



Manuel d'utilisation Force.com: Salesforce Summer '10

Manuel d'utilisation Force.com



Table des matières

À propos du Manuel d'utilisation Force.com.....	3
Didacticiel n° 1 : Création d'une application Entrepôt.....	5
Étape 1 : Création d'un objet personnalisé Article.....	5
Étape 2 : Ajout des champs Description, Prix et Inventaire total.....	7
Étape 3 : Création d'un onglet.....	9
Étape 4 : Création d'une sous-application Entrepôt.....	10
Étape 5 : Création d'un enregistrement Article.....	11
Résumé.....	12
Didacticiel n° 2 : Ajout de relations.....	13
Étape 1 : Création d'un objet personnalisé Relevé de facture.....	13
Étape 2 : Création d'un objet Élément de ligne.....	16
Étape 3 : Association des objets.....	17
Étape 4 : Ajout de relevés de facture à la sous-application.....	18
Étape 5 : Création d'un enregistrement Facture.....	18
Résumé.....	20
Didacticiel n° 3 : Utilisation de formules et de règles de validation.....	21
Étape 1 : Calcul d'une valeur pour chaque Élément de ligne.....	21
Étape 2 : Calcul d'une valeur pour le Relevé de facture avec un champ récapitulatif de cumul.....	22
Étape 3 : Contrôle d'inventaire avec une règle de validation.....	23
Étape 4 : Test de la règle de validation.....	25
Étape 5 : Amélioration de la règle de validation.....	26
Résumé.....	27
Didacticiel n° 4 : Utilisation de workflow et d'approbations.....	28
Étape 1 : Renseignement du prix d'élément de ligne à l'aide d'une règle de workflow.....	29
Étape 2 : Mise à jour de l'inventaire total lors d'une commande.....	30
Étape 3 : Test des règles de workflow.....	32
Étape 4 : Création d'un modèle d'e-mail.....	33
Étape 5 : Création d'un processus d'approbation.....	34
Étape 6 : Création d'un profil de sécurité personnalisé.....	36
Étape 7 : Création d'un utilisateur.....	37
Étape 8 : Test du processus d'approbation.....	38
Résumé.....	40
Didacticiel n° 5 : Ajout d'une logique commerciale avec Apex.....	41
Étape 1 : Création d'une définition de déclencheur Apex.....	42
Étape 2 : Définition d'une variable de liste.....	43
Étape 3 : Itération sur une liste et modification du prix.....	44
Étape 4 : Test du déclencheur.....	45
Résumé.....	46
Didacticiel n° 6 : Ajout de tests à l'application.....	47
Étape 1 : Création d'une classe de test Apex.....	48
Étape 2 : Ajout de méthodes de test à la classe.....	48

Étape 3 : Ecriture du code exécutant le déclencheur.....	49
Étape 4 : Exécution du test.....	50
Étape 5 : Affichage de la couverture de code et amélioration des tests.....	51
Résumé.....	53
Didacticiel n° 7 : Élaboration d'une interface utilisateur personnalisée en utilisant Visualforce.....	54
Étape 1 : Activation du Mode de développement de Visualforce.....	54
Étape 2 : Création d'une page Visualforce.....	55
Étape 3 : Ajout d'une ressource statique Feuille de style.....	56
Étape 4 : Ajout d'un contrôleur à la page.....	58
Étape 5 : Affichage de la Feuille d'inventaire en tant que page Visualforce.....	59
Résumé.....	61
Didacticiel n° 8 : Création d'une page Web publique en utilisant les Sites.....	62
Étape 1 : Création d'une page de catalogue de produits.....	62
Étape 2 : Enregistrement d'un nom de domaine Force.com.....	64
Étape 3 : Création d'un site Force.com.....	65
Étape 4 : Configuration et test du site.....	65
Résumé.....	67
Didacticiel n° 9 : Création d'une boutique.....	68
Étape 1 : Création d'un contrôleur.....	68
Étape 2 : Ajout de méthodes au contrôleur.....	69
Étape 3 : Création de la boutique.....	70
Étape 4 : Bonus - Mise à jour de la page avec AJAX.....	72
Résumé.....	73
Didacticiel n° 10 : Création d'une sous-application de bureau avec Adobe Flash Builder for Force.com.....	75
Étape 1 : Configuration des informations personnelles Salesforce.com.....	76
Étape 2 : Création d'une Configuration de porte-documents hors-ligne.....	77
Étape 3 : Génération de votre WDSL Entreprise et téléchargement d'un certificat client.....	79
Étape 4 : Installation et lancement de Adobe Flash Builder for Force.com.....	80
Étape 5 : Création d'un projet Force.com Stratus.....	80
Étape 6 : Définition des attributs du composant Stratus Application.....	82
Étape 7 : Création d'un composant Window Suivi Inventaire.....	83
Étape 8 : Test de la sous-application Suivi Inventaire.....	87
Résumé.....	89
Annexe.....	90
Création et connexion à un environnement test sandbox.....	90
Création d'un projet dans l'IDE Force.com.....	90
Distribution de votre application.....	91

À propos du Manuel d'utilisation Force.com

Le Manuel d'utilisation Force.com présente comment créer une application dans une série de didacticiels. Vous pouvez utiliser la plate-forme Force.com pour élaborer toutes sortes d'applications, mais la plupart des applications partagent certaines caractéristiques, notamment :

- Une base de données pour modéliser les informations dans l'application
- Une logique commerciale et un workflow pour effectuer des tâches spécifiques dans certaines conditions
- Une interface utilisateur pour présenter les données et les fonctionnalités aux utilisateurs connectés à votre application
- Un site Internet public pour montrer les données et les fonctionnalités sur le Web



Les didacticiels sont axés sur l'élaboration d'un système très simple de gestion d'un entrepôt. Vous allez développer entièrement l'application, c'est-à-dire que vous allez commencer par élaborer un modèle de base de données pour le suivi des articles. Vous allez ensuite ajouter une logique commerciale : des règles de validation pour s'assurer que le stock est suffisant, un workflow pour mettre à jour l'inventaire lors d'une vente, des approbations pour envoyer des notifications par e-mail pour les valeurs de facturation importantes et une logique de déclenchement pour mettre à jour les prix dans les factures en cours. Une fois la base de données et la logique commerciale terminées, vous allez créer une interface utilisateur pour montrer l'inventaire d'un produit au personnel, un site Web public pour présenter un catalogue de produits, puis les premiers éléments d'une boutique simple. Pour le développement hors ligne et l'intégration à la sous-application, nous avons ajouté un dernier didacticiel qui présente l'utilisation de Adobe Flash Builder for Force.com.

Chaque didacticiel repose sur les didacticiels précédents selon une progression logique dans le développement de l'application, et présente en même temps une fonctionnalité particulière de la plate-forme. Cela paraît beaucoup, mais vous allez rapidement apprécier la simplicité des points abordés.

Choix d'un environnement de développement

Le manuel est conçu pour être utilisé dans une organisation Developer Edition qui offre l'ensemble des fonctionnalités et fonctions requises. Pour vous abonner gratuitement à une organisation Developer Edition, accédez à la page developer.force.com/join.

Vous pouvez toutefois utiliser la version Free Edition, en tenant compte des points ci-dessous.

- Certains didacticiels nécessitent l'utilisation d'un environnement test sandbox. Il est par conséquent plus facile de terminer tous les didacticiels au sein d'une organisation test sandbox. Si vous n'avez pas encore créé d'environnement test sandbox, reportez-vous à [Création et connexion à un environnement test Sandbox](#) à la page 90.
- Vous ne pouvez pas utiliser la version Free Edition pour terminer le Didacticiel n° 10 : Création d'une sous-application de bureau avec Adobe Flash Builder for Force.com. En tant qu'aperçu pour développeur, Adobe Flash Builder for Force.com est disponible uniquement dans Developer Edition.
- Les images du manuel peuvent être légèrement différentes de celles affichées sur votre écran. Par exemple, Developer Edition utilise un onglet Démarrer ici, tandis que Free Edition inclut un onglet Premiers pas. Dans la plupart des cas, les différences sont insignifiantes.

- Une fois l'application terminée, vous pouvez migrer les modifications depuis votre environnement test vers un environnement de production. Pour plus d'informations, reportez-vous à [Distribution de votre application](#) à la page 91.

Pour vous abonner à une organisation Free Edition, accédez à salesforce.com/form/signup/freeforce-platform.jsp.

Organisation du manuel

Les didacticiels de ce manuel se complètent et s'inscrivent dans une progression logique.

Didacticiel	Niveau	Durée estimée
Créer la sous-application		
Didacticiel n° 1 : Création d'une application Entrepôt	Débutant	20 à 30 minutes
Didacticiel n° 2 : Ajout de relations	Débutant	20 à 30 minutes
Ajouter une logique		
Didacticiel n° 3 : Utilisation de formules et de règles de validation	Débutant	20 à 30 minutes
Didacticiel n° 4 : Utilisation de workflow et d'approbations	Intermédiaire	30 à 45 minutes
Didacticiel n° 5 : Ajout d'une logique commerciale avec Apex	Avancé	20 à 30 minutes
Ecrire des tests		
Didacticiel n° 6 : Ajout de tests à l'application	Avancé	20 à 30 minutes
Étendre l'interface utilisateur		
Didacticiel n° 7 : Élaboration d'une interface utilisateur personnalisée en utilisant Visualforce	Intermédiaire	30 à 45 minutes
Didacticiel n° 8 : Création d'une page Web publique en utilisant les Sites	Intermédiaire	20 à 30 minutes
Didacticiel n° 9 : Création d'une boutique	Intermédiaire	30 minutes
Intégration hors ligne		
Didacticiel n° 10 : Création d'une sous-application de bureau avec Adobe Flash Builder for Force.com.	Avancé	30 minutes
Annexe		
Création et connexion à un environnement test sandbox		
Création d'un projet dans l'IDE Force.com		
Distribution de votre application		

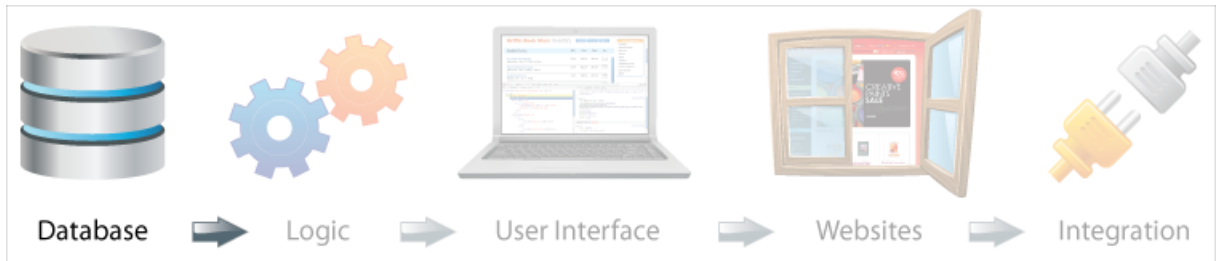
En savoir plus

Chaque étape se termine par une section facultative En savoir plus. Pour avancer rapidement, passez à l'étape suivante. Si toutefois vous souhaitez approfondir, ces sections contiennent de nombreuses informations utiles et des étapes supplémentaires pour étoffer votre sous-application.

- Pour en savoir plus sur Force.com et accéder à un vaste ensemble de ressources, visitez le site Developer Force à l'adresse developer.force.com.
- La dernière version de ce Manuel et des didacticiels complémentaires permettant d'étendre la sous-application sont disponibles à la page developer.force.com/workbook.

