte erstellt werden. Im Anschluss daran können diese in einer Anwendung zusammengefasst werden, mit der Sie Informationen zu Ihren Merchandise-Artikeln erstellen und nachverfolgen können. Die Anwendung ist zwar noch

lange nicht vollständig, bietet jedoch bereits viele integrierte Funktionen, wie das Anzeigen und Erstellen von Merchandise-Artikeln.

Im nächsten Schritt wird eine Rechnung erstellt, um die Warenlagerbewegungen nachzuverfolgen.

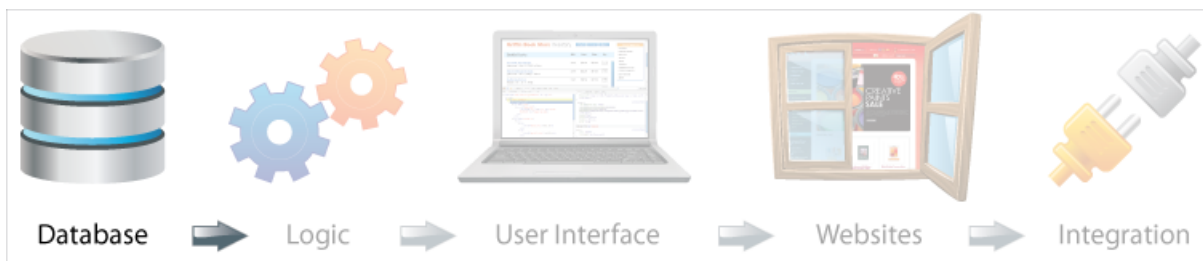
**Siehe auch:**

[\*Lernprogramm 1: Erstellen einer Warenlageranwendung\*](#)

[\*Schritt 5: Erstellen eines Merchandise-Datensatzes\*](#)

# Lernprogramm 2: Hinzufügen von Beziehungen

**Niveau:** Anfänger; **Dauer:** 20–30 Minuten



In diesem Lernprogramm werden zwei neue Objekte erstellt: eine Rechnungsanweisung und ein Belegposten. Diese werden dann in einer Master-Detail-Beziehung zueinander in Beziehung gesetzt. Mit dieser Beziehung können Sie verschiedene Belegposten erstellen und sie einer einzelnen Rechnungsanweisung zuweisen. Jeder Belegposten weist darauf hin, wie viele Einheiten eines bestimmten Merchandise-Artikels verkauft wurden. Daher fügen Sie auch eine Beziehung zwischen Merchandise- und Belegpostenobjekten hinzu.

Wenn Sie mit relationalen Datenbanken vertraut sind, können Sie sich Beziehungen als Fremdschlüssel vorstellen. Der Unterschied liegt darin, dass die zugrunde liegende Implementierung von Force.com übernommen wird. Sie müssen lediglich einrichten, wie die Objekte miteinander in Beziehung stehen.

## Voraussetzungen

### Warenlageranwendung

Zunächst müssen Sie die einfache Warenlageranwendung wie in [Lernprogramm 1: Erstellen einer Warenlageranwendung](#) auf Seite 6 beschrieben erstellen.

## Siehe auch:

[Schritt 1: Erstellen eines benutzerdefinierten Rechnungsanweisungsobjekts](#)

[Schritt 2: Erstellen eines Belegpostenobjekts](#)

[Schritt 3: Verknüpfen der Objekte](#)

[Schritt 4: Hinzufügen von Rechnungsanweisungen zur Anwendung](#)

[Schritt 5: Erstellen eines Rechnungsdatensatzes](#)

[Zusammenfassung](#)

## Schritt 1: Erstellen eines benutzerdefinierten Rechnungsanweisungsobjekts

Zum Verschieben von Artikeln in oder aus dem Warenlager ist eine Rechnungsanweisung erforderlich. In diesem Schritt erstellen Sie eine Rechnungsanweisung mit eindeutiger Nummer, einem Status und einer Beschreibung.

1. Klicken Sie auf **Setup** ► **Erstellen** ► **Objekte**.
2. Klicken Sie auf **Neues benutzerdefiniertes Objekt** und füllen Sie die benutzerdefinierte Objektdefinition aus.
  - Geben Sie im Feld `Label` den Wert `Invoice Statement` ein.

- Geben Sie im Feld **Plural Label** den Wert `Invoice Statements` ein.
- Geben Sie im Feld **Record Name** den Wert `Invoice Number` ein.
- Wählen Sie im Feld **Data Type** den Wert `Auto Number` aus.
- Geben Sie im Feld **Display Format** den Wert `INV-{0000}` ein.
- Geben Sie im Feld **Starting Number** den Wert `1` ein.

Custom Object Information ! = Required Information

The singular and plural labels are used in tabs, page layouts, and reports.

**Label**  **Example:** Account

**Plural Label**  **Example:** Accounts

The Object Name is used when referencing the object via the API.

**Object Name**  **Example:** Account

**Description**

**Context-Sensitive Help Setting**

Open the standard Salesforce.com Help & Training window

Open a window using a custom s-control

Open a window using a Visualforce page

**Content Name**

---

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

**Record Name**  **Example:** Account Name

**Data Type**

**Display Format**  **Example:** A-{0000} [What Is This?](#)

**Starting Number**

3. Belassen Sie die anderen Werte, wie sie sind, und klicken Sie auf **Speichern**.
4. Fügen Sie ein Statusfeld hinzu.
  - a. Blättern Sie nach unten zur Themenliste "Benutzerdefinierte Felder und Beziehungen" und klicken Sie auf **Neu**.
  - b. Wählen Sie für "Data Type" die Option `Picklist` aus und klicken Sie auf **Weiter**.
  - c. Füllen Sie die Details des benutzerdefinierten Felds aus.
    - Geben Sie im Feld **Field Label** den Wert `Status` ein.
    - Geben Sie die folgenden Auswahllistenwerte im bereitgestellten Feld ein, wobei jeder Wert für sich in einer Zeile steht.

Open Closed Negotiating Pending

- Aktivieren Sie das Kontrollkästchen `Use first value as default value`.

Step 2. Enter the details Step 2 of 4

Previous Next Cancel

Field Label: Status

Please enter the list of values for the picklist field below. Each value should be separated by a new line.

Open  
Closed  
Negotiating  
Pending

Sort values alphabetically, not in the order entered. Values will be displayed alphabetically everywhere.

Use first value as default value

Field Name: Status

Description:

Help Text:

Previous Next Cancel

- d. Klicken Sie auf **Weiter**.
  - e. Wählen Sie für die Feldebene Sicherheit die Option **Schreibschutz** aus und klicken Sie dann auf **Weiter**.
  - f. Klicken Sie auf **Speichern und neu**.
5. Erstellen Sie nun ein optionales Beschreibungsfeld.
- a. Wählen Sie im Feld "Data Type" die Option `Text Area` aus und klicken Sie auf **Weiter**.
  - b. Geben Sie in den Feldern `Field Label` und `Field Name` den Wert `Description` ein.
  - c. Klicken Sie auf **Weiter**, nehmen Sie die Standardwerte an und klicken Sie nochmals auf **Weiter**.
  - d. Klicken Sie auf **Speichern**, um die Detailseite für das Objekt "Rechnungsanweisung" anzuzeigen.

Das Objekt "Rechnungsanweisung" sollte nun wie in der folgenden Abbildung zwei benutzerdefinierte Felder aufweisen.

Custom Fields & Relationships		New	Field Dependencies	Custom Fields & Relationships Help	
Action	Field Label	Data Type	Controlling Field	Modified By	
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">Description</a>	Text Area(255)		Admin User, 10/6/2009 9:37 PM	
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Replace</a>	<a href="#">Status</a>	Picklist		Admin User, 10/6/2009 9:34 PM	

### Weitere Informationen

Beachten Sie, dass der Datentyp des Rechnungsnamens mit "Auto Number" festgelegt und des Weiteren ein Anzeigeformat angegeben wurde. Die Plattform weist nun jedem neuen eindeutigen Datensatz, der erstellt wird, automatisch eine Nummer zu. Dabei wird mit der Startnummer begonnen, die Sie angeben. Die Anzeige entspricht dem folgenden Format: INV-002.

### Siehe auch:

[Lernprogramm 2: Hinzufügen von Beziehungen](#)  
[Schritt 2: Erstellen eines Belegpostenobjekts](#)

## Schritt 2: Erstellen eines Belegpostenobjekts

Jede Rechnung besteht aus mehreren Belegpostenobjekten, die die Menge der Merchandise-Artikel darstellen, die zu einem bestimmten Preis verkauft werden. Zunächst erstellen Sie das Belegpostenobjekt, danach verknüpfen Sie es mit den Objekten "Rechnungsanweisung" und "Merchandise".

1. Klicken Sie auf **Setup** ► **Erstellen** ► **Objekte**.
2. Klicken Sie auf **Neues benutzerdefiniertes Objekt** und füllen Sie die benutzerdefinierte Objektdefinition aus.
  - Geben Sie im Feld `Label` den Wert `Line Item` ein.
  - Geben Sie im Feld `Plural Label` den Wert `Line Items` ein.
  - Ändern Sie das Feld `Record Name` in `Line Item Number`.
  - Belassen Sie im Feld `Data Type` den Wert `Text`.

3. Belassen Sie die anderen Werte, wie sie sind, und klicken Sie auf **Speichern**.
4. Fügen Sie ein schreibgeschütztes Stückpreisfeld hinzu. Das Feld ist schreibgeschützt, da der Wert im weiteren Verlauf des Lernprogramms aus dem Merchandise-Objekt abgerufen wird. Das Feld trägt die Bezeichnung "Unit Price" (Stückpreis), damit es nicht mit dem Preisfeld des Merchandise-Objekts verwechselt wird.
  - a. Blättern Sie nach unten zur Themenliste "Benutzerdefinierte Felder und Beziehungen" und klicken Sie auf **Neu**.
  - b. Wählen Sie im Feld `Data Type` die Option `Currency` aus und klicken Sie auf **Weiter**.
  - c. Füllen Sie die Details des benutzerdefinierten Felds aus.
    - Geben Sie im Feld `Field Label` den Wert `Unit Price` ein.
    - Geben Sie im Feld `Length` den Wert `16` und für `Decimal Places` den Wert `2` ein.
  - d. Klicken Sie auf **Weiter**.
  - e. Wählen Sie für alle Profile die Option **Schreibschutz** aus, klicken Sie auf **Weiter** und dann auf **Speichern und neu**.



**Anmerkung:** Falls die Option **Schreibschutz** nicht verfügbar ist, haben Sie möglicherweise aus Versehen die Option **Erforderlich** aktiviert. Klicken Sie auf **Zurück** und deaktivieren Sie die Option **Erforderlich**.

5. Zum Hinzufügen eines Felds der verkauften Einheiten führen Sie ähnliche Schritte aus.
  - a. Wählen Sie im Feld `Data Type` die Option `Number` aus und klicken Sie auf **Weiter**.
  - b. Geben Sie im Feld `Field Label` den Wert `Units Sold` ein und klicken Sie auf **Weiter**.
  - c. Akzeptieren Sie die Standardwerte durch Klicken auf **Weiter**.
  - d. Klicken Sie auf **Speichern**, um zur Detailseite des benutzerdefinierten Belegpostenobjekts zurückzukehren.

### Weitere Informationen

Zu diesem Zeitpunkt liegen bereits drei benutzerdefinierte Objekte vor: Merchandise, Rechnungsanweisung und Belegposten. Für jedes dieser Objekte haben Sie benutzerdefinierte Felder für Text, Zahlen und Währung erstellt. Alle diese Felder weisen eine Gemeinsamkeit auf: die Werte werden vom Benutzer bereitgestellt. Darüber hinaus haben Sie zwei benutzerdefinierte Felder mit vom System generierten Werten erstellt: Die Auswahlliste "Status" mit dem Standardwert "Open" und das Feld "Invoice\_Number", das automatisch vom Datentyp "Auto Number" zugewiesen wird. Im nächsten Schritt erstellen Sie zwei weitere Felder. Im Gegensatz zu den bisherigen Feldern rufen diese ihre Werte aus anderen Objekten ab.

### Siehe auch:

[Lernprogramm 2: Hinzufügen von Beziehungen](#)

[Schritt 1: Erstellen eines benutzerdefinierten Rechnungsanweisungsobjekts](#)

[Schritt 3: Verknüpfen der Objekte](#)

## Schritt 3: Verknüpfen der Objekte

Da nun alle Objekte des Datenmodells vorhanden sind, werden diese in Beziehung miteinander gesetzt. Der Belegposten bezieht sich sowohl auf eine Rechnungsanweisung (Anweisungen bestehen aus mehreren Belegposten) als auch auf Merchandise (Belegposten rufen ihren Preis aus Merchandise ab).

1. Blättern Sie auf der Detailseite des Belegpostenobjekts nach unten bis zur Themenliste "Benutzerdefinierte Felder & Beziehungen" und klicken Sie auf **Neu**.
2. Wählen Sie im Feld `Data Type` die Option **Master-Detail-Beziehung** aus und klicken Sie auf **Weiter**.
3. Wählen Sie im Feld `Related To` das benutzerdefinierte Objekt **Merchandise** aus und klicken Sie auf **Weiter**.
4. Übernehmen Sie in den nächsten drei Fenstern die Standardwerte durch Klicken auf **Weiter**.
5. Deaktivieren Sie das Kontrollkästchen neben `Merchandise Layout`, sodass in der Themenliste "Merchandise" keine Belegposten angezeigt werden.
6. Klicken Sie auf **Speichern und neu**.
7. Wählen Sie im Feld `Data Type` die Option **Master-Detail-Beziehung** aus und klicken Sie auf **Weiter**.
8. Wählen Sie im Feld **Related To** das benutzerdefinierte Objekt **Rechnungsanweisung** aus und klicken Sie auf **Weiter**.
9. Übernehmen Sie in den nächsten Fenstern die Standardwerte durch Klicken auf **Weiter** und klicken Sie dann auf **Speichern**, um zum Detailfenster des Bestandselements zurückzukehren.

### Weitere Informationen

Sie haben soeben zwei Master-Detail-Beziehungen erstellt und sichergestellt, dass die Rechnungsanweisungsdatensätze mit den Datensätzen der Rechnungsbelegposten sowie die Rechnungsbelegposten mit "Merchandise" verknüpft sind.

Master-Detail-Beziehungen unterstützen darüber hinaus Rollup-Zusammenfassungsfelder, sodass Sie Informationen über die untergeordneten Datensätze zusammenfassen können. Diese Funktion wird später im Lernprogramm verwendet.

### Siehe auch:

[Lernprogramm 2: Hinzufügen von Beziehungen](#)

[Schritt 2: Erstellen eines Belegpostenobjekts](#)

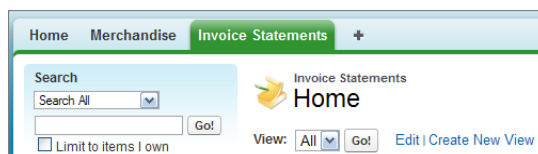
[Schritt 4: Hinzufügen von Rechnungsanweisungen zur Anwendung](#)

## Schritt 4: Hinzufügen von Rechnungsanweisungen zur Anwendung

Wie zuvor beim benutzerdefinierten Merchandise-Objekt soll eine Registerkarte für das Objekt "Rechnungsanweisung" erstellt und zur Warenlageranwendung hinzugefügt werden. Dadurch wird die Benutzeroberfläche zugänglich gemacht, die Force.com automatisch für dieses Objekt erstellt.

1. Klicken Sie im Setup-Bereich auf **Erstellen** ► **Registerkarten**.
2. Klicken Sie in der Themenliste "Registerkarten für benutzerdefinierte Objekte" auf **Neu**, um den Assistenten für neue benutzerdefinierte Registerkarten zu starten.
3. Wählen Sie in der Dropdown-Liste **Object** die Option **Rechnungsanweisung** aus.
4. Klicken Sie für die Option **Tab Style** auf die Nachschlageschaltfläche und wählen Sie das Symbol **Formular** aus.
5. Übernehmen Sie die restlichen Standardwerte und klicken Sie auf **Weiter** und dann nochmals auf **Weiter**.
6. Deaktivieren Sie auf der Seite "Zu benutzerdefinierten Anwendungen hinzufügen" alle Kontrollkästchen außer **Warehouse**. Dadurch wird die Registerkarte "Rechnungsanweisungen" zur Warenlageranwendung hinzugefügt.
7. Klicken Sie auf **Speichern**, um das Erstellen der Registerkarte abzuschließen.

Die Registerkarte "Rechnungsanweisungen" ist nun Teil der Anwendung.



### Siehe auch:

[Lernprogramm 2: Hinzufügen von Beziehungen](#)

[Schritt 3: Verknüpfen der Objekte](#)

[Schritt 5: Erstellen eines Rechnungsdatensatzes](#)

## Schritt 5: Erstellen eines Rechnungsdatensatzes

Wie Sie im Verlauf des Lernprogramms feststellen konnten, generiert die Plattform automatisch eine Benutzeroberfläche für die von Ihnen erstellten Objekte, sodass Sie Datensätze anzeigen, bearbeiten, löschen und aktualisieren können. Da Sie die Objekte darüber hinaus verknüpft haben, bietet die Benutzeroberfläche die Möglichkeit, auch zwischen verknüpften Datensätzen hin- und herzunavigieren.

1. Klicken Sie auf die Registerkarte "Rechnungsanweisungen".
2. Klicken Sie auf **Neu**.

3. Geben Sie im Feld `Description` den Wert `First Invoice` ein und klicken Sie dann auf **Speichern**.

Die Detailseite Ihrer Rechnungsanweisung sollte wie folgt aussehen.

Beachten Sie, dass der Rechnung automatisch eine Nummer zugewiesen wurde und die Benutzeroberfläche darunter die leere Themenliste "Belegposten" aufweist. Die Rechnungsanweisung ist über ein Master-Detail-Feld mit dem Belegposten verknüpft. Die Beziehung wurde auf diese Weise erstellt und daher weist die Detailseite "Rechnungsanweisung" eine Themenliste auf. Als Nächstes wird der Rechnung ein Belegposten hinzugefügt.

1. Klicken Sie auf **Neuer Belegposten**.

2. Füllen Sie die Felder aus.

- Geben Sie im Feld `Line Item Number` den Wert `1` ein.
- Geben Sie im Feld `Unit Price` den Wert `10` ein.
- Geben Sie im Feld `Units Sold` den Wert `4` ein.
- Klicken Sie für das Feld `Merchandise` auf die Nachschlageschaltfläche und wählen Sie ein Produkt aus.

3. Klicken Sie auf **Speichern**.

### Weitere Informationen

Sie fragen sich möglicherweise, warum es sich bei "Line Item Number" um ein Textfeld handelt, wenn in der Tat eine Zahl eingegeben wird. Warum sollte das Feld nicht wie Rechnungsanweisungen automatisch nummeriert werden, wenn Belegposten nummeriert werden? Knapp zusammengefasst lässt sich sagen, dass das Arbeiten mit Text bei Datensätzen einfacher ist, und dieses Lernprogramm soll einfach nachzuvollziehen sein. Wenn Sie möchten, können Sie Belegposten als automatisch

nummerierte Felder erstellen. Dadurch wird jedoch [Lernprogramm 8: Erstellen einer öffentlichen Webseite mit Sites](#) etwas komplexer.

**Siehe auch:**

[Lernprogramm 2: Hinzufügen von Beziehungen](#)

[Schritt 4: Hinzufügen von Rechnungsanweisungen zur Anwendung](#)

[Zusammenfassung](#)

## Zusammenfassung

In diesem Lernprogramm haben Sie Beziehungen zwischen den Objekten in Ihrem Datenmodell erstellt. Die Beziehungen entsprechen Fremdschlüsseln in relationalen Datenbanken auf abstrakteren Ebenen. Sie können sich auf das Wesentliche (die Beziehungen) konzentrieren und müssen sich nicht mit der zugrunde liegenden Implementierung auseinandersetzen. Die Master-Detail-Beziehung ermöglicht das Zusammenfassen von Informationen und im nächsten Lernprogramm erfahren Sie, wie der Preis einzelner Rechnungsbelegposten in der Rechnungsanweisung zusammengefasst werden kann. Beziehungen weisen noch weitere Vorteile auf: Sie können in einer Benutzeroberfläche und einer Abfragesprache zu den verknüpften Datensätzen navigieren. Auch hierzu erhalten Sie später noch Informationen.

Nach dem Erstellen der einfachen Anwendung können Sie mit Formeln und Validierungsregeln wie im [Lernprogramm 3: Verwenden von Formeln und Validierungsregeln](#) auf Seite 23 beschrieben Geschäftslogik hinzufügen.

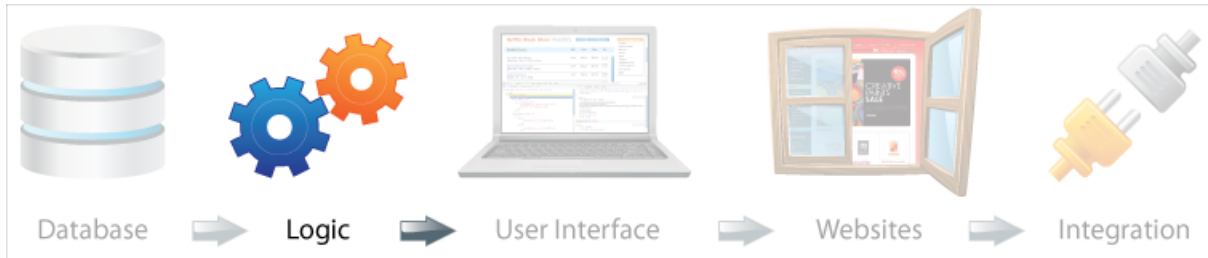
**Siehe auch:**

[Lernprogramm 2: Hinzufügen von Beziehungen](#)

[Schritt 5: Erstellen eines Rechnungsdatensatzes](#)

# Lernprogramm 3: Verwenden von Formeln und Validierungsregeln

Niveau: Anfänger; Dauer: 20–30 Minuten



Mit der Force.com-Plattform können Sie Formeln und Feldvalidierungsregeln erstellen, um die Verwaltung der in der Anwendung eingegebenen Daten zu unterstützen und ihre Qualität zu verbessern. Sowohl bei Formelfeldern als auch Feldvalidierungsregeln werden integrierte Funktionen eingesetzt, die das automatische Bearbeiten von Daten, Validieren von Daten und Berechnen anderer, auf den Daten basierenden Werten ermöglichen. Die in Formelfeldern und Feldvalidierungsregeln verwendeten Funktionen ähneln sehr stark jenen Funktionen, die bei der Tabellenkalkulation genutzt werden, bei der auf Werte in anderen Zellen des Tabellenblatts verwiesen, Berechnungen durchgeführt und Ergebnisse zurückgegeben werden können. Im Gegensatz dazu verweisen Sie mit Formelfeldern und Feldvalidierungsregeln auf Felder in den Datensätzen Ihrer Anwendung.

In diesem Lernprogramm optimieren Sie die Warenlageranwendung durch das Hinzufügen eines Formelfelds zum automatischen Berechnen des Gesamtwerts eines jeden Belegpostens. Darüber hinaus verwenden Sie diesen neuen Belegpostenwert und eine Eigenschaft der Master-Detail-Beziehungen zum automatischen Berechnen eines Rechnungswerts. Abschließend fügen Sie noch eine Formel zum Durchführen einer Bestandsprüfung hinzu, um sicherzustellen, dass nur für Merchandise-Artikel Belegposten erstellt werden können, die tatsächlich auf Lager sind.

## Voraussetzungen

### Warenlageranwendung

Zunächst benötigen Sie eine einfache Warenlageranwendung und müssen Beziehungen hinzufügen. Dieser Vorgang wird unter [Lernprogramm 2: Hinzufügen von Beziehungen](#) auf Seite 15 beschrieben.

## Siehe auch:

[Schritt 1: Berechnen eines Werts für jeden Belegposten](#)

[Schritt 2: Berechnen der Werte für die Rechnungsanweisung mit einem Rollup-Zusammenfassungsfeld](#)

[Schritt 3: Prüfen des Bestands mit einer Validierungsregel](#)

[Schritt 4: Testen der Validierungsregel](#)

[Schritt 5: Optimieren der Validierungsregel](#)

[Zusammenfassung](#)

## Schritt 1: Berechnen eines Werts für jeden Belegposten

Im ersten Schritt dieses Lernprogramms fügen Sie ein neues berechnetes Feld mit der Bezeichnung "Value" (Wert) zum Belegposten hinzu. Mit diesem Feld wird die Anzahl der Elemente mit dem Preis multipliziert. Es fungiert als Gesamtsumme für jeden Belegposten.

1. Navigieren Sie zur Seite des benutzerdefinierten Objekts "Belegposten", indem Sie auf **Setup** ► **Erstellen** ► **Objekte** ► **Belegposten** klicken.
2. Blättern Sie nach unten zur Themenliste "Benutzerdefinierte Felder und Beziehungen" und klicken Sie auf **Neu**.
3. Wählen Sie als Feldtyp **Formula** aus und klicken Sie auf **Weiter**.
4. Geben Sie in den Feldern **Field Label** und **Field Name** den Wert **Value** ein.
5. Wählen Sie im Feld **Formula Return Type** die Option **Währung** aus.
6. Klicken Sie auf **Weiter**.
7. Wählen Sie in der Dropdown-Liste **Insert Merge Field** die Option **Unit\_Price** aus. Nun sollte im Textfeld **Unit\_Price\_\_c** angezeigt werden.
8. Klicken Sie auf die Dropdown-Liste **Operator einfügen** und wählen Sie **Multiplizieren** aus.
9. Wählen Sie in der Dropdown-Liste "Insert Merge Field" die Option **Verkaufte Einheiten** aus. Nun sollte im Textfeld **Unit\_Price\_\_c \* Units\_Sold\_\_c** angezeigt werden.

10. Klicken Sie auf **Weiter**, klicken Sie nochmals auf **Weiter** und klicken Sie dann auf **Speichern**.

Wenn Sie zur Detailseite des benutzerdefinierten Objekts zurückkehren, wird dort das neue Feld mit der Bezeichnung "Value" angezeigt.

Custom Fields & Relationships				
Action	Field Label	Data Type	Controlling Field	Modified By
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">Invoice Statement</a>	Master-Detail(Invoice Statement)		Mario.Korf, 10/22/2009 9:35 PM
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">Merchandise</a>	Master-Detail(Merchandise)		Mario.Korf, 10/22/2009 9:34 PM
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">Unit Price</a>	Currency(16, 2)		Mario.Korf, 10/22/2009 9:30 PM
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">Units Sold</a>	Number(18, 0)		Mario.Korf, 10/22/2009 9:30 PM
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">Value</a>	Formula (Currency)		Mario.Korf, 10/22/2009 9:53 PM

### Weitere Informationen

Wie hier gezeigt, eignet sich der Feldtyp Formel ideal zum automatischen Ableiten von Feldwerten von anderen Werten. Die eingegebene Formel ist verhältnismäßig einfach: eine einfache Multiplikation zweier Feldwerte des gleichen Datensatzes. Es gibt darüber hinaus die Registerkarte "Erweiterte Formel", mit der Sie diese Formeln für weit mehr Aktionen einsetzen können.

### Siehe auch:

[Lernprogramm 3: Verwenden von Formeln und Validierungsregeln](#)

[Schritt 2: Berechnen der Werte für die Rechnungsanweisung mit einem Rollup-Zusammenfassungsfeld](#)

## Schritt 2: Berechnen der Werte für die Rechnungsanweisung mit einem Rollup-Zusammenfassungsfeld

Nachdem nun der Gesamtwert eines jeden Belegpostens vorliegt, ist es sinnvoll, diese alle zusammenzurechnen, um die Gesamtsumme der Rechnung zu erhalten. Da die Belegposten eine Master-Detail-Beziehung mit der Rechnungsanweisung aufweisen, können wir ein Rollup-Zusammenfassungsfeld verwenden, um diesen Wert zu berechnen. Bei einer Rollup-Zusammenfassung handelt es sich um ein spezielles Feld, mit dem Informationen zu Detailobjekten (untergeordneten Objekten) zusammengefasst werden können. In diesem Fall möchten Sie die Summe der Werte aller Belegposten bilden.

1. Navigieren Sie zurück zur Seite des benutzerdefinierten Objekts "Rechnungsanweisung", indem Sie auf **Setup ► Erstellen ► Objekte ► Rechnungsanweisung** klicken.
2. Blättern Sie nach unten zur Themenliste "Benutzerdefinierte Felder und Beziehungen" und klicken Sie auf **Neu**.
3. Wählen Sie als Feldtyp **Roll-Up Summary** aus und klicken Sie auf **Weiter**.
4. Geben Sie im Feld **Field Label** den Wert **Invoice Value** ein und klicken Sie auf **Weiter**.
5. Wählen Sie in der Liste **Summarized Object** die Option **Belegposten** aus.
6. Wählen Sie für das Feld **Roll Up Type** die Option **Summe** aus.
7. Wählen Sie in der Liste **Field to Aggregate** die Option **Wert** aus.
8. Stellen Sie sicher, dass der Bildschirm dem folgenden Beispiel entspricht, und klicken Sie dann auf **Weiter**.

9. Klicken Sie nochmals auf **Weiter** und dann auf **Speichern**.

### Weitere Informationen

Wenn Sie zurück zu einem Rechnungsanweisungsdatensatz navigieren, wird das neue Rollup-Zusammenfassungsfeld mit dem Gesamtwert aller Rechnungsbelegposten angezeigt. Wenn in einem Feld eine Sanduhr angezeigt wird, warten Sie einen kurzen Moment und aktualisieren Sie dann die Anzeige. Sie können diese neue Funktion testen, indem Sie einen neuen Belegposten hinzufügen.