Didacticiel n° 1 : Création d'une application Entrepôt

Niveau : Débutant ; **Durée :** 20 à 30 minutes



Cette application repose sur ce que vous vendez : des articles. Vous commencez par créer un objet de données qui permet de suivre tous les éléments relatifs à un article spécifique, par exemple son nom, sa description, son prix, etc. Dans la plate-forme Force.com, ces objets de données sont appelés *objets personnalisés*. Si vous maîtrisez le concept de bases de données, vous pouvez comparer un objet personnalisé à un tableau.

Un objet contient des champs standard, ainsi que des écrans qui permettent de répertorier, d'afficher et de modifier les informations relatives à l'objet. Vous pouvez toutefois ajouter vos propres champs afin de suivre ou de répertorier tout élément de votre choix. Une fois ce didacticiel terminé, vous disposerez d'une application fonctionnelle dotée de son propre menu, d'un onglet et d'un objet personnalisé permettant de suivre le nom, la description et le prix de vos articles, ainsi que d'écrans d'affichage et de modification de ces informations.

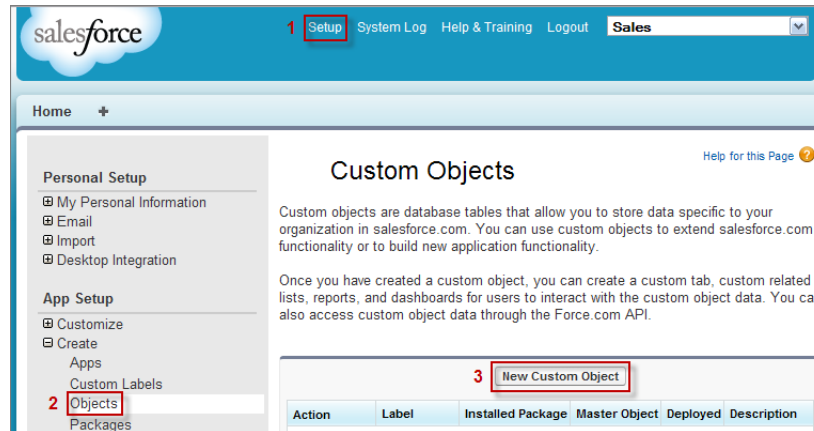
Voir aussi :

- [Étape 1 : Création d'un objet personnalisé Article](#)
- [Étape 2 : Ajout des champs Description, Prix et Inventaire total](#)
- [Étape 3 : Création d'un onglet](#)
- [Étape 4 : Création d'une sous-application Entrepôt](#)
- [Étape 5 : Création d'un enregistrement Article](#)
- [Résumé](#)

Étape 1 : Création d'un objet personnalisé Article

La plate-forme Force.com inclut des objets standard prédéfinis, mais vous pouvez également créer des objets personnalisés afin de suivre tout élément de votre choix. La première étape consiste à créer un objet personnalisé capable de contenir les informations sur l'article.

1. Lancez un navigateur et accédez à la page <https://login.salesforce.com>.
2. Saisissez votre nom d'utilisateur (sous forme d'adresse de messagerie) et votre mot de passe.
3. Cliquez sur **Configuration** dans le coin supérieur droit, puis sur **Créer** ➤ **Objets** dans le menu latéral pour afficher la page Objets personnalisés.



4. Cliquez sur **Nouvel objet personnalisé** pour afficher l'assistant Nouvel objet personnalisé.
5. Renseignez la définition de l'objet personnalisé.
 - Dans le champ Étiquette, saisissez Article.
 - Dans le champ Étiquette - pluriel, saisissez Articles.
 - Laissez les autres champs inchangés.

Custom Object Definition Edit Save Save & New Cancel

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

6. Cliquez sur **Enregistrer** pour terminer la création de l'objet.

En savoir plus

Une fois le nouvel objet personnalisé enregistré, la page de détails de l'objet personnalisé s'ouvre. Observez l'image ci-dessous pour vous familiariser avec l'objet personnalisé Article.

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 Delete

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

4 No custom fields defined

1. Page de détails Article : cette page affiche toutes les informations utiles relatives à votre objet personnalisé Article. Vous allez bientôt ajouter des champs, des relations, des règles de validation et d'autres fonctionnalités simples pour cet objet personnalisé.
2. Nom API : lors de la création de l'objet Article, vous n'avez spécifié aucun nom d'API, il a été automatiquement généré. Ce nom sert à référencer l'objet dans la programmation. Tous les objets personnalisés se terminent par __c, qui permet de les différencier des objets standard et de faciliter leur identification.
3. Champs standard : certains champs, appelés champs standard, sont automatiquement générés. Par exemple, l'objet Article contient un champ standard Responsable, qui suit automatiquement le créateur de chaque enregistrement.
4. Champs personnalisés : vous n'avez pas encore défini de champ personnalisé, vous le ferez à l'étape suivante. Si vous avez deviné que les noms des champs personnalisés se terminent par __c, bien vu !

Voir aussi :

[Didacticiel n° 1 : Création d'une application Entrepôt](#)

[Étape 2 : Ajout des champs Description, Prix et Inventaire total](#)

Étape 2 : Ajout des champs Description, Prix et Inventaire total

Un objet article doit contenir les champs qui permettent de suivre diverses informations, notamment la description, le coût et la quantité en stock. Vous pouvez ajouter des champs afin de répertorier ou de suivre tout élément de votre choix.

Suivez les étapes ci-dessous afin de créer des champs personnalisés pour la description, le prix et l'inventaire.

1. Créez le champ Description.
 - a. Accédez à la liste associée Champs personnalisés & relations.
 - b. Cliquez sur **Nouveau** pour lancer l'assistant Nouveau champ personnalisé.
 - c. Dans la zone Type de données, sélectionnez `Zone de texte`, puis cliquez sur **Suivant**.

d. Renseignez les détails des champs personnalisés.

- Dans le champ Étiquette du champ, saisissez Description et appuyez sur la touche Tabulation.
- Le champ Nom du champ doit être automatiquement renseigné avec le nom de champ : Description.
- Sélectionnez la case Obligatoire. L'une de vos règles commerciales indique que vous ne pouvez pas créer d'article sans fournir une description. Rendre le champ Description obligatoire satisfait cette règle commerciale.
- Vous pouvez renseigner les champs facultatifs Description et Texte d'aide. Il est recommandé de renseigner ces champs, mais nous les laissons de côté dans ce didacticiel pour gagner du temps.
- Ne renseignez pas le champ Valeur par défaut.

e. Cliquez sur **Suivant**, acceptez les valeurs par défaut, puis cliquez de nouveau sur **Suivant**.

f. Cliquez sur **Enregistrer et Nouveau** pour enregistrer le champ Description et revenir à la première étape de l'assistant.

2. Créez le champ Prix.

a. Dans la zone Type de données, sélectionnez Devise, puis cliquez sur **Suivant**.

b. Renseignez les détails des champs personnalisés :

- Dans le champ Étiquette du champ, saisissez Prix.
- Dans le champ Longueur, saisissez 16, puis saisissez 2 dans le champ Décimales.
- Sélectionnez la case Obligatoire.

c. Laissez les autres champs inchangés, puis cliquez sur **Suivant**.

d. À l'étape suivante, acceptez les valeurs par défaut, puis cliquez sur **Suivant**.

e. À l'étape suivante, cliquez sur **Enregistrer et nouveau** pour enregistrer le champ Prix et revenir à la première étape de l'assistant.



Remarque: Cette sous-application ne contient qu'un seul champ Prix que nous allons utiliser pour le cours et le prix de détail. Vous pouvez cependant créer un autre champ Cours si vous le souhaitez, il suffit de répéter les étapes ci-dessus et d'utiliser le nom Cours.

3. Créez le champ Inventaire total.

- a. Dans la zone Type de données, sélectionnez Numéro, puis cliquez sur **Suivant**.
- b. Renseignez les détails des champs personnalisés :
 - Dans le champ Étiquette du champ, saisissez Inventaire total.
 - Sélectionnez la case Obligatoire.
- c. Acceptez les valeurs par défaut des autres champs, puis cliquez sur **Suivant**.
- d. À l'étape suivante, acceptez les valeurs par défaut, puis cliquez sur **Suivant**.
- e. Cliquez sur **Enregistrer** pour créer le champ Inventaire et revenir à la page Objet personnalisé Article.

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

À ce stade, nous disposons d'une présentation claire de nos articles d'inventaire, avec un nom, une description et un prix. Nous avons également enregistré la quantité (Inventaire total) dont nous disposons pour chaque article.

En savoir plus

Peut-être avez-vous remarqué que lors de la saisie de l'Étiquette du champ, le système a automatiquement renseigné le champ Nom du champ. Pourquoi avez-vous besoin des deux champs ? Une Étiquette du champ est visible par l'utilisateur, il doit être facile à lire et peut contenir des espaces. Le Nom du champ est utilisé en interne, dans le code, et ne doit pas contenir d'espace ou des caractères spéciaux. Par exemple, le champ étiqueté « Client ph# » serait nommé Client_ph dans le code (le système remplace les espaces par des caractères de soulignement, et supprime les caractères # et :).

Voir aussi :

- [Didacticiel n° 1 : Création d'une application Entrepôt](#)
- [Étape 1 : Création d'un objet personnalisé Article](#)
- [Étape 3 : Création d'un onglet](#)

Étape 3 : Création d'un onglet

Les onglets permettent de rechercher et d'organiser aisément les objets et les enregistrements. Dans cette étape, vous allez créer un onglet pour l'objet personnalisé Article. Lorsque les utilisateurs cliquent sur cet onglet, ils peuvent créer, afficher et modifier les enregistrements.

1. Dans le menu de Configuration, cliquez sur **Créer ► Onglets**.
2. Dans la liste associée Onglets de l'objet personnalisé, cliquez sur **Nouveau** pour lancer l'assistant Onglet Nouvel objet personnalisé.
3. Dans la liste déroulante Objet, sélectionnez Article.
4. Cliquez sur l'icône de recherche en regard du champ Style d'onglet, puis sélectionnez l'icône Boîte.

New Custom Object Tab

Step 1. Enter the Details

Choose the custom object for this new custom tab. Fill in other details.

Select an existing custom object or [create a new custom object now](#).

Object: Merchandise

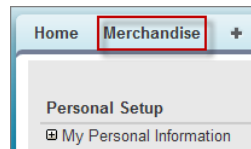
Tab Style: Box

(Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab.

Splash Page Custom Link: --None--

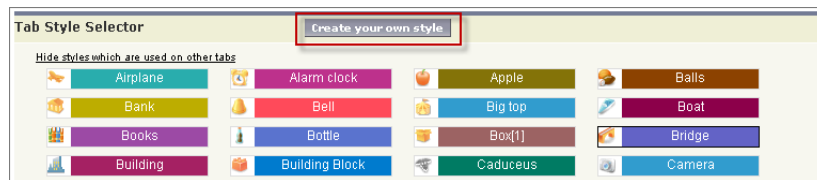
5. Acceptez les autres valeurs par défaut, puis cliquez sur **Suivant**.
6. Cliquez sur **Suivant**, puis sur **Enregistrer** pour terminer la création de l'onglet.

Une fois l'onglet créé, il s'affiche en haut de l'écran.



En savoir plus

Lors de la création d'un onglet, vous n'êtes pas limité à un ensemble d'icônes et de couleurs standard, vous pouvez sélectionner la couleur de votre choix et une image personnalisée en cliquant sur **Créer votre propre style**.



Voir aussi :

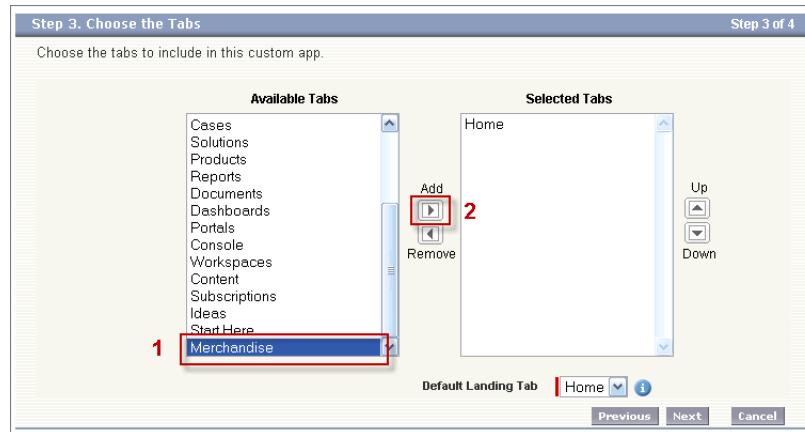
- [Didacticiel n° 1 : Création d'une application Entrepôt](#)
- [Étape 2 : Ajout des champs Description, Prix et Inventaire total](#)
- [Étape 4 : Création d'une sous-application Entrepôt](#)

Étape 4 : Création d'une sous-application Entrepôt

Une application est un ensemble d'onglets. Dans cette étape, vous allez créer une application Entrepôt et lui ajouter un onglet. Vous lui ajouterez plus tard des onglets supplémentaires.

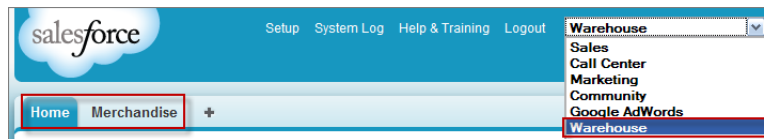
1. Cliquez sur **Configuration** ► **Créer** ► **Sous-applications**.
2. Cliquez sur **Nouveau** pour lancer l'assistant Nouvelle sous-application personnalisée.
3. Renseignez les détails des champs de la sous-application.
 - Dans le champ Étiquette de sous-application, saisissez Entrepôt.
 - Dans le champ Nom de la sous-application, saisissez Entrepôt.

4. Cliquez sur **Suivant**.
5. Acceptez les valeurs par défaut de cette page, en acceptant le logo par défaut pour la sous-application, puis cliquez sur **Suivant**.
6. Dans la liste Onglets disponibles, sélectionnez l'onglet Articles, puis cliquez sur **Ajouter** pour l'ajouter à la liste des onglets sélectionnés.



7. Laissez le champ Onglet de destination par défaut défini sur l'onglet Accueil, puis cliquez sur **Suivant**.
8. Sélectionnez la case **Visible** pour rendre l'application disponible pour tous les profils d'utilisateur.
9. Cliquez sur **Enregistrer** pour créer l'application Entrepôt.

Une fois l'application créée, elle s'affiche dans le menu Sous-applications Force.com, en haut à droite de la page.



En savoir plus

Les onglets de cette application ne doivent pas obligatoirement être associés. Vous pouvez modifier les applications personnalisées afin de regrouper à un emplacement commun les onglets que vous utilisez le plus souvent. Par exemple, si vous avez souvent recours à l'onglet Démarrer ici, vous pouvez l'ajouter à la sous-application Entrepôt. Vous pouvez basculer vers l'une des applications que vous avez créées, achetées ou installées en sélectionnant simplement son nom dans le menu.

Voir aussi :

[Didacticiel n° 1 : Création d'une application Entrepôt](#)

[Étape 3 : Création d'un onglet](#)

[Étape 5 : Création d'un enregistrement Article](#)

Étape 5 : Création d'un enregistrement Article

À ce stade, vous avez créé une application qui fonctionne. Lors de la définition d'un objet dans Force.com, la plate-forme génère automatiquement une interface utilisateur qui permet de créer, de lire, de mettre à jour et de supprimer des enregistrements. Observez maintenant le fonctionnement de votre sous-application.

1. Sélectionnez l'application Entrepôt dans le menu des sous-applications Force.com.

2. Cliquez sur l'onglet Articles, puis cliquez sur **Nouveau** pour créer un produit.
3. Renseignez tous les champs.

- Dans le champ Nom de l'article, saisissez Wee Jet.
- Dans le champ Description, saisissez Un petit avion.
- Dans le champ Prix, saisissez 9,99.
- Dans le champ Inventaire total, saisissez 2000.

comme indiqué sur l'image suivante.

4. Cliquez sur **Enregistrer**.

En savoir plus

Votre entrepôt peut fonctionner avec un seul article, mais il serait plus réaliste de lui ajouter des articles. Créez davantage d'articles. Astuce : utilisez le bouton **Enregistrer et Nouveau** pour créer rapidement des enregistrements.

Voir aussi :

[Didacticiel n° 1 : Création d'une application Entrepôt](#)

[Étape 4 : Création d'une sous-application Entrepôt](#)

[Résumé](#)

Résumé

Félicitations ! Vous venez de créer une application Entrepôt qui permet de suivre les produits que vous possédez en stock. Comme vous l'avez constaté, il est facile d'utiliser l'interface en ligne pour créer un objet personnalisé, des champs et un onglet, puis de les organiser dans une application permettant de créer et de suivre les informations associées à vos articles. Bien que l'application soit très incomplète, elle contient déjà de nombreuses fonctionnalités intégrées, notamment la possibilité d'afficher et de créer des articles.

L'étape suivante présente l'utilisation d'une facture afin de suivre les mouvements d'entrée et de sortie des articles dans l'entrepôt.

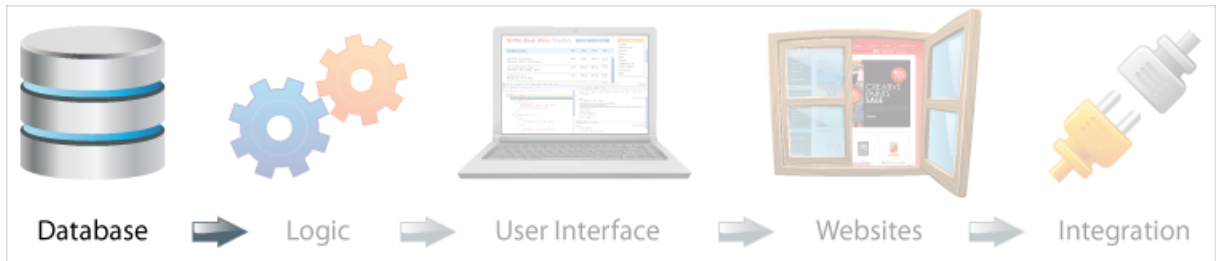
Voir aussi :

[Didacticiel n° 1 : Création d'une application Entrepôt](#)

[Étape 5 : Création d'un enregistrement Article](#)

Didacticiel n° 2 : Ajout de relations

Niveau : Débutant ; Durée : 20 à 30 minutes



Ce didacticiel présente comment créer deux objets, un relevé de facture et un élément de ligne, puis comment les associer dans une relation principal-détails. Cette relation permet de composer plusieurs éléments de ligne et de les associer à un seul relevé de facture. Chaque élément de ligne indique la quantité d'unités vendues pour un article particulier. Vous allez par conséquent ajouter une relation entre les articles et les objets d'éléments de ligne.

Si vous maîtrisez le concept de bases de données relationnelles, vous pouvez comparer une relation à des clés étrangères, à la différence que Force.com effectue la mise en œuvre sous-jacente pour vous, vous vous contentez de définir les associations entre les objets.

Prérequis

Sous-application Entrepôt

Vous devez commencer par créer une application Entrepôt de base en suivant les instructions du [Didacticiel n° 1 : Création d'une application Entrepôt](#) à la page 5.

Voir aussi :

[Étape 1 : Création d'un objet personnalisé Relevé de facture](#)

[Étape 2 : Création d'un objet Élément de ligne](#)

[Étape 3 : Association des objets](#)

[Étape 4 : Ajout de relevés de facture à la sous-application](#)

[Étape 5 : Création d'un enregistrement Facture](#)

[Résumé](#)

Étape 1 : Création d'un objet personnalisé Relevé de facture

Un relevé de facture est requis pour les mouvements d'entrée et de sortie d'inventaire dans l'entrepôt. Dans cette étape, vous allez créer un relevé de facture avec un numéro unique, un statut et une description.

1. Cliquez sur **Configuration** ► **Créer** ► **Objets**.
2. Cliquez sur **Nouvel objet personnalisé**, puis définissez l'objet personnalisé.
 - Dans le champ Étiquette, saisissez Relevé de facture
 - Dans le champ Étiquette - pluriel, saisissez Relevés de facture
 - Dans le champ Nom de l'enregistrement, saisissez Numéro de facture
 - Dans le champ Type de données, sélectionnez Numéro automatique

- Dans le champ Format d'affichage, saisissez FACT-{0000}
- Dans le champ Numéro de début, saisissez 1.

Custom Object Information ! = Required Information

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

Label	Invoice Statement	Example: Account
Plural Label	Invoice Statements	Example: Accounts

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

Object Name	Invoice_Statement	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 -None-

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	Invoice Number	Example: Account Name
--------------------	----------------	------------------------------

Data Type Auto Number

Display Format	INV-{0000}	Example: A-{0000} What is This?
-----------------------	------------	--

Starting Number 1

3. Laissez les autres valeurs inchangées, puis cliquez sur **Enregistrer**.

4. Ajoutez un champ Statut.

- a. Accédez à la liste associée Champs personnalisés & relations, puis cliquez sur **Nouveau**.
- b. Dans la zone Type de données, sélectionnez Liste de sélection, puis cliquez sur **Suivant**.
- c. Renseignez les détails des champs personnalisés.
 - Dans le champ Étiquette du champ, saisissez Statut.
 - Saisissez les valeurs de liste de sélection suivantes dans la zone fournie, une entrée par ligne.

En cours Fermé Négociation En attente

- Sélectionnez la case Utiliser la première valeur comme valeur par défaut.

Step 2. Enter the details Step 2 of 4

Previous Next Cancel

Field Label ⓘ

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 ⓘ

Description

Help Text

Previous Next Cancel

- d. Cliquez sur **Suivant**.
 - e. Dans la zone Sécurité au niveau du champ, sélectionnez **Lecture seule**, puis cliquez sur **Suivant**.
 - f. Cliquez sur **Enregistrer et Nouveau**.
5. Créez maintenant un champ Description facultatif.
 - a. Dans la zone Type de données, sélectionnez Zone de texte, puis cliquez sur **Suivant**.
 - b. Dans les champs Étiquette du champ et Nom du champ, saisissez Description.
 - c. Cliquez sur **Suivant**, acceptez les valeurs par défaut, puis cliquez de nouveau sur **Suivant**.
 - d. Cliquez sur **Enregistrer** pour accéder à la page de détails de l'objet Relevé de facture.

Votre objet Relevé de facture doit désormais inclure deux champs personnalisés, comme illustré ici.

Custom Fields & Relationships		New	Field Dependencies	Custom Fields & Relationships Help ⓘ	
Action	Field Label	Data Type	Controlling Field	Modified By	
Edit Del	Description	TextArea(255)		Admin User, 10/6/2009 9:37 PM	
Edit Del Replace	Status	Picklist		Admin User, 10/6/2009 9:34 PM	

En savoir plus

Notez comment nous avons défini le type de données du nom de la facture sur Numéro automatique et défini un Format d'affichage. La plate-forme attribue automatiquement un numéro à chaque enregistrement unique créé, en commençant par le numéro de début que vous avez spécifié. Le format affiché se présente comme suit : FACT-002.

Voir aussi :

[Didacticiel n° 2 : Ajout de relations](#)

[Étape 2 : Création d'un objet Élément de ligne](#)

Étape 2 : Création d'un objet Élément de ligne

Chaque facture est composée d'un nombre d'éléments de ligne de facture, qui représente la quantité d'articles vendus à un prix spécifique. Vous allez créer l'objet Élément de ligne, puis l'associer aux objets Relevé de facture et Articles.