### Siehe auch:

- [Lernprogramm 3: Verwenden von Formeln und Validierungsregeln](#)
- [Schritt 1: Berechnen eines Werts für jeden Belegposten](#)
- [Schritt 3: Prüfen des Bestands mit einer Validierungsregel](#)

## Schritt 3: Prüfen des Bestands mit einer Validierungsregel

Die Felder, die Sie in Objekten definieren, können Validierungsregeln aufweisen, die in der gleichen Formelsprache geschrieben sind, die Sie zum Erstellen des Formelfelds verwendet haben. Die Validierungsregeln können verwendet werden, um zu bestimmen, welcher Eingabebereich gültig ist, und um eine Meldung für den Benutzer anzuzeigen, wenn ein Feldwert nicht gültig ist. Die Fehlerbedingungsformeln sollten bei der Evaluierung "true" zurückgeben, wenn dem Benutzer eine Meldung angezeigt werden soll.

1. Navigieren Sie zurück zur Seite des benutzerdefinierten Objekts "Belegposten", indem Sie auf **Setup ► Erstellen ► Objekte ► Belegposten** klicken.
2. Blättern Sie nach unten zur Themenliste "Validierungsregeln" und klicken Sie auf **Neu**.
3. Geben Sie im Feld `Rule Name` den Wert `Order in stock` ein.
4. Klicken Sie im Bereich "Fehlerbedingungsformel" auf **Feld einfügen**, um das Popup-Fenster "Feld einfügen" zu öffnen.
  - a. Wählen Sie in der ersten Spalte `Line Item >` aus.
  - b. Wählen Sie in der zweiten Spalte `Merchandise >` aus.
  - c. Wählen Sie in der dritten Spalte `Total Inventory` aus.
  - d. Klicken Sie auf **Einfügen**.
  - e. Geben Sie das Kleinerzeichen `<` ein, sodass die Formel wie folgt aussieht:

```
Merchandise__r.Total_Inventory__c <
```

- f. Klicken Sie nochmals auf **Feld einfügen**.
- g. Wählen Sie in der ersten Spalte `Line Item >` aus.
- h. Wählen Sie in der zweiten Spalte `Units Sold` aus.
- i. Klicken Sie auf **Einfügen** und überprüfen Sie, ob der Code wie folgt aussieht.

```
Merchandise__r.Total_Inventory__c < Units_Sold__c
```

5. Klicken Sie auf **Syntax prüfen**, um sicherzustellen, dass keine Fehler vorliegen. Beheben Sie etwaige Fehler, bevor Sie fortfahren.
6. Geben Sie im Feld `Error Message` den Wert `You have ordered more items than we have in stock` ein.
7. Wählen Sie unter `Error Location` den Wert `Field` und in der Dropdown-Liste dann `Units Sold` aus.

8. Klicken Sie auf **Speichern**.

### Weitere Informationen

Sie können im Bereich "Fehlerbedingungsformel" direkt Formeln eingeben. Wie Sie hier jedoch gesehen haben, können Sie die verfügbaren Objekte mühelos verschieben und die für die Formel erforderlichen Komponenten suchen. Analysieren wir nun die erstellte Formel.

- `Merchandise__r`: Da das Merchandise-Objekt eine Beziehung mit dem Belegpostenobjekt aufweist, stellt die Plattform automatisch ein Beziehungsfeld bereit, mit dem Sie von einem Belegpostendatensatz zu einem Merchandise-Datensatz navigieren können. Hierbei handelt es sich um die Funktion des Elements `Merchandise__r`.
- `Total_Inventory__c`: Dieses Feld haben Sie erstellt, um die gesamte Menge des Bestands eines Merchandise-Datensatzes nachzuerfolgen.
- `Merchandise__r.Total_Inventory__c`: Hierdurch wird das System angewiesen, den Wert des Felds "Total Inventory" des entsprechenden Merchandise-Datensatzes abzurufen.
- `Units_Sold__c`: Dies bezieht sich auf das Feld "Units Sold" des aktuellen (Belegposten-)Datensatzes.

Zusammengefasst prüft die Formel, ob der Gesamtbestand des entsprechenden Merchandise-Datensatzes unter der Anzahl der verkauften Einheiten liegt. Wie auf der Seite "Fehlerbedingungsformel" angezeigt wird, müssen Sie eine Formel angeben, die den Wert "true" aufweist, wenn eine Fehlermeldung angezeigt werden soll. Dies entspricht genau den Anforderungen in diesem Fall: Der Wert ist nur "true", wenn der Gesamtbestand unter den verkauften Einheiten liegt.

### Siehe auch:

[Lernprogramm 3: Verwenden von Formeln und Validierungsregeln](#)

[Schritt 2: Berechnen der Werte für die Rechnungsanweisung mit einem Rollup-Zusammenfassungsfeld](#)

[Schritt 4: Testen der Validierungsregel](#)

## Schritt 4: Testen der Validierungsregel

Nun wird die Validierungsregel getestet, die wir in Schritt 3 erstellt haben.

1. Klicken Sie auf die Registerkarte "Rechnungsanweisungen" und wählen Sie eine vorhandene Rechnung aus.
2. Blättern Sie nach unten bis "Belegposten" und klicken Sie auf **Bearbeiten** neben einem der Belegposten.
3. Ändern Sie den Wert `Units Sold` in eine Zahl, die über der Menge im Bestand liegt.
4. Klicken Sie auf **Speichern**. Es sollte eine Fehlermeldung angezeigt werden, die angibt, dass nicht genügend Artikel auf Lager sind. Klicken Sie auf **Abbrechen**.

The screenshot shows a web form titled "Line Item Edit" for item 1. It contains fields for "Line Item Number" (1), "Unit Price" (10.00), "Units Sold" (30.000), "Merchandise" (Wee Jet), and "Invoice Statement" (INV-0001). A red error message is displayed at the top: "Error: Invalid Data. Review all error messages below to correct your data." A more specific error message is shown below the "Units Sold" field: "Error: You have ordered more items than we have in stock." Buttons for "Save", "Save & New", and "Cancel" are present at the top and bottom of the form.

5. Klicken Sie auf **Abbrechen**.

### Siehe auch:

[Lernprogramm 3: Verwenden von Formeln und Validierungsregeln](#)

[Schritt 3: Prüfen des Bestands mit einer Validierungsregel](#)

[Schritt 5: Optimieren der Validierungsregel](#)

## Schritt 5: Optimieren der Validierungsregel

Die Logik in der Validierungsregel weist Ungenauigkeiten auf! Stellen Sie sich vor, ein Datensatz wird bearbeitet und die Anzahl der verkauften Einheiten gesenkt. In diesem Fall muss der Bestand nicht geprüft werden. Oder die Anzahl der verkauften Einheiten wird erhöht. Dann sollte nur geprüft werden, ob die Anzahl der zusätzlichen Einheiten vorrätig ist. Sie können die Validierungsregel so optimieren, dass diese Fälle berücksichtigt werden, indem Sie die `ISNEW()`-Funktion in der Formel verwenden, mit der bestimmt wird, ob ein neuer Datensatz erstellt wird. Falls es sich nicht um einen neuen Datensatz handelt, müssen Sie die `PRIORVALUE()`-Funktion verwenden, damit der vorherige Wert des Feldes vor der Bearbeitung angegeben wird.

1. Navigieren Sie zurück zur Seite des benutzerdefinierten Objekts "Belegposten", indem Sie auf **Setup** ► **Erstellen** ► **Objekte** ► **Belegposten** klicken.
2. Blättern Sie nach unten zur Themenliste "Validierungsregeln" und klicken Sie neben `Order in stock` auf **Bearbeiten**.
3. Ersetzen Sie die vorhandene Formel mit der folgenden.

```
IF( ISNEW(), Merchandise__r.Total_Inventory__c < Units_Sold__c , IF ( Units_Sold__c < PRIORVALUE(Units_Sold__c), FALSE, Merchandise__r.Total_Inventory__c < (Units_Sold__c - PRIORVALUE(Units_Sold__c)) ) )
```

4. Klicken Sie auf **Syntax prüfen**, um sicherzustellen, dass keine Fehler vorliegen.
5. Klicken Sie auf **Speichern**.

## Weitere Informationen

Sehen wir uns diese Formel etwas genauer an.

- `IF` ist eine Bedingung, die zur Laufzeit sicherstellt, dass je nach Bedingung eine der beiden Optionen verwendet wird.
- `ISNEW` ist die Bedingung. Diese Funktion weist den Wert "true" auf, wenn ein neuer Datensatz erstellt wird. Anderenfalls lautet der Wert "false".
- Wenn `ISNEW` den Wert "true" aufweist, dann wird wie zuvor lediglich der Bestand geprüft. Bei neuen Datensätzen muss sichergestellt werden, dass nicht mehr verkauft wird, als Artikel im Bestand vorrätig sind.
- Wenn `ISNEW` den Wert "false" aufweist, ist sichergestellt, dass eine Aktualisierung eines vorhandenen Datensatzes durchgeführt wird und kein neuer Datensatz erstellt wird. Daher wird eine weitere Bedingungsüberprüfung durchgeführt, dieses Mal, um zu ermitteln, ob die Anzahl der Artikel erhöht oder gesenkt wurde, indem der Wert mit der `PRIORVALUE`-Funktion mit dem vorherigen Wert verglichen wird.
- Wenn der vorherige Wert größer war, wurde der Datensatz aktualisiert und die Anzahl der Artikel gesenkt. Da ausreichend Artikel vorrätig sind, wird `FALSE` zurückgegeben, um anzugeben, dass kein Validierungsfehler vorliegt.
- Wenn der vorherige Wert kleiner war, wurde die Anzahl der Einheiten angehoben. In diesem Fall muss also geprüft werden, ob der Bestand ausreicht, um die Differenz zwischen der neuen Anzahl an Artikeln und der alten Anzahl zu decken.

## Siehe auch:

[Lernprogramm 3: Verwenden von Formeln und Validierungsregeln](#)

[Schritt 4: Testen der Validierungsregel](#)

[Zusammenfassung](#)

## Zusammenfassung

In diesem Lernprogramm haben Sie Formelfelder und Validierungsregeln erstellt, die die Daten der Anwendung optimieren und validieren. Bei der ersten Formel haben Sie eine Zwischensumme für jeden Belegposten erstellt, indem der Preis eines Produkts mit der Anzahl der bestellten Einheiten multipliziert wurde. Eine Rechnung konnte aus mehreren Belegposten zusammengestellt werden, sodass Sie ein Rollup-Zusammenfassungsfeld erstellt haben, um die Zwischensummen der Belegposten automatisch zusammenzufassen.

Darüber hinaus haben Sie erfahren, wie eine Validierungsregel definiert wird, um sicherzustellen, dass genügend Artikel vorrätig sind. Die erste Version der Regel war einfach, Sie haben diese jedoch optimiert, damit sie sowohl mit neuen Datensätzen als auch vorhandenen Datensätzen mit Aktualisierungen kompatibel ist. Diese Art Anpassung kann bei allen Feldern in der Anwendung angewendet werden, wenn Sie sicherstellen möchten, dass die eingegebenen Daten bestimmten Kriterien genügen.

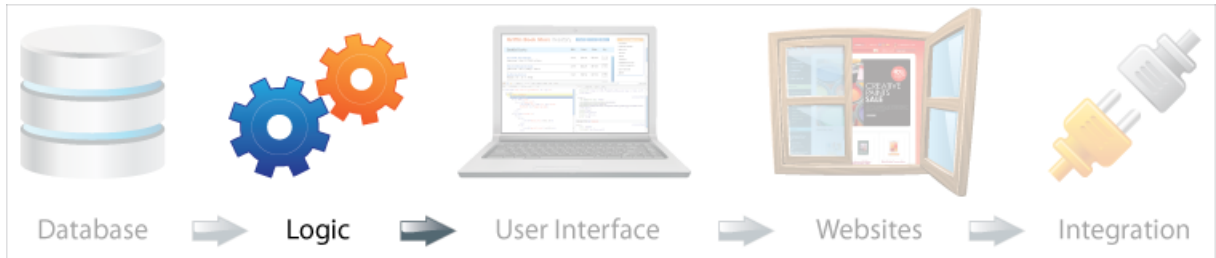
## Siehe auch:

[Lernprogramm 3: Verwenden von Formeln und Validierungsregeln](#)

[Schritt 5: Optimieren der Validierungsregel](#)

# Lernprogramm 4: Verwenden von Workflows und Genehmigungen

**Niveau:** Fortgeschrittene Anfänger; **Dauer:** 30-45 Minuten



Ihr Unternehmen kann mit standardisierten internen Verfahren und automatisierten Geschäftsprozessen effizienter arbeiten. In der Force.com-Anwendung können Sie Workflowregeln und Genehmigungsprozesse verwenden, um Ihre Verfahren und Prozesse zu automatisieren. Mit diesen wird nicht nur Zeit eingespart, sondern auch eine höhere Konsistenz und Übereinstimmung bei der Geschäftspraxis Ihres Unternehmens erzielt.

Workflowregeln können basierend auf Zeitauslösern, Kriterien und Formeln Aktionen wie E-Mail-Benachrichtigungen, Feldaktualisierungen und ausgehenden Nachrichten Aufgaben auslösen. Genehmigungsprozesse können alle Genehmigungen in Ihrer Organisation, von einfachen bis zu komplexen, automatisieren.

In diesem Lernprogramm erstellen und testen Sie zwei Workflowregeln. Mit der ersten wird der aktuelle Preis eines Merchandise-Artikels in den neu erstellten Belegpostendatensatz übertragen. Die zweite Workflowregel aktualisiert den Bestand immer dann, wenn ein Produktbelegposten aktualisiert wird. Des Weiteren erstellen Sie einen Genehmigungsprozess, bei dem die ausdrückliche Genehmigung eines Abteilungsleiters erforderlich wird, wenn eine Rechnung einen Wert über \$ 2000 aufweist.

## Voraussetzungen

### Formeln und Validierung

Zunächst müssen Sie das Rollup-Zusammenfassungsfeld und die Validierungsregeln erstellen, wie in [Lernprogramm 3: Verwenden von Formeln und Validierungsregeln](#) auf Seite 23 beschrieben.

## Siehe auch:

- [Schritt 1: Auffüllen des Belegpostenpreises mit einer Workflowregel](#)
- [Schritt 2: Aktualisieren des Gesamtbestands bei der Erteilung von Aufträgen](#)
- [Schritt 3: Testen der Workflowregeln](#)
- [Schritt 4: Erstellen einer E-Mail-Vorlage](#)
- [Schritt 5: Erstellen eines Genehmigungsprozesses](#)
- [Schritt 6: Erstellen eines benutzerdefinierten Sicherheitsprofils](#)
- [Schritt 7: Erstellen eines Benutzers](#)
- [Schritt 8: Testen des Genehmigungsprozesses](#)
- [Zusammenfassung](#)

## Schritt 1: Auffüllen des Belegpostenpreises mit einer Workflowregel

Zu diesem Zeitpunkt haben Sie Belegposten- und Merchandise-Objekte mit Preisfeldern erstellt. Auch haben Sie das Feld für den Belegpostenpreis als schreibgeschützt definiert. Dies trat in der Benutzeroberfläche allerdings u. U. nicht zutage, da es sich beim Standardbenutzer um Administratoren handelt, die Zugriff auf alle Daten haben. Nunmehr erstellen Sie eine Workflowregel zum Auffüllen des Stückpreises der Zeile mit dem Stückpreis des Merchandise-Artikels zum Zeitpunkt des Erstellens des Belegpostens. Mit einer deklarativen Workflowregel kann diese komplexe Logik verhältnismäßig leicht erstellt werden.

1. Wählen Sie die Optionsfolge **Setup ► Erstellen ► Workflow & Genehmigungen ► Workflowregeln**. Klicken Sie auf **Weiter**, wenn die Seite "Grundlegendes zu Workflow" angezeigt wird.
2. Klicken Sie auf der Seite "Alle Workflow-Regeln" auf **Neue Regel**.
3. Wählen Sie in Schritt 1 des Assistenten für Workflowregeln `Line Item` als Objekt aus und klicken Sie auf **Weiter**.
4. Geben Sie im Feld `Rule Name` den Wert `Line Item Created` ein.
5. Wählen Sie für `Evaluation Criteria` den Wert `Only when a record is created` aus.
6. Wählen Sie im Feld `Rule Criteria` den Wert `formula evaluates to true` aus.
7. Geben Sie im Textfeld `true` ein.
8. Vergewissern Sie sich, dass die Seite der folgenden entspricht, und klicken Sie dann auf **Speichern & Weiter**.

The screenshot shows the 'Edit Rule' configuration interface. Key elements include:

- Object:** Line Item
- Rule Name:** Line Item Created
- Evaluation Criteria:** Only when a record is created (selected)
- Rule Criteria:** formula evaluates to true
- Example:** OwnerId <=> LastModifiedById evaluates to true when the person who last modified the record is not the record owner.
- Functions List:** ABS, AND, BEGINS, BLANKVALUE, BR, CASE
- Text Field:** true

9. Klicken Sie bei Schritt 3 des Assistenten für Workflowregeln im Abschnitt "Sofortige Workflow-Aktionen" auf **Workflow-Aktion hinzufügen** und wählen Sie `New Field Update` aus. Der Assistent für neue Feldaktualisierungen wird geöffnet.
10. Geben Sie im Feld `Name` den Wert `Insert Merchandise Price` ein.
11. Wählen Sie in der Dropdown-Liste `Field to Update` im ersten Feld den Wert `Line Item` und im zweiten den Wert `Unit Price` aus.
12. Wählen Sie `Use a formula to set the new value` aus.
13. Klicken Sie auf **Formel-Editor anzeigen**.
14. Klicken Sie auf **Feld einfügen**.
15. Wählen Sie in der ersten Spalte `Line Item >`, in der zweiten `Merchandise >` und in der dritten `Price` aus.
16. Klicken Sie auf **Einfügen**.

17. Klicken Sie auf **Speichern**, um den Assistenten für neue Feldaktualisierungen zu schließen und zu Schritt 3 des Workflowassistenten zurückzukehren.
18. Klicken Sie auf der Seite "Workflow-Aktionen angeben" auf **Fertig**.
19. Klicken Sie auf der Seite "Workflow-Regel" auf **Aktivieren**.



**Wichtig:** Es wird häufig vergessen, neue Workflowregeln zu aktivieren. Wenn die Regel nicht aktiv ist, werden ihre Kriterien beim Erstellen oder Speichern von Datensätzen nicht evaluiert.

### Weitere Informationen

Wenn nun ein neuer Belegposten erstellt, gespeichert und angezeigt wird, bemerken Sie, dass der Stückpreis des Belegpostens automatisch mit dem Preis des verknüpften Merchandise-Datensatzes festgelegt wird. Denken Sie daran, dass Ihnen als Administrator ein Eingabefeld für den Preis angezeigt wird. Da dieses Feld schreibgeschützt ist, wird es den Benutzern nicht angezeigt.

Der Grund dafür, dass diese Workflowregel nur beim Erstellen von Datensätzen und niemals danach ausgeführt wird, liegt darin begründet, dass der Rechnungspreis eines Artikels nicht erhöht werden soll, nachdem der Artikel zur Rechnung hinzugefügt wurde. Kunden wären irritiert, wenn sie plötzlich feststellten, dass der Preis eines Artikels höher als beworben ist. Im Gegensatz dazu wären die Kunden froh, wenn eine Aktualisierung mit dem neuen Preis durchgeführt würde, wenn der Preis eines Artikels sinkt. Diese Funktion wird zu einem späteren Zeitpunkt implementiert.

Sie wundern sich vielleicht, warum wir eine Regel erstellt haben, deren Formel immer den Wert "true" ergibt. Dies ist so, weil die Feldaktualisierung immer vorgenommen werden soll, wenn ein Datensatz erstellt wird, wozu der Wert stets "true" lauten muss. Es kann auch eine Formel zum Evaluieren eines Ausdrucks verwendet werden, was in diesem Fall nicht erforderlich war.

### Siehe auch:

[Lernprogramm 4: Verwenden von Workflows und Genehmigungen](#)  
[Schritt 2: Aktualisieren des Gesamtbestands bei der Erteilung von Aufträgen](#)

## Schritt 2: Aktualisieren des Gesamtbestands bei der Erteilung von Aufträgen

Im Idealfall wird der Gesamtbestand der Merchandise-Datensätze automatisch verwaltet. Beim Erstellen von neuen Rechnungen (die standardmäßig mit dem Status "Offen" erstellt werden) sollen alle Aktualisierungen der Belegposten im Feld des Gesamtbestands widergespiegelt werden, das für das Merchandise-Objekt erstellt wurde. Dies kann mit einer weiteren Workflowregel erreicht werden. Dabei muss nur darauf geachtet werden, den Wert des Gesamtbestands zu ändern. Beim Erstellen eines neuen Belegpostens muss der Gesamtbestand um die Anzahl der verkauften Artikel reduziert werden. Wenn hingegen ein Belegposten aktualisiert wird, muss der Gesamtbestand um die Differenz zwischen Anzahl der Artikel zuvor und neu verkaufter Artikel angepasst werden.

1. Wählen Sie die Optionsfolge **Setup ► Erstellen ► Workflow & Genehmigungen ► Workflowregeln**.
2. Klicken Sie auf **Weiter**, wenn die Seite "Grundlegendes zu Workflow" angezeigt wird. Fahren Sie anderenfalls mit dem nächsten Schritt fort.
3. Klicken Sie auf der Seite "Alle Workflow-Regeln" auf **Neue Regel**.
4. Wählen Sie in Schritt 1 des Assistenten für Workflowregeln **Line Item** als Objekt aus und klicken Sie auf **Weiter**.
5. Geben Sie im Feld **Rule Name** den Wert **Line Item Updated** ein.
6. Wählen Sie für **Evaluation Criteria** den Wert **Every time a record is created or edited** aus.
7. Wählen Sie im Feld **Rule Criteria** den Wert **criteria are met** aus.
8. Wählen Sie in der Dropdown-Liste **Field** die Option **Invoice Statement: Status** aus. Wählen Sie dann für **Operator** den Wert **equals** aus. Klicken Sie unter **Value** auf das Nachschlagesymbol und wählen Sie **Open** aus. Klicken Sie auf **Ausgewählte einfügen**.
9. Vergewissern Sie sich, ob die Seite der folgenden entspricht, und klicken Sie dann auf **Speichern & Weiter**.

Field	Operator	Value	
Invoice Statement Status	equals	Open	AND
-None-	-None-		AND
-None-	-None-		AND
-None-	-None-		AND
-None-	-None-		

10. Klicken Sie bei Schritt 3 des Assistenten für Workflowregeln im Abschnitt "Sofortige Workflow-Aktionen" auf **Workflow-Aktion hinzufügen** und wählen Sie **New Field Update** aus. Der Assistent für neue Feldaktualisierungen wird geöffnet.
11. Geben Sie im Feld **Name** den Wert **Update Stock Inventory** ein.
12. Wählen Sie in der ersten Dropdown-Liste **Field to Update** den Wert **Merchandise** aus. Wählen Sie in der zweiten den Wert **Total Inventory** aus.
13. Wählen Sie **Use a formula to set the new value** aus.
14. Klicken Sie auf **Formel-Editor anzeigen**.

15. Geben Sie folgenden Code ein.

```
IF ( ISNEW(), Merchandise__r.Total_Inventory__c - Units_Sold__c ,
Merchandise__r.Total_Inventory__c - (Units_Sold__c - PRIORVALUE(Units_Sold__c)) )
```

16. Klicken Sie auf **Syntax prüfen** und nehmen Sie ggf. Korrekturen vor.

17. Klicken Sie auf **Speichern**, um den Assistenten für neue Feldaktualisierungen zu schließen und zu Schritt 3 des Workflowassistenten zurückzukehren.

18. Klicken Sie auf der Seite "Workflow-Aktionen angeben" auf **Fertig**.

19. Klicken Sie auf der Seite "Workflow-Regel" auf **Aktivieren**.

### Siehe auch:

[Lernprogramm 4: Verwenden von Workflows und Genehmigungen](#)

[Schritt 1: Auffüllen des Belegpostenpreises mit einer Workflowregel](#)

[Schritt 3: Testen der Workflowregeln](#)

## Schritt 3: Testen der Workflowregeln

Sie können prüfen, ob die Workflowregeln wie gewünscht funktionieren, indem Sie einen neuen Belegposten erstellen. Überprüfen Sie anschließend, ob dessen Stückpreis automatisch festgelegt wird und der Gesamtbestand im Lager reduziert wurde.

1. Klicken Sie auf die Registerkarte "Rechnungsanweisungen".
2. Klicken Sie auf den Rechnungsdatensatz, den Sie zuvor erstellt haben.
3. Wählen Sie `New Line Item` aus und geben Sie die folgenden Werte ein.
  - Geben Sie im Feld `Line Item Number` den Wert 2 ein.
  - Lassen Sie das Feld `Unit Price` leer.
  - Legen Sie `Units Sold` mit 1000 fest.

- Klicken Sie neben dem Feld `Merchandise` auf das Nachschlagesymbol und wählen Sie den Merchandise-Datensatz "Wee Jet" aus.
4. Klicken Sie auf **Speichern**.
  5. Klicken Sie auf Belegposten 2. Beachten Sie, dass der Stückpreis mit dem gleichen Wert festgelegt wurde wie der Datensatz "Wee Jet".
  6. Öffnen Sie nun den Merchandise-Datensatz "Wee Jet" durch Klicken auf den Link **Wee Jet**. Beachten Sie, dass `Total Inventory` von 2000 (bzw. von dem von Ihnen eingegebenen Wert) auf 1000 verringert wurde.

### Siehe auch:

[Lernprogramm 4: Verwenden von Workflows und Genehmigungen](#)

[Schritt 2: Aktualisieren des Gesamtbestands bei der Erteilung von Aufträgen](#)

[Schritt 4: Erstellen einer E-Mail-Vorlage](#)

## Schritt 4: Erstellen einer E-Mail-Vorlage

Eine der Geschäftsregeln, die Sie in Ihrem Workflow durchsetzen möchten, bedingt, dass alle Rechnungen mit einem Gesamtwert von über \$ 2000 von einem Manager genehmigt werden müssen. Zum Implementieren dieser Regel müssen Sie zwei weitere Elemente erstellen: eine E-Mail, die an den Manager gesendet wird, wenn eine Rechnung einen Wert von \$ 2000 überschreitet, und einen Genehmigungsprozess, den der Manager einhalten muss.



**Anmerkung:** Falls es sich bei Ihrer Standardwährung nicht um US-Dollar handelt, nehmen Sie einfach an, dass die bei Ihnen gebräuchliche Währung verwendet wird.

In diesem Schritt erstellen Sie eine E-Mail-Vorlage, die von der Workflowregel zum Generieren und Senden der E-Mail verwendet wird. In den nächsten Schritten erstellen Sie den Genehmigungsprozess.

1. Klicken Sie auf **Setup** ► **Kommunikationsvorlagen** ► **E-Mail-Vorlagen**.
2. Klicken Sie auf **Neue Vorlage**.
3. Wählen Sie in Schritt 1 des Assistenten für E-Mail-Vorlagen den Wert `Text` aus und klicken Sie auf **Weiter**.
4. Wählen Sie `Available For Use` aus.
5. Geben Sie im Feld `Email Template Name` den Wert `Large Invoice Template` ein.
6. Geben Sie im Feld `Subject` den Wert `A large invoice has been submitted` ein.
7. Geben Sie im Feld `Email Body` folgenden Code und Text ein.

```
{!Invoice_Statement__c.OwnerFullName} submitted for approval an Invoice Statement that
totalled {!Invoice_Statement__c.Invoice_Value__c} on
{!Invoice_Statement__c.LastModifiedDate}
```

8. Überprüfen Sie, dass die Anzeige dem folgenden Beispiel entspricht, und klicken Sie dann auf **Speichern**.

The screenshot shows a 'Step 2 of 2' dialog for creating a new text email template. The 'Email Template Information' section includes the following fields:

- Folder:** Unified Public Email Templates
- Available For Use:**
- Email Template Name:** Large Invoice Template
- Template Unique Name:** Large\_Invoice\_Templat
- Encoding:** General US & Western Europe (ISO-8859-1, ISO-LATIN-1)
- Description:** (empty)
- Subject:** A large invoice has been submitted
- Email Body:** {{Invoice\_Statement\_\_c.OwnerFullName}} submitted for approval an Invoice Statement that totalled {{Invoice\_Statement\_\_c.Invoice\_Value\_\_c}} on {{Invoice\_Statement\_\_c.LastModifiedDate}}

### Weitere Informationen

Der eben eingegebene E-Mail-Text unterstützt Briefvorlagenfelder wie `{!Invoice_Statement__c.Invoice_Value__c}`. Zum Zeitpunkt der E-Mail-Generierung wird dieser Wert dynamisch mit dem tatsächlichen Wert der Rechnung ersetzt, der den Genehmigungsprozess generiert hat.

### Siehe auch:

[Lernprogramm 4: Verwenden von Workflows und Genehmigungen](#)

[Schritt 3: Testen der Workflowregeln](#)

[Schritt 5: Erstellen eines Genehmigungsprozesses](#)

## Schritt 5: Erstellen eines Genehmigungsprozesses

In diesem Schritt erstellen Sie einen Genehmigungsprozess, bei dem die ausdrückliche Genehmigung eines Abteilungsleiters erforderlich wird, wenn eine Rechnung einen Wert über \$ 2000 aufweist. Das Erstellen und Verwenden eines Genehmigungsprozesses ist ebenso einfach wie das Erstellen einer Workflowregel.

1. Wählen Sie die Optionsfolge **Setup > Erstellen > Workflow & Genehmigungen > Genehmigungsprozesse**.
2. Wählen Sie in der Dropdown-Liste **Manage Approval Processes For** den Wert **Invoice Statement** aus.
3. Klicken Sie auf **Neuen Genehmigungsprozess erstellen** und wählen Sie in der Dropdown-Liste die Option **Use Jump Start Wizard** aus.
4. Geben Sie im Feld **Name** den Wert **Large Invoice Value** ein.
5. Klicken Sie neben dem Feld **Approval Assignment Email Template** auf das Nachschlagesymbol und wählen Sie den eben erstellten Wert **Large Invoice Template** aus.
6. Geben Sie die folgenden Werte im Bereich "Eingabekriterien festlegen" ein.
  - a. Wählen Sie in der Dropdown-Liste **Field** die Option **Invoice Value** aus.
  - b. Wählen Sie in der Dropdown-Liste **Operator** den Wert **greater than** aus.
  - c. Geben Sie im Feld **value** den Wert **2000** ein.
7. Wählen Sie **Automatically assign an approver using a standard or custom hierarchy field** aus und wählen Sie für das Hierarchiefeld den Wert **Manager** aus. Dadurch wird sichergestellt, dass der Manager des Benutzers

als Genehmiger zugewiesen wird, wenn dieser Benutzer den Genehmigungsprozess startet. Im nächsten Schritt erstellen Sie einen Benutzer mit dem Namen "Bob Smith".

The screenshot shows the configuration for an approval process. Key elements include:

- Name:** Large Invoice Value
- Unique Name:** Large\_Invoice\_Value
- Approval Assignment Email Template:** Large Invoice Template
- Specify Entry Criteria:** A table with columns 'Field', 'Operator', and 'Value'. The first row is 'Invoice Value', 'greater than', and '2000'.
- Select Approver:** The option 'Automatically assign an approver using a standard or custom hierarchy field' is selected, with 'Manager' chosen in the dropdown.

8. Klicken Sie auf **Speichern**.
9. Es wird eine Warnmeldung angezeigt, dass der Genehmigungsprozess aktiviert werden muss. Klicken Sie auf **OK**. Das Erstellen des Genehmigungsprozesses ist abgeschlossen. Bevor dieser jedoch ausgeführt werden kann, müssen Sie definieren, wie mit Datensätzen verfahren wird, wenn diese zum ersten Mal übermittelt, genehmigt und wenn diese abgelehnt werden.
10. Klicken Sie auf **Seite mit Genehmigungsprozessdetails anzeigen**.
11. Erstellen Sie neue Feldaktualisierungsaktionen, indem Sie auf **Neu hinzufügen** klicken und für jede Themenliste in der folgenden Tabelle **Field Update** auswählen. Konfigurieren Sie jede Feldaktualisierungsaktion wie gezeigt.

Themenliste	Name	Zu aktualisierendes Feld	Auswahllistenoptionen
Aktionen beim ursprünglichen Einreichen	Set Initial Approval Status	Status	Wählen Sie A specific value und dann Pending aus.
Aktionen für endgültige Genehmigung	Set Final Approval Status	Status	Wählen Sie A specific value und dann Closed aus.
Endgültige Ablehnungen	Set Final Rejection Status	Status	Wählen Sie A specific value und dann Open aus.

12. Klicken Sie auf **Speichern**.
13. Klicken Sie auf **Genehmigungsprozesse**, um zur Genehmigungsliste zurückzukehren.
14. Klicken Sie auf **Aktivieren**, um den Genehmigungsprozess zu aktivieren.

## Weitere Informationen

Einigen der Genehmigungsschritte sind Standardaktionen zugeordnet. Die Aktionen beim ursprünglichen Einreichen und die Aktionen für die endgültige Genehmigung sperren den Datensatz. Bei der Aktion für die endgültige Ablehnung hingegen wird er entsperrt. Dies ist aus folgenden Gründen sinnvoll: Wenn ein Benutzer einen Datensatz zur Genehmigung einreicht, sollte dieser in der Zwischenzeit nicht geändert werden. Ebenso sollte der Datensatz entsperrt werden, wenn dieser abgelehnt wird, sodass Änderungen vorgenommen werden können.

## Siehe auch:

[Lernprogramm 4: Verwenden von Workflows und Genehmigungen](#)

[Schritt 4: Erstellen einer E-Mail-Vorlage](#)

[Schritt 6: Erstellen eines benutzerdefinierten Sicherheitsprofils](#)

## Schritt 6: Erstellen eines benutzerdefinierten Sicherheitsprofils

Ein Profil ist eine Sammlung mit Berechtigungen und anderen Einstellungen, die einem Benutzer oder einer Benutzergruppe zugeordnet werden. In Ihrer Organisation sind bereits einige Standardprofile definiert. Beim Erstellen eines benutzerdefinierten Objekts sind die Berechtigungen zum Zugreifen auf dieses Objekt ("Lesen", "Erstellen", "Bearbeiten" und "Löschen") für die meisten Profile deaktiviert. Diese standardmäßige Sicherheitseinstellung gewährleistet, dass nicht alle Benutzer sofort Zugriff auf benutzerdefinierte Objekte und deren Daten erhalten, da die Berechtigung ausdrücklich erteilt werden muss. Diese Berechtigungen können in benutzerdefinierten Profilen, nicht jedoch in Standardprofilen geändert werden.

In diesem Schritt erstellen Sie ein benutzerdefiniertes Profil, das Sie Benutzern zuweisen können, die Zugriff auf die benutzerdefinierten Objekte in der Warenlageranwendung benötigen. Danach aktivieren Sie im neuen benutzerdefinierten Profil Objektberechtigungen, damit die Benutzer auf die Objekte zugreifen können. Den neuen Benutzer erstellen Sie im nächsten Schritt.

1. Klicken Sie auf **Setup ► Benutzer verwalten ► Profile**.
2. Klicken Sie auf **Neu**.
3. Wählen Sie im Feld `Clone Profile` den Wert "Standard User" aus. Wählen Sie bei Verwendung der Free Edition den Wert "Force.com - Free User" aus.
4. Geben Sie im Feld `User License Profile Name` den Wert `Basic User` ein.
5. Klicken Sie auf der Detailseite des neuen Profils auf **Bearbeiten**.
6. Führen Sie einen Bildlauf an das Ende der Seite zur Bearbeitung des Profils durch und wählen Sie unter den benutzerdefinierten Objektberechtigungen die Felder **Lesen**, **Erstellen**, **Bearbeiten** und **Löschen** für die Objekte "Rechnungsanweisungen", "Belegposten" und "Merchandise" aus.
7. Überprüfen Sie, dass die Anzeige dem folgenden Beispiel entspricht, und klicken Sie dann auf **Speichern**.

Custom Object Permissions						
	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Invoice Statements	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Line Items	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Merchandise	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## Weitere Informationen

Sie haben soeben erfahren, wie einfach es ist, ein benutzerdefiniertes Profil zu erstellen und zu bearbeiten. Wenn viele Profile bearbeitet werden müssen, können Sie die erweiterte Profilverwaltung verwenden, um eine benutzerdefinierte Listenansicht der Profile anzuzeigen und dann die Profile über die Listenansicht zu bearbeiten. Weitere Informationen finden Sie unter "Bearbeiten von Profilen mithilfe von Profillisten" in der Salesforce.com-Online-Hilfe.

## Siehe auch:

[Lernprogramm 4: Verwenden von Workflows und Genehmigungen](#)

[Schritt 5: Erstellen eines Genehmigungsprozesses](#)

[Schritt 7: Erstellen eines Benutzers](#)

## Schritt 7: Erstellen eines Benutzers

Beim Erstellen einer Force.com-Anwendung wird diese automatisch so aktiviert, dass sich mehrere Benutzer anmelden können. Jetzt können Sie die Anwendung anpassen, indem Sie sie so konfigurieren, dass Sie in Abhängigkeit vom Profil des sich anmeldenden Benutzers ein anderes Verhalten aufweist. Sie können einer bestimmten Benutzergruppe etwa schreibgeschützten Zugriff für Felder gewähren oder diese vollständig ausblenden. Sie können auch eine strikte Datenfreigabe durchsetzen, sodass einem Benutzer nur seine Datensätze oder die seines Vorgesetzten angezeigt werden.

In diesem Schritt erstellen Sie einen neuen Benutzer und verknüpfen diesen über das Managerfeld mit Ihrem aktuellen Benutzeraccount. Diese Konfiguration wird verwendet, um sicherzustellen, dass beim Erstellen einer Rechnung durch den neuen Benutzer unter bestimmten Bedingungen diese an seinen Vorgesetzten weitergeleitet wird.

1. Klicken Sie auf **Setup** ► **Benutzer verwalten** ► **Benutzer**.
2. Klicken Sie auf der Seite "Alle Benutzer" auf **Neuer Benutzer**.
3. Geben Sie folgende Informationen ein:

- Geben Sie im Feld `First Name` den Wert `Bob` ein.
- Geben Sie im Feld `Last Name` den Wert `Smith` ein.
- Geben Sie im Feld `Alias` den Wert `bsmith` ein.
- Geben Sie im Feld `Email` Ihre eigene E-Mail-Adresse ein, sodass Sie die Genehmigungsanfragen erhalten, die an "Bob Smith" weitergeleitet werden.
- Das Feld `Username` wird standardmäßig mit Ihrer E-Mail-Adresse festgelegt, Sie müssen jedoch einen eindeutigen Benutzernamen für "Bob" in Form einer frei erfundenen E-Mail-Adresse erstellen.



**Anmerkung:** Notieren Sie sich den Benutzernamen für den Benutzer "Bob" (also seine frei erfundene E-Mail-Adresse), da Sie sich in Kürze als dieser Benutzer anmelden.

- Wählen Sie im Feld `Manager` den Benutzer aus, den Sie beim Anmelden bei Ihrer Organisation erstellt haben.
- Wählen Sie im Feld **Benutzerlizenz** den Wert `Salesforce` aus.
- Wählen Sie im Feld `Profile` den Wert `Basic User` aus.

#### 4. Klicken Sie auf **Speichern**.

Sie sollten nun eine E-Mail mit der Bestätigung des Erstellens des neuen Benutzers erhalten. Sie müssen noch einen Setup-Schritt zum Abschließen des Genehmigungsprozesses durchführen. Melden Sie sich also noch nicht als "Bob Smith" an, da Sie sich sonst sofort wieder als Administrator anmelden müssen.

In der Free Edition müssen Sie vor der Anmeldung und Verwendung der Warenlageranwendung als "Bob" das Standardbenutzerprofil so bearbeiten, dass die Warenlageranwendung verwendet wird. Free Edition-Benutzer können nur auf eine Anwendung zugreifen, sodass es sich bei dieser um den Standard handeln muss. (Überspringen Sie diesen Schritt, wenn Sie die Developer Edition verwenden.)

1. Klicken Sie auf **Setup** ► **Benutzer verwalten** ► **Profile**.
2. Klicken Sie neben dem Profil `Basic User` auf **Bearbeiten**.
3. Wählen Sie neben `Warehouse` den Wert `Default` aus.

#### Siehe auch:

[Lernprogramm 4: Verwenden von Workflows und Genehmigungen](#)

[Schritt 6: Erstellen eines benutzerdefinierten Sicherheitsprofils](#)

[Schritt 8: Testen des Genehmigungsprozesses](#)

## Schritt 8: Testen des Genehmigungsprozesses

Nun kann der Genehmigungsprozess zusammengestellt und in der Praxis angezeigt werden. Wenn der Benutzer "Bob Smith" eine Rechnung erstellt und deren Gesamtwert über \$ 2000 liegt, kann er in der Rechnungsanweisung auf die neue Schaltfläche zum Einreichen für eine Genehmigung klicken. Die Anwendung sendet unter Verwendung der zuvor erstellten E-Mail-Vorlage eine E-Mail an seinen Manager und sperrt den Datensatz, sodass keine weiteren Änderungen mehr vorgenommen werden können.

Nach dem Anmelden kann der Manager die Rechnung genehmigen oder ablehnen. Wenn der Manager die Rechnung genehmigt, soll deren Status mit dem Wert "Closed" aktualisiert werden. Wenn die Rechnung hingegen abgelehnt wird, soll der ursprüngliche Status "Open" beibehalten werden. Bei praktischen Anwendungen kann es der Geschäftsprozess erforderlich machen, dass unter anderem eine E-Mail an den Rechnungsinhaber mit der Bitte um Bestätigung gesendet wird.

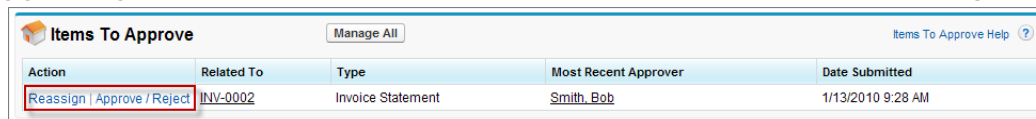
Stellen Sie vor dem Testen des Genehmigungsprozesses sicher, dass auf Ihrer Website Elemente angezeigt werden können, die einer Genehmigung bedürfen.

1. Klicken Sie auf **Setup** ► **Anpassen** ► **Startseite** ► **Startseitenlayouts**.
2. Klicken Sie neben Ihrem Startseitenlayout auf **Bearbeiten**. (Klicken Sie bei Verwendung der Free Edition neben VolunteerForce Home Page auf **Bearbeiten**.)
3. Wählen Sie die Option **Items to Approve** aus, falls diese nicht bereits ausgewählt ist.
4. Klicken Sie auf **Weiter** und dann auf **Speichern**.

Beim Entwickeln der Anwendung waren Sie als Administrator angemeldet. Um einen neuen Datensatz zu erstellen und den Genehmigungsprozess zu testen, müssen Sie sich vom Administratoraccount abmelden und sich als Standardbenutzer anmelden.

1. Melden Sie sich als "Bob Smith" an, also als der Benutzer, den Sie zuvor erstellt haben. Wenn Sie sich zum ersten Mal als Benutzer "Bob" anmelden, müssen Sie unter Umständen das Kennwort ändern.
2. Wählen Sie die Warenlageranwendung aus.
3. Klicken Sie auf die Registerkarte "Rechnungsanweisungen".
4. Klicken Sie auf **Neu** und erstellen Sie eine Rechnungsanweisung.
5. Fügen Sie mit der Option **Neuer Belegposten** einen neuen Belegposten und ausreichend verkaufte Einheiten hinzu, sodass die Rechnung einen Wert von \$ 2000 übersteigt.
6. Klicken Sie auf **Speichern**.
7. Klicken Sie auf **Für Genehmigung einreichen**.
8. Melden Sie sich von der Anwendung ab.

Eine E-Mail wird an den festgelegten Manager des Benutzers "Bob Smith" gesendet, bei dem es sich im Rahmen dieses Lernprogramms um Sie als Administrator handelt. Da Sie der Manager sind, können Sie Ihren Posteingang überprüfen, um dies zu bestätigen. Wenn Sie sich als Administrator bei der Anwendung anmelden, wird auf der Registerkarte "Startseite" die ausstehende Genehmigungsanfrage angezeigt. (Möglicherweise müssen Sie einen Bildlauf durchführen, damit diese angezeigt wird.) Der Datensatz ist gesperrt, bis der Manager die Rechnung genehmigt. Mit dem Feld "Genehmigung" kann der Manager die Rechnung genehmigen oder ablehnen und den Workflow fortführen. Fahren Sie fort und **Genehmigen** Sie sie.

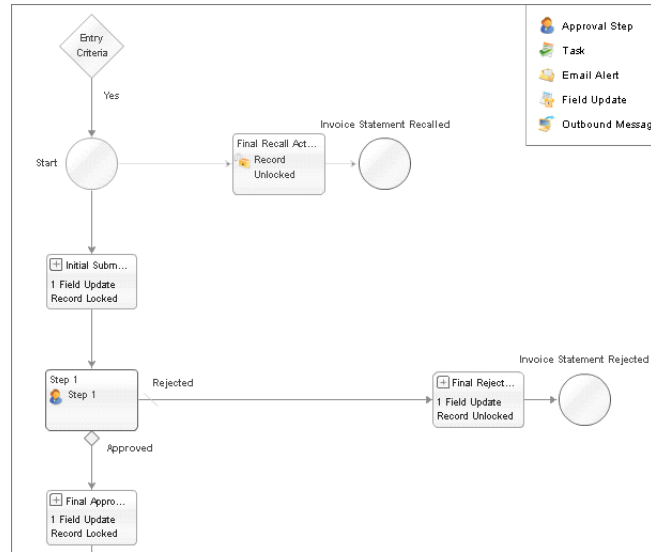


Action	Related To	Type	Most Recent Approver	Date Submitted
Reassign   Approve / Reject	INV-0002	Invoice Statement	Smith, Bob	1/13/2010 9:28 AM

## Weitere Informationen

Auf der Detailseite "Genehmigungsprozess" wird ein anschauliches Diagramm des Genehmigungsprozesses und der Aktionen angezeigt, die bei jedem Schritt ausgelöst werden.

1. Klicken Sie im bereits bekannten Bereich **Setup** auf **Genehmigungsprozess** und wählen Sie den erstellten Prozess aus.
2. Klicken Sie auf **Diagramm anzeigen**. Das Genehmigungsprozessdiagramm wird in einem separaten Browserfenster angezeigt.

**Siehe auch:**

[Lernprogramm 4: Verwenden von Workflows und Genehmigungen](#)  
[Schritt 7: Erstellen eines Benutzers](#)  
[Zusammenfassung](#)

**Zusammenfassung**

Workflowregeln und Genehmigungsprozesse unterstützen Sie beim Automatisieren Ihrer Geschäftsabläufe und Durchsetzen Ihrer Standards. Die Workflowregel aktualisiert automatisch Preise und Bestand über verschiedene Objekte hinweg. Der Genehmigungsprozess wird immer ausgeführt, wenn bestimmte Bedingungen vorliegen, und sendet eine E-Mail-Benachrichtigung an den angegebenen Empfänger.



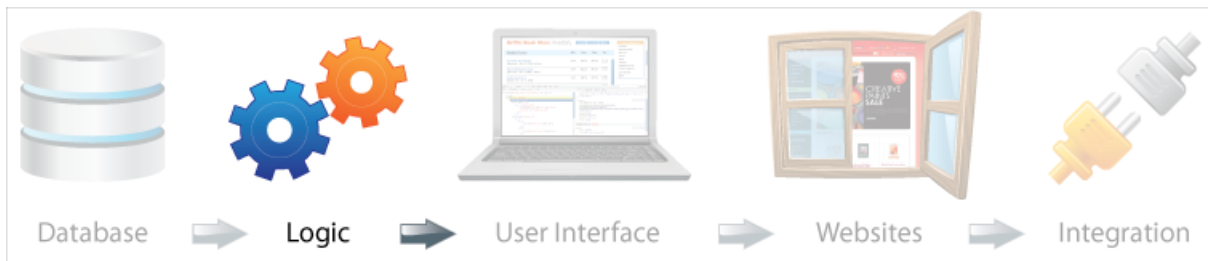
**Wichtig:** Bei Verwendung der Free Edition ist es bei den folgenden Lernprogrammen erforderlich, dass die Entwicklung in einer Sandbox durchgeführt wird. Befolgen Sie die Anweisungen unter [Erstellen einer Sandbox und Anmelden bei dieser](#) auf Seite 93, wenn die Entwicklung noch nicht in einer Sandbox durchgeführt wird. Die Sandbox-Instanz dupliziert keine Datensätze. Daher müssen Sie nach dem Anmelden den Datensatz "Wee Jet" wie in [Schritt 5: Erstellen eines Merchandise-Datensatzes](#) auf Seite 13 und die Rechnungsanweisung wie in [Schritt 5: Erstellen eines Rechnungsdatensatzes](#) auf Seite 20 beschrieben neu erstellen.

**Siehe auch:**

[Lernprogramm 4: Verwenden von Workflows und Genehmigungen](#)  
[Schritt 8: Testen des Genehmigungsprozesses](#)

# Lernprogramm 5: Hinzufügen von Geschäftslogik mit Apex

**Niveau:** Fortgeschritten; **Dauer:** 20–30 Minuten



Apex ist eine streng typisierte, objektorientierte und Java™-ähnliche Programmiersprache, die auf Force.com ausgeführt wird. Apex wird zum Hinzufügen programmatischer Geschäftslogik zu Anwendungen verwendet. Sie können sie z. B. zum Schreiben von Auslösern, Webservices und Programmsteuerfeldern in der Benutzeroberflächenebene Ihrer Anwendung verwenden.

Sie haben mit der deklarativen Workflow-Umgebung bereits Geschäftslogik hinzugefügt. In diesem Lernprogramm erstellen Sie mit Apex weitere Logik, die sich ideal eignet, um mit mehreren Datensätzen und komplexer Logik zu arbeiten.

Bei dem Geschäftsvorgang, an dem Sie arbeiten, muss der sinkende Preis von Merchandise-Artikeln an die Kunden weitergegeben werden. Dazu erstellen Sie einen Apex-Auslöser, mit dem alle offenen Rechnungen aktualisiert werden, wenn der Merchandise-Preis sinkt. Ein Auslöser ist ein Satz mit Code, der zu einem bestimmten Zeitpunkt während des Lebenszyklus eines Datensatzes ausgelöst wird. In diesem Fall erstellen Sie einen Auslöser, der nach dem Aktualisieren eines Merchandise-Datensatzes ausgelöst wird.

Anwendungen können auf zwei Arten auf Force.com erstellt werden: Sie können entweder wie bisher die Onlineumgebung verwenden oder die Force.com-IDE einsetzen. Falls die IDE noch nicht installiert ist, kann diese in wenigen Momenten eingerichtet werden. Falls Sie jedoch bereits mit der IDE vertraut sind, wissen Sie, dass diese Syntaxhervorhebung, Codeeinklick und viele weitere Funktionen aufweist, die die Entwicklung insbesondere im Team zu einem äußerst produktiven Vorgang machen.



**Anmerkung:** Beim Einsatz der Free Edition muss sämtliche Entwicklungsarbeit in der Sandbox durchgeführt werden. Falls Sie noch keine Sandbox erstellt haben, finden Sie unter [Erstellen einer Sandbox und Anmelden bei dieser](#) auf Seite 93 weitere Informationen.

## Voraussetzungen

### Grundlegendes Wissen

Bei diesem Lernprogramm ist es hilfreich, über grundlegendes Wissen zu objektorientierten Programmiersprachen wie Java oder C# zu verfügen, dies ist aber nicht zwingend erforderlich. Dieses Lernprogramm kann auch in der Force.com-IDE abgeschlossen werden, sodass sich Kenntnisse zu Eclipse als nützlich erweisen können. Auch diese sind jedoch nicht zwingend erforderlich.

### Softwareanforderungen

Falls Sie möchten, können Sie für dieses Lernprogramm die Force.com-IDE verwenden. Wenn Sie dies tun möchten, benötigen Sie Eclipse 3.3 oder 3.4 und das Force.com-IDE-Plugin: [wiki.developerforce.com/index.php/Force.com\\_IDE](http://wiki.developerforce.com/index.php/Force.com_IDE). Informationen zum Erstellen eines Projekts finden Sie unter [Erstellen eines Projekts in der Force.com-IDE](#) auf Seite 93.

## Berechtigungen "Entwicklermodus", "Alle Daten modifizieren" und "Autor-Apex"

Stellen Sie sicher, dass Sie entsprechende Berechtigungen zum Erstellen von Apex-Klassen haben, da im Rahmen dieses Lernprogramms Arbeiten mit Apex anfallen.

### Siehe auch:

[Schritt 1: Erstellen einer Apex-Auslöserdefinition](#)

[Schritt 2: Definieren einer Listenvariablen](#)

[Schritt 3: Durchgehen der Liste und Ändern des Preises](#)

[Schritt 4: Testen des Auslösers](#)

[Zusammenfassung](#)

## Schritt 1: Erstellen einer Apex-Auslöserdefinition

Zunächst müssen Sie die Auslöserdefinition erstellen, die den Namen des Auslösers, das betroffene Objekt und eine Aktion enthält, die den Auslöser auslöst. Auslöser können auf der Weboberfläche oder in der Force.com-IDE erstellt werden. Für beide Tools sind Anweisungen vorhanden.

So erstellen Sie einen Auslöser auf der Weboberfläche:

1. Klicken Sie in der Randleiste auf **Setup** ► **Erstellen** ► **Objekte**.
2. Klicken Sie auf Ihr benutzerdefiniertes Objekt **Merchandise**.
3. Führen Sie auf der Detailseite "Merchandise" einen Bildlauf nach unten bis **Auslöser** durch und klicken Sie auf **Neu**.
4. Ersetzen Sie <name> und <events>, sodass der Code wie folgt aussieht:

```
trigger HandleProductPriceChange on Merchandise__c (after update) {
}
```

5. Klicken Sie bei Verwendung der Weboberfläche auf **Schnellspeicherung**, wodurch Ihre Änderungen gespeichert werden und Sie mit dem Bearbeiten fortfahren können. Durch Speichern Ihrer Änderungen zu diesem Zeitpunkt wird des Weiteren überprüft, ob Sie den Code korrekt eingegeben haben. Bei Syntaxfehlern lässt das System nämlich keine Speichervorgänge zu.

So erstellen Sie den Auslöser in der Force.com-IDE:

1. Klicken Sie im Package Explorer mit der rechten Maustaste auf Ihr Projekt und klicken Sie dann auf **Neu** ► **Apex-Auslöser**.
2. Geben Sie im Dialogfeld als Name `HandleProductPriceChange` ein.
3. Wählen Sie in der Dropdown-Liste `Object` die Option `Merchandise__c` aus.



**Anmerkung:** Klicken Sie auf **Objekte aktualisieren**, wenn dieses Objekt nicht in der Liste der Objekte angezeigt wird. (Ignorieren Sie die Popup-Warnung mit Informationen dazu, wie ein Vorgang ausgewählt werden muss.)

4. Wählen Sie `after update` aus.

Apex Trigger Properties

Name:

Version: 16.0

Object:  Refresh Objects

Apex Trigger Operations

before insert  before update  before delete  after insert

after update  after delete  after undelete

5. Klicken Sie auf **Fertig stellen** und die Datei wird im Editor geöffnet.

### Weitere Informationen

Noch führt der Auslöser keine Vorgänge aus. Sie können nun jedoch jede Logik verwenden, die ausgeführt werden soll, wenn ein Merchandise-Datensatz aktualisiert wird. Vor den Informationen zur Auslöserlogik finden Sie hier noch Hinweise zu den einzelnen Bestandteilen der Auslöserdefinition.

- `HandleProductPriceChange`: Der Name des Auslösers.
- `on Merchandise__c`: Das Objekt, auf den sich der Auslöser bezieht (in diesem Fall Ihr benutzerdefiniertes Objekt `Merchandise__c`).
- `(after update)`: Die Aktion, die den Auslöser auslöst. Apex-Auslöser werden nach Datenaktionen wie Einfüge-, Aktualisierungs- und Löschvorgängen ausgelöst, und zwar vor oder nach einem dieser Ereignisse. Der hier definierte Auslöser wird nach dem Aktualisieren eines Datensatzes ausgelöst.
- `{ }`: Der von geschweiften Klammern umschlossene Code wird als *Text* bezeichnet und gibt an, welche Vorgänge durch den Auslöser ausgeführt werden. Im nächsten Schritt wird der Text des Auslösers erstellt.

### Siehe auch:

[Lernprogramm 5: Hinzufügen von Geschäftslogik mit Apex](#)  
[Schritt 2: Definieren einer Listenvariablen](#)

## Schritt 2: Definieren einer Listenvariablen

Als Erstes müssen Sie eine Listenvariable definieren, die eine Liste der Belegposten enthält. Es werden nicht alle Belegposten benötigt, sodass Sie nur diejenigen auswählen, die sich in der Menge der Datensätze befinden, die das Ausführen dieses Codes auslösen und den Status "Negotiating" aufweisen.

1. Geben Sie in den geschweiften Klammern Ihrer Auslöserdefinition den folgenden Kommentar ein und definieren Sie dann eine Listenvariable.

```
// update invoice line items associated with open invoices List<Line_Item__c> openLineItems = [ ] ;
```



**Anmerkung:** Um den Rahmen dieses Handbuchs nicht zu sprengen, sind nicht zu allen Elementen Informationen vorhanden. Es hat sich jedoch als sinnvolle Vorgehensweise erwiesen, den Code zu kommentieren.

2. Geben Sie in den geschweiften Klammern eine Abfrage ein, die Informationen aus Ihrem benutzerdefinierten Objekt "Belegposten" abrufen.

```
List<Line_Item__c> openLineItems = [SELECT j.Unit_Price__c, j.Merchandise__r.Price__c FROM Line_Item__c j WHERE j.Invoice_Statement__r.Status__c = 'Negotiating' AND j.Merchandise__r.id IN :Trigger.new FOR UPDATE];
```



**Anmerkung:** Damit die Codebeispiele auf einer Seite angezeigt werden, sind Zeilenumbrüche an Stellen enthalten, an denen diese nicht zwingend erforderlich sind. Bei Apex werden Leerstellen ignoriert. Wenn das Format Ihres Codes nicht exakt dem hier gezeigten entspricht, ist dies kein Problem. Darüber hinaus wird bei Apex die Groß- und Kleinschreibung nicht berücksichtigt. Wenn Sie also lieber die Kleinschreibung verwenden möchten, können Sie beispielsweise `select` statt `SELECT` eingeben.

## Weitere Informationen

Die Liste `openLineItems` enthält eine Liste der Datensätze aus Ihrem benutzerdefinierten Objekt `Line_Item__c`. Der Inhalt dieser Liste wird von der Abfrage in den geschweiften Klammern bestimmt, die in der Salesforce-Objektabfragesprache (Salesforce Object Query Language, SOQL) geschrieben ist. Im Folgenden finden Sie detaillierte Informationen zu dieser Abfrage.