1. Cliquez sur **Configuration** ► **Créer** ► **Objets**.
2. Cliquez sur **Nouvel objet personnalisé**, puis définissez l'objet personnalisé.
 - Dans le champ **Étiquette**, saisissez **Élément de ligne**
 - Dans le champ **Étiquette - pluriel**, saisissez **Éléments de ligne**
 - Modifiez le **Nom de l'enregistrement en Numéro de l'élément de ligne**.
 - Laissez le champ **Type de données défini sur Texte**.

3. Laissez les autres valeurs inchangées, puis cliquez sur **Enregistrer**.
4. Ajoutez un champ **Prix unitaire** en lecture seule. Ce champ est en lecture seule car la valeur sera récupérée à partir de l'objet Article dans un autre didacticiel. Nous l'appelons **Prix unitaire** pour le différencier du champ **Prix** de l'objet Article.
 - a. Accédez à la liste associée **Champs personnalisés & relations**, puis cliquez sur **Nouveau**.
 - b. Dans la zone **Type de données**, sélectionnez **Devise**, puis cliquez sur **Suivant**.
 - c. Renseignez les détails des champs personnalisés.
 - Dans le champ **Étiquette** du champ, saisissez **Prix unitaire**.
 - Dans le champ **Longueur**, saisissez **16**, puis saisissez **2** dans le champ **Décimales**.
 - d. Cliquez sur **Suivant**.
 - e. Sélectionnez **Lecture seule** pour tous les profils, cliquez sur **Suivant**, puis cliquez sur **Enregistrer et Nouveau**.



Remarque: Si l'option **Lecture seule** n'est pas disponible, vous avez probablement sélectionné par erreur **Obligatoire**. Cliquez sur **Précédent** et désactivez **Obligatoire**.

5. Répétez les étapes ci-dessus pour ajouter un champ **Unités vendues**.

- a. Dans la zone **Type de données**, sélectionnez **Numéro**, puis cliquez sur **Suivant**.
- b. Dans le champ **Étiquette** du champ, saisissez **Unités vendues**, puis cliquez sur **Suivant**.
- c. Cliquez sur **Suivant**, en acceptant les valeurs par défaut.
- d. Cliquez sur **Enregistrer** pour revenir à la page de détails de l'objet personnalisé **Élément de ligne**.

En savoir plus

À ce stade, vous avez créé trois objets personnalisés : **Article**, **Relevé d'inventaire** et **Élément de ligne**. Dans chacun de ces objets, vous avez créé des champs personnalisés pour représenter du texte, des chiffres et une devise. Tous ces champs ont un élément en commun : les valeurs sont fournies par l'utilisateur. Vous avez également créé deux champs personnalisés contenant des valeurs générées par le système : la liste de sélection **Statut** qui s'ouvre sur **En cours** et le champ **Numéro de facture** qui est automatiquement attribué par le type de données **Numérotation automatique**. À l'étape suivante, vous allez créer deux champs supplémentaires. Contrairement aux champs précédents, leurs valeurs proviennent d'autres objets.

Voir aussi :

[Didacticiel n° 2 : Ajout de relations](#)

[Étape 1 : Création d'un objet personnalisé Relevé de facture](#)

[Étape 3 : Association des objets](#)

Étape 3 : Association des objets

Vous disposez maintenant de tous les objets représentant le modèle de données et vous souhaitez par conséquent les associer. L'**Élément de ligne** est associé à un **Relevé de facture** (un relevé est composé d'un nombre d'éléments de ligne) et à un **Article** (dans un élément de ligne, le prix provient de l'article).

1. Dans la page de détails de l'objet **Élément de ligne**, accédez à la liste associée **Champs personnalisés & relations**, puis cliquez sur **Nouveau**.
2. Dans la zone **Type de données**, sélectionnez **Relation principal-détails**, puis cliquez sur **Suivant**.
3. Dans le champ **Associé à**, sélectionnez votre objet personnalisé **Article**, puis cliquez sur **Suivant**.
4. Acceptez les valeurs par défaut dans les trois écrans qui suivent en cliquant sur **Suivant**.
5. Désactivez la case en regard de **Présentation de Article** afin de ne pas afficher **Éléments de ligne** dans la liste associée **Article**.
6. Cliquez sur **Enregistrer et Nouveau**.
7. Dans la zone **Type de données**, sélectionnez **Relation principal-détails**, puis cliquez sur **Suivant**.
8. Dans le champ **Associé à**, sélectionnez votre objet personnalisé **Relevé de facture**, puis cliquez sur **Suivant**.
9. Acceptez les valeurs par défaut dans les écrans qui suivent en cliquant sur **Suivant**, puis cliquez sur **Enregistrer** pour revenir à la page de détails **Élément de ligne**.

En savoir plus

Vous venez de créer deux relations principal-détails en associant les enregistrements **Relevé de facture** aux enregistrements **Éléments de ligne de facture** et en associant les **Éléments de ligne de facture** aux **Articles**. Les relations principal-détails

prennent également en charge les champs récapitulatifs de cumul, qui permettent de regrouper les informations sur les enregistrements enfant. Vous utiliserez cette fonctionnalité dans un didacticiel ultérieur.

Voir aussi :

[Didacticiel n° 2 : Ajout de relations](#)

[Étape 2 : Création d'un objet Élément de ligne](#)

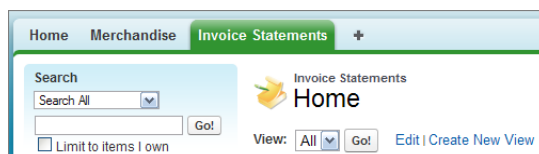
[Étape 4 : Ajout de relevés de facture à la sous-application](#)

Étape 4 : Ajout de relevés de facture à la sous-application

Comme vous l'avez fait pour l'objet personnalisé Article, vous allez créer un onglet pour l'objet Relevé de facture et l'ajouter à votre sous-application Entrepôt. Cet onglet affichera l'interface utilisateur que Force.com génère automatiquement pour cet objet.

1. Dans le menu de Configuration, cliquez sur **Créer ► Onglets**.
2. Dans la liste associée Onglets de l'objet personnalisé, cliquez sur **Nouveau** pour lancer l'assistant Onglet Nouvel objet personnalisé.
3. Dans la liste déroulante **Objet**, sélectionnez **Relevé de facture**.
4. Cliquez sur l'icône de recherche en regard du champ **Style d'onglet**, puis sélectionnez l'icône **Formulaire**.
5. Acceptez les autres valeurs par défaut, puis cliquez sur **Suivant** deux fois.
6. Dans la page **Ajouter aux applications personnalisées**, désactivez toutes les cases à l'exception de **Entrepôt**. L'onglet Relevés de facture est ajouté à votre sous-application Entrepôt.
7. Cliquez sur **Enregistrer** pour terminer la création de l'onglet.

Votre onglet Relevés de facture fait désormais partie de votre sous-application.



Voir aussi :

[Didacticiel n° 2 : Ajout de relations](#)

[Étape 3 : Association des objets](#)

[Étape 5 : Création d'un enregistrement Facture](#)

Étape 5 : Création d'un enregistrement Facture

Comme vous l'avez constaté dans le didacticiel précédent, la plate-forme génère automatiquement une interface utilisateur pour les objets que vous créez pour vous permettre d'afficher, de modifier, de supprimer et de mettre à jour les enregistrements. Comme vous avez également associé les objets, l'interface utilisateur représente également un outil de navigation entre les enregistrements associés.

1. Cliquez sur l'onglet Relevés de facture.
2. Cliquez sur **Nouveau**.

3. Dans le champ *Description*, saisissez *Première facture*, puis cliquez sur **Enregistrer**.

La page de détails de votre relevé de facture doit se présenter comme dans l'illustration ci-dessous.

Notez que le numéro de facture a été automatiquement attribué et que l'interface utilisateur affiche une liste associée Éléments de ligne vide dessous. Le Relevé de facture est lié à l'Élément de ligne via un champ principal-détails. Vous avez créé cette relation qui explique la présence d'une liste associée dans la page de détails Relevé de facture. Vous allez ensuite ajouter un élément de ligne à la facture.

1. Cliquez sur **Nouvel élément de ligne**.

2. Renseignez les champs.

- Dans le champ *Numéro de l'élément de ligne*, saisissez 1.
- Dans le champ *Prix unitaire*, saisissez 10.
- Dans le champ *Unités vendues*, saisissez 4.
- Cliquez sur l'icône de recherche en regard du champ *Article*, puis sélectionnez un produit.

3. Cliquez sur **Enregistrer**.

En savoir plus

Vous vous demandez peut-être pourquoi le *Numéro de l'élément de ligne* est un champ de texte alors que vous saisissez des valeurs. Si les éléments de ligne sont numérotés, pourquoi ne pas définir le champ sur la numérotation automatique, comme les Relevés de facture ? Pour répondre simplement, il est plus facile d'utiliser du texte avec des enregistrements et ce didacticiel

doit rester convivial. Si vous le souhaitez, vous pouvez définir le champ Élément de ligne sur la numérotation automatique, ce qui rendra toutefois le [Didacticiel n° 8, Création d'une page Web publique en utilisant les Sites](#), plus complexe.

Voir aussi :

[Didacticiel n° 2 : Ajout de relations](#)

[Étape 4 : Ajout de relevés de facture à la sous-application](#)

[Résumé](#)

Résumé

Dans ce didacticiel, vous avez créé des relations entre les objets dans votre modèle de données. Les relations fonctionnent comme des clés étrangères dans des bases de données relationnelles, mais de façon plus abstraite, en vous laissant gérer les éléments importants (les relations) plutôt que la mise en œuvre sous-jacente. Les relations principal-détails permettent de regrouper les informations. Le didacticiel suivant présente comment calculer le prix de chaque élément de ligne de facture dans le relevé de facture. Les relations offrent également l'avantage suivant : vous pouvez accéder aux enregistrements associés dans une interface utilisateur et un langage de requête. Nous en reparlerons.

Vous venez de créer l'application de base ; vous pouvez maintenant ajouter une logique commerciale à l'aide de formules et de règles de validation dans le [Didacticiel n° 3 : Utilisation de formules et de règles de validation](#) à la page 21.

Voir aussi :

[Didacticiel n° 2 : Ajout de relations](#)

[Étape 5 : Création d'un enregistrement Facture](#)

Didacticiel n° 3 : Utilisation de formules et de règles de validation

Niveau : Débutant ; Durée : 20 à 30 minutes



La plate-forme Force.com permet de créer des formules et des règles de validation de champ qui facilitent la gestion et l'optimisation de la qualité des données saisies dans votre sous-application. Les champs de formule et les règles de validation de champ utilisent des fonctions intégrées qui permettent de manipuler automatiquement vos données, de les valider et de calculer d'autres valeurs basées sur vos données. Les fonctions que vous utilisez dans les champs de formule et les règles de validation de champ ressemblent à celles que vous utilisez dans un tableur pour référencer des valeurs dans d'autres champs d'une feuille de calcul, effectuer des calculs et renvoyer un résultat. Toutefois, avec les champs de formule et les règles de validation de champ, vous référencez des champs dans les enregistrements de votre sous-application.

Dans ce didacticiel, vous allez améliorer la sous-application Entrepôt en ajoutant un champ de formule qui calcule automatiquement la valeur totale de chaque élément de ligne. Vous allez également utiliser cette nouvelle valeur d'élément de ligne et une propriété des relations principal-détails pour calculer automatiquement le total d'une facture. Pour terminer, vous allez ajouter une formule qui effectue un contrôle d'inventaire, afin de vous empêcher de créer un élément de ligne pour une quantité d'articles supérieure à votre stock.