- **SELECT:** Gibt an, welche Objektfelder abgerufen werden.
- **FROM:** Gibt an, auf welche Objekte zugegriffen wird. Das "j" in `Line_Item__c j` ist ein Alias: Mit dieser praktischen Verknüpfung kann mit "j" auf `Line_Item__c` verwiesen werden.
- **WHERE:** Der Beginn der Bedingungsanweisung. Im vorliegenden Beispiel sollen nur diejenigen Datensätze zurückgegeben werden, deren Status "Negotiating" lautet.
- **AND:** Die zweite Bedingung der Anweisung. Diese ruft nur die eindeutigen IDs der Datensätze ab, die neu sind. Im Code wird die spezielle Variable `Trigger.new` verwendet, die automatisch mit den Kennzeichnern der Datensätze initialisiert wird, die aktualisiert werden.
- **FOR UPDATE:** Hiermit wird angegeben, dass die Datensätze auf der Plattform gesperrt werden sollen, wodurch andere Programme bzw. Benutzer daran gehindert werden, diese zu aktualisieren. Die Sperre bleibt aktiv, bis der Auslöser abgeschlossen ist.

## Siehe auch:

[Lernprogramm 5: Hinzufügen von Geschäftslogik mit Apex](#)

[Schritt 1: Erstellen einer Apex-Auslöserdefinition](#)

[Schritt 3: Durchgehen der Liste und Ändern des Preises](#)

## Schritt 3: Durchgehen der Liste und Ändern des Preises

Im vorhergehenden Schritt wurde eine Liste der Belegposten erstellt und in der Variablen `openLineItems` gespeichert. Nun kann die Liste mit der Schleife `for` durchgegangen und der Originalpreis von Artikeln geändert werden, wenn der neue Preis niedriger liegt.

1. Definieren Sie eine `for`-Schleife.

```
for (Line_Item__c li: openLineItems) { }
```

2. Geben Sie in den geschweiften Klammern eine bedingte `if`-Anweisung ein.

```
for (Line_Item__c li: openLineItems) { if ( li.Merchandise__r.Price__c < li.Unit_Price__c ) { } }
```

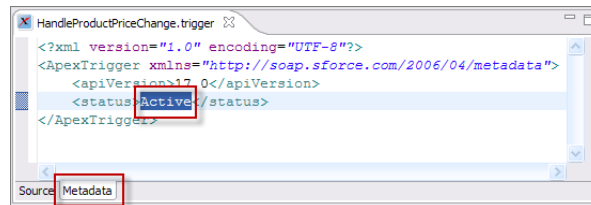
3. Geben Sie in den geschweiften Klammern der Schleife `for` den Code ein, mit dem der Preis aktualisiert wird.

```
for (Line_Item__c li: openLineItems) { if ( li.Merchandise__r.Price__c < li.Unit_Price__c ) { li.Unit_Price__c = li.Merchandise__r.Price__c; } }
```

4. Die Schleife `for` ist fertig gestellt. Nun sollen die Belegposten aktualisiert werden. Fügen Sie vor der letzten geschweiften Klammer des Auslösers Folgendes hinzu.

```
update openLineItems;
```

5. Während des Entwickelns sind Auslöser in der Regel nicht aktiviert. Um den Auslöser zu testen, muss dieser aktiviert werden.
- Aktivieren Sie bei Verwendung der Weboberfläche das Kontrollkästchen **Ist aktiv**, das nach dem Speichern des Auslösers verfügbar ist.
  - Klicken Sie bei Verwendung der IDE auf die Registerkarte **Metadaten** und ändern Sie den Statuswert in **Active**.



6. Stellen Sie sicher, dass Ihr Code dem folgenden Beispiel entspricht, und klicken Sie dann auf **Speichern**.

```
trigger HandleProductPriceChange on Merchandise__c (after update) {
    List<Line_Item__c> openLineItems = [SELECT j.Unit_Price__c, j.Merchandise__r.Price__c
    FROM Line_Item__c j WHERE j.Invoice_Statement__r.Status__c = 'Negotiating' AND
    j.Merchandise__r.id IN :Trigger.new FOR UPDATE];

    for (Line_Item__c li: openLineItems) { if ( li.Merchandise__r.Price__c <
    li.Unit_Price__c ) { li.Unit_Price__c = li.Merchandise__r.Price__c; } } update
    openLineItems; }
```

### Weitere Informationen

Mit der letzten Anweisung (`update openLineItems`) werden die Datensätze in der Datenbank aktualisiert. Diese Anweisung ist verhältnismäßig einfach aufgebaut, aber die `for`-Schleife bedarf einer näheren Betrachtung.

- `for (Line_Item__c li: openLineItems) { }:` Geht die Liste des offenen Belegpostens durch. Beim Durchgehen wird der aktuelle Belegpostens der Variablen `li` zugewiesen.
- `if (li.Merchandise__r.Price__c < li.Unit_Price__c ) { }:` Prüft, ob der Preis des Merchandise-Datensatzes unter dem aktuellen Preis liegt. Aktionen sollen nur dann ausgeführt werden, wenn dies der Fall ist.
- `li.Unit_Price__c = li.Merchandise__r.Price__c;:` Aktualisiert schließlich den Stückpreis des aktuellen Belegpostens (der Variablen `li` zugewiesen) mit dem neuen Merchandise-Preis.

### Siehe auch:

[Lernprogramm 5: Hinzufügen von Geschäftslogik mit Apex](#)

[Schritt 2: Definieren einer Listenvariablen](#)

[Schritt 4: Testen des Auslösers](#)

## Schritt 4: Testen des Auslösers

Nun testen wir den Auslöser in der Anwendung. Zunächst müssen Sie eine neue Rechnungsanweisung erstellen und mindestens ein Produkt bestellen bzw. ein bestehendes bearbeiten. Als einzige Bedingung müssen Sie das Statusfeld der Rechnungsanweisung in "Negotiating" ändern. Dann senken Sie den Stückpreis des Merchandise-Artikels, der in einem der Belegposten dieser Rechnung verwendet wird, und prüfen, ob die Belegposten- und Rechnungswerte aktualisiert werden.

1. Klicken Sie auf die Registerkarte "Rechnungsanweisungen".
2. Klicken Sie auf den Namen einer bestehenden Rechnungsanweisung.
3. Ändern Sie den Status in Negotiating und klicken Sie auf **Speichern**.
4. Achten Sie auf den Gesamtwert der Rechnung. Klicken Sie nun auf einen Belegposten und achten Sie auf seinen Stückpreis.
5. Klicken Sie auf die Registerkarte "Merchandise" und wählen Sie den Datensatz "Wee Jet" (oder einen beliebigen anderen Artikel in den Belegposten) aus.
6. Bearbeiten Sie den Datensatz durch Senken des Stückpreises. Ändern Sie den Wert in `.01`, um eine drastische Senkung zu verwenden.
7. Kehren Sie zur Rechnung und zum Belegposten zurück.
8. Beachten Sie, dass die Werte des Belegpostens und der Rechnung automatisch geändert wurden.

### Weitere Informationen

Sie haben sicher bemerkt, dass dieser Auslöser nicht besonders effektiv ist: Es wird eine Liste aller Belegpostendatensätze initialisiert und bearbeitet, die zu den Rechnungsanweisungen mit dem Status "Negotiation" zählen. Nur weil ein Merchandise-Datensatz aktualisiert wurde, muss dies allerdings noch nicht bedeuten, dass der Auslöser den Datensatz aktualisiert. Sie haben beispielsweise die Beschreibung des Merchandise-Datensatzes aktualisiert oder seinen Preis erhöht, wobei in beiden Fällen der Auslöser nicht ausgelöst werden muss. Im Idealfall sollte dieser Auslöser den Satz `Trigger.new` (die Liste der Merchandise-Datensätze, die aktualisiert wurden) durchgehen und dabei einen neuen Satz dieser Datensätze erstellen, die dem Preiskriterium entsprechen, und erst dann die entsprechenden Belegpostendatensätze abrufen.

### Siehe auch:

[Lernprogramm 5: Hinzufügen von Geschäftslogik mit Apex](#)

[Schritt 3: Durchgehen der Liste und Ändern des Preises](#)

[Zusammenfassung](#)

## Zusammenfassung

In diesem Lernprogramm haben Sie auf der Anwendung aufgebaut, indem Sie Geschäftslogik hinzufügen, mit der alle offenen Rechnungen aktualisiert werden, wenn der Stückpreis eines Merchandise-Artikels sinkt. Dazu haben Sie einen Auslöser definiert, der immer dann ausgeführt wird, wenn ein Merchandise-Datensatz mit einem niedrigeren Stückpreis gespeichert wird. In solchen Fällen, wenn Sie also mehrere Datensätze bei Vorliegen einer bestimmten Bedingung aktualisieren müssen, sind Auslöser sehr nützlich. Mit einem Objekt können mehrere Auslöser verknüpft sein, die bei unterschiedlichen Ereignissen und Bedingungen ausgelöst werden.

Ein wichtiger Bestandteil beim Erstellen von Apex-Auslösern ist sicherzustellen, dass automatische Tests vorhanden sind, die überprüfen, ob die Auslöser wie gewünscht ausgeführt werden. Weitere Informationen zum Testen erhalten Sie im [Lernprogramm 6: Hinzufügen von Tests zur Anwendung](#) auf Seite 50.

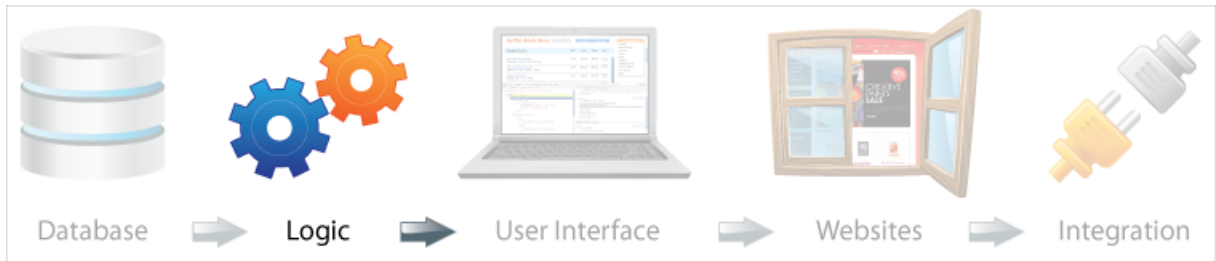
**Siehe auch:**

*[Lernprogramm 5: Hinzufügen von Geschäftslogik mit Apex](#)*

*[Schritt 4: Testen des Auslösers](#)*

# Lernprogramm 6: Hinzufügen von Tests zur Anwendung

**Niveau:** Fortgeschritten; **Dauer:** 20–30 Minuten



Das Testen stellt einen der wichtigsten Schritte beim Entwickeln einer Anwendung dar. Durch Tests wird sichergestellt, dass der Code wie erwartet ausgeführt wird und Ressourcen nicht unnötigerweise in Anspruch genommen werden. Von Tests profitieren nicht nur Sie (Sie können mehr Vertrauen in die Anwendungen setzen, die Sie schreiben), sondern auch die Plattform. Daher wird für die Plattform vorausgesetzt, dass Sie alle Apex-Codes testen, bevor diese in einer Produktionsumgebung bereitgestellt werden können. Tests werden auch vor dem Veröffentlichen von neuen Versionen der Plattform ausgeführt, um sicherzustellen, dass keine Probleme bezüglich der Rückwärtskompatibilität eingeführt werden.

Um die Tests zu vereinfachen, unterstützt Apex *Einheitentests*, die das Codeverhalten und das erwartete Ergebnis programmatisch validieren. Alle Apex-Codes weisen Einschränkungen auf, die bestimmen, wie viele Ressourcen verwendet werden können. Beispielsweise gibt es eine festgelegte Begrenzung der Anzahl an Abfragen, die in einem Auslöser durchgeführt werden können. Qualitativ hochwertige Tests stellen sicher, dass sich der Code nicht nur wie erwartet verhält, sondern dass er auch diese Einschränkungen nicht überschreitet. In diesem Lernprogramm schreiben Sie einen Einheitentest und führen diesen aus, der den eben erstellten Apex-Auslöser ausführt.

## Voraussetzungen

### Apex-Lernprogramm

Zunächst müssen Sie einen Apex-Auslöser wie in [Lernprogramm 5: Hinzufügen von Geschäftslogik mit Apex](#) auf Seite 43 beschrieben erstellen.

### Softwareanforderungen

Falls Sie möchten, können Sie für dieses Lernprogramm die Force.com-IDE verwenden. Wenn Sie dies tun möchten, benötigen Sie Eclipse 3.3 oder 3.4 und das Force.com-IDE-Plugin: [wiki.developerforce.com/index.php/Force.com\\_IDE](http://wiki.developerforce.com/index.php/Force.com_IDE). Informationen zum Erstellen eines Projekts finden Sie unter [Erstellen eines Projekts in der Force.com-IDE](#) auf Seite 93.

## Siehe auch:

- [Schritt 1: Erstellen einer Apex-Testklasse](#)
- [Schritt 2: Hinzufügen von Testmethoden zur Klasse](#)
- [Schritt 3: Schreiben von Code zum Ausführen des Auslösers](#)
- [Schritt 4: Ausführen des Tests](#)
- [Schritt 5: Anzeigen der Codeabdeckung und Verbessern der Tests](#)
- [Zusammenfassung](#)

## Schritt 1: Erstellen einer Apex-Testklasse

Alle Einheitentests sind in Apex-Klassen enthalten. In diesem Schritt erstellen Sie die Klasse, die die Einheitentests enthält.

So erstellen Sie eine neue Apex-Klasse auf der Weboberfläche:

1. Klicken Sie auf **Setup** ► **Entwickeln** ► **Apex-Klassen**.
2. Klicken Sie auf **Neu**.
3. Geben Sie im Editorfenster den folgenden Code ein.

```
@isTest private class TestHandleProductPriceChange { }
```

4. Klicken Sie auf **Schnellspeicherung**, um Ihre Eingabe zu speichern und die Bearbeitung fortzusetzen.

So erstellen Sie eine neue Apex-Klasse in der Force.com-IDE:

1. Klicken Sie in der IDE mit der rechten Maustaste auf den Projektordner. Klicken Sie dann auf **Neue** ► **Apex-Klasse**.
2. Geben Sie auf der Seite "Apex-Klasse erstellen" als Name `TestHandleProductPriceChange` ein.
3. Wählen Sie im Feld `Template` die Option `Test Class` aus.
4. Klicken Sie zum Erstellen der Klasse auf **Fertig stellen**.

### Weitere Informationen

Durch die Bemerkung `@isTest` wird Force.com angewiesen, dass jeglicher Code in der Apex-Klasse Code ist, der den restlichen Code testet. Nun erstellen Sie Testmethoden in dieser Klasse, die die eigentlichen Tests durchführen.

### Siehe auch:

- [Lernprogramm 6: Hinzufügen von Tests zur Anwendung](#)
- [Schritt 2: Hinzufügen von Testmethoden zur Klasse](#)

## Schritt 2: Hinzufügen von Testmethoden zur Klasse

Nun können Sie eine Methode zur Klasse hinzufügen, die den eigentlichen Testvorgang durchführt. Der erstellte Auslöser, der getestet werden soll, wird nur ausgeführt, wenn Datensätze aktualisiert werden. Als Erstes müssen Sie also Testdatensätze in der Datenbank anlegen. Die Methode soll Rechnungsanweisungen sowie Merchandise- und Belegpostendatensätze erstellen und diese zur Datenbank hinzufügen.

1. Beginnen Sie mit dem Erstellen der Testmethode, die die drei Testverfahren enthalten soll. Geben Sie in den geschweiften Klammern den folgenden Code ein.

```
static testMethod void testPriceChange() {
}
```



**Anmerkung:** Ersetzen Sie den generierten Kommentar sowie Code, wenn Sie die IDE verwenden.

2. Fügen Sie in den geschweiften Klammern den folgenden Code hinzu, mit dem eine Rechnungsanweisung erstellt und in der Datenbank eingefügt wird.

```
Invoice_Statement__c invoice = new Invoice_Statement__c(Status__c = 'Negotiating'); insert invoice;
```

3. Fügen Sie direkt unter `insert invoice` den Code zum Erstellen neuer Merchandise-Datensätze hinzu.

```
Merchandise__c[] products = new Merchandise__c[] { new Merchandise__c(Name = 'item 1', Description__c = 'test product 1', Price__c = 10, Total_Inventory__c = 10), new Merchandise__c(Name = 'item 2', Description__c = 'test product 2', Price__c = 11, Total_Inventory__c = 10) }; insert products;
```

4. Geben Sie dann direkt unter `insert products` den Code zum Hinzufügen von Belegposten ein.

```
Line_Item__c[] lineItems = new Line_Item__c[] { new Line_Item__c(Invoice_Statement__c = invoice.id, Merchandise__c = products[0].id, Unit_Price__c = 10, Units_Sold__c = 3), new Line_Item__c(Invoice_Statement__c = invoice.id, Merchandise__c = products[1].id, Unit_Price__c = 11, Units_Sold__c = 6) }; insert lineItems;
```

Nun haben Sie die Daten eingerichtet. Beachten Sie, dass die erstellte Rechnungsanweisung den Status "Negotiating" aufweist, wobei es sich um die Bedingung handelt, die zum Auslösen des Auslösers erforderlich ist. Jetzt müssen Sie den Code schreiben, der den anderen Bedingungen zum Auslösen des Auslösers entspricht: Sie müssen den Preis eines Merchandise-Artikels senken. Dieser Vorgang wird im nächsten Schritt durchgeführt.

### Siehe auch:

[Lernprogramm 6: Hinzufügen von Tests zur Anwendung](#)

[Schritt 1: Erstellen einer Apex-Testklasse](#)

[Schritt 3: Schreiben von Code zum Ausführen des Auslösers](#)

## Schritt 3: Schreiben von Code zum Ausführen des Auslösers

Der Code in der Testmethode weist bislang noch kaum eine Funktion auf: Der Auslöser wird dadurch nicht ausgelöst, da dieser Vorgang nur durchgeführt wird, wenn der Preis eines Merchandise-Datensatzes aktualisiert wird.

1. Fügen Sie nach `insert lineItems;` die folgenden Zeilen hinzu.

```
products[0].price__c = 20; Test.startTest(); update products; Test.stopTest();
```



**Anmerkung:** Mit diesem Code wird der Preis eines der Merchandise-Datensätze geändert, indem er angehoben wird. Daraufhin wird `startTest()` aufgerufen, wobei es sich um einen Bestandteil der Teststruktur handelt, der den Punkt im Testcode markiert, an dem der Test tatsächlich beginnt. Das System betrachtet den Setupcode und Datenbankvorgänge davor nicht als Bestandteil des Tests. Durch das Testen wird des Weiteren sichergestellt, dass der Code gegen keine regulierenden Einschränkungen verstößt, mit denen die Ressourcen beschränkt werden, die vom Apex-Code verwendet werden können. Durch Angeben der Stelle, an der der eigentliche Test beginnt und endet, rechnet das System den Setupcode nicht zu den regulierenden Einschränkungen hinzu. Dadurch ist sichergestellt, dass der Test genauer wiedergibt, wie sich der Code in der Praxis auswirkt.

- Nach dem Aktualisieren der Produkte müssen Sie nun prüfen, ob sich der Preis in den Belegposten geändert hat. Idealerweise sollte dies nicht der Fall sein, da der Preis im Setupcode angehoben wurde. Fügen Sie nach `Test.stopTest()`; den folgenden Code hinzu.

```
lineItems = [SELECT id, unit_price__c FROM Line_Item__c WHERE id IN :lineItems];
system.assert(lineItems[0].unit_price__c == 10);
```

- Stellen Sie sicher, dass Ihr Code dem folgenden Beispiel entspricht, und klicken Sie dann auf **Speichern**.

```
@isTest private class TestHandleProductPriceChange {
    static testMethod void testPriceChange() { Invoice_Statement__c invoice = new
    Invoice_Statement__c (Status__c = 'Negotiating'); insert invoice;

        Merchandise__c[] products = new Merchandise__c[] { new Merchandise__c (Name =
    'item 1', Description__c = 'test product 1', Price__c = 10, Total_Inventory__c = 10),
    new Merchandise__c (Name = 'item 2', Description__c = 'test product 2', Price__c = 11,
    Total_Inventory__c = 10) }; insert products; Line_Item__c[] lineItems = new Line_Item__c[]
    { new Line_Item__c (Invoice_Statement__c = invoice.id, Merchandise__c = products[0].id,
    Unit_Price__c = 10, Units_Sold__c = 3), new Line_Item__c (Invoice_Statement__c =
    invoice.id, Merchandise__c = products[1].id, Unit_Price__c = 11, Units_Sold__c = 6) };
    insert lineItems;

        products[0].price__c = 20; Test.startTest(); update products; Test.stopTest();

        lineItems = [SELECT id, unit_price__c FROM Line_Item__c WHERE id IN :lineItems];
    system.assert(lineItems[0].unit_price__c == 10); } }
```

### Siehe auch:

[Lernprogramm 6: Hinzufügen von Tests zur Anwendung](#)

[Schritt 2: Hinzufügen von Testmethoden zur Klasse](#)

[Schritt 4: Ausführen des Tests](#)

## Schritt 4: Ausführen des Tests

Force.com weist eine Teststruktur auf, mit der Sie Tests ausführen und auch die Codeabdeckung prüfen können. Nun führen Sie die Tests aus und analysieren die sich daraus ergebende Codeabdeckung.



**Anmerkung:** In diesen Schritten wird erläutert, wie Tests auf der Weboberfläche ausgeführt werden. Sie können Tests allerdings auch in der Force.com-IDE ausführen, indem Sie im Package Explorer mit der rechten Maustaste auf den Klassennamen klicken und die Option **Force.com > -Tests ausführen** auswählen. Im restlichen Teil dieses Lernprogramms wird die Weboberfläche verwendet, Sie sollten aber dennoch in der Lage sein, die Schritte ganz mühelos mit der IDE nachzuvollziehen.

- Navigieren Sie zu Ihrer Testklasse, indem Sie auf **Setup > Entwickeln > Apex-Klassen** und dann auf Ihre Testklasse `TestHandleProductsPriceChange` klicken.
- Klicken Sie auf **Test ausführen** und Sie sehen ein Ergebnis, das in etwa dem folgenden entspricht.

Apex Test Result		Help for this Page ?
<b>Summary</b>		
Test Class	TestHandleProductPriceChange	
Tests Run	1	
Test Failures	0	
Code Coverage Total %	80	
Total Time (ms)	1384.0	
<b>Test Successes</b>		
Method Name	Total Time (ms)	
TestHandleProductPriceChange.testPriceChange	1384.0	
<b>Code Coverage</b>		
Trigger Code Coverage		
Trigger Name	Coverage %	
HandleProductPriceChange	80	
<b>Debug Log</b>		
*** Beginning Test 1: TestHandleProductPriceChange.static testMethod void testPriceChange()		
20091023171640.243:Class.TestHandleProductPriceChange.testPriceChange: line 6, column 10: Insert: SUBJECT:Invoice_Statement__c		
20091023171640.243:Class.TestHandleProductPriceChange.testPriceChange: line 6, column 10: DML Operation executed in		

Aus diesem Ergebnis lassen sich einige wichtige Dinge herauslesen.

- Es gibt an, ob die Tests erfolgreich waren oder nicht. Wenn die boolesche Bedingung in der Anweisung `system.assert` fehlschlägt, die Sie zum Test hinzugefügt haben, würde dieses Ergebnis hier angezeigt werden. Das Hinzufügen vieler Vermögenswerte ist eine hervorragende Möglichkeit, das erwartete Verhalten des Codes zu testen.
- Dadurch sind Details zum Ausführen des Tests verfügbar. Durch die Analyse des Debug-Protokolls erfahren Sie beispielsweise, dass die Validierungsregel `Order_in_Stock` ausgelöst wurde. Darüber hinaus erfahren Sie, welche Datensätze erstellt wurden, wie viele Abfragen ausgeführt wurden und noch vieles mehr.
- Die Codeabdeckung wird angegeben, also wie viele Zeilen des Codes in einer anderen Klasse oder einem anderen Auslöser ausgeführt wurden.

Beachten Sie des Weiteren, dass die Ergebnisseite anzeigt, dass eine 80%ige Abdeckung des Auslösers `HandleProductPriceChange` erzielt wurde. Diese Codeabdeckung reicht für eine Bereitstellung aus, allerdings wäre eine 100%ige Abdeckung wünschenswert. Im nächsten Schritt wird die Codeabdeckung analysiert, um zu ermitteln, an welchen Stellen weitere Testfälle hinzugefügt werden müssen, um eine 100%ige Abdeckung zu erzielen.

### Siehe auch:

[Lernprogramm 6: Hinzufügen von Tests zur Anwendung](#)

[Schritt 3: Schreiben von Code zum Ausführen des Auslösers](#)

[Schritt 5: Anzeigen der Codeabdeckung und Verbessern der Tests](#)

## Schritt 5: Anzeigen der Codeabdeckung und Verbessern der Tests

Sie haben zwei Codesätze geschrieben. Zum einen den Auslöser, der als *Produktionscode* bezeichnet wird, und den Code in der Testklasse, der als *Testcode* bezeichnet wird. Der Begriff *Codeabdeckung* bezieht sich darauf, wie groß der Teil des Produktionscodes ist, der vom Testcode abgedeckt wird. In anderen Worten: Wenn eine Person den Testcode ausführt, wird dadurch der gesamte oder nur ein Teil des Produktionscodes ausgeführt? Wenn nur ein Teil des Codes ausgeführt wird, kann dies bedeuten, dass der Produktionscode Fehler im nicht getesteten Teil des Codes enthält. Dieses Konzept kann noch anschaulicher gemacht werden, indem man eine grafische Darstellung der Codeabdeckung verwendet.



**Anmerkung:** Falls Sie für dieses Lernprogramm die Force.com-IDE verwendet haben, so verwenden Sie sie auch weiterhin. Wenn Sie sowohl in der IDE als auch auf der Weboberfläche gleichzeitig entwickeln, müssen Sie jedes Mal eine Synchronisierung mit dem Server durchführen, wenn Sie zwischen den Tools wechseln. Synchronisierungen können sehr leicht durchgeführt werden (klicken Sie mit der rechten Maustaste auf die Klasse und wählen Sie die Option **Force.com ► Mit Server synchronisieren** aus), noch einfacher ist es jedoch, diese gänzlich zu vermeiden!

1. Klicken Sie im Abschnitt zur Codeabdeckung auf **80**.

Code Coverage	
▼ Trigger Code Coverage	
Trigger Name	Coverage %
HandleProductPriceChange	80

2. Die Seite "Codeabdeckung" wird geöffnet. Mit der blauen Markierung werden Zeilen des Codes hervorgehoben, die infolge der Testmethode *abgedeckt* (ausgeführt) wurden. Mit der roten Markierung werden Zeilen hervorgehoben, die *nicht* ausgeführt wurden. In diesem Fall wurde Zeile 12 nicht ausgeführt, da der Preis eines Merchandise-Artikels nicht gesenkt, sondern angehoben wurde.

Code Coverage	
HandleProductPriceChange (Code Covered: 80%)	
line	executions source
1	trigger HandleProductPriceChange on Merchandise__c (after update) {
2	
3	1 List<Line_Item__c> openLineItems =
4	{SELECT j.Unit_Price__c, j.Merchandise__r.Price__c
5	FROM Line_Item__c j
6	WHERE j.Invoice_Statement__r.Status__c = 'Negotiating'
7	AND j.Merchandise__r.id IN :Trigger.new
8	FOR UPDATE};
9	
10	1 for (Line_Item__c li: openLineItems) {
11	2 if ( li.Merchandise__r.Price__c < li.Unit_Price__c ){
12	0 li.Unit_Price__c = li.Merchandise__r.Price__c;
13	}
14	}
15	1 update openLineItems;
16	}

3. Diese rote Markierung weist sehr anschaulich darauf hin, dass der Test noch nicht abgeschlossen ist. Bearbeiten Sie also die Testmethode, um die Codeabdeckung zu verbessern. Schließen Sie das Fenster "Codeabdeckung" und klicken Sie dann auf **Setup ► Entwickeln ► Apex-Klassen** und wählen Sie die Testklasse **TestHandleProductPriceChange** aus.
4. Klicken Sie auf **Bearbeiten**.
5. Ersetzen Sie den Code ab `products[0]` bis zur geschweiften Klammer mit folgendem Code.

```
products[0].price__c = 20; // raise price products[1].price__c = 5; // lower price
Test.startTest(); update products; Test.stopTest();

lineItems = [SELECT id, unit_price__c FROM Line_Item__c WHERE id IN :lineItems];
System.assert(lineItems[0].unit_price__c == 10); // unchanged
System.assert(lineItems[1].unit_price__c == 5); // changed!!
```

6. Klicken Sie auf **Speichern**.
7. Klicken Sie nun nochmals auf **Test ausführen** und Sie sehen, dass die Codeabdeckung nun 100 % beträgt.

Summary	
Test Class	TestHandleProductPriceChange
Tests Run	1
Test Failures	0
Code Coverage Total %	100
Total Time (ms)	929.0

**Siehe auch:**

[Lernprogramm 6: Hinzufügen von Tests zur Anwendung](#)

[Schritt 4: Ausführen des Tests](#)

[Zusammenfassung](#)

**Zusammenfassung**

In diesem Lernprogramm haben Sie Tests für den Apex-Auslöser erstellt und erfahren, wie Sie die integrierten Testtools dabei unterstützen können, eine Codeabdeckung von 100 % zu erzielen. Das Erstellen von Einheitentests beim Entwickeln ist für die Bereitstellung erforderlich und stellt darüber hinaus auf lange Sicht den Schlüssel zu erfolgreichen Entwicklungen dar.

Es ist jedoch wichtig zu beachten, dass die Codeabdeckung nicht das einzige Ziel von Tests ist. Im Idealfall sollten Sie auch alle möglichen Szenarien testen und der eben geschriebene Testfall deckt einen möglichen Fall nicht ab. Derzeit werden nur Rechnungsanweisungen geprüft, die den Status "Negotiating" aufweisen. Es sollten jedoch auch andere Statuswerte geprüft werden. Wenn eine Rechnungsanweisung den Status "Negotiating" z. B. *nicht* aufweist, sollte das Anheben oder Senken des Preises keine Auswirkungen haben.

**Siehe auch:**

[Lernprogramm 6: Hinzufügen von Tests zur Anwendung](#)

[Schritt 5: Anzeigen der Codeabdeckung und Verbessern der Tests](#)

# Lernprogramm 7: Erstellen einer benutzerdefinierten Benutzeroberfläche mit Visualforce

**Niveau:** Fortgeschrittene Anfänger; **Dauer:** 30-45 Minuten



Visualforce ist eine komponentenbasierte Benutzeroberflächenstruktur für die Force.com-Plattform. In den vorhergehenden Lernprogrammen haben Sie die Anwendung durch eine automatisch generierte Benutzeroberfläche erstellt und erweitert. Mit Visualforce können Sie die Benutzeroberfläche in weit größerem Maße kontrollieren, da Sie eine Anzeigestruktur erhalten, die eine tagbasierte Markupsprache ähnlich HTML, eine Bibliothek wiederverwendbarer sowie erweiterbarer Komponenten und ein Apex-basiertes Steuerelementmodell umfasst. Visualforce unterstützt den Designstil Model-View-Controller (MVC) für Benutzeroberflächen und ist hochflexibel.

In diesem Lernprogramm verwenden Sie Visualforce, um eine neue Oberfläche für die Warenlageranwendung zu erstellen, mit der eine Bestandszählerseite angezeigt wird, die das Aufführen des gesamten Merchandise-Bestands sowie das Aktualisieren der einzelnen Stückzahlen ermöglicht. Zweck der Zählerseite ist es, das Computersystem mit der physischen Anzahl der Merchandise-Artikel zu aktualisieren, falls diese abweichen.

## Voraussetzungen

### Grundlegendes Wissen

Bei diesem Lernprogramm ist es hilfreich, über grundlegendes Wissen zu Markupsprachen wie HTML und XML zu verfügen, dies ist aber nicht zwingend erforderlich.

### Siehe auch:

- [Schritt 1: Aktivieren des Visualforce-Entwicklungsmodus](#)
- [Schritt 2: Erstellen einer Visualforce-Seite](#)
- [Schritt 3: Hinzufügen einer statischen Stylesheet-Ressource](#)
- [Schritt 4: Hinzufügen eines Steuerelements zur Seite](#)
- [Schritt 5: Anzeigen der Bestandszählerseite als Visualforce-Seite](#)
- [Zusammenfassung](#)

## Schritt 1: Aktivieren des Visualforce-Entwicklungsmodus

Mit dem Entwicklungsmodus wird ein Visualforce-Seiteneditor im Browser eingebettet. Damit kann Code angezeigt und gleichzeitig eine Vorschau der Seite erstellt werden. Außerdem wird mit dem Entwicklungsmodus ein Apex-Editor zum Bearbeiten von Steuerelementen und Erweiterungen hinzugefügt.

1. Klicken Sie auf **Setup** ► **Meine persönlichen Daten** ► **Persönliche Daten**.

2. Klicken Sie auf **Bearbeiten**.
3. Aktivieren Sie das Kontrollkästchen Entwicklungsmodus und klicken Sie auf **Speichern**.

User Detail		Edit	Change Password
Name	Admin User	Role	
Alias	AUser	User License	Salesforce
Email	mkoft@salesforce.com	Profile	System Administrator
Username	workbook2009@de.com	Active	<input checked="" type="checkbox"/>
Community Nickname	workbook20091.2547864026304688E12	Marketing User	<input checked="" type="checkbox"/>
Delegated Approver		Offline User	<input checked="" type="checkbox"/>
Manager		Mobile User	<input checked="" type="checkbox"/>
Title		Mobile Configuration	
Company	Darkforce	Accessibility Mode	<input type="checkbox"/>
Department		Color-Blind Palette on Charts	<input type="checkbox"/>
Division		Send Apex Warning Emails	<input type="checkbox"/>
Address	1 Market St San Francisco, CA 94105 US	Salesforce CRM Content User	<input type="checkbox"/>
Time Zone	(GMT-08:00) Pacific Daylight Time (America/Los_Angeles)	Receive Salesforce CRM Content Email Alerts	<input checked="" type="checkbox"/>
Locale	English (United States)	Receive Salesforce CRM Content Alerts as Daily Digest	<input checked="" type="checkbox"/>
Language	English	Development Mode	<input checked="" type="checkbox"/>
Newsletter	<input checked="" type="checkbox"/>	Allow Forecasting	<input checked="" type="checkbox"/>
Admin Newsletter	<input checked="" type="checkbox"/>	Call Center	

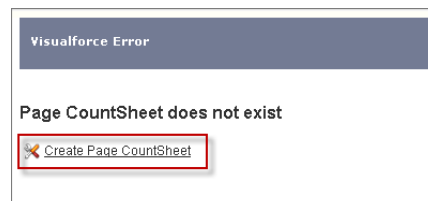
### Siehe auch:

[Lernprogramm 7: Erstellen einer benutzerdefinierten Benutzeroberfläche mit Visualforce](#)  
[Schritt 2: Erstellen einer Visualforce-Seite](#)

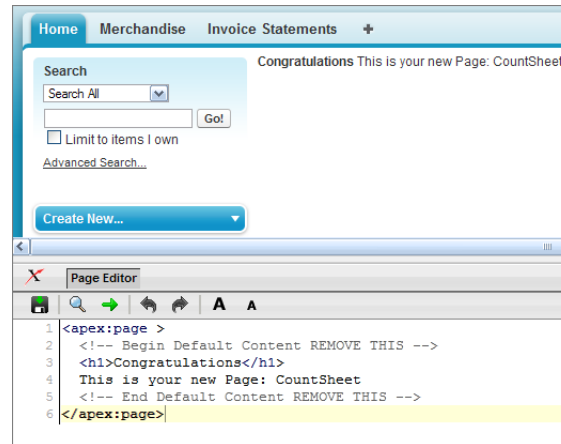
## Schritt 2: Erstellen einer Visualforce-Seite

In diesem Schritt erstellen Sie eine Visualforce-Seite, die als Bestandszählerseite fungiert.

1. Fügen Sie im Browser den Text `/apex/CountSheet` zum URL der Salesforce.com-Instanz hinzu. Wenn Ihre Salesforce.com-Instanz beispielsweise `https://na1.salesforce.com` lautet, wäre der neue URL `https://na1.salesforce.com/apex/CountSheet`. Es wird eine Fehlermeldung angezeigt: "Seite 'CountSheet' ist nicht vorhanden"



2. Klicken Sie auf den Link **Seite 'CountSheet' erstellen**, um die neue Seite zu erstellen.
3. Klicken Sie unten links auf der Seite auf den Link "Seiteneditor". Die Registerkarte "Seiteneditor" zeigt den Code und eine Vorschau der neuen Seite an, auf der Standardtext enthalten ist. Die Anzeige sollte wie folgt aussehen.



- Natürlich soll die Überschrift der Seite nicht "Herzlichen Glückwunsch" lauten, also ändern Sie den Inhalt des Tags `<h1>` in `Inventory Count Sheet`. Fahren Sie dann mit dem Entfernen der Kommentare fort. Der Code der Seite sollte nun wie folgt aussehen.

```
<apex:page>
<h1>Inventory Count Sheet</h1>
</apex:page>
```

- Klicken Sie oben im Seiteneditor auf **Speichern**. Die Seite wird neu geladen und Ihre Änderungen werden angezeigt.

### Weitere Informationen

Beachten Sie, dass der Code der Seite stark standardmäßigem HTML ähnelt. Dies liegt daran, dass Visualforce-Seiten HTML-Tags wie `<h1>` und Visualforce-spezifische Tags verwenden, die mit `<apex:>` beginnen.

### Siehe auch:

- [Lernprogramm 7: Erstellen einer benutzerdefinierten Benutzeroberfläche mit Visualforce](#)
- [Schritt 1: Aktivieren des Visualforce-Entwicklungsmodus](#)
- [Schritt 3: Hinzufügen einer statischen Stylesheet-Ressource](#)

## Schritt 3: Hinzufügen einer statischen Stylesheet-Ressource

Sie möchten natürlich, dass die Warenlageranwendung professionell aussieht, also verwenden Sie ein benutzerdefiniertes Stylesheet (eine CSS-Datei), um Farbe, Schriftart und Anordnung des Texts auf der Seite anzugeben. Bei den meisten Webseiten und von einer Vielzahl von Webdesignern wird für diesen Zweck die Standardinternettechnologie CSS eingesetzt, welche wir für Sie erstellt haben. Damit Ihre Seiten auf ein Stylesheet verweisen können, müssen Sie dieses als *statische Ressource* hochladen. Eine statische Ressource ist eine Datei oder Dateisammlung, die auf Force.com gespeichert wird. Sobald das Stylesheet als statische Ressource hinzugefügt wurde, kann von allen Ihren Visualforce-Seiten aus auf dieses verwiesen werden.

So fügen Sie ein Stylesheet als statische Ressource hinzu:

- Wechseln Sie in Ihrem Browser zu [developer.force.com/workbook/styles](https://developer.force.com/workbook/styles). Laden Sie die Datei herunter und speichern Sie sie auf Ihrem Desktop.
- Klicken Sie in der Anwendung auf **Setup** ► **Entwickeln** ► **Statische Ressourcen** und dann auf **Neu**.

3. Geben Sie im Feld Name den Wert `styles` ein.
4. Klicken Sie auf **Durchsuchen** und suchen Sie nach der heruntergeladenen Datei `styles.zip`.
5. Wählen Sie in der Auswahlliste Cache Control die Option `Public` aus.

6. Klicken Sie auf **Speichern**.



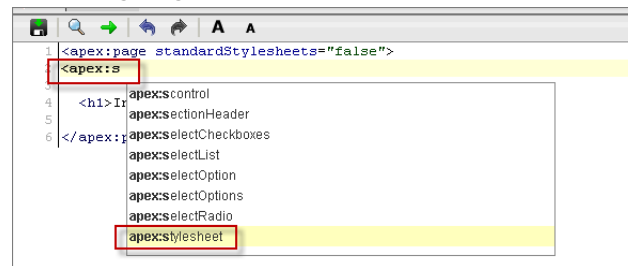
**Anmerkung:** Wenn die Visualforce-Seiten auf einer öffentlichen Website veröffentlicht werden, wird von Force.com ein globales Content Delivery Network aus Cacheservern genutzt, um Kopien der statischen Dateien zu speichern.

Nun müssen Sie die Visualforce-Seite so bearbeiten, dass auf das Stylesheet verwiesen wird.

1. Fügen Sie wie beim Erstellen der Seite den Text `/apex/CountSheet` zum URL der Salesforce.com-Instanz hinzu.
2. Bearbeiten Sie die Attribute des Tags `<apex:page>` und geben Sie den folgenden Code ein, um das standardmäßige Stylesheet sowie die Kopfzeile und die Randleiste zu entfernen.

```
<apex:page standardStylesheets="false" showHeader="false" sidebar="false">
```

3. Nun müssen Sie auf der Seite angeben, wo sich das Stylesheet befindet. Sie fügen also nach dem ersten Tag `<apex:page>` eine neue Zeile hinzu und geben `<apex:` ein.
4. Der Editor weist automatische Vervollständigung von Code auf, sodass eine Dropdown-Liste der Elemente angezeigt wird, die in diesem Kontext verfügbar sind. Beginnen Sie mit der Eingabe von `stylesheet` und wählen Sie die Option `apex:stylesheet` aus, sobald diese angezeigt wird.



5. Geben Sie nun wie gezeigt den Speicherort des Stylesheet an.

```
<apex:stylesheet value="{!URLFOR($Resource.styles, 'styles.css') }"/>
```

6. Prüfen Sie, ob der Code wie folgt aussieht:

```
<apex:page standardStylesheets="false" showHeader="false" sidebar="false">
<apex:stylesheet value="{!URLFOR($Resource.styles, 'styles.css') }"/>
<h1>Inventory Count Sheet</h1>
</apex:page>
```

7. Klicken Sie oben im Seiteneditor auf **Speichern**.

Beachten Sie, dass die angezeigte Seite nun stark verändert wurde. Der Titel wird an anderer Stelle in einem anderen Schrifttyp angezeigt und standardmäßige Kopfzeile sowie Randleiste werden nicht mehr angezeigt.

### Weitere Informationen

Gehen wir noch etwas genauer auf den Stylesheet-Code ein.

- `$Resources` ist eine globale Variable, auf die Visualforce zugreifen kann. Mit `$Resource.styles` verweisen Sie auf die Ressource "styles", die Sie zuvor erstellt haben.
- Mit der Funktion `URLFOR()` wird der Speicherort der statischen Ressource sowie einer Datei in der Ressource angegeben. Des Weiteren wird der URL berechnet, der auf der finalen Seite generiert werden sollte. Falls Ihnen die Syntax bekannt vorkommt, liegt das daran, dass diese beim dynamischen Evaluieren von Werten bereits verwendet wurde, als die Visualforce-Seite dargestellt wurde.

### Siehe auch:

[Lernprogramm 7: Erstellen einer benutzerdefinierten Benutzeroberfläche mit Visualforce](#)

[Schritt 2: Erstellen einer Visualforce-Seite](#)

[Schritt 4: Hinzufügen eines Steuerelements zur Seite](#)

## Schritt 4: Hinzufügen eines Steuerelements zur Seite

Dank des Designstils Model-View-Controller (MVC) von Visualforce ist es ein Leichtes, die Anzeige und deren Stil von der zugrunde liegenden Datenbank und Logik zu trennen. In MVC interagiert die Ansicht (die Visualforce-Seite) mit einem Steuerelement. In unserem Fall handelt es sich beim Steuerelement in der Regel um eine Apex-Klasse, die einige Funktionalität auf der Seite verfügbar macht. Das Steuerelement kann etwa die Logik enthalten, die beim Klicken auf eine Schaltfläche ausgeführt werden soll. Steuerelemente interagieren in der Regel darüber hinaus mit dem Modell (der Datenbank) und machen dabei die Daten verfügbar, die in der Ansicht angezeigt werden sollen.

Alle Force.com-Objekte weisen standardmäßig Standardsteuerelemente auf, die zum Interagieren mit den Daten verwendet werden können, die mit dem Objekt verknüpft sind. In vielen Fällen müssen Sie den Code für das Steuerelement also nicht selbst schreiben. Sie können die Standardsteuerelemente erweitern, um neue Funktionen hinzuzufügen, oder von Grund auf benutzerdefinierte Steuerelemente erstellen. In diesem Lernprogramm verwenden Sie das Standardsteuerelement.

1. Falls der Seiteneditor auf der Visualforce-Seite nicht geöffnet ist, klicken Sie auf **Seiteneditor**, um die Seite zu bearbeiten.
2. Bearbeiten Sie den Code so, dass das Standardsteuerelement "Merchandise\_\_c" aktiviert wird, indem Sie zunächst das Tag `<apex:page>` bearbeiten. Leerzeichen werden vom Editor nicht beachtet, sodass Sie den Text in einer neuen Zeile eingeben können.

```
<apex:page standardStylesheets="false" showHeader="false" sidebar="false"
standardController="Merchandise__c">
```

3. Fügen Sie als Nächstes die Definition für das Standardlistensteuerelement hinzu.

```
<apex:page standardStylesheets="false" showHeader="false" sidebar="false"
standardController="Merchandise__c" recordSetVar="products">
```

4. Klicken Sie oben im Seiteneditor auf **Speichern**. Auf der Seite werden keine Änderungen angezeigt. Da Sie jedoch angegeben haben, dass auf der Seite ein Steuerelement verwendet werden soll, und die Variable `products` definiert haben, ist die Variable im Hauptteil der Seite verfügbar und stellt eine Liste der Merchandise-Datensätze dar.

## Weitere Informationen

Das Attribut `recordSetVar` aktiviert ein Standardlistensteuerelement, das weitere Steuerelementunterstützung zum Aufführen einiger Datensätze mit Paginierung bietet. Dessen mit "products" festgelegter Wert stellt sicher, dass die neue Variable `products` eine Reihe Datensätze enthält, die auf der Visualforce-Seite angezeigt werden.

## Siehe auch:

[Lernprogramm 7: Erstellen einer benutzerdefinierten Benutzeroberfläche mit Visualforce](#)

[Schritt 3: Hinzufügen einer statischen Stylesheet-Ressource](#)

[Schritt 5: Anzeigen der Bestandszählerseite als Visualforce-Seite](#)

## Schritt 5: Anzeigen der Bestandszählerseite als Visualforce-Seite

Nun ist die gesamte Funktionalität vorhanden, um die Visualforce-Seite auszuarbeiten. Es wird eine Tabelle aller Merchandise-Datensätze zusammen mit jeweils einem Eingabefeld angezeigt, sodass die Anzahl des Bestands aktualisiert werden kann.

1. Geben Sie in der Zeile unterhalb des Tags `</h1>` den Wert `<apex:f` ein und heben Sie `<apex:form>` hervor, wenn diese Option in der Dropdown-Liste angezeigt wird. Mit dem Formular können Aktualisierungen auf der Seite vorgenommen werden.
2. Drücken Sie die Eingabetaste und beachten Sie, dass die öffnenden und schließenden Tags vom System erstellt werden.
3. Setzen Sie den Cursor zwischen die Tags und erstellen Sie eine Datentabelle. Geben Sie `<apex:d` ein und drücken Sie die Eingabetaste, um in der Dropdown-Liste `dataTable` auszuwählen.
4. Zu diesem Zeitpunkt müssen Sie Attribute zum Tag `dataTable` hinzufügen. Das Wertattribut gibt an, welche Liste mit Elementen die Komponente `dataTable` durchgehen soll. Das Attribut `var` weist jedes Element der Liste für jede einzelne Iteration der Variablen `pitem` zu. Das Attribut `rowClasses` weist CSS-Stilbezeichnungen sich abwechselnden Zeilen zu. Geben Sie in mindestens einer Zeile im Tag Folgendes ein.

```
<apex:dataTable value="{!products}" var="pitem" rowClasses="odd,even">
```

5. Jetzt definieren Sie jede einzelne Spalte und bestimmen, von wo die Daten abgerufen werden, indem das entsprechende Feld in der Variablen `pitem` nachgeschlagen wird. Geben Sie folgenden Code zwischen den öffnenden und schließenden `dataTable`-Tags ein.

```
<apex:dataTable value="{!products}" var="pitem" rowClasses="odd,even"> <apex:column
headerValue="Product"> <apex:outputText value="{!pitem.name}"/> </apex:column>
</apex:dataTable>
```

6. Klicken Sie auf **Speichern** und die Tabelle wird angezeigt.

Inventory Count Sheet	
Product	
Wee Jet	

Das Attribut `headerValue` stellt lediglich einen Kopfzeilentitel für die Spalte bereit und darunter wird eine Liste mit Zeilen angezeigt, je eine für jeden Merchandise-Datensatz. Der Ausdruck `{!pitem.name}` gibt an, dass das Namensfeld der aktuellen Zeile angezeigt werden soll.

7. Fügen Sie nunmehr nach dem schließenden Tag der ersten Spalte zwei weitere Spalten hinzu.

```
<apex:column headerValue="Inventory"> <apex:outputField
value="{!pitem.Total_Inventory__c}"/> </apex:column> <apex:column headerValue="Physical
Count"> <apex:inputField value="{!pitem.Total_Inventory__c}"/> </apex:column>
```



**Anmerkung:** Bei der zweiten Spalte handelt es sich um ein `inputField`-Feld, kein `outputField`-Feld. Das Feld `inputField` zeigt einen Wert an, welcher jedoch auch geändert werden kann.

8. Klicken Sie auf **Speichern** und Sie erhalten eine Bestandszählerseite! Diese führt alle Merchandise-Datensätze auf, zeigt den aktuellen Bestand an und stellt ein Eingabefeld für die physische Anzahl bereit.
9. Fügen Sie als abschließendes Detail eine Schaltfläche hinzu, mit der die physische Anzahl einer jeden Zeile bearbeitet und die Werte auf der Seite aktualisiert werden können. Geben Sie hierfür direkt über der Zeile `</apex:form>` folgenden Code ein.

```
<br/> <apex:commandButton action="{!quicksave}" value="Update Counts" />
```

Inventory Count Sheet

Product	Inventory	Physical Count
Wee Jet	400	<input type="text" value="400"/>

### Weitere Informationen

- Die Komponente `dataTable` erstellt eine Tabelle mit Zeilen, die jeweils durch Iteration einer Liste abgerufen wird. Das für diese Seite verwendete Standardsteuerelement wurde mit `Merchandise__c` und `recordSetVar` mit `products` festgelegt. Demzufolge füllt das Steuerelement automatisch die Produktlistenvariable mit Merchandise-Datensätzen auf, die von der Datenbank abgerufen wurden. Dabei handelt es sich um die Liste, die die Komponente `dataTable` verwendet.
- Es wird eine Möglichkeit benötigt, beim Durchgehen der Liste auf die aktuelle Zeile zu verweisen. Die Anweisung `var="pitem"` weist die Variable `pitem` zu, die die aktuelle Zeile umfasst.
- Die Attribute `rowClasses` und `styleClass` verwenden einfach einige der Stile der CSS-Formatvorlage, die in der statischen Ressource geladen wurden. Diese können problemlos entfernt werden, was lediglich dazu führt, dass die Anzeige weniger ansprechend erscheint.
- Jedes Standardsteuerelement weist mehrere Methoden auf, die für alle Force.com-Objekte vorhanden sind. Mit der Komponente `commandButton` wird die Schaltfläche angezeigt und die Methode `quicksave` für das Standardsteuerelement aufgerufen, mit der die Werte der Datensätze aktualisiert werden. Im vorliegenden Fall wird die physische Anzahl des Produkts aktualisiert und eine Schnellspeicherung durchgeführt, wodurch das Produkt mit der neuen Anzahl aktualisiert wird.

- In diesem Beispiel wird zwar keine Paginierung verwendet, die Funktionalität ist jedoch gegeben. Wenn genügend Datensätze für eine Paginierung vorhanden sind, fügen Sie folgenden Code nach dem Wert `commandButton` hinzu, um ein Blättern durch Seiten zu ermöglichen.

```
<apex:commandLink action="{!next}" value="Next" rendered="{!hasNext}" />
```

### Siehe auch:

[Lernprogramm 7: Erstellen einer benutzerdefinierten Benutzeroberfläche mit Visualforce](#)

[Schritt 4: Hinzufügen eines Steuerelements zur Seite](#)

[Zusammenfassung](#)

## Zusammenfassung

Herzlichen Glückwunsch! Sie haben eine neue Oberfläche für die Warenlageranwendung erstellt, indem Sie eine Visualforce-Seite erstellt haben, auf der ein Standardsteuerelement verwendet wird. Die Seite lässt sich umfangreich anpassen. Sie können z. B. leicht ändern, welche Daten in jeder Zeile angezeigt werden, indem Sie die Spaltenkomponenten bearbeiten. Des Weiteren werden auf der Seite viele Funktionen eingesetzt, die das Standardsteuerelement im Hintergrund bereitstellt. So fragt das Steuerelement etwa automatisch die Datenbank ab, sucht nach allen Merchandise-Datensätzen und weist diese der Variablen `products` zu. Über die Methode zur schnellen Speicherung können Datensätze auch gespeichert werden.

Da Sie nun einige Visualforce-Komponenten erstellt haben, können Sie diese Funktionalität auf einer öffentlichen Website öffentlich zugänglich machen. Informationen dazu finden Sie im [Lernprogramm 8: Erstellen einer öffentlichen Webseite mit Sites](#).

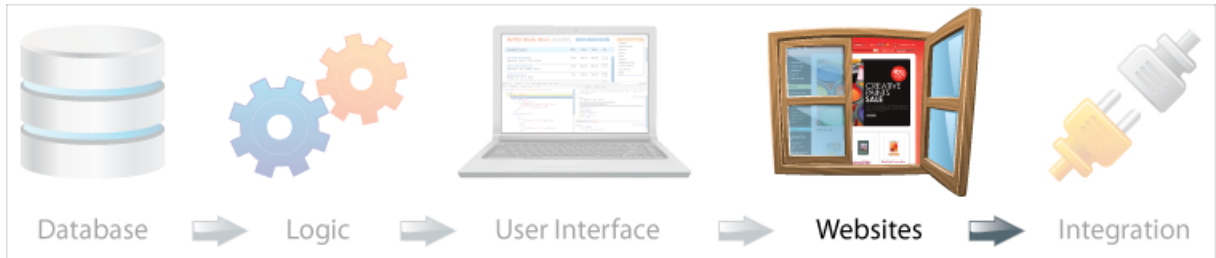
### Siehe auch:

[Lernprogramm 7: Erstellen einer benutzerdefinierten Benutzeroberfläche mit Visualforce](#)

[Schritt 5: Anzeigen der Bestandszählerseite als Visualforce-Seite](#)

# Lernprogramm 8: Erstellen einer öffentlichen Webseite mit Sites

**Niveau:** Fortgeschrittene Anfänger; **Dauer:** 20–30 Minuten



Bei der eben erforderlichen Anwendung ist zur Verwendung eine Anmeldung erforderlich. Force.com unterstützt zwar den Einsatz Ihrer Anwendung in einer Umgebung mit mehreren Benutzern, von Zeit zu Zeit muss der Anmeldevorgang jedoch eliminiert werden. Dies ist z. B. bei öffentlichen Websites der Fall.

Mit Sites können Sie öffentliche Websites und Anwendungen erstellen, die direkt in Ihre Organisation integriert sind. Die Benutzer müssen sich dabei nicht mit einem Benutzernamen und einem Kennwort anmelden. Sie können alle Informationen, die in Ihrer Organisation gespeichert sind, über einen beliebigen firmenspezifischen URL öffentlich zugänglich machen. Außerdem können Sie die Seiten der Website entsprechend dem Erscheinungsbild Ihrer Marke gestalten. Da die Sites auf Force.com-Servern gehostet werden, bestehen keinerlei Probleme hinsichtlich der Datenintegration. Und da die Sites auf systemeigenen Seiten aufgebaut werden, werden die erfassten Informationen automatisch validiert. Sie können den Benutzern auch ermöglichen, sich über Ihre öffentliche Site übergangslos bei einem verknüpften Kundenportal zu registrieren oder anzumelden.

In diesem Lernprogramm erstellen Sie eine Visualforce-Seite, aktivieren Sites für Ihre Organisation, registrieren Ihren Force.com-Domännennamen und veröffentlichen die erstellte Visualforce-Seite im Internet als öffentlichen Produktkatalog.

## Voraussetzungen

### Visualforce-Lernprogramm

Zunächst müssen Sie wie in [Lernprogramm 7: Erstellen einer benutzerdefinierten Benutzeroberfläche mit Visualforce](#) auf Seite 57 beschrieben die Visualforce-Seite erstellen.

## Siehe auch:

- [Schritt 1: Erstellen einer Produktkatalogseite](#)
- [Schritt 2: Registrieren eines Force.com-Domännennamens](#)
- [Schritt 3: Erstellen einer Force.com-Site](#)
- [Schritt 4: Konfigurieren und Testen der Site](#)
- [Zusammenfassung](#)

## Schritt 1: Erstellen einer Produktkatalogseite

In diesem Schritt duplizieren Sie die zuvor erstellte Bestandszählerseite. Auf der neuen Produktkatalogseite werden eine Beschreibung und der Preis anstelle von Bestand und Anzahl angezeigt.

1. Klicken Sie auf **Setup** ► **Entwickeln** ► **Seiten**. Falls Sie noch mit dem Visualforce-Editor arbeiten, klicken Sie in Ihrem Browser auf **Zurück**, bis der Bereich "Setup" angezeigt wird.
2. Wählen Sie die zuvor erstellte Seite **CountSheet** aus.
3. Klicken Sie auf **Duplizieren**.
4. Ändern Sie auf der Registerkarte "Seiteneditor" die Felder **Label** und **Name** in **Catalog**.
5. Ändern Sie im Seiteneditor den Inhalt des Tags `<h1>` in **Merchandise Catalog**.
6. Suchen Sie im Editor nach den folgenden Zeilen.

```
<apex:column headerValue="Inventory"> <apex:outputField
value="{!pitem.Total_Inventory__c}"/> </apex:column>
```

7. Ändern Sie **Inventory** in **Description** und **Total\_Inventory\_\_c** in **Description\_\_c**. Die Tabelle wird beibehalten, die Tabellenüberschrift und die enthaltenen Daten werden jedoch geändert. Der Code sollte wie folgt aussehen.

```
<apex:column headerValue="Description"> <apex:outputField value="{!pitem.Description__c}"/>
</apex:column>
```

8. Nun führen Sie bei der Textspalte eine ähnliche Änderung durch, um den Preis anzuzeigen. Ändern Sie **Physical Count** in **Price**. Ändern Sie dann **inputfield** in **outputfield**. Ändern Sie dann wie gezeigt **Total\_Inventory\_\_c** in **Price\_\_c**.

```
<apex:column headerValue="Price"> <apex:outputField value="{!pitem.Price__c}"/>
</apex:column>
```

9. Entfernen Sie abschließend die öffnenden und schließenden Tags `<apex:form>` und `<apex:commandButton>`, da auf dieser Seite keine Eingaben zulässig sein sollen.
10. Überprüfen Sie, dass Ihr Code dem folgenden Beispiel entspricht, und klicken Sie dann auf **Speichern**.

```
<apex:page standardStylesheets="false" showHeader="false" sidebar="false"
standardController="Merchandise__c" recordSetVar="products" >

  <apex:stylesheet value="{!URLFOR($Resource.styles, 'styles.css')}/>

  <h1>Merchandise Catalog</h1>

  <apex:dataTable value="{!products}" var="pitem" rowClasses="odd,even"> <apex:column
headerValue="Product"> <apex:outputText value="{!pitem.name}"/> </apex:column> <apex:column
headerValue="Description"> <apex:outputField value="{!pitem.Description__c}"/>
</apex:column> <apex:column headerValue="Price"> <apex:outputField
value="{!pitem.Price__c}"/> </apex:column> </apex:dataTable>

</apex:page>
```

## Weitere Informationen

Bei diesem Schritt müssen einige wichtige Dinge beachtet werden.