Prérequis

Sous-application Entrepôt

Vous devez commencer par créer une application Entrepôt de base en suivant les instructions du [Didacticiel n° 2 : Ajout de relations](#) à la page 13.

Voir aussi :

[Étape 1 : Calcul d'une valeur pour chaque Élément de ligne](#)

[Étape 2 : Calcul d'une valeur pour le Relevé de facture avec un champ récapitulatif de cumul](#)

[Étape 3 : Contrôle d'inventaire avec une règle de validation](#)

[Étape 4 : Test de la règle de validation](#)

[Étape 5 : Amélioration de la règle de validation](#)

[Résumé](#)

Étape 1 : Calcul d'une valeur pour chaque Élément de ligne

Dans cette étape, vous allez ajouter un nouveau champ calculé appelé Valeur à l'élément de ligne. Ce champ multipliera le nombre d'articles par le prix et agira comme un total pour chaque élément de ligne.

1. Accédez à l'objet personnalisé Élément de ligne en cliquant sur **Configuration** ► **Créer** ► **Objets** ► **Élément de ligne**.
2. Accédez à la liste associée Champs personnalisés & relations, puis cliquez sur **Nouveau**.
3. Choisissez le type de champ **Formule**, puis cliquez sur **Suivant**.
4. Dans les champs **Étiquette** du champ et **Nom** du champ, saisissez **Valeur**.
5. Dans la zone **Type de renvoi de la formule**, sélectionnez **Devise**.
6. Cliquez sur **Suivant**.
7. Dans la liste déroulante **Insérer un champ de fusion**, sélectionnez **Prix unitaire**. La valeur `Prix_unitaire__c` s'affiche dans la zone de texte.
8. Cliquez sur la liste déroulante **Insérer un opérateur**, puis sélectionnez **Multiplication**.
9. Dans la liste déroulante **Insérer un champ de fusion**, sélectionnez **Unités vendues**. La valeur `Prix_unitaire__c * Unit_s_vendues__c` s'affiche dans la zone de texte.

10. Cliquez sur **Suivant** deux fois, puis cliquez sur **Enregistrer**.

Lorsque vous revenez à la page de détails de l'objet personnalisé, notez le nouveau champ appelé **Valeur**.

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

En savoir plus

Le type de champ **Formule** est très pratique pour calculer automatiquement les valeurs de champ à partir d'autres valeurs, comme nous l'avons fait ici. La formule que vous avez saisie ne présentait pas de difficulté : une simple multiplication de deux valeurs de champ dans le même enregistrement. Il existe également un onglet **Formule avancée** qui permet de créer des formules plus complexes.

Voir aussi :

[Didacticiel n° 3 : Utilisation de formules et de règles de validation](#)

[Étape 2 : Calcul d'une valeur pour le Relevé de facture avec un champ récapitulatif de cumul](#)

Étape 2 : Calcul d'une valeur pour le Relevé de facture avec un champ récapitulatif de cumul

Vous disposez maintenant du total de chaque élément de ligne ; il est naturel d'ajouter ces totaux afin d'obtenir le total de la facture. Puisque ces éléments de ligne ont une relation principal-détails avec le relevé de facture, nous pouvons utiliser un champ récapitulatif de cumul pour calculer cette valeur. Le récapitulatif de cumul est un type de champ spécial qui permet de regrouper les informations relatives aux objets (enfant) de détails associés. Dans ce cas, vous calculez la valeur de chaque élément de ligne.

1. Revenez à la page de l'objet personnalisé Relevé de facture en cliquant sur **Configuration** ► **Créer** ► **Objets** ► **Relevé de facture**.
2. Accédez à la liste associée Champs personnalisés & relations, puis cliquez sur **Nouveau**.
3. Sélectionnez le type de champ Récapitulatif de cumul, puis cliquez sur **Suivant**.
4. Dans le champ Étiquette du champ, saisissez Valeur de facture, puis cliquez sur **Suivant**.
5. Dans la liste Objet synthétisé, sélectionnez **Éléments de ligne**
6. Dans le champ Sélectionner le type de cumul, sélectionnez **SUM**.
7. Dans la liste Champ à synthétiser, sélectionnez **Valeur**.
8. Vérifiez que votre écran se présente comme dans l'illustration ci-dessous, puis cliquez sur **Suivant**.

9. Cliquez de nouveau sur **Suivant**, puis cliquez sur **Enregistrer**.

En savoir plus

Si vous revenez à l'enregistrement du relevé de facture, notez le nouveau champ de récapitulatif de cumul qui affiche la valeur totale de tous les éléments de ligne de facture. Si un sablier est affiché en regard du champ, attendez quelques secondes, puis actualisez la page. Vous pouvez tester cette nouvelle fonctionnalité en ajoutant un autre élément de ligne.

Voir aussi :

[Didacticiel n° 3 : Utilisation de formules et de règles de validation](#)

[Étape 1 : Calcul d'une valeur pour chaque Élément de ligne](#)

[Étape 3 : Contrôle d'inventaire avec une règle de validation](#)

Étape 3 : Contrôle d'inventaire avec une règle de validation

Les champs que vous définissez dans des objets peuvent être dotés de règles de validation écrits dans le même langage de formule que vous avez utilisé pour créer le champ de formule. Les règles de validation peuvent servir à déterminer quelle plage de saisie est valide et à afficher un message à l'utilisateur lorsqu'une valeur de champ n'est pas valide. L'évaluation des formules de condition d'erreur doit être True lorsque vous souhaitez afficher un message à l'utilisateur.

1. Revenez à la page de l'objet personnalisé Élément de ligne en cliquant sur **Configuration** ► **Créer** ► **Objets** ► **Élément de ligne**.
2. Accédez à la liste associée Règles de validation, puis cliquez sur **Nouveau**.

3. Dans le champ Nom de règle, saisissez `Commande en stock`.
4. Dans la zone Formule de condition d'erreur, cliquez sur **Insérer un champ** pour ouvrir la fenêtre contextuelle Insérer un champ.
 - a. Sélectionnez `Elément de ligne >` dans la première colonne.
 - b. Sélectionnez `Article >` dans la deuxième colonne.
 - c. Sélectionnez `Inventaire total` dans la troisième colonne.
 - d. Cliquez sur **Insérer**.
 - e. Tapez le symbole inférieur à `<` pour obtenir la formule :

```
Article__r.Inventaire_total__c <
```

- f. Cliquez de nouveau sur **Insérer un champ**.
- g. Sélectionnez `Elément de ligne >` dans la première colonne.
- h. Sélectionnez `Unités vendues` dans la deuxième colonne.
- i. Cliquez sur **Insérer**, puis vérifiez que le code se présente comme suit :

```
Article__r.Inventaire_total__c < Unit_s_vendues__c
```

5. Cliquez sur **Vérifier la syntaxe** pour vous assurer de l'absence d'erreur. Si des erreurs sont détectées, corrigez-les avant de continuer.
6. Dans le champ Message d'erreur, saisissez `Vous avez commandé un nombre d'articles supérieur à la quantité en stock`.
7. Pour l'option Emplacement de l'erreur, activez **Champ**, puis sélectionnez `Unités vendues` dans la liste déroulante.

8. Cliquez sur **Enregistrer**.

En savoir plus

Vous pouvez saisir une formule directement dans la zone Formule de condition d'erreur, cependant, comme vous l'avez constaté ici, vous pouvez aisément parcourir les objets disponibles et sélectionner les composants nécessaires pour la formule. Analysons la formule que vous avez créée.

- `Article__r` : puisque l'objet Article est associé à l'objet Élément de ligne, la plate-forme fournit automatiquement un champ de relation qui permet de naviguer depuis un enregistrement Élément de ligne vers un enregistrement Article, ce que fait le champ `Article__r`.
- `Inventaire_total__c` : il s'agit du champ que vous avez créé pour suivre le montant total du stock dans un enregistrement Article.
- `Article__r.Inventaire_total__c` : indique au système de récupérer la valeur du champ Inventaire total dans un enregistrement Article associé.
- `Unit_s_vendues__c` : correspond au champ Unités vendues dans l'enregistrement actuel (Élément de ligne).

Une fois regroupés, ces éléments constituent une formule qui vérifie si l'inventaire total dans l'enregistrement Article associé est inférieur à la quantité d'unités vendues. Comme indiqué à la page Formule de condition d'erreur, vous devez fournir une formule qui renvoie True si vous souhaitez afficher une erreur. La formule est vraie lorsque l'inventaire total est inférieur aux unités vendues.

Voir aussi :

[Didacticiel n° 3 : Utilisation de formules et de règles de validation](#)

[Étape 2 : Calcul d'une valeur pour le Relevé de facture avec un champ récapitulatif de cumul](#)

[Étape 4 : Test de la règle de validation](#)

Étape 4 : Test de la règle de validation

Vous allez maintenant tester la règle de validation créée à l'étape 3.

1. Cliquez sur l'onglet Relevés de facture, puis sélectionnez une facture existante.
2. Accédez à la zone Éléments de ligne, puis cliquez sur **Modifier** en regard d'un ou des éléments de ligne.
3. Modifiez la valeur Unités vendues avec un chiffre supérieur à la quantité en stock.
4. Cliquez sur **Enregistrer**. Une erreur devrait s'afficher indiquant que la quantité d'articles en stock est insuffisante. Cliquez sur **Annuler**.

The screenshot shows the 'Line Item Edit' interface for line item 1. The 'Units Sold' field contains the value '30,000', which is highlighted in red. A red error message is displayed below the field: 'Error: You have ordered more items than we have in stock.' Above the form, a general error message reads: 'Error: Invalid Data. Review all error messages below to correct your data.' The form includes fields for Line Item Number (1), Unit Price (10.00), Merchandise (Wee Jet), and Invoice Statement (INV-0001). Buttons for 'Save', 'Save & New', and 'Cancel' are visible at the top and bottom of the form.

5. Cliquez sur **Annuler**.

Voir aussi :

[Didacticiel n° 3 : Utilisation de formules et de règles de validation](#)

[Étape 3 : Contrôle d'inventaire avec une règle de validation](#)

[Étape 5 : Amélioration de la règle de validation](#)

Étape 5 : Amélioration de la règle de validation

La logique de la formule de validation est imparfaite ! Imaginez que vous modifiez un enregistrement et diminuez la quantité d'unités vendues. Dans ce cas, aucun contrôle de l'inventaire n'est nécessaire. À l'inverse, si vous augmentez la quantité d'unités vendues, il suffit de vérifier que les articles supplémentaires sont disponibles en stock. Vous pouvez améliorer la règle de validation en lui ajoutant ces scénarios en utilisant la fonction ISNEW() dans votre formule, qui détermine si vous créez un enregistrement. S'il ne s'agit pas d'un nouvel enregistrement, vous devez utiliser la fonction PRIORVALUE() qui indique la valeur précédente du champ, avant sa modification.

1. Revenez à la page de l'objet personnalisé Élément de ligne en cliquant sur **Configuration** > **Créer** > **Objets** > **Élément de ligne**.
2. Dans la liste associée Règles de validation, cliquez sur **Modifier** en regard du champ `Commande en stock`.
3. Remplacez la formule existante par la formule suivante :

```
IF( ISNEW(), Article__r.Inventaire_total__c < Unit_s_vendues__c , IF ( Unit_s_vendues__c < PRIORVALUE(Unit_s_vendues__c), FALSE, Article__r.Inventaire_total__c < (Unit_s_vendues__c - PRIORVALUE(Unit_s_vendues__c)) ) )
```

4. Cliquez sur **Vérifier la syntaxe** pour vous assurer de l'absence d'erreur.
5. Cliquez sur **Enregistrer**.

En savoir plus

Examinons cette formule plus en détails.