- Viele Komponenten der Force.com-Plattform können dupliziert werden. Wie Sie soeben erfahren haben, ist es einfach, eine Visualforce-Seite zu duplizieren.
- Über die Standardsteuerelemente für die Visualforce-Seite kann auf alle Ihre Daten zugegriffen werden. Durch einfaches Ändern einiger Werte in einer Tabelle können Daten aus unterschiedlichen Feldern angezeigt werden.

- Erinnern Sie sich, wie einfach es ist, ein Eingabefeld in ein Ausgabefeld zu ändern? Das Visualforce-Standardsteuerelement ist in der Tat sehr leistungsstark, Sie können das Steuerelement aber auch erweitern und benutzerdefinierte Funktionen und sogar von Grund auf Ihr eigenes Steuerelement erstellen.

### Siehe auch:

[Lernprogramm 8: Erstellen einer öffentlichen Webseite mit Sites](#)

[Schritt 2: Registrieren eines Force.com-Domännennamens](#)

## Schritt 2: Registrieren eines Force.com-Domännennamens

Die eindeutige Force.com-Domäne, die als Host Ihrer Website dient, wird aus dem eindeutigen *Domänenpräfix* gebildet, das Sie registrieren, sowie `force.com`. Wenn Sie beispielsweise "unternehmensname" als Domänenpräfix wählen, lautet der Domänenname `http://www.unternehmensname.force.com`.

Registrieren Sie zunächst wie folgt die Force.com-Domäne für Ihr Unternehmen.

1. Klicken Sie auf **Setup** ► **Entwickeln** ► **Sites**.
2. Geben Sie einen eindeutigen Namen für Ihre Force.com-Domäne ein. Dieser Name darf nur Unterstriche und alphanumerische Zeichen enthalten und muss in Ihrer Organisation eindeutig sein. Er muss mit einem Buchstaben beginnen, darf keine Leerzeichen enthalten und nicht mit einem Unterstrich enden oder zwei aufeinanderfolgende Unterstriche enthalten. salesforce.com empfiehlt, den Namen Ihres Unternehmens oder eine Abwandlung davon zu verwenden, beispielsweise `meinunternehmensname`.



**ACHTUNG:** Sie können Ihren Force.com-Domännennamen nach der Registrierung nicht mehr ändern.

In der Free Edition werden Domännennamen automatisch zugewiesen und können nicht aktualisiert werden. Erstellen Sie eine benutzerdefinierte Webadresse, wenn Sie einen benutzerdefinierten Domännennamen verwenden möchten.

3. Klicken Sie auf **Verfügbarkeit prüfen**, um zu bestätigen, dass der von Ihnen eingegebene Domänenname bisher noch nicht vorhanden ist. Wenn er bereits vorhanden ist, werden Sie aufgefordert, ihn zu ändern.
4. Lesen Sie die Nutzungsbedingungen für Sites und akzeptieren Sie sie durch Aktivieren des zugehörigen Kontrollkästchens.
5. Klicken Sie auf **Meine Force.com-Domain registrieren**. Nachdem Sie die Nutzungsbedingungen akzeptiert und Ihre Force.com-Domäne registriert haben, werden die Änderungen, die sich auf die Site-Erstellung beziehen, im Einrichtungs-Aktivierungsprotokoll und in der Themenliste "Site-Verlauf" Ihrer Organisation aufgezeichnet. Bis die Registrierung wirksam wird, können 48 Stunden vergehen.

Herzlichen Glückwunsch! Nun können Sie beginnen, Ihre erste Force.com-Website zu erstellen.



## Siehe auch:

[Lernprogramm 8: Erstellen einer öffentlichen Webseite mit Sites](#)

[Schritt 1: Erstellen einer Produktkatalogseite](#)

[Schritt 3: Erstellen einer Force.com-Site](#)

## Schritt 3: Erstellen einer Force.com-Site

Nach dem Registrieren einer Domäne können Sie die Visualforce-Merchandise-Katalogseite zur Startseite Ihrer neuen Site machen.

1. Wechseln Sie zur Seite "Sites" durch Klicken auf **Setup** ► **Entwickeln** ► **Sites**.
2. Klicken Sie auf **Neu**. Die Seite "Site bearbeiten" wird angezeigt.
3. Geben Sie auf der Seite "Site bearbeiten" die Details zur Site ein:
  - a. Geben Sie in den Feldern `Site Label` und `Site Name` den Wert `Catalog` ein.
  - b. Geben Sie im Feld `Active Site Home Page` den Wert `Catalog` ein.
  - c. Aktivieren Sie das Kontrollkästchen `Active`.

New Site

Save Cancel

Site Label: Catalog

Site Name: Catalog

Site Description:

Site Contact: Admin User

Default Web Address: http://workbook-developer-edition.na1.force.com/

Active:

Active Site Home Page: Catalog

Inactive Site Home Page: InMaintenance

Site Template: SiteTemplate

Site Robots.txt:

Site Favorite Icon:

Analytics Tracking Code:

Enable Feeds:

Save Cancel

4. Klicken Sie auf **Speichern**.

### Siehe auch:

[Lernprogramm 8: Erstellen einer öffentlichen Webseite mit Sites](#)

[Schritt 2: Registrieren eines Force.com-Domännennamens](#)

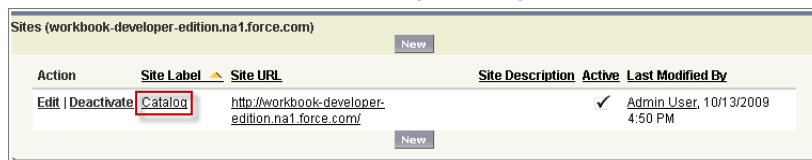
[Schritt 4: Konfigurieren und Testen der Site](#)

## Schritt 4: Konfigurieren und Testen der Site

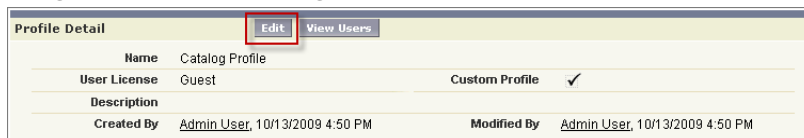
Nach dem Erstellen und Festlegen der Site als Standardseite können Sie diese nun bald testen.

Die Plattform ist mit einigen konservativen Kontrollen für die Datensicherheit ausgestattet. Eine der Kontrollen stellt sicher, dass auch auf öffentlichen Seiten keine Daten angezeigt werden, sofern diese Funktion nicht explizit aktiviert wird. In diesem Schritt schalten Sie diese Sicherheitseinstellung für das Merchandise-Objekt um und besuchen dann Ihre neue Site.

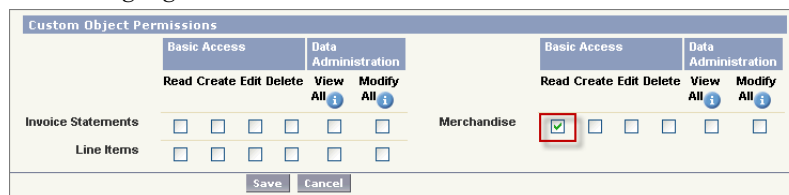
1. Wechseln Sie zur Liste "Sites" durch Klicken auf **Setup ► Entwickeln ► Sites**.
2. Klicken Sie auf den Link `Site URL` für die Site "Produktkatalog". Je nach Browser wird durch diese Aktion entweder eine neue Registerkarte oder ein neues Fenster geöffnet. Statt der Seite "Merchandise-Katalog" wird allerdings groß die Seite "Autorisierung erforderlich" angezeigt. Dies liegt daran, dass der anonymen Websiteanzeige noch keine Berechtigung zum Anzeigen der Daten gewährt wurde, die auf der Seite angezeigt werden. Beheben Sie dies, indem Sie zurück zur Seite "Setup" wechseln.
3. Klicken Sie auf den Link `Site Label`, der die Bezeichnung "Catalog" aufweisen sollte.



4. Klicken Sie auf **Einstellungen für öffentlichen Zugriff** und dann im Abschnitt "Profildetails" auf **Bearbeiten**.



5. Blättern Sie nach unten zum Abschnitt "Benutzerdefinierte Objektberechtigungen" und wählen Sie für das Merchandise-Objekt die Berechtigung `Read` aus.



6. Klicken Sie auf **Speichern**.
7. Kehren Sie nun zu Ihrer Website zurück und aktualisieren Sie den Browser. Ihre Seite sollte angezeigt werden.

Merchandise Catalog		
Product	Description	Price
Wee Jet	A small plane	\$5.00



**Anmerkung:** Wenn die Meldung "Autorisierung erforderlich" noch immer angezeigt wird, haben Sie im vorherigen Schritt die Active Site Home Page nicht mit Catalog festgelegt.

### Weitere Informationen

- Sie haben der Site eine einzelne Visualforce-Seite zugewiesen und sie als Startseite für diese Site festgelegt. Sie können mit dem Erstellen weiterer Visualforce-Seiten fortfahren und diese auch der Site zuweisen. Wenn Sie beispielsweise ein Seitenmem zur Site hinzufügen, können Sie mit einem URL auf dieses zugreifen, die in etwa `http://workbook-developer-edition.na1.force.com/Mem` lautet (je nach Ihrem Domännennamen natürlich).
- Force.com speichert die Anzahl der Seitenaufrufe einer Developer Edition-Site und setzt auf der Site eine tägliche Bandbreite durch.
- Mit Sites-Sites können Sie ein globales Content Delivery Network nutzen, um raschen Zugriff und das Zwischenspeichern der Seiten Ihrer Site in Produktionsumgebungen zu ermöglichen. Sie müssen lediglich die Komponente `page` so bearbeiten, dass diese eine Anweisung zum Zwischenspeichern enthält.

```
<apex:page cache="true" expires="600">
```

### Siehe auch:

[Lernprogramm 8: Erstellen einer öffentlichen Webseite mit Sites](#)

[Schritt 3: Erstellen einer Force.com-Site](#)

[Zusammenfassung](#)

## Zusammenfassung

Herzlichen Glückwunsch, Sie haben eine öffentliche Website erstellt! Jede Person weltweit kann auf die Website zugreifen, die Sie soeben erstellt haben, und Ihre Homepage anzeigen, ohne sich dafür anmelden zu müssen. Zum Erstellen der Website haben Sie eine Visualforce-Seite dupliziert und dann Ihre Developer Edition-Organisation für die Sites-Funktionalität aktiviert. Danach haben Sie einen eindeutigen Force.com-Domännennamen registriert, die Seite als Ihre Startseite registriert, die Sicherheit angepasst und die Website getestet, um ihre Funktionalität zu überprüfen.

Da nun eine Website mit einem Produktkatalog vorhanden ist, ist es sinnvoll, im nächsten Schritt den Benutzern eine Möglichkeit zu eröffnen, Merchandise-Artikel zu bestellen. Dies wird im [Lernprogramm 9: Erstellen einer Ladenseite](#) auf Seite 71 durchgeführt.

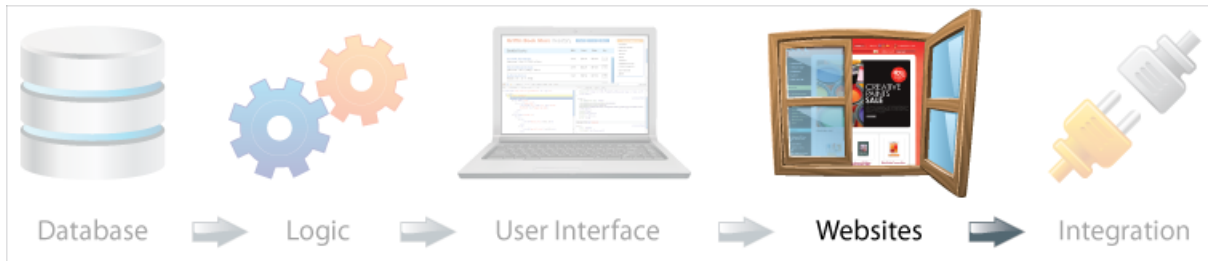
### Siehe auch:

[Lernprogramm 8: Erstellen einer öffentlichen Webseite mit Sites](#)

[Schritt 4: Konfigurieren und Testen der Site](#)

# Lernprogramm 9: Erstellen einer Ladenzeile

**Niveau:** Fortgeschrittene Anfänger; **Dauer:** 30 Minuten



Apex ist eine objektorientierte Programmiersprache. Mit Apex können Klassen und Methoden erstellt, Abrufe an die Datenbank gerichtet, Webservices erstellt, E-Mails gesendet und viele weitere Aufgaben durchgeführt werden. In diesem Lernprogramm erstellen Sie eine einfache Visualforce-Ladenzeilenseite, bei der ein Steuerelement verwendet wird, das Sie in Apex schreiben. Dabei erstellen Sie eine Apex-Klasse, lernen weitere Apex-Syntax kennen und erfahren, wie Werte zwischen einer Visualforce-Seite und ihrem Steuerelement übertragen werden können.

## Voraussetzungen

### Warenlageranwendung

Zunächst benötigen Sie eine einfache Warenlageranwendung und müssen Beziehungen hinzufügen. Dieser Vorgang wird unter [Lernprogramm 2: Hinzufügen von Beziehungen](#) auf Seite 15 beschrieben.

### Softwareanforderungen

Falls Sie möchten, können Sie für dieses Lernprogramm die Force.com-IDE verwenden. Wenn Sie dies tun möchten, benötigen Sie Eclipse 3.3 oder 3.4 und das Force.com-IDE-Plugin: [wiki.developerforce.com/index.php/Force.com\\_IDE](http://wiki.developerforce.com/index.php/Force.com_IDE). Informationen zum Erstellen eines Projekts finden Sie unter [Erstellen eines Projekts in der Force.com-IDE](#) auf Seite 93.

### Berechtigungen "Entwicklermodus", "Alle Daten modifizieren" und "Autor-Apex"

Stellen Sie sicher, dass Sie entsprechende Berechtigungen zum Erstellen von Apex-Klassen haben, da im Rahmen dieses Lernprogramms Arbeiten mit Apex anfallen.

## Siehe auch:

- [Schritt 1: Erstellen eines Steuerelements](#)
- [Schritt 2: Hinzufügen von Methoden zum Steuerelement](#)
- [Schritt 3: Erstellen der Ladenzeile](#)
- [Schritt 4: Zusätzlicher Schritt – Aktualisieren der Seite mit AJAX](#)
- [Zusammenfassung](#)

## Schritt 1: Erstellen eines Steuerelements

Anstelle des Standardsteuerelements aus dem vorangegangenen Lernprogramm schreiben Sie dieses Mal den Steuerelementcode selbst. Steuerelemente rufen auf einer Visualforce-Seite anzuzeigende Daten in der Regel ab und umfassen Code, der auf Seitenaktionen hin ausgeführt wird. Hierzu zählt z. B. das Klicken auf eine Befehlsschaltfläche.

Die einfache Ladenzeile weist zwei Methoden auf: `getProducts()`, die die Produkte zurückgibt, die in der Ladenzeile angezeigt werden; und `shop()`, die ausgeführt wird, wenn Produkte ausgewählt wurden.

So erstellen Sie die Klasse auf der Weboberfläche:

1. Klicken Sie auf **Setup** ► **Entwickeln** ► **Apex-Klassen**.
2. Klicken Sie auf **Neu**.
3. Fügen Sie den folgenden Code als Definition der Klasse hinzu und klicken Sie dann auf **Schnellspeicherung**.

```
public class StoreFront {
}
```

So erstellen Sie die Klasse in der Force.com-IDE:

1. Klicken Sie in der IDE mit der rechten Maustaste auf den Projektordner und wählen Sie **Neu** ► **Apex-Klasse** aus.
2. Geben Sie auf der Seite "Apex-Klasse erstellen" als Name `StoreFront` an.
3. Klicken Sie auf **Fertig stellen**.

Nun haben Sie eine einfache Klasse für das Steuerelement erstellt. Sie weist keine Methoden oder Felder auf, sodass Sie diese im nächsten Schritt hinzufügen.

### Siehe auch:

[Lernprogramm 9: Erstellen einer Ladenzeile](#)

[Schritt 2: Hinzufügen von Methoden zum Steuerelement](#)

## Schritt 2: Hinzufügen von Methoden zum Steuerelement

Im letzten Schritt haben Sie ein benutzerdefiniertes Steuerelement erstellt. Jetzt fügen Sie Methoden hinzu. Sie beginnen mit dem Hinzufügen der Methode `shop()`. Diese weist zunächst noch keine Funktion auf, ist jedoch als Platzhalter für die Aktion erforderlich, die beim Klicken auf die Schaltfläche "Shop" durchgeführt wird. Darüber hinaus sollen mehrere Merchandise-Artikel mit Anzahl angezeigt werden, damit der Endbenutzer auswählen kann, wie viele Einheiten eines jeden Artikels gekauft werden sollen.

1. Fügen Sie den folgenden Code zur Klasse hinzu (direkt nach der Zeile `public class StoreFront` {}).

```
public PageReference shop() { return null; }
```

2. Definieren Sie mit der aktuellen Klasse (in der nächsten Zeile) ein Feld, das die Liste `DisplayMerchandise` umfasst, sowie eine neue innere Klasse mit der Bezeichnung `DisplayMerchandise`, die die Daten enthält.

```
DisplayMerchandise[] products; public class DisplayMerchandise { public Merchandise__c merchandise { get; set; } public Decimal count { get; set; } public DisplayMerchandise(Merchandise__c item) { this.merchandise = item; } }
```

3. Definieren Sie in der nächsten Zeile die Methode `getProducts()`, die die Produkte initialisiert.

```
public DisplayMerchandise[] getProducts() { if (products == null) { products = new DisplayMerchandise[]{}; for (Merchandise__c item : [SELECT id, name, description__c,
```

```
price__c FROM Merchandise__c WHERE Total_Inventory__c > 0]) { products.add(new
DisplayMerchandise(item)); } } return products; }
```

4. Überprüfen Sie, ob Ihr Code dem folgenden Beispiel entspricht, und klicken Sie dann auf **Speichern**.

```
public class StoreFront { public PageReference shop() { return null; }

    DisplayMerchandise[] products;

    public class DisplayMerchandise { public Merchandise__c merchandise { get; set; }
public Decimal count { get; set; } public DisplayMerchandise(Merchandise__c item) {
this.merchandise = item; } }

    public DisplayMerchandise[] getProducts() { if (products == null) { products = new
DisplayMerchandise[]{}; for (Merchandise__c item : [SELECT id, name, description__c,
price__c FROM Merchandise__c WHERE Total_Inventory__c > 0]) { products.add(new
DisplayMerchandise(item)); } } return products; } }
```

Das Definieren eines einfachen Steuerelements ist hiermit abgeschlossen. Es werden standardmäßige Apex-Klassen und -Methoden verwendet, um auf die Datenbank zuzugreifen.

### Weitere Informationen

Die Klasse `DisplayMerchandise` führt für den bereits in der Datenbank vorhandenen `Merchandise`-Typ einen Wrap-Vorgang durch und fügt dabei ein neues Dezimalfeld hinzu. Mit dem Steuerfeld können Sie eine neue `DisplayMerchandise`-Instanz erstellen, indem ein vorhandener `Merchandise`-Datensatz übergeben wird. Die Instanzvariable `products` ist als Liste der `DisplayMerchandise`-Instanzen definiert.

Die Methode `getProducts()` führt eine Abfrage aus (der Text in geschweiften Klammern, welcher auch als SOQL-Abfrage bezeichnet wird) und gibt alle `Merchandise`-Datensätze zurück, die einen positiven Gesamtbestand aufweisen. Daraufhin werden die von der Abfrage zurückgegebenen Datensätze durchgegangen und zu einer Liste der `DisplayMerchandise`-Produkte hinzugefügt, die im Anschluss zurückgegeben wird.

### Siehe auch:

- [Lernprogramm 9: Erstellen einer Ladenzeile](#)
- [Schritt 1: Erstellen eines Steuerelements](#)
- [Schritt 3: Erstellen der Ladenzeile](#)

## Schritt 3: Erstellen der Ladenzeile

Bei der Ladenzeile wird eine weitere Visualforce-Seite verwendet, die jetzt erstellt wird.

1. Fügen Sie in der Adressleiste des Browsers am Ende der Instanz `/apex/StoreFront` hinzu. Beispiel: `https://na1.salesforce.com/apex/StoreFront`. Die bereits bekannte Fehlermeldung wird angezeigt.
2. Klicken Sie auf den Link **Seiten-StoreFront erstellen**.
3. Klicken Sie unten auf die Registerkarte "Seiteneditor".
4. Ersetzen Sie den Text im Editor mit dem folgenden.

```
<apex:page standardStylesheets="false" showHeader="false" sidebar="false"
controller="StoreFront" > <apex:stylesheet value="{!URLFOR($Resource.styles,
'styles.css')}" /> <h1>Store Front</h1> <apex:form>
```

```
</apex:form> </apex:page>
```



**Anmerkung:** Das Attribut `controller="StoreFront"` stellt sicher, dass die Visualforce-Seite, die Sie erstellen, die Apex-Klasse `StoreFront` als Steuerelement verwendet.

- Nun können Sie den Inhalt des Formulars zwischen den öffnenden und schließenden Formulartags eingeben. Sie erstellen eine Tabelle, die die Produkte durchgeht, sowie eine Schaltfläche, die die Methode `shop` aufruft. Geben Sie in der nächsten Zeile diesen Code zwischen den öffnenden und schließenden `form`-Tags ein.

```
<apex:dataTable value="{!products}" var="pitem" rowClasses="odd,even"> <apex:column
headerValue="Product"> <apex:outputText value="{!pitem.merchandise.name}"/> </apex:column>
<apex:column headerValue="Price"> <apex:outputText value="{!pitem.merchandise.price__c}"/>
</apex:column> <apex:column headerValue="#Items"> <apex:inputText value="{!pitem.count}"/>
</apex:column> </apex:dataTable> <br /> <apex:commandButton action="{!shop}" value="Buy"
/>
```

- Klicken Sie auf **Speichern**. Die Ladenzeilenseite sollte wie folgt aussehen.

Store Front

Product	Price	#Items
Wee Jet	5.0	<input style="width: 50px;" type="text"/>

---

Page Editor    Controller

```

1 <apex:page standardStylesheets="false" showHeader="false" sidebar="false"
2   controller="StoreFront" >
3   <apex:stylesheet value="{!URLFOR($Resource.styles, 'styles.css')}"/>
4   <h1>Store Front</h1>
5   <apex:form>
6     <apex:dataTable value="{!products}" var="pitem" rowClasses="odd,even">
7       <apex:column headerValue="Product">
8         <apex:outputText value="{!pitem.merchandise.name}"/>
9       </apex:column>
10      <apex:column headerValue="Price">
11        <apex:outputText value="{!pitem.merchandise.price__c}"/>
12      </apex:column>
13      <apex:column headerValue="#Items">
14        <apex:inputText value="{!pitem.count}"/>
15      </apex:column>
16    </apex:dataTable>
17    <br />
18    <apex:commandButton action="{!shop}" value="Buy" />
19  </apex:form>
20 </apex:page>
```

## Weitere Informationen

In diesem Schritt wurde eine Menge neuer Stoff behandelt. Gehen wir auf diesen ein.

- Das Wertattribut von `dataTable` ist mit "products" festgelegt, wodurch angegeben wird, dass die Tabelle eine Liste mit der Bezeichnung "products" durchgehen soll. Da ein benutzerdefiniertes Steuerelement verwendet wird, sucht Visualforce im Apex-Steuerelement automatisch nach einer Methode mit der Bezeichnung `getProducts()`.
- Die Methode `getProducts()` im Steuerelement gibt eine Reihe der `DisplayMerchandise`-Objekte zurück. Jedes dieser Objekte bildet eine neue Zeile und wird auf der Visualforce-Seite beim Durchgehen von `dataTable` der `pitem`-Variablen zugewiesen.

- Die Visualforce-Seite weist die Aktion `{!shop}` auf. Da es sich um eine Aktion handelt, wird im Apex-Steuerelement eine Methode mit dem exakt gleichen Namen verwendet.

### Siehe auch:

[Lernprogramm 9: Erstellen einer Ladenzeile](#)

[Schritt 2: Hinzufügen von Methoden zum Steuerelement](#)

[Schritt 4: Zusätzlicher Schritt – Aktualisieren der Seite mit AJAX](#)

## Schritt 4: Zusätzlicher Schritt – Aktualisieren der Seite mit AJAX

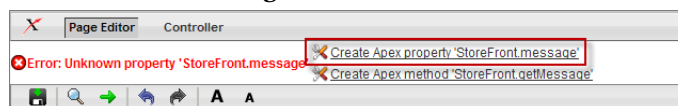
Dieser Schritt ist optional. Es wird gezeigt, wie Visualforce Daten transparent an das Steuerelement zurückgibt, wo diese verarbeitet werden können. Die Methode `shop()` beispielsweise, die Sie im Steuerelement geschrieben haben, hat Zugriff auf die Merchandise-Anzahl, die vom Endbenutzer auf der Visualforce-Seite eingegeben wurde. In diesem Schritt wird gezeigt, wie Sie auf diese Daten im Steuerelement zugreifen. Dabei werden diese Daten auf der Seite einfach nochmals (mit einer eleganten AJAX-Seitenaktualisierung) angezeigt. Bei praktischen Anwendungen kann die Bestellung gesendet und ein Webservice aufgerufen werden, die Artikel können zu einem Warenkorb hinzugefügt und viele weitere Aktionen können durchgeführt werden.

1. Navigieren Sie mit dem URL zur Visualforce-Seite, z. B. `https://na1.salesforce.com/apex/StoreFront`.
2. Klicken Sie auf die Registerkarte "Seiteneditor".
3. Bearbeiten Sie die Seite, indem Sie nach Zeile `</apex:form>` folgenden Code eingeben.

```
<apex:outputPanel id="msg">{!message}</apex:outputPanel>
```

Sie haben soeben einen als `msg` identifizierten Ausgabebereich erstellt, der ein Element des Steuerelements mit der Bezeichnung "message" anzeigt. Dieses wurde allerdings noch nicht erstellt.

4. Klicken Sie auf **Speichern**. Über den Seiteneditor wird ermittelt, dass keine Methode oder Eigenschaft mit der Bezeichnung "message" vorhanden ist. Daher wird die Aufforderung angezeigt, eine solche zu erstellen.
5. Klicken Sie auf **Apex-Eigenschaft "StoreFront.message" erstellen**.



6. Ändern Sie den Tag `commandButton` nun so, dass das Attribut `reRender` enthalten ist.

```
<apex:commandButton action="{!shop}" reRender="msg" value="Buy" />
```

Sie haben die Visualforce-Seite jetzt so angepasst, dass eine vorhandene AJAX-Aktualisierung verwendet wird. Dadurch wird der Bereich "msg" nach Abruf der Methode `shop()` des Steuerelements aktualisiert. Bearbeiten Sie die Methode `shop()` nun so, dass die im Bereich angezeigte Meldungseigenschaft mit einer Liste der ausgewählten Elemente aktualisiert wird.

1. Klicken Sie neben der Registerkarte "Seiteneditor" auf die Registerkarte "Steuerelement".
2. Suchen Sie nach der Methode `shop()` (Zeilen 4 bis 6) und ersetzen Sie sie mit folgendem Code.

```
public PageReference shop() { message = 'You bought: '; for (DisplayMerchandise p: products) { if (p.count > 0) { message += p.merchandise.name + ' (' + p.count + ') ' ; } } return null; }
```



**Anmerkung:** Beachten Sie, dass bei diesem Code schlicht die Variable `products` verwendet wird. Visualforce stellt automatisch sicher, dass die auf der Benutzeroberfläche geänderten Daten von den Daten in der Variablen `products` widergespiegelt werden.

3. Klicken Sie auf das Symbol **Speichern**.
4. Testen Sie den Warenkorb nun. Fügen Sie zu einem der Merchandise-Artikel eine Zahl hinzu und klicken Sie auf **Erwerben**. Unterhalb der Merchandise-Liste wird ein Textfeld angezeigt, das die Menge der erworbenen Produkte angibt.

Product	Price	#Items
Wee Jet	1.0	3.0

Buy

You bought: Wee Jet (3)

### Weitere Informationen

Durch dieses einfache Hinzufügen ist eine leistungsstarke Funktionalität verfügbar.

- Wie Sie in diesem Schritt erfahren haben, hat Visualforce automatisch die Datenänderungen im Formular bei der Produktvariable widergespiegelt. Diese Funktionalität ist äußerst leistungsfähig und ermöglicht es Ihnen, rasch Formulare und andere komplexe Eingabeseiten zu erstellen.
- Beim Klicken auf die Schaltfläche **Erwerben** wird der Bereich aktualisiert, ohne dass der gesamte Bildschirm aktualisiert wird. Der dazu erforderliche AJAX-Effekt, der in der Regel eine komplexe JavaScript-Bearbeitung erforderlich macht, wurde mit einem einfachen `reRender`-Attribut erzielt.

### Siehe auch:

[Lernprogramm 9: Erstellen einer Ladenzeile](#)

[Schritt 3: Erstellen der Ladenzeile](#)

[Zusammenfassung](#)

## Zusammenfassung

Apex ist eine leistungsstarke objektorientierte Sprache mit vielen Funktionen, die in der Regel in ähnlichen Sprachen wie Java anzutreffen sind. In diesem Lernprogramm haben Sie eine Apex-Klasse erstellt und einige Funktionen der Sprache verwendet, wie Reihen, Iteration und Abfragen der Datenbank. Wenn Sie den zusätzlichen Schritt durchgeführt haben, haben Sie die Visualforce-Seite so bearbeitet, dass die zum Steuerelement zurückgegebenen Daten verwendet werden, um mit einer AJAX-Aktualisierung eine Meldung anzuzeigen, ohne dass dazu die gesamte Seite aktualisiert wird.