- IF est une condition qui exécute l'une des deux alternatives selon la condition.
- ISNEW est la condition. Cette fonction renvoie True si vous créez un enregistrement, sinon elle renvoie False.
- Si ISNEW est True, vous contrôlez simplement l'inventaire comme vous l'avez fait précédemment. Pour les nouveaux enregistrements, vous devez vérifier que vous n'avez pas vendu plus d'articles que vous en avez en stock.
- Si ISNEW est False, vous savez que vous effectuez une mise à jour d'un enregistrement existant, vous ne créez pas un enregistrement. Par conséquent, vous effectuez une autre vérification conditionnelle afin de déterminer si la quantité d'unités a augmenté ou diminué, en la comparant à la valeur précédente avec la fonction PRIORVALUE.
- Si la valeur précédente est supérieure, vous avez mis à jour l'enregistrement et diminué ses unités. Puisque votre stock est suffisant, vous renvoyez FALSE pour indiquer l'absence d'échec de validation.

- Si la valeur précédente est inférieure, vous avez augmenté la quantité d'unités et vous devez vérifier si l'inventaire est suffisant pour couvrir la différence entre la nouvelle valeur et l'ancienne.

Voir aussi :

[Didacticiel n° 3 : Utilisation de formules et de règles de validation](#)

[Étape 4 : Test de la règle de validation](#)

[Résumé](#)

Résumé

Dans ce didacticiel, vous avez créé des champs de formule et des règles de validation qui améliorent et valident les données de votre sous-application. Pour la première formule, vous avez créé un sous-total pour chaque élément de ligne en multipliant le prix d'un produit par la quantité d'unités commandées. Comme une facture peut inclure plusieurs éléments de ligne, vous avez créé un champ récapitulatif de cumul afin d'ajouter automatiquement les sous-totaux d'éléments de ligne.

Vous avez également appris comment définir une règle de validation pour vérifier la disponibilité des articles en stock. La première version de cette règle était simple, mais vous l'avez améliorée pour l'utiliser avec des enregistrements nouveaux et avec des enregistrements existants comprenant des mises à jour. Ces types de modifications peuvent s'appliquer à n'importe quel champ de votre sous-application afin de vérifier la conformité des données avec critères spécifiques.

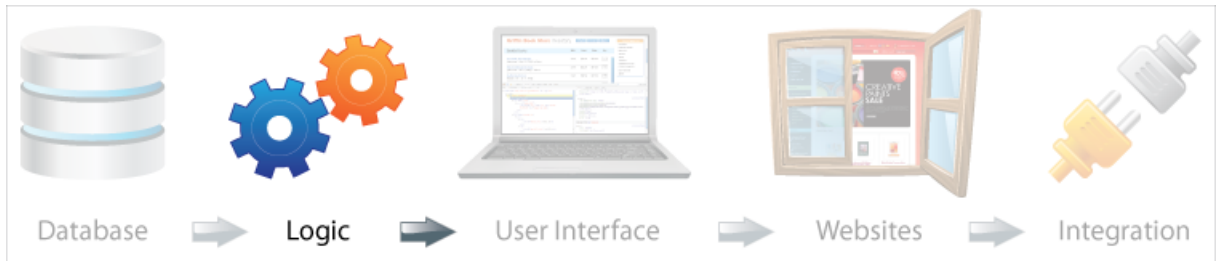
Voir aussi :

[Didacticiel n° 3 : Utilisation de formules et de règles de validation](#)

[Étape 5 : Amélioration de la règle de validation](#)

Didacticiel n° 4 : Utilisation de workflow et d'approbations

Niveau : Intermédiaire ; **Durée :** 30 à 45 minutes



Votre société travaille plus efficacement si elle bénéficie de procédures internes normalisées et de processus commerciaux automatisés. Dans votre sous-application Force.com, vous pouvez utiliser des règles de workflow et des processus d'approbation afin d'automatiser vos procédures et processus. Ils permettent de gagner du temps, mais également d'imposer cohérence et conformité dans les pratiques commerciales de votre société.

Les règles de workflow peuvent lancer des actions (telles que des alertes par e-mail, des tâches, des mises à jour de champ et des messages sortants) basées sur des déclencheurs temporels, des critères et des formules. Les processus d'approbation peuvent automatiser toutes les approbations de votre organisation, qu'ils soient simples ou complexes.

Dans ce didacticiel, vous allez créer et tester deux règles de workflow. Le premier transmet le prix actuel d'un article vers l'enregistrement d'élément de ligne créé, et le deuxième actualise l'inventaire lorsque vous mettez à jour un élément de ligne de produit. Vous allez également créer un processus d'approbation qui demande l'approbation explicite d'un responsable si la facture dépasse 2 000 €.

Prérequis

Formules et validation

Vous devez commencer par créer un champ récapitulatif de cumul et des règles de validation en suivant les instructions du [Didacticiel n° 3 : Utilisation de formules et de règles de validation](#) à la page 21.

Voir aussi :

[Étape 1 : Renseignement du prix d'élément de ligne à l'aide d'une règle de workflow](#)

[Étape 2 : Mise à jour de l'inventaire total lors d'une commande](#)

[Étape 3 : Test des règles de workflow](#)

[Étape 4 : Création d'un modèle d'e-mail](#)

[Étape 5 : Création d'un processus d'approbation](#)

[Étape 6 : Création d'un profil de sécurité personnalisé](#)

[Étape 7 : Création d'un utilisateur](#)

[Étape 8 : Test du processus d'approbation](#)

[Résumé](#)

Étape 1 : Renseignement du prix d'élément de ligne à l'aide d'une règle de workflow

À ce stade, vous avez créé l'Élément de ligne et les objets Articles avec des champs Prix. Vous avez également marqué le champ Prix d'élément de ligne en lecture seule, même si vous ne l'avez pas remarqué dans l'interface car l'utilisateur par défaut est un administrateur ayant accès à toutes les données. Vous allez maintenant créer une règle de workflow pour renseigner le prix unitaire de la ligne avec celui de l'article au moment de la création de l'élément de ligne. Vous pouvez créer aisément cette logique complexe avec une règle de workflow déclarative.

1. Cliquez sur **Configuration** ► **Créer** ► **Workflow & approbations** ► **Règles de workflow**. Si la page Présentation du workflow s'affiche, cliquez sur **Continuer**.
2. Dans la page Toutes les règles de workflow, cliquez sur **Nouvelle règle**.
3. À l'étape 1 de l'assistant Règle de workflow, sélectionnez l'objet **Élément de ligne**, puis cliquez sur **Suivant**.
4. Dans le champ **Nom de règle**, saisissez **Élément de ligne créé**.
5. Dans la zone **Critères d'évaluation**, sélectionnez **Uniquement lors de la création d'un enregistrement**.
6. Dans le champ **Critères de la règle**, sélectionnez l'évaluation de la formule est positive.
7. Dans la zone de texte, saisissez **true**.
8. Assurez-vous que la page se présente comme dans l'illustration ci-dessous, puis cliquez sur **Enregistrer et Suivant**.

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

- Object:** Line Item
- Rule Name:** Line Item Created
- Evaluation Criteria:** Evaluate rule is set to 'Only when a record is created'.
- Rule Criteria:** Run this rule if the following: formula evaluates to true.
- Formula Field:** true
- Functions List:** Includes ABS, AND, BEGINS, BLANKVALUE, BR, CASE.

9. À l'étape 3 de l'assistant Règle de workflow, dans la section **Actions de workflow immédiates**, cliquez sur **Ajouter une action de workflow**, puis sélectionnez **Nouvelle mise à jour du champ**. L'assistant **Nouvelle mise à jour du champ** s'ouvre.
10. Dans le champ **Nom**, saisissez **Insérer le prix de l'article**.
11. Dans la liste déroulante **Champ à mettre à jour**, sélectionnez **Élément de ligne** dans la première case et **Prix unitaire** dans la deuxième.
12. Sélectionnez **Utiliser une formule pour définir la nouvelle valeur**.
13. Cliquez sur **Afficher l'éditeur de formule**.
14. Cliquez sur **Insérer un champ**.
15. Sélectionnez **Élément de ligne** > dans la première colonne, **Article** > dans la deuxième et **Prix** dans la troisième.
16. Cliquez sur **Insérer**.

17. Cliquez sur **Enregistrer** pour fermer l'assistant Nouvelle mise à jour du champ et revenir à l'étape 3 de l'assistant du workflow.
18. Dans la page Spécifier les actions de workflow, cliquez sur **Terminé**.
19. Dans la page Règle de workflow, cliquez sur **Activer**.



Important: Ne pas activer une nouvelle règle de workflow est une erreur fréquente. Si la règle n'est pas active, ses critères ne sont pas évalués lorsque les enregistrements sont créés ou enregistrés.

En savoir plus

Si vous créez maintenant un nouvel élément de ligne et l'affichez, vous pouvez constater que son prix unitaire est automatiquement défini sur le prix de l'enregistrement de l'article associé. Puisque vous êtes un administrateur, un champ de saisie s'affiche pour le prix. Vos utilisateurs ne voient pas ce champ car il est en lecture seule.

Cette règle de workflow s'exécute uniquement lors de la création d'un enregistrement (jamais après) car nous ne voulons pas augmenter le prix indiqué sur la facture une fois l'article ajouté à la facture. Le client serait pour le moins surpris de découvrir un prix supérieur à celui annoncé. Toutefois, si le prix baissait, le client apprécierait d'être tenu informé de tout changement. Vous définirez cette fonction ultérieurement.

Vous vous demandez peut-être pourquoi nous avons créé une règle dont les formules sont toujours évaluées sur True. C'est parce que nous souhaitons mettre à jour le champ chaque fois qu'un enregistrement est créé, ce qui est toujours vrai. Vous pouvez utiliser une formule pour évaluer une expression, ce qui n'était pas nécessaire dans notre exemple.

Voir aussi :

[Didacticiel n° 4 : Utilisation de workflow et d'approbations](#)

[Étape 2 : Mise à jour de l'inventaire total lors d'une commande](#)

Étape 2 : Mise à jour de l'inventaire total lors d'une commande

Vous allez préférer que la gestion de l'inventaire total des enregistrements d'articles soit automatique. Lors de la création de factures (dont le statut par défaut est « En cours »), vous allez vouloir refléter les mises à jour des éléments de ligne dans le

champ Inventaire total que vous avez créé dans l'objet Article. Vous pouvez le faire avec une autre règle de workflow. La seule difficulté est la modification de la valeur de l'inventaire total. Si vous créez un élément de ligne, vous devez diminuer l'inventaire total par le nombre d'unités vendues. Cependant, si vous mettez à jour un élément de ligne, vous devez ajuster l'inventaire total avec la différence entre l'ancienne et la nouvelle quantités d'unités vendues.

1. Cliquez sur **Configuration** ► **Créer** ► **Workflow & approbations** ► **Règles de workflow**.
2. Si l'écran Présentation du workflow s'affiche, cliquez sur **Continuer** ou passez à l'étape suivante.
3. Dans la page Toutes les règles de workflow, cliquez sur **Nouvelle règle**.
4. À l'étape 1 de l'assistant Règle de workflow, sélectionnez l'objet **Elément de ligne**, puis cliquez sur **Suivant**.
5. Dans le champ Nom de règle, saisissez **Elément de ligne mis à jour**.
6. Dans la zone Critères d'évaluation, sélectionnez **À chaque création ou modification d'un enregistrement**.
7. Dans le champ Critères de la règle, sélectionnez critères suivants sont remplis.
8. Dans la liste déroulante Champ, sélectionnez **Relevé de facture: Statut**. Dans la liste déroulante Opérateur, sélectionnez **égal à**. En regard de Valeur, cliquez sur l'icône de recherche, puis sélectionnez **En cours**. Cliquez sur **Insérer les éléments sélectionnés**.
9. Assurez-vous que l'écran se présente comme dans l'illustration ci-dessous, puis cliquez sur **Enregistrer et Suivant**.

The screenshot shows the 'Edit Rule' configuration interface. The 'Rule Name' is 'Line Item Updated'. The 'Evaluation Criteria' section shows 'Evaluate rule' set to 'Every time a record is created or edited'. The 'Rule Criteria' section shows a table with the following configuration:

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

10. À l'étape 3 de l'assistant Règle de workflow, dans la section Actions de workflow immédiates, cliquez sur **Ajouter une action de workflow**, puis sélectionnez **Nouvelle mise à jour du champ**. L'assistant Nouvelle mise à jour du champ s'ouvre.
11. Dans le champ Nom, saisissez **Mettre à jour l'inventaire du stock**.
12. Dans la première liste déroulante, Champ à mettre à jour, sélectionnez **Article**. Dans la deuxième, sélectionnez **Inventaire total**.
13. Sélectionnez **Utiliser une formule pour définir la nouvelle valeur**.
14. Cliquez sur **Afficher l'éditeur de formule**.
15. Saisissez le code suivant :

```
IF ( ISNEW(), Article__r.Inventaire_total__c - Unit_s_vendues__c ,
Article__r.Inventaire_total__c - (Unit_s_vendues__c - PRIORVALUE(Unit_s_vendues__c)) )
```

16. Cliquez sur **Vérifier la syntaxe** et corrigez toute erreur éventuelle.

17. Cliquez sur **Enregistrer** pour fermer l'assistant Nouvelle mise à jour du champ et revenir à l'étape 3 de l'assistant du workflow.
18. Dans la page Spécifier les actions de workflow, cliquez sur **Terminé**.
19. Dans la page Règle de workflow, cliquez sur **Activer**.

Voir aussi :

[Didacticiel n° 4 : Utilisation de workflow et d'approbations](#)

[Étape 1 : Renseignement du prix d'élément de ligne à l'aide d'une règle de workflow](#)

[Étape 3 : Test des règles de workflow](#)

Étape 3 : Test des règles de workflow

Vous pouvez vérifier le fonctionnement de vos workflow en créant un élément de ligne, puis vous assurer que son prix unitaire est automatiquement défini et que l'inventaire total du stock a diminué.

1. Cliquez sur l'onglet Relevés de facture.
2. Cliquez sur l'enregistrement de facture créé précédemment.
3. Cliquez sur **Nouvel élément de ligne**, puis saisissez les valeurs ci-dessous.
 - Dans le champ **Numéro de l'élément de ligne**, saisissez 2.
 - Laissez le champ **Prix unitaire** vide.
 - Définissez **Unités vendues** sur 1000.
 - En regard du champ **Article**, cliquez sur l'icône de recherche, puis sélectionnez l'enregistrement de l'article Wee Jet.
4. Cliquez sur **Enregistrer**.
5. Cliquez sur l'élément de ligne 2. Notez que le prix unitaire est défini sur une valeur identique à celle de l'enregistrement Wee Jet.

- Ouvrez maintenant l'enregistrement d'article Wee Jet en cliquant sur le lien **Wee Jet**. Notez que l'`Inventaire total` est passé de 2 000 à 1 000.

Voir aussi :

Didacticiel n° 4 : Utilisation de workflow et d'approbations

Étape 2 : Mise à jour de l'inventaire total lors d'une commande

Étape 4 : Création d'un modèle d'e-mail

Étape 4 : Création d'un modèle d'e-mail

L'une des règles commerciales que vous souhaitez appliquer à votre workflow est l'approbation du responsable pour toutes les factures dont le total dépasse 2 000 €. Pour mettre en œuvre cette règle, vous devez créer deux éléments supplémentaires : un e-mail à envoyer au responsable lorsqu'une facture dépasse 2 000 € et un processus d'approbation que le responsable doit suivre.



Remarque: Si votre devise par défaut n'est pas définie sur Euros, appliquez votre devise à cet exemple.

Dans cette étape, vous allez créer le modèle d'e-mail que la règle de workflow doit utiliser pour générer et envoyer l'e-mail. Vous allez créer le processus d'approbation dans les étapes suivantes.

- Cliquez sur **Configuration** ► **Modèles de communication** ► **Modèles de message**.
- Cliquez sur **Nouveau modèle**.
- À l'étape 1 de l'assistant Modèle du message, sélectionnez `Texte`, puis cliquez sur **Suivant**.
- Activez la case `Disponible`.
- Dans le champ `Nom` du modèle de message, saisissez `Modèle facture importante`.
- Dans le champ `Objet`, saisissez `Une facture importante a été soumise`.
- Dans le champ `Corps` du message, saisissez le code et le texte suivants :

```
{!Relev_de_facture__c.OwnerFullName} a soumis pour approbation un Relevé de facture qui
dépasse le total {!Relev_de_facture__c.Valeur_de_facture__c} du
{!Relev_de_facture__c.LastModifiedDate}
```

- Vérifiez que votre écran se présente comme dans l'illustration ci-dessous, puis cliquez sur **Enregistrer**.

The screenshot shows a 'Step 2. Text Email Template: New Template' dialog box. It contains the following fields and values:

- 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)

En savoir plus

Le texte du corps de l'e-mail que vous avez saisi prend en charge les champs de fusion, par exemple `{!Relev_de_facture__c.Valeur_de_facture__c}`. Lors de la génération de l'e-mail, cette valeur est remplacée dynamiquement par la valeur réelle de la facture qui a généré le processus d'approbation.

Voir aussi :

[Didacticiel n° 4 : Utilisation de workflow et d'approbations](#)

[Étape 3 : Test des règles de workflow](#)

[Étape 5 : Création d'un processus d'approbation](#)

Étape 5 : Création d'un processus d'approbation

Dans cette étape, vous allez créer un processus d'approbation qui nécessite l'approbation explicite du responsable si une facture dépasse 2 000 €. La création et l'utilisation d'un processus d'approbation sont aussi faciles que la création d'une règle de workflow.

1. Cliquez sur **Configuration** ► **Créer** ► **Workflow & approbations** ► **Processus d'approbation**.
2. Dans la liste déroulante **Gérer les processus d'approbation pour**, sélectionnez **Relevé de facture**.
3. Cliquez sur **Créer un processus d'approbation** et sélectionnez l'option **Utiliser l'assistant de démarrage automatique** dans la liste déroulante.
4. Dans le champ **Nom**, saisissez **Valeur facture importante**.
5. En regard du champ **Modèle d'e-mail d'attribution de l'approbation**, cliquez sur l'icône de recherche et sélectionnez le **Modèle facture importante** que vous venez de créer.
6. Saisissez les valeurs suivantes dans la zone **Spécifier les critères d'entrée**.
 - a. Dans la liste déroulante **Champ**, sélectionnez **Valeur de facture**.
 - b. Dans la liste déroulante **Opérateur**, sélectionnez **supérieur à**.
 - c. Dans le champ **Valeur**, saisissez **2000**.
7. Sélectionnez l'option **Attribuer automatiquement un approbateur à l'aide d'un champ de hiérarchie personnalisé ou standard**, puis **Responsable** dans le champ de relation hiérarchique. Cette option permet

d'attribuer le rôle d'approbateur au responsable de l'utilisateur qui démarre le processus d'approbation. Vous allez créer un utilisateur Alain Durand à l'étape suivante.

8. Cliquez sur **Enregistrer**.
9. Un avertissement vous invite à activer le processus d'approbation. Cliquez sur **OK**. Vous avez terminé la création du processus d'approbation, mais avant de pouvoir l'exécuter, vous devez définir les actions exécutées lors de la soumission initiale, de l'approbation et du refus des enregistrements.
10. Cliquez sur **Afficher la page de détails sur le processus d'approbation**.
11. Créez des actions de mise à jour de champ en cliquant sur **Ajouter une nouvelle action** et en sélectionnant **Mise à jour** du champ pour chaque liste associée du tableau ci-dessous. Configurez chaque action de mise à jour de champ comme indiqué.

Liste associée	Nom	Champ à mettre à jour	Options de liste de sélection
Actions de soumission initiale	Définir le statut d'approbation initiale	Statut	Sélectionner Une valeur spécifique, puis En attente.
Actions d'approbation finale	Définir le statut d'approbation finale	Statut	Sélectionner Une valeur spécifique, puis Fermé.
Actions de refus final	Définir le statut de refus final	Statut	Sélectionner Une valeur spécifique, puis En cours.

12. Cliquez sur **Enregistrer**.
13. Cliquez sur **Processus d'approbation** pour revenir à la liste des approbations.
14. Cliquez sur **Activer** pour activer le processus d'approbation.

En savoir plus

Des actions par défaut sont associées à certaines étapes d'approbation. Les actions de soumission initiale et d'approbation finale verrouillent l'enregistrement, alors que l'action de refus final déverrouille l'enregistrement. Ceci est logique : si un enregistrement est soumis pour approbation, il ne doit pas être modifiable durant l'approbation. S'il est refusé, il doit être déverrouillé pour lui apporter des modifications.

Voir aussi :

[Didacticiel n° 4 : Utilisation de workflow et d'approbations](#)

[Étape 4 : Création d'un modèle d'e-mail](#)

[Étape 6 : Création d'un profil de sécurité personnalisé](#)

Étape 6 : Création d'un profil de sécurité personnalisé

Un profil est un ensemble d'autorisations et d'autres paramètres associés à un utilisateur ou à un groupe d'utilisateurs. Votre organisation est associée à des profils standard prédéfinis. Si vous créez un objet personnalisé, les autorisations d'accès à cet objet (« Lire », « Créer », « Modifier » et « Supprimer ») sont désactivées pour la plupart des profils. Ce paramètre de sécurité par défaut permet de bloquer l'accès de tous les utilisateurs aux objets personnalisés et à leurs données tant que l'autorisation n'est pas explicitement accordée. Vous pouvez modifier ces autorisations dans des profils personnalisés, mais pas dans des profils standard.

Dans cette étape, vous allez créer un profil personnalisé que vous pouvez attribuer aux utilisateurs qui doivent accéder aux objets personnalisés de la sous-application Entrepôt. Dans le nouveau profil personnalisé, vous allez ensuite activer les autorisations d'objet pour permettre aux utilisateurs d'accéder aux objets. Vous créez un utilisateur à l'étape suivante.

1. Cliquez sur **Configuration** ► **Gestion des utilisateurs** ► **Profil**.
2. Cliquez sur **Nouveau**.
3. Dans le champ `Profil` existant, sélectionnez `Utilisateur standard`. Si vous utilisez la version Free Edition, sélectionnez `Force.com - Free User`.
4. Dans le champ `Licence utilisateur` `Nom du profil`, saisissez `Utilisateur de base`.
5. Dans la page de détails du nouveau profil, cliquez sur **Modifier**.
6. En bas de la page, dans la zone `Autorisations de l'objet personnalisé`, activez les cases **Lecture**, **Créer**, **Modifier** et **Supprimer** pour les objets `Relevés de facture`, `Éléments de ligne` et `Articles`.
7. Vérifiez que votre écran se présente comme dans l'illustration ci-dessous, puis cliquez sur **Enregistrer**.

Custom Object Permissions													
	Basic Access				Data Administration			Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All		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"/>	Merchandise	<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"/>							

En savoir plus

Vous venez de créer et de modifier aisément un profil personnalisé. Pour modifier de nombreux modèles, vous pouvez utiliser la gestion de profils avancée afin de créer une vue de liste personnalisée de vos profils et de les modifier dans la vue. Pour plus d'informations, reportez-vous à « Modification de profils à l'aide de listes de profils » dans l'aide en ligne de Salesforce.com.