Als Nächstes erstellen Sie möglicherweise ein authentifiziertes Portal, sodass sich Benutzer bei Ihrer Website anmelden, Benutzername und Kennwort erstellen und Merchandise-Artikel erwerben können. Dazu müssten sie ein Portal erstellen. Dieser Vorgang wird in diesem Workbook jedoch nicht abgedeckt. Allerdings sind detaillierte Schritte vorhanden, die aufzeigen, wie ein Portal erstellt wird, das mit der zuvor erstellten Anwendung kompatibel ist. Besuchen Sie für das Portallernprogramm [developer.force.com/workbook](http://developer.force.com/workbook). Dort finden Sie auch weitere Ergänzungen zum Workbook.

Darüber hinaus möchten Sie die Anwendung als Nächstes möglicherweise bereitstellen, entweder als Paket zum Vertrieb oder als Bereitstellung in einer Produktionsorganisation. Weitere Informationen finden Sie unter [Bereitstellen Ihrer Anwendung](#) auf Seite 94.

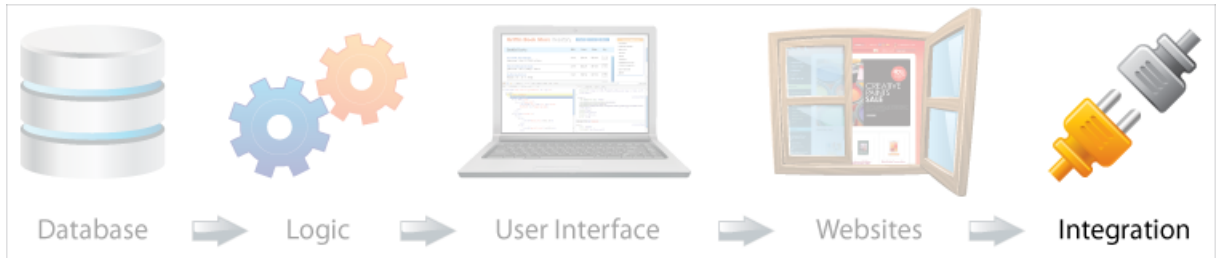
**Siehe auch:**

*[Lernprogramm 9: Erstellen einer Ladenzeile](#)*

*[Schritt 4: Zusätzlicher Schritt – Aktualisieren der Seite mit AJAX](#)*

# Lernprogramm 10: Erstellen einer Desktopanwendung mit Adobe Flash Builder für Force.com

**Niveau:** Fortgeschritten; **Dauer:** 30 Minuten



Adobe Flash Builder für Force.com ist eine Toolsammlung, mit der Sie Force.com Stratus-Anwendungen entwickeln können, also Desktopanwendungen, die in der Adobe Integrated Runtime-(AIR®-)Umgebung ausgeführt werden und Force.com-Logik sowie Datenbankfunktionen nutzen. Force.com Stratus-Anwendungen können sowohl *online* als auch *offline* verwendet werden und eignen sich somit ideal für Benutzer, die nicht immer Zugriff auf das Internet haben, aber trotzdem auf Ihre Force.com-Anwendungen zugreifen können müssen.

In diesem Lernprogramm wird aufgezeigt, wie Adobe Flash Builder für Force.com zum Erstellen einer einfachen Force.com Stratus-Anwendung mit der Bezeichnung Inventory Tracker verwendet wird, mit der offline auf Merchandise-Daten zugegriffen werden kann. Mit der Anwendung können Manager den Bestand auf Tablet-PCs nachverfolgen, während sie sich in Warenlagern mit schwankender Wi-Fi-Abdeckung aufhalten. Diese umfasst die folgenden standardmäßigen Force.com Stratus-Anwendungsfunktionen.

- Ein Anmeldebildschirm, auf dem die Benutzer zur Eingabe ihrer Anmeldeinformationen aufgefordert werden und mit dem die Verbindung mit Force.com hergestellt wird, wenn eine Verbindung mit dem Internet vorhanden ist.
- Eine Konfliktlösungsoberfläche, mit der die Benutzer auf einfache Weise Datenkonflikte lösen können, die auftreten, wenn Werte in der Force.com Stratus-Anwendung gegen Validierungsregeln verstoßen oder mit Änderungen anderer Benutzer in Konflikt stehen.
- Eine Statusleiste, die den Benutzern anzeigt, ob die Force.com Stratus-Anwendung mit dem Internet verbunden ist, und die Anzahl der ungelösten Datenkonflikte und Fehler angibt.

Name	Description	Price	Total Inventory
Fender Lonestar Strat	Pearly gates pickguard, Seymour	899	2
ESP LTD MH-401	A carved quilted maple top on a m	399	12
Les Paul Studio	A carved top and humbucking pick	900	4
Ibanez Roadster	Bass guitar	395	7
MR-0001	Chair	99	0
Paul Reed Smith SE Custom 22	Genuine maple top with flamed m	1299	6
Wee Jet	A small airplane	3	981

Zum Erstellen der Anwendung verwenden wir Adobe Flash-Technologien wie MXML- und ActionScript-Komponenten sowie Force.com Stratus-Klassen, die Adobe Flash Builder für Force.com generiert. Zum Erstellen der Anwendung Inventory Tracker benötigen Sie jedoch keine MXML- und ActionScript-Kenntnisse. In diesem Lernprogramm ist sämtlicher Code enthalten, den Sie zum Erstellen Ihrer ersten funktionsfähigen Force.com Stratus-Anwendung benötigen.



**ACHTUNG:** Adobe Flash Builder für Force.com ist derzeit nur als Developer-Vorschau in Developer Edition-Organisationen verfügbar. Bevor die Anwendung allgemein verfügbar ist, kann sich die Funktionalität noch grundlegend ändern. Verwenden Sie Adobe Flash Builder für Force.com in der Developer-Version nicht zum Erstellen von Anwendungen für Produktionsumgebungen.

### Siehe auch:

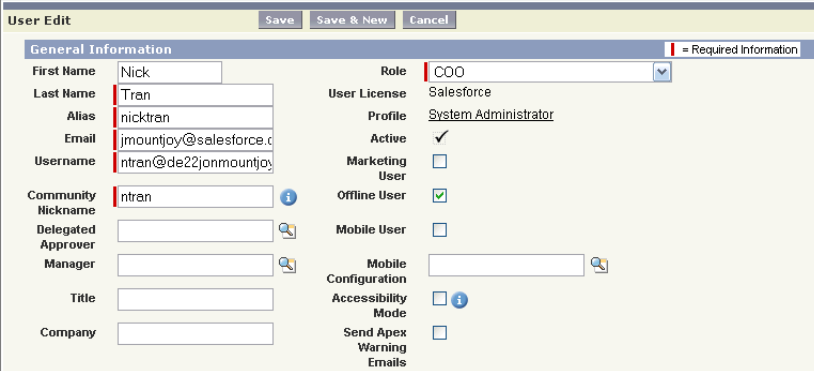
- [Schritt 1: Konfigurieren Ihrer persönlichen Salesforce.com-Daten](#)
- [Schritt 2: Erstellen einer Offline-Aktenkofferkonfiguration](#)
- [Schritt 3: Erstellen Ihrer Unternehmens-WSDL und Herunterladen eines Client-Zertifikats](#)
- [Schritt 4: Installieren und Starten von Adobe Flash Builder für Force.com](#)
- [Schritt 5: Erstellen eines Force.com Stratus-Projekts](#)
- [Schritt 6: Festlegen der Attribute für die Stratus-Anwendungskomponente](#)
- [Schritt 7: Erstellen der Inventory Tracker-Fensterkomponente](#)
- [Schritt 8: Testen der Inventory Tracker-Anwendung](#)
- [Zusammenfassung](#)

## Schritt 1: Konfigurieren Ihrer persönlichen Salesforce.com-Daten

Bei Force.com Stratus-Anwendungen wird die Connect Offline-Technologie in Force.com verwendet. Zum Entwickeln oder Verwenden einer Force.com Stratus-Anwendung muss die Option `Offline User` (Offline-Benutzer) in Ihren persönlichen Salesforce.com-Daten aktiviert sein.

1. Klicken Sie in Salesforce.com auf **Setup** ► **Benutzer verwalten** ► **Benutzer**.

- Klicken Sie neben Ihrem Namen auf **Bearbeiten**.
- Aktivieren Sie das Kontrollkästchen **Offline Benutzer** (Offline-Benutzer).



The screenshot shows the 'User Edit' window with the following details:

- Buttons:** Save, Save & New, Cancel
- General Information:**
  - First Name: Nick
  - Last Name: Tran
  - Alias: nicktran
  - Email: jmountjoy@salesforce.com
  - Username: ntran@de22jonmountjoy.com
  - Community Nickname: ntran
  - Delegated Approver: [empty]
  - Manager: [empty]
  - Title: [empty]
  - Company: [empty]
- Role:** COO
- User License:** Salesforce
- Profile:** System Administrator
- Active:**
- Marketing User:**
- Offline User:**
- Mobile User:**
- Mobile Configuration:** [empty]
- Accessibility Mode:**
- Send Apex Warning Emails:**

- Klicken Sie auf **Speichern**.

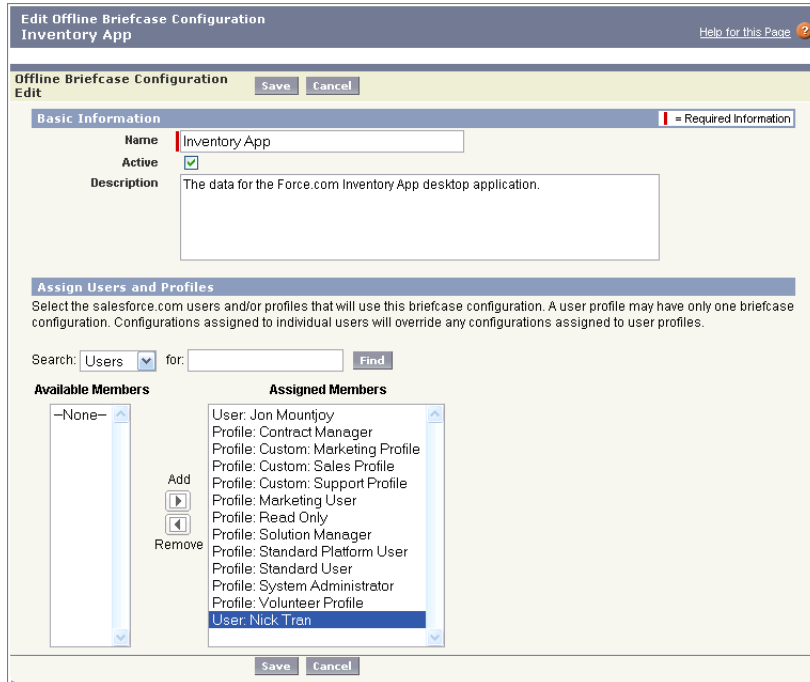
#### Siehe auch:

[Lernprogramm 10: Erstellen einer Desktopanwendung mit Adobe Flash Builder für Force.com](#)  
[Schritt 2: Erstellen einer Offline-Aktenkofferkonfiguration](#)

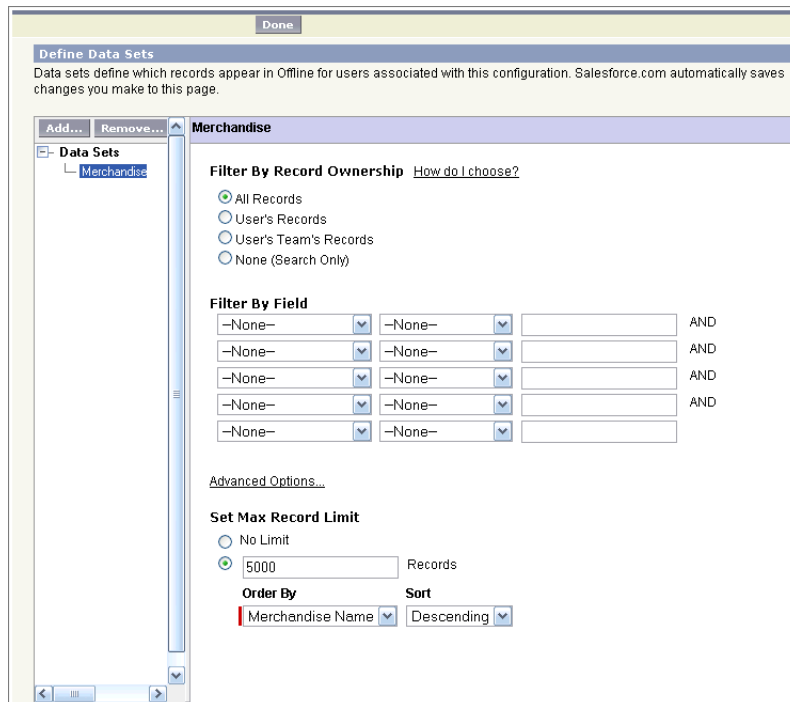
## Schritt 2: Erstellen einer Offline-Aktenkofferkonfiguration

Mit der Force.com-Desktopanwendung können Sie nur auf die Force.com-Daten zugreifen, die in der Ihnen in Salesforce.com zugewiesenen Offline-Aktenkofferkonfiguration angegeben sind. Die Anwendung Inventory Tracker benötigt Zugriff auf Merchandise-Daten, erstellen Sie also eine Offline-Aktenkofferkonfiguration mit dem Objekt "Merchandise".

- Klicken Sie in Salesforce.com auf **Setup** ► **Desktop-Verwaltung** ► **Offline-Aktenkofferkonfiguration**.
- Klicken Sie auf **Neue Offline-Aktenkofferkonfiguration**.
- Geben Sie einen Namen für die Offline-Aktenkofferkonfiguration ein, z. B. Inventory Tracker-Anwendungsdaten.
- Aktivieren Sie das Kontrollkästchen **Aktiv**.
- Wählen Sie Ihren Namen in der Liste **Verfügbare Mitglieder** aus und klicken Sie auf **Hinzufügen**, um ihn in die Liste **Zugewiesene Mitglieder** zu verschieben.



6. Klicken Sie auf **Speichern**. Die Detailseite "Offline-Aktenkofferkonfiguration" wird angezeigt.
7. Klicken Sie in der Themenliste "Datensets" auf **Bearbeiten**.
8. Klicken Sie auf **Hinzufügen...**
9. Wählen Sie **Merchandise** aus und klicken Sie auf **OK**.
10. Wählen Sie in der Auswahlliste **Sortieren** nach **Merchandise-Name** aus.



11. Klicken Sie auf **Fertig**.

## Weitere Informationen

Offline-Aktenkofferkonfigurationen sind Parametersätze, die die in Force.com Stratus-Anwendungen und Force.com Connect Offline verfügbaren Datensätze bestimmen. Connect Offline ist eine Clientanwendung, die Ihnen den Zugriff auf eine Teilmenge von Force.com-Daten ermöglicht, und zwar über dieselbe browserbasierte Schnittstelle wie das Online-System, jedoch ohne Internetverbindung. Mit Connect Offline können Sie Accounts, Aktivitäten, Kontakte, Opportunities, Leads sowie benutzerdefinierte Objektdatensätze (einschließlich Beziehungsgruppen) anzeigen, bearbeiten, erstellen und löschen. Sie können auch Produkte und Pläne zu Opportunities hinzufügen und aktualisieren.

Sie können mehrere Aktenkofferkonfigurationen erstellen und jede davon mit einem anderen Benutzer und Profil verknüpfen, um den Bedürfnissen unterschiedlicher Arten von Offline-Benutzern gleichzeitig gerecht zu werden. So kann beispielsweise eine Konfiguration Leads und Opportunities für Vertriebsmitarbeiter umfassen, während eine andere Konfiguration Accounts und zugehörige Opportunities für Kundenbeauftragte umfasst.

## Siehe auch:

[Lernprogramm 10: Erstellen einer Desktopanwendung mit Adobe Flash Builder für Force.com](#)

[Schritt 1: Konfigurieren Ihrer persönlichen Salesforce.com-Daten](#)

[Schritt 3: Erstellen Ihrer Unternehmens-WSDL und Herunterladen eines Client-Zertifikats](#)

## Schritt 3: Erstellen Ihrer Unternehmens-WSDL und Herunterladen eines Client-Zertifikats

Webservices werden häufig für Integrationen eingesetzt und Force.com umfasst einen stabilen Funktionssatz zur Unterstützung ein- und ausgehender Webserviceanfragen. Darüber hinaus erstellt Force.com automatisch einen Webservice-Endpunkt, der den Zugriff auf die Daten in Ihrer Anwendung ermöglicht. *Unternehmens-WSDL* bezieht sich auf Web Services Definition Language. Diese Sprache beschreibt die Force.com-Objekte. Für gewöhnlich wird bei der Integration in andere Systeme nichts weiter benötigt, da diese Sprache alle Objekte und den Webservice vollständig beschreibt.

Mit Adobe Flash Builder für Force.com kann die Unternehmens-WSDL in Entwicklungsprojekte importiert werden. Beim Importieren der WSDL erstellt Adobe Flash Builder für Force.com ActionScript-Klassen für jedes Objekt in der Unternehmens-WSDL, wodurch im ActionScript-Code auf die Force.com-Objekte verwiesen werden kann.

Zum Erstellen einer Klasse für das Merchandise-Objekt importieren wir die Unternehmens-WSDL zu einem späteren Zeitpunkt in das Entwicklungsprojekt. Speichern Sie für den Moment eine lokale Kopie der aktuellen Unternehmens-WSDL. Laden Sie des Weiteren ein Salesforce.com-Client-Zertifikat herunter, mit dem der Zugriff von Client-Anwendungen auf Force.com autorisiert wird.

1. Klicken Sie in Salesforce.com auf **Setup ► Entwickeln ► API**.
2. Klicken Sie auf **Unternehmens-WSDL generieren**.
3. Klicken Sie im Browser auf **Datei ► Seite speichern unter**.
4. Benennen Sie die Datei mit `unternehmens.wsdl` und klicken Sie auf **Speichern**.
5. Klicken Sie auf **Client-Zertifikat generieren**, um das Client-Zertifikat für den Computer zu generieren und zu speichern.

## Siehe auch:

[Lernprogramm 10: Erstellen einer Desktopanwendung mit Adobe Flash Builder für Force.com](#)

[Schritt 2: Erstellen einer Offline-Aktenkofferkonfiguration](#)

[Schritt 4: Installieren und Starten von Adobe Flash Builder für Force.com](#)

## Schritt 4: Installieren und Starten von Adobe Flash Builder für Force.com

Mit Adobe Flash Builder für Force.com wird eine eigenständige Version von Adobe Flash Builder installiert, der integrierten Entwicklungsumgebung von Adobe, das auf der Eclipse-Plattform basiert.

1. Laden Sie die ZIP-Datei für Adobe Flash Builder für Force.com von [developer.force.com/flashbuilder](http://developer.force.com/flashbuilder) herunter.
2. Entpacken Sie die ZIP-Datei.
3. Doppelklicken Sie auf das Installationsprogramm.
4. Klicken Sie nach dem Abschließen der Installation auf die Adobe Flash Builder für Force.com-Verknüpfung auf dem Desktop.

### Siehe auch:

[Lernprogramm 10: Erstellen einer Desktopanwendung mit Adobe Flash Builder für Force.com](#)

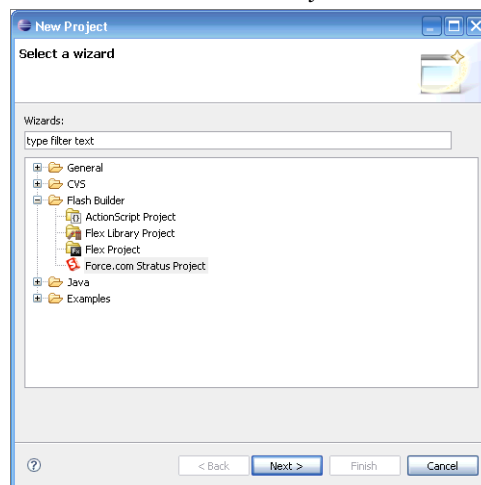
[Schritt 3: Erstellen Ihrer Unternehmens-WSDL und Herunterladen eines Client-Zertifikats](#)

[Schritt 5: Erstellen eines Force.com Stratus-Projekts](#)

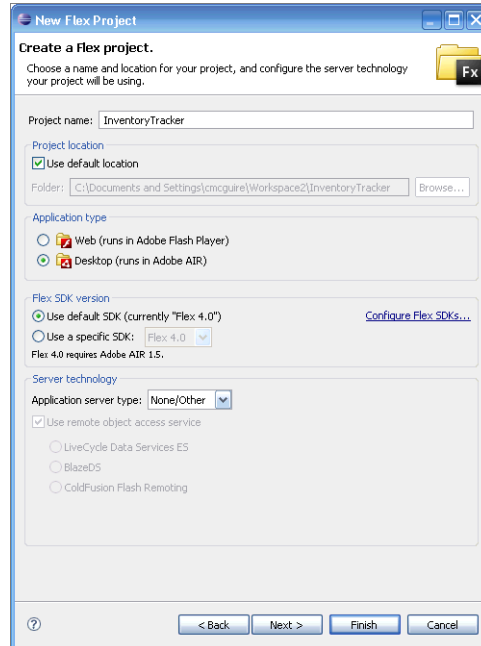
## Schritt 5: Erstellen eines Force.com Stratus-Projekts

Nach dem Installieren von Adobe Flash Builder für Force.com ist in Flash Builder ein neuer Projekttyp mit der Bezeichnung "Force.com Stratus-Projekt" verfügbar. Mit Force.com Stratus-Projekten können Sie Ihre Unternehmens-WSDL importieren und automatisch ActionScript-Klassen für Ihre Salesforce.com-Objekte erstellen. Basierend auf Ihre WSDL bietet Flash Builder des Weiteren Hinweise zum Code und Compilerwarnungen.

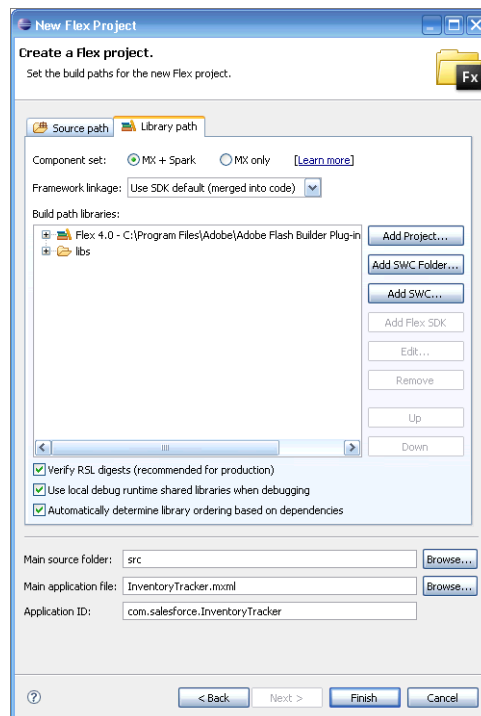
1. Wählen Sie in Flash Builder **Datei** ► **Neues Projekt** aus. Der Assistent für neue Projekte wird geöffnet.
2. Wählen Sie im Flash Builder-Ordner das "Force.com Stratus-Projekt" aus und klicken Sie auf **Weiter**.



3. Geben Sie als Projektnamen `Inventory Tracker` ein und wählen Sie `Desktop (runs in Adobe AIR)` aus.
4. Klicken Sie auf **Weiter**.

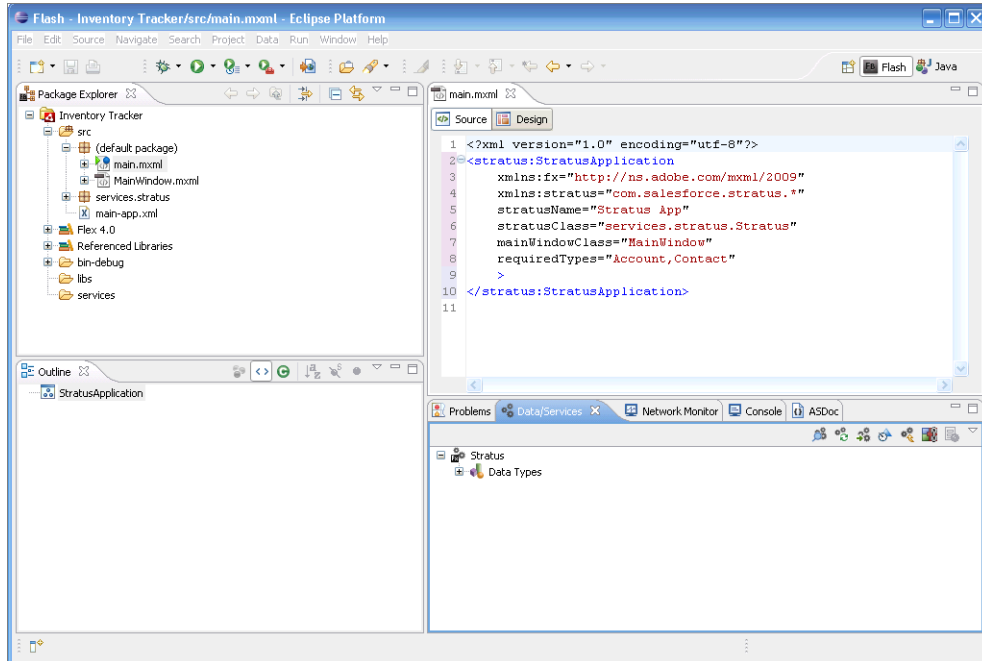


5. Belassen Sie für den Ausgabeordner die Standardeinstellungen und klicken Sie auf **Weiter**.
6. Geben Sie im Feld `Application ID` `com.salesforce.InventoryTracker` ein. Die AIR-Laufzeitumgebung und das Betriebssystem zum Identifizieren der Anwendung. Verwenden Sie die umgekehrte Domännennamennotation, um eine eindeutige Anwendungs-ID zu gewährleisten.
7. Klicken Sie auf **Fertig stellen**, um fortzufahren.



8. Wählen Sie Ihre Unternehmens-WSDL aus und klicken Sie dann auf **Weiter**.

Nach dem Erstellen eines Force.com Stratus-Projekts wird im Flash Builder Package Explorer die folgende Verzeichnisstruktur angezeigt.



### Siehe auch:

- [Lernprogramm 10: Erstellen einer Desktopanwendung mit Adobe Flash Builder für Force.com](#)
- [Schritt 4: Installieren und Starten von Adobe Flash Builder für Force.com](#)
- [Schritt 6: Festlegen der Attribute für die Stratus-Anwendungskomponente](#)

## Schritt 6: Festlegen der Attribute für die Stratus-Anwendungskomponente

Jede Force.com Stratus-Anwendung ist in einer Stratus-Anwendungskomponente enthalten. Die Stratus-Anwendungskomponente verbindet die Force.com Stratus-Anwendung mit Force.com, stellt die Force.com-Anmeldung und Authentifizierungsfunktion bereit und übernimmt die anfängliche Datensynchronisierung zwischen Force.com und der Force.com Stratus-Anwendung.

Force.com Stratus-Projekte weisen eine standardmäßige `main.mxml`-Datei auf, die eine Stratus-Anwendungsvorlage umfasst, die Sie für Ihre Inventory Tracker-Anwendung anpassen können. Bearbeiten Sie die Attribute des `stratus:StratusApplication`-Elements in dieser Datei.

1. Öffnen Sie die Datei `main.mxml`, die sich unter **src** ► **(Standardpaket)** befindet, sofern sie nicht bereits geöffnet ist.
2. Legen Sie das Attribut `stratusName` mit `Inventory Tracker` fest. Dadurch wird gesteuert, welcher Name in der Titelleiste der Anwendung angezeigt wird.
3. Legen Sie das Attribut `requiredTypes` mit `Merchandise__c` fest. Mit diesem Attribut wird bestimmt, auf welche Objekte in der Force.com-Anwendung die Force.com Stratus-Anwendung verweist.
4. Der finale Code sollte wie folgt aussehen.

```
<?xml version="1.0" encoding="utf-8"?> <stratus:StratusApplication
xmlns:fx="http://ns.adobe.com/mxml/2009" xmlns:stratus="com.salesforce.stratus.*"
stratusName="Inventory Tracker" stratusClass="services.stratus.Stratus"
mainWindowClass="MainWindow" requiredTypes="Merchandise__c" > </stratus:StratusApplication>
```

5. Speichern Sie die Datei.

### Siehe auch:

[Lernprogramm 10: Erstellen einer Desktopanwendung mit Adobe Flash Builder für Force.com](#)

[Schritt 5: Erstellen eines Force.com Stratus-Projekts](#)

[Schritt 7: Erstellen der Inventory Tracker-Fensterkomponente](#)

## Schritt 7: Erstellen der Inventory Tracker-Fensterkomponente

Die Benutzeroberfläche von Inventory Tracker wird zwei Bestandteile aufweisen. Der untere Teil wird eine Liste mit Merchandise-Datensätzen aufweisen. Wenn Sie einen Datensatz in dieser Liste auswählen, werden die Details des Datensatzes im oberen Teil angezeigt. Sie können auf die Datensatzdetails klicken, um sie zu bearbeiten. Mit der Schaltfläche **Speichern** unterhalb der Datensatzdetails können Sie diese Änderungen in Force.com übernehmen.

Die Logik für die Hauptfunktion der Inventory Tracker-Anwendung besteht aus ActionScript und MXML, bei denen zwei Adobe-Komponentensätze (Spark und Halo) sowie Stratus-Komponenten genutzt werden. Stratus-Komponenten sind von Adobe Flash Builder für Force.com erstellte Komponenten. Sie vereinfachen den Prozess des Zugreifens auf Force.com-Logik und -Daten im Code.

Sie geben die Anwendungslogik in der Standardfensterkomponente ein, die Adobe Flash Builder für Force.com beim Erstellen eines Force.com Stratus-Projekts erstellt. Die Standardfensterkomponente befindet sich in der Datei `MainWindow.mxml`. Der Standardcode lautet:

```
<?xml version="1.0" encoding="utf-8"?> <s:Window xmlns:fx="http://ns.adobe.com/mxml/2009"
xmlns:s="library://ns.adobe.com/flex/spark" xmlns:mx="library://ns.adobe.com/flex/halo"
xmlns:stratus="com.salesforce.stratus.*" showStatusBar="false" backgroundColor="#e7e7e7"
width="800" height="600" windowComplete="onWindowComplete()" >

  <s:layout> <s:VerticalLayout/> </s:layout> <s:VGroup width="100%" height="100%"
paddingBottom="10" paddingLeft="10" paddingRight="10" paddingTop="10"> <s:Label text="Your
content goes here"/> </s:VGroup> <fx:Script> <![CDATA[ import
com.salesforce.stratus.StratusMessage; private function onWindowComplete():void {

  }new StratusMessage(StratusMessage.STATUS_INFO, "Application initialized").showAsStatus();
// your code goes here      }      ]]> </fx:Script> <stratus:StatusBar/> </s:Window>
```

Der Standardcode umfasst Folgendes :

- Die Spark-Komponente `Window`, die die Namespaces, Hintergrundfarbe und Abmessungen für die Force.com Stratus-Anwendung angibt. Des Weiteren umfasst die Komponente auch das Attribut `windowComplete`, mit dem die Funktion bestimmt wird, die beim Anmelden eines Benutzers ausgeführt wird. Das Attribut `showStatusBar`, mit der bestimmt wird, ob die Anwendung die Flash Builder-Standardstatusleiste anzeigt, ist mit "false" festgelegt, damit stattdessen die Stratus-Statusleiste angezeigt wird.
- Die Spark-Komponente `Layout`, die die Layoutelemente gemäß ihrer einzelnen Einstellungen anordnet.
- Die Spark-Komponente `VerticalLayout`, die die Layoutelemente in vertikaler Reihenfolge von oben nach unten anordnet und dabei optional Abstände zwischen den Elementen und Abstand um die Reihenfolge der Elemente einfügt.
- Die Spark-Komponente `VGroup`, die eine vertikale Gruppierung der Benutzeroberflächenelemente erstellt.
- Die Spark-Komponente `Label`, die eine Bezeichnung für die vertikale Gruppierung erstellt.

- Die XML-Komponente `Script`, in der ActionScript eingegeben werden kann. Diese Komponente umfasst ActionScript-Standardelemente, mit denen die Daten beim Öffnen der Force.com Stratus-Anwendung initialisiert werden.
- Die Stratus-Komponente `StatusBar`, die unten in der Force.com Stratus-Anwendung eine Statusleiste anzeigt.

Bearbeiten Sie den Standardcode wie folgt, um die Inventory Tracker-Anwendungslogik zu implementieren.

1. Entfernen Sie die Komponente `Label`.

```
<s:Label text="Your content goes here" />
```

2. Fügen Sie in der Komponente `vGroup` die Stratus-Komponente `FieldContainer` hinzu. Die Stratus-Komponenten `FieldContainer` gruppieren Daten und vereinfachen den Prozess des Bearbeitens mehrerer Felder in einem einzelnen Vorgang. Darüber hinaus werden mit der hier erstellten Komponente `FieldContainer` die Details des ausgewählten Datensatzes oben im Bildschirm angezeigt.

Legen Sie die Komponentenbreite von `FieldContainer` mit 100% fest und geben Sie für die ID `_editFieldContainer` ein. Zu einem späteren Punkt im Code verweisen Sie auf die `FieldContainer`-ID.

```
<stratus:FieldContainer id="_editFieldContainer" width="100%"> </stratus:FieldContainer>
```

3. Fügen Sie in der Komponente `FieldContainer` eine Komponente `LabelAndField` für jedes Feld des Merchandise-Objekts hinzu, das Sie anzeigen möchten, und verweisen Sie auf jedes Feld mit seinem internen Force.com-API-Namen. Die Komponente `LabelAndField` ist eine Stratus-Komponente, die einen Salesforce.com-Feldwert mit seiner Bezeichnung anzeigt. Die Felder, die Sie mit der Komponente `FieldAndLabel` erstellen, funktionieren in Force.com Stratus-Anwendungen automatisch auf die gleiche Weise, wie sie dies in der Salesforce.com-Benutzeroberfläche tun. Beispielsweise zeigen Datenfelder einen Kalender an, wenn darauf geklickt wird. Die Felder akzeptieren auch Feldabhängigkeiten und bieten automatisch Inline-Bearbeitung, Hover-Details, Fehlerbenachrichtigungen und Hilfetext beim Infosymbol.

```
<stratus:LabelAndField field="Merchandise__c.Name" /> <stratus:LabelAndField  
field="Merchandise__c.Description__c" /> <stratus:LabelAndField  
field="Merchandise__c.Price__c" /> <stratus:LabelAndField  
field="Merchandise__c.Total_Inventory__c" />
```

4. Stellen Sie nach dem Hinzufügen der `LabelAndField`-Komponenten sicher, dass Sie das schließende `FieldContainer`-Element hinzufügen, sofern Sie dies nicht bereits getan haben.

```
</stratus:FieldContainer>
```

5. Fügen Sie nun eine Spark-Standardschaltfläche hinzu, benennen Sie sie mit **Speichern** und konfigurieren Sie sie so, dass eine Funktion mit der Bezeichnung `onEditSaveClick` ausgeführt wird. Diese Funktion schreiben Sie zu einem späteren Zeitpunkt.

```
<s:Button label="Save" click="onEditSaveClick()" />
```

6. Unter der Schaltfläche **Speichern** soll in der Inventory Tracker-Benutzeroberfläche eine Liste der Merchandise-Datensätze angezeigt werden. Zum Anzeigen der Liste verwenden Sie die ActionScript-Standardkomponente `DataGrid`. Geben Sie den Code wie folgt ein.

```
<mx:DataGrid id="dataGrid" width="100%" height="100%" dataProvider="{_gridDataProvider}"  
itemClick="onDataGridItemClick()"> <mx:columns> <mx:DataGridColumn dataField="Name" />  
<mx:DataGridColumn dataField="Description__c" headerText="Description" /> <mx:DataGridColumn
```

```
dataField="Price__c" headerText="Price"/> <mx:DataGridColumn
dataField="Total_Inventory__c" headerText="Total Inventory"/> </mx:columns> </mx:DataGrid>
```

Beachten Sie, dass das Attribut `dataProvider` auf eine Variable verweist, die später erstellt wird. Auch das Attribut `itemClick` verweist auf eine Funktion, die später erstellt wird.

- Die finale Komponente im Element `VGroup` ist die Stratus-Komponente `StatusBar`, die Teil des Standardcodes ist. Belassen Sie die Komponente unverändert und stellen Sie sicher, dass Sie das schließende Element `VGroup` hinzufügen.

```
<stratus:StatusBar/> </s:VGroup>
```

- Die Benutzeroberfläche der Inventory Tracker-Anwendung ist nun vorhanden, Sie müssen jedoch das ActionScript-Element hinzufügen, das die Salesforce.com-Daten abrufen und die Daten in der Force.com Stratus-Anwendung speichert. Beginnen Sie das Erstellen des ActionScript-Elements durch Importieren der Klassen, die Sie für das ActionScript-Element im `<![CDATA[-Abschnitt im Element <fx:Script>` benötigen. Der Code sieht wie folgt aus:

```
import mx.collections.ArrayCollection; import mx.data.IManaged; import
com.salesforce.stratus.StratusWrapper; import com.salesforce.stratus.StratusResponder;
import services.stratus.Merchandise__c; import com.salesforce.stratus.StratusMessage;
```

Die Klassen lauten wie folgt.

- `ArrayCollection`: Ein ActionScript-Wrapper, der ein Array als Sammlung darstellt, auf die zugegriffen und die bearbeitet werden kann.
  - `IManaged`: Eine ActionScript-Schnittstelle, die den Vertrag für ein verwaltetes Objekt bereitstellt.
  - `StratusWrapper`: Eine Stratus-Komponente, die den Wrap-Vorgang für die Funktionalität des Data Management Service (DMS) von Adobe durchführt. DMS ist eine Schlüsselkomponente der Adobe AIR-Plattform, die ein Modell für die Verwaltung der Datensynchronisierung zwischen Client und Server bereitstellt.
  - `StratusResponder`: Eine Stratus-Komponente, die sicherstellt, dass NULL-Antwortdienstfunktionen zu keinen internen Problemen führen.
  - `Merchandise__c`: Die ActionScript-Klasse des Merchandise-Objekts, das von Adobe Flash Builder für Force.com erstellt wird.
  - `StratusMessage`: Eine Stratus-Komponente, die standardmäßige Meldungen zu Kontextfehlern und Informationen bietet, z. B. bei Datenkonflikten.
- Erstellen Sie eine Variable, die die Merchandise-Daten umfasst. Im Code wird diese Variable verwendet, um die Daten in der Liste anzuzeigen.

```
[Bindable] private var _gridDataProvider:ArrayCollection = null;
```

- Erstellen Sie eine Variable, die im Code verwendet werden kann, um auf die DMS-Funktionalität zuzugreifen.

```
private var _stratusWrapper:StratusWrapper;
```

- Die Funktion `onWindowComplete()` im Standardcode fragt die Datenbank beim Laden der Anwendung ab. Bearbeiten Sie diese Funktion so, dass speziell Merchandise-Daten zurückgegeben werden. Die vollständige Funktion sieht folgendermaßen aus:

```
public function onWindowComplete():void { new StratusMessage(StratusMessage.STATUS_INFO,
"Application initialized").showAsStatus(); _stratusWrapper = StratusWrapper.getInstance();
```

```
_stratusWrapper.query("select * from Merchandise__c", new StratusResponder(
function(rows:ArrayCollection):void { _gridDataProvider = rows; }, null )); }
```

Sie werden sich erinnern, dass am Anfang dieses Schritts im Lernprogramm im Element Window auf eine Funktion mit der Bezeichnung `Attribut WindowComplete` verwiesen wird.

- Erstellen Sie eine Funktion mit der Bezeichnung `onEditSaveClick`, mit der die von Benutzern über die Benutzeroberfläche vorgenommenen Änderungen an den Daten wieder in den Speicher kopiert werden.

```
private function onEditSaveClick():void {
_editFieldContainer.fieldCollection.updateObject(new StratusResponder(commitToDB, null));
}
```

Dies ist die Funktion, die Benutzer durch Klicken auf die Schaltfläche **Speichern** auslösen. Diese Funktion führt auch eine einfache Validierung der Daten durch, die Benutzer eingeben. Wenn die Daten gültig sind, ruft diese Funktion die Funktion `commitToDB` auf.

- Erstellen Sie die Funktion `commitToDB`, mit der die Daten im Speicher an die Datenbank übertragen und die Daten aus `FieldContainer` gelöscht werden.

```
private function commitToDB(managedObject:IManaged):void {
_stratusWrapper.save(managedObject, new StratusResponder(
function(result:StratusMessage):void { _editFieldContainer.fieldCollection.clear(); },
null )); }
```

- Erstellen Sie eine Funktion, die die `FieldContainer`-Daten in der Force.com Stratus-Anwendung anzeigt. Verweisen Sie mit der ID auf `FieldContainer`, die Sie zuvor festgelegt haben (`_editFieldContainer`).

```
private function onDataGridItemClick():void { var merchandise:Merchandise__c =
_dataGrid.selectedItem as Merchandise__c;
_editFieldContainer.fieldCollection.render(merchandise); }
```

- Speichern Sie die Datei.

Der finale Code sollte wie folgt aussehen.

```
<?xml version="1.0" encoding="utf-8"?> <s:Window xmlns:fx="http://ns.adobe.com/mxml/2009"
xmlns:s="library://ns.adobe.com/flex/spark" xmlns:mx="library://ns.adobe.com/flex/halo"
xmlns:stratus="com.salesforce.stratus.*" showStatusBar="false" width="800" height="600"
backgroundColor="#e7e7e7" windowComplete="onWindowComplete()">

<s:layout> <s:VerticalLayout/> </s:layout>

<s:VGroup width="100%" height="100%" paddingBottom="10" paddingLeft="10" paddingRight="10"
paddingTop="10"> <stratus:FieldContainer id="_editFieldContainer" width="100%">
<stratus:LabelAndField field="Merchandise__c.Name" /> <stratus:LabelAndField
field="Merchandise__c.Description__c" /> <stratus:LabelAndField
field="Merchandise__c.Price__c" /> <stratus:LabelAndField
field="Merchandise__c.Total_Inventory__c" /> </stratus:FieldContainer> <s:Button label="Save"
click="onEditSaveClick()" /> <mx:DataGrid id="_dataGrid" width="100%" height="100%"
dataProvider="{_gridDataProvider}" itemClick="onDataGridItemClick()"> <mx:columns>
<mx:DataGridColumn dataField="Name"/> <mx:DataGridColumn dataField="Description__c"
headerText="Description"/> <mx:DataGridColumn dataField="Price__c" headerText="Price"/>
<mx:DataGridColumn dataField="Total_Inventory__c" headerText="Total Inventory"/> </mx:columns>
</mx:DataGrid> <stratus:StatusBar/> </s:VGroup>

<fx:Script> <![CDATA[ import mx.collections.ArrayCollection; import mx.data.IManaged;
import com.salesforce.stratus.StratusWrapper; import com.salesforce.stratus.StratusResponder;
import com.salesforce.stratus.StratusMessage; import services.stratus.Merchandise__c;
```

```

[Bindable] private var _gridDataProvider:ArrayCollection = null; private var
_stratusWrapper:StratusWrapper;

public function onWindowComplete():void { new StratusMessage(StratusMessage.STATUS_INFO,
"Application initialized").showAsStatus(); _stratusWrapper = StratusWrapper.getInstance();
_stratusWrapper.query("select * from Merchandise__c", new StratusResponder(
function(rows:ArrayCollection):void { _gridDataProvider = rows; }, null )); }

private function onEditSaveClick():void {
_editFieldContainer.fieldCollection.updateObject(new StratusResponder(commitToDB, null));
}

private function commitToDB(managedObject:IManaged):void {
_stratusWrapper.save(managedObject, new StratusResponder( function(result:StratusMessage):void
{ _editFieldContainer.fieldCollection.clear(); }, null )); }

private function onDataGridItemClick():void { var merchandise:Merchandise__c =
_dataGrid.selectedItem as Merchandise__c;
_editFieldContainer.fieldCollection.render(merchandise); } ]]> </fx:Script> </s:Window>

```

### Siehe auch:

[Lernprogramm 10: Erstellen einer Desktopanwendung mit Adobe Flash Builder für Force.com](#)

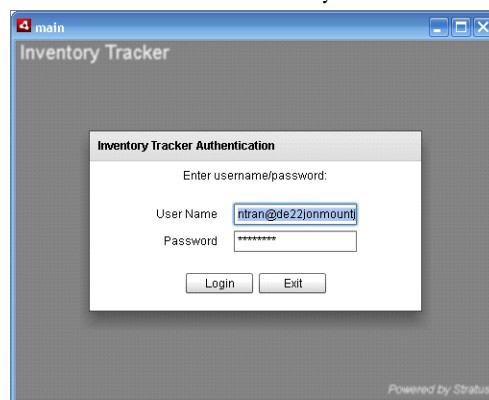
[Schritt 6: Festlegen der Attribute für die Stratus-Anwendungskomponente](#)

[Schritt 8: Testen der Inventory Tracker-Anwendung](#)

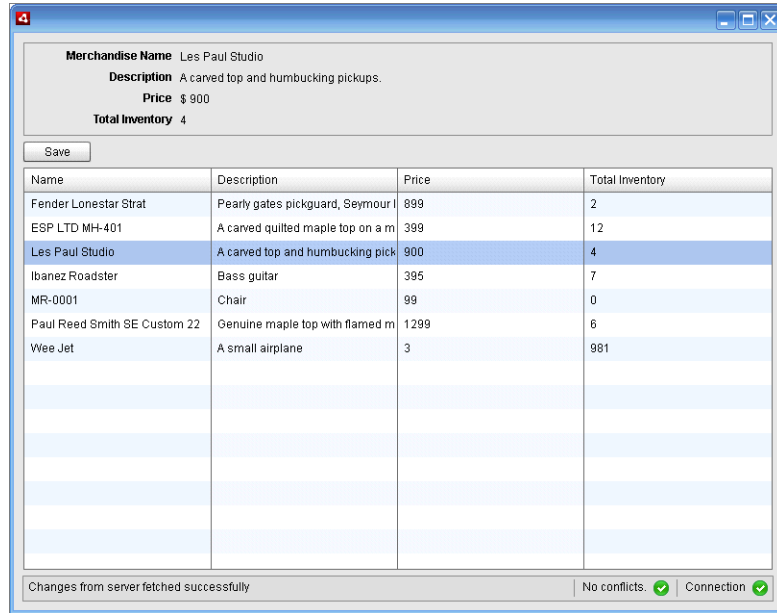
## Schritt 8: Testen der Inventory Tracker-Anwendung

Die Inventory Tracker-Anwendung kann nun getestet werden.

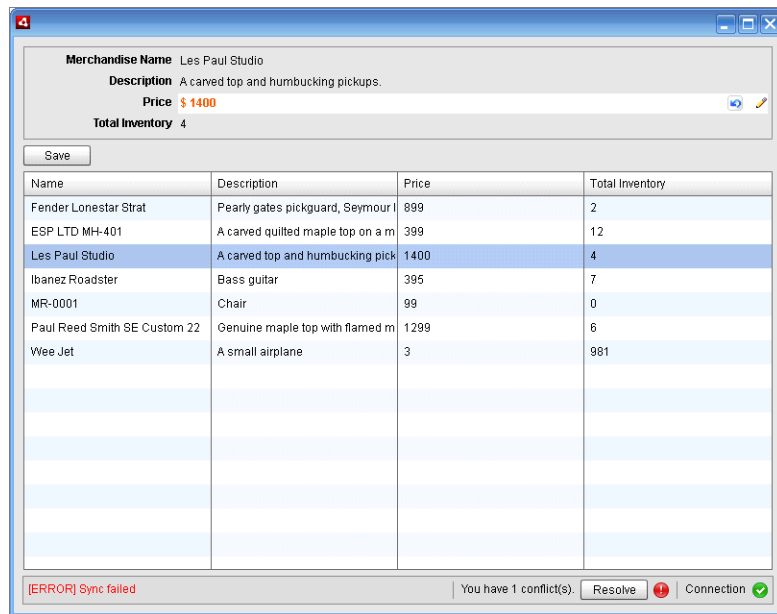
1. Stellen Sie sicher, dass der Computer eine aktive Verbindung mit dem Internet aufweist.
2. Klicken Sie in Flash Builder mit der rechten Maustaste auf das Inventory Tracker-Projekt und wählen Sie **Ausführen als Desktopanwendung** aus. Der Anmeldebildschirm der Inventory Tracker-Anwendung wird angezeigt.



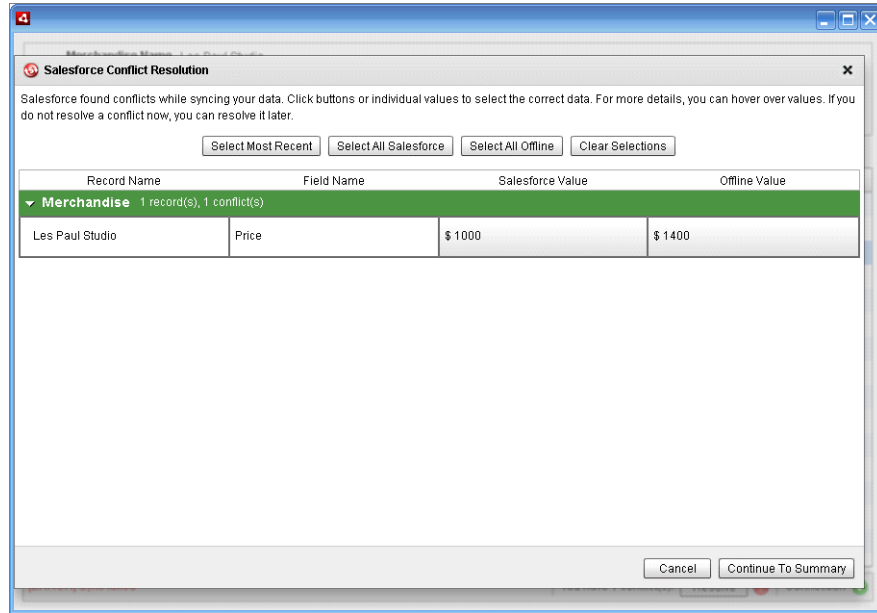
3. Geben Sie im Anmeldebildschirm den Benutzernamen und das Kennwort ein, das Sie zum Anmelden bei der Force.com-Anwendung verwenden, und klicken Sie auf **Anmelden**. Die Inventory Tracker-Benutzeroberfläche wird angezeigt.



- Wählen Sie einen Datensatz in der Liste aus und ändern Sie seine Daten. Klicken Sie dann auf **Speichern**.
- Melden Sie sich bei Ihrer Force.com-Anwendung an und prüfen Sie, ob Ihre Änderung angezeigt wird.
- Ändern Sie ein Feld in Ihrer Force.com-Anwendung und klicken Sie dann auf **Speichern**, um die Funktion zum Lösen von Datenkonflikten zu testen. Ändern Sie dann das gleiche Feld in der Inventory Tracker-Anwendung in einen anderen Wert. Wenn Sie in der Inventory Tracker-Anwendung auf **Speichern** klicken, achten Sie auf die Statusleiste, die Sie darauf hinweist, dass ein Konflikt vorliegt.



- Klicken Sie in der Statusleiste auf **Auflösen**, um die Oberfläche zum Lösen von Konflikten zu starten.



Die Oberfläche zum Lösen von Konflikten zeigt sowohl die Werte von Salesforce.com als auch der Inventory Tracker-Anwendung an. Wählen Sie den Wert aus, den Sie behalten möchten, und klicken Sie dann auf **Weiter zur Zusammenfassung**.

#### Siehe auch:

[Lernprogramm 10: Erstellen einer Desktopanwendung mit Adobe Flash Builder für Force.com](#)

[Schritt 7: Erstellen der Inventory Tracker-Fensterkomponente](#)

[Zusammenfassung](#)

## Zusammenfassung

In diesem Lernprogramm haben Sie eine einfache Force.com Stratus-Anwendung erstellt, mit der Benutzer Force.com-Daten sowohl *mit* als auch *ohne* Internetverbindung anzeigen und bearbeiten können. Sie zeigt die Grundlagen von Adobe Flash Builder für Force.com ohne viel Codierungsaufwand auf. Wie Sie sich sicherlich vorstellen können, können Sie Adobe Flash Builder für Force.com zur Erstellung von weit leistungsfähigeren Force.com Stratus-Anwendungen verwenden, um die Art und Weise zu verbessern, wie Benutzer mit Force.com-Anwendungen interagieren.

#### Siehe auch:

[Lernprogramm 10: Erstellen einer Desktopanwendung mit Adobe Flash Builder für Force.com](#)

[Schritt 8: Testen der Inventory Tracker-Anwendung](#)

# ANHANG

## Erstellen einer Sandbox und Anmelden bei dieser

Beim Einsatz der Free Edition sollte sämtliche Entwicklungsarbeit in der Sandbox durchgeführt werden. So erstellen Sie eine Sandbox:

1. Melden Sie sich bei der Free Edition-Organisation an.
2. Klicken Sie auf **Setup** ► **Datenverwaltung** ► **Sandbox**.
3. Klicken Sie auf **Neue Sandbox**.
4. Geben Sie unter **Name** einen Namen an, beispielsweise **Entwicklung**.
5. Klicken Sie auf **Version starten**.

Das Starten einer Version der Produktionsorganisation nimmt etwas Zeit in Anspruch. Aktualisieren Sie Ihren Browser und achten Sie auf das Feld **Status**, um den Fortschritt anzuzeigen. Falls Sie sich nicht mit dem Beobachten des Fortschritts aufhalten möchten, können Sie andere Arbeiten aufgreifen und Sie werden dann per E-Mail benachrichtigt, wenn der Vorgang abgeschlossen ist.

Melden Sie sich an, sobald die Sandbox verfügbar ist:

1. Klicken Sie auf **Setup** ► **Datenverwaltung** ► **Sandbox**, wenn Sie während des Erstellens der Sandbox-Version zu einer anderen Seite navigiert sind.
2. Klicken Sie neben Ihrer Sandbox auf **Anmelden**.
3. Im Browser wird eine Anmeldeseite geöffnet. Geben Sie Ihr Kennwort ein und klicken Sie auf **Anmelden**.

### Weitere Informationen

Nach dem Anmelden bemerken Sie möglicherweise, dass das Feld **User Name** von Ihrem regulären Benutzernamen abweicht. Force.com hängt Ihren Sandbox-Namen automatisch an Ihren Benutzernamen an, sodass Sie sich bei mehreren Sandbox-Instanzen anmelden können. Darüber hinaus haben Sie möglicherweise bemerkt, dass in Ihrem Browser ein anderer URL angezeigt wird. Dies liegt daran, dass jede Sandbox eine eigene Instanz aufweist, die von jener abweicht, die von Ihrer regulären Organisation verwendet wird.

## Erstellen eines Projekts in der Force.com-IDE

So erstellen Sie ein Projekt in der Force.com-IDE:

1. Wählen Sie **Datei** ► **Neu** ► **Force.com-Projekt** aus.
2. Geben Sie die Anmeldeinformationen Ihrer Organisation ein.
  - a. Geben Sie einen Namen für das Projekt ein. In einer Organisation können mehrere Projektname vorhanden sein, es ist also hilfreich, jedes Projekt mit einem beschreibenden Namen identifizieren zu können. Nennen Sie dieses Projekt **Workbook**.
  - b. Geben Sie den Benutzernamen ein, mit dem Sie sich als Administrator anmelden. Der Benutzername liegt in Form einer E-Mail-Adresse vor.

- c. Geben Sie Ihr Kennwort ein.
- d. Wenn Sie zum Entwickeln eine Sandbox-Instanz verwenden, wählen Sie im Feld `Environment` in der Dropdown-Liste den Wert `Sandbox` aus.
- e. Klicken Sie auf **Weiter**.



**Anmerkung:** Wenn der Fehler "LOGIN\_MUST\_USE\_SECURITY\_TOKEN" angezeigt wird, wird die aktuelle IP nicht von einem vertrauenswürdigen Netzwerkbereich erkannt. Sie müssen das Sicherheitstoken zurücksetzen. Klicken Sie auf der Salesforce.com-Benutzeroberfläche auf **Setup > Meine persönlichen Daten > Mein Sicherheitstoken zurücksetzen** und befolgen Sie die dortigen Anweisungen.

3. Klicken Sie auf der Seite "Projekthinhalte" auf **Fertig stellen**, um die Apex- und Visualforce-Komponenten abzurufen.

## Bereitstellen Ihrer Anwendung

Da Ihre Anwendung nun vollständig ist, möchten Sie sicher Informationen zu den verschiedenen Möglichkeiten des Bereitstellens erfahren. Wenn diese Anwendung entwickelt wurde, um in mehreren Organisationen installiert zu werden, muss ein *Paket* erstellt werden. Wenn diese Anwendung nur für eine Organisation entwickelt wurde, wird sie *bereitgestellt*.

### Informationen zu Paketen

Pakete sind wie Container, die Komponenten, Code und Anwendungen enthalten können. Sie können sie verwenden, um eine Anwendung an mehrere Organisationen zu verteilen. Die Force.com-Plattform kennt zwei Pakettypen: *nicht verwaltet* und *verwaltet*.

- Nicht verwaltete Pakete werden verwendet, um Kopien der gebündelten Komponenten inklusive Quellcode zu verteilen. Hierbei handelt es sich in der Regel um kostenlose Pakete ohne Upgradeoption. Nachdem ein nicht verwaltetes Paket in einer anderen Organisation installiert wurde, haben Sie als Herausgeber keine Möglichkeit mehr zu kontrollieren, was mit den enthaltenen Komponenten geschieht.
- Verwaltete Pakete werden verwendet, um Kopien der gebündelten Komponenten so zu verteilen, dass Sie als Herausgeber die Kontrolle über das geistige Eigentum und die Lizenzierung haben. Bei verwalteten Paketen können Sie darüber hinaus auf einfache Weise Upgrades an Kunden verteilen, bei denen ältere Versionen der Anwendung installiert sind. Verwaltete Pakete werden bevorzugt, wenn es um das Veröffentlichen von Anwendungen auf Force.com AppExchange geht.

Weitere Informationen zu Paketen finden Sie unter "Übersicht über Pakete" in der Force.com-Onlinehilfe.

**Informationen zum Bereitstellen**

Wenn Sie den Code in einer Produktionsorganisation bereitstellen möchten, für die Sie über Anmeldedaten verfügen, sind mehrere Optionen verfügbar:

- Wenn Sie mit Befehlszeilenschnittstellen vertraut sind, verwenden Sie zur Bereitstellung das Tool für die Force.com-Migration. Weitere Informationen finden Sie unter [wiki.developerforce.com/index.php/Migration\\_Tool\\_Guide](http://wiki.developerforce.com/index.php/Migration_Tool_Guide).
- Wenn Sie eine grafische Oberfläche bevorzugen, verwenden Sie die Force.com-IDE. Weitere Informationen finden Sie unter [wiki.developerforce.com/index.php/Force.com\\_IDE](http://wiki.developerforce.com/index.php/Force.com_IDE).