Voir aussi :

[Didacticiel n° 4 : Utilisation de workflow et d'approbations](#)

[Étape 5 : Création d'un processus d'approbation](#)

[Étape 7 : Création d'un utilisateur](#)

Étape 7 : Création d'un utilisateur

Lors de la création d'une application Force.com, elle est automatiquement activée pour permettre la connexion de plusieurs utilisateurs. Vous pouvez désormais personnaliser l'application en la configurant selon le profil de l'utilisateur connecté. Vous pouvez par exemple accorder un accès en lecture seule aux champs à un groupe d'utilisateurs spécifique, ou les rendre invisibles. Vous pouvez également imposer un strict partage de données pour permettre à un utilisateur d'afficher ses enregistrements ou ceux de son responsable.

Dans cette étape, vous allez créer un utilisateur et l'associer à votre compte d'utilisateur actuel via le champ Responsable. Cette configuration permet de s'assurer que si le nouvel utilisateur crée une facture respectant certaines conditions, la facture est acheminée vers son responsable.

1. Cliquez sur **Configuration** ► **Gestion des utilisateurs** ► **Utilisateurs**.
2. Dans la page Tous les utilisateurs, cliquez sur **Nouvel utilisateur**.
3. Saisissez les informations suivantes :

- Dans le champ **Prénom**, saisissez **Alain**.
- Dans le champ **Nom**, saisissez **Durand**.
- Dans le champ **Alias**, saisissez **adurand**.
- Dans le champ **Adresse e-mail**, saisissez votre adresse e-mail pour vous permettre de recevoir les demandes d'approbation acheminées vers **Alain Durand**.
- Le champ **Nom d'utilisateur** est renseigné par défaut avec votre adresse e-mail, mais vous devez créer un nom d'utilisateur unique pour **Alain**, sous la forme d'une adresse e-mail imaginaire.



Remarque: Notez le nom d'utilisateur d'Alain (son adresse e-mail imaginaire), vous allez bientôt vous connecter sous son identité.

- Dans le champ **Responsable**, sélectionnez l'utilisateur créé lors de l'inscription à l'organisation.
- Dans le champ **Licence utilisateur**, sélectionnez **Salesforce**.
- Dans le champ **Profil**, sélectionnez **Utilisateur de base**.

4. Cliquez sur **Enregistrer**.

Vous devriez recevoir un message de confirmation de la création du nouvel utilisateur. Vous devez effectuer une étape supplémentaire pour terminer le processus d'approbation. Ne vous connectez-pas immédiatement à Alain Durand, car vous avez encore besoin de votre accès administrateur.

Dans la version Free Edition, pour permettre à Alain Durand de se connecter et d'utiliser la sous-application Entrepôt, vous devez modifier le profil Utilisateur de base en conséquence. Les utilisateurs de Free Edition ayant accès à une seule sous-application, Entrepôt doit être sélectionnée par défaut (si vous utilisez Developer Edition, passez cette étape).

1. Cliquez sur **Configuration** ► **Gestion des utilisateurs** ► **Profils**.
2. En regard du profil Utilisateur de base, cliquez sur **Modifier**.
3. En regard de Entrepôt, sélectionnez Par défaut.

Voir aussi :

[Didacticiel n° 4 : Utilisation de workflow et d'approbations](#)

[Étape 6 : Création d'un profil de sécurité personnalisé](#)

[Étape 8 : Test du processus d'approbation](#)

Étape 8 : Test du processus d'approbation

Vous pouvez maintenant préparer le processus d'approbation et l'exécuter. Si Alain Durand crée une facture avec un total supérieur à 2 000 €, il peut cliquer sur le bouton Soumettre pour approbation dans le relevé de facture. La sous-application envoie un e-mail à son responsable (à l'aide du modèle d'e-mail créé précédemment) et verrouille l'enregistrement, empêchant ainsi Alain d'effectuer de nouvelles modifications.

Lorsqu'il se connecte, le responsable peut approuver (ou refuser) la facture. S'il l'approuve, vous pouvez mettre à jour le statut de la facture sur Fermé. Dans le cas contraire, vous conservez le statut initial En cours. Dans un scénario réel, votre processus commercial peut exiger l'envoi d'un e-mail au propriétaire de la facture et demander une confirmation, entre autres.

Avant de tester le processus d'approbation, vous devez vous assurer que votre page d'accueil peut afficher les éléments nécessitant une approbation.

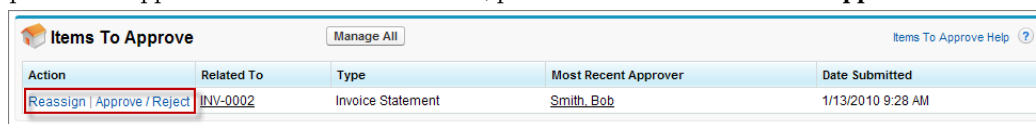
1. Cliquez sur **Configuration** ► **Personnalisation** ► **Accueil** ► **Présentations de page d'accueil**.
2. Cliquez sur **Modifier** en regard de votre présentation de page d'accueil (si vous utilisez Free Edition, cliquez sur **Modifier** en regard de VolunteerForce Home Page).

3. Sélectionnez l'option **Éléments à approuver**, si elle est désactivée.
4. Cliquez sur **Suivant**, puis sur **Enregistrer**.

Pendant le développement de l'application, vous êtes connecté en tant qu'administrateur. Pour créer un enregistrement et tester le processus d'approbation, vous devez vous déconnecter du compte administrateur et vous connecter en tant qu'utilisateur standard.

1. Connectez-vous en tant qu'Alain Durand. S'il s'agit de votre première connexion sous cette identité, vous serez probablement invité à changer de mot de passe.
2. Sélectionnez la sous-application Entrepôt.
3. Cliquez sur l'onglet Relevés de facture.
4. Cliquez sur **Nouveau**, puis créez un relevé de facture.
5. Ajoutez un **Nouvel élément de ligne** et une quantité d'unités suffisante pour obtenir un total de facture supérieur à 2 000 €.
6. Cliquez sur **Enregistrer**.
7. Cliquez sur **Soumettre pour approbation**.
8. Déconnectez-vous de l'application.

Un e-mail est envoyé à l'administrateur désigné d'Alain Durand, soit vous-même dans ce didacticiel. Comme vous êtes administrateur, vous pouvez vérifier cet envoi dans votre boîte de réception. Lorsque vous vous connectez à l'application en tant qu'administrateur, vous pouvez accéder à la demande d'approbation en attente sous l'onglet Accueil (faites défiler la page pour l'afficher). L'enregistrement est verrouillé jusqu'à l'approbation de la facture par le responsable. Un champ Approbation permet au responsable d'approuver ou de refuser la facture, puis de continuer le workflow. **Approuvez** la facture.

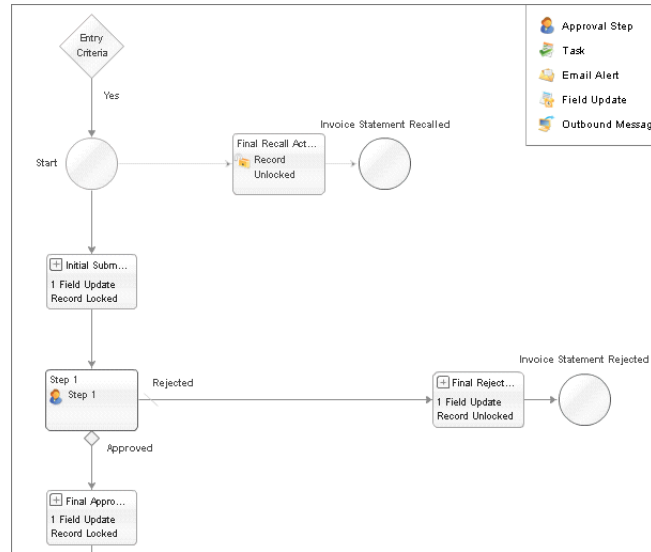


Action	Related To	Type	Most Recent Approver	Date Submitted
Reassign Approve / Reject	INV-0002	Invoice Statement	Smith, Bob	1/13/2010 9:28 AM

En savoir plus

La page de détails Processus d'approbation permet d'afficher un diagramme de processus d'approbation ainsi que les actions déclenchées à chaque étape.

1. Dans la **Configuration**, cliquez sur **Processus d'approbation**, puis sélectionnez le processus que vous avez créé.
2. Cliquez sur **Afficher le diagramme**. Le diagramme de processus d'approbation s'affiche dans une nouvelle fenêtre de navigateur.

**Voir aussi :**

[Didacticiel n° 4 : Utilisation de workflow et d'approbations](#)

[Étape 7 : Création d'un utilisateur](#)

[Résumé](#)

Résumé

Les règles de workflow et les processus d'approbation permettent d'automatiser vos processus commerciaux et d'appliquer vos normes. La règle de workflow met automatiquement à jour le prix et l'inventaire dans différents objets. Le processus d'approbation est exécuté dès qu'une condition est remplie et envoie une alerte par e-mail au destinataire désigné.



Important: Si vous utilisez la version Free Edition, les didacticiels qui suivent nécessitent un développement dans un environnement test sandbox. Si vous ne développez pas dans un environnement test sandbox, suivez les instructions de l'étape [Création et connexion à un environnement test sandbox](#) à la page 90. La copie de votre environnement test sandbox ne duplique pas les enregistrements pour vous. Par conséquent, une fois connecté, créez de nouveau l'enregistrement Wee Jet de l'[Étape 5 : Création d'un enregistrement Article](#) à la page 11 ainsi que le Relevé de facture de l'[Étape 5 : Création d'un enregistrement Facture](#) à la page 18.

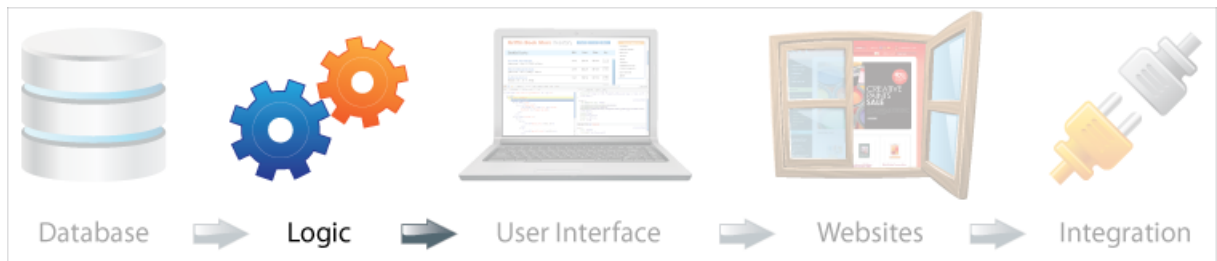
Voir aussi :

[Didacticiel n° 4 : Utilisation de workflow et d'approbations](#)

[Étape 8 : Test du processus d'approbation](#)

Didacticiel n° 5 : Ajout d'une logique commerciale avec Apex

Niveau : Avancé ; Durée : 20 à 30 minutes



Apex est un langage de programmation fortement typé, orienté objet et de type Java™ qui s'exécute sous Force.com. Apex permet d'ajouter une logique commerciale programmée aux applications. Par exemple, vous pouvez l'utiliser pour rédiger des déclencheurs, des services Web et des contrôleurs de programme dans la couche interface utilisateur de votre sous-application.

Vous avez déjà ajouté une logique commerciale en utilisant l'environnement de workflow déclaratif. Dans ce didacticiel, vous allez créer une logique supplémentaire en utilisant le langage Apex, qui est idéal pour gérer de multiples enregistrements et une logique complexe.

L'étude de cas sur laquelle nous allons nous appuyer stipule que lors de la baisse du prix d'un article, les clients bénéficient de la réduction. Pour cela, vous créez un déclencheur Apex qui met à jour toutes les factures en cours dès que le prix de l'article diminue. Un déclencheur est un ensemble de codes qui s'exécutent à un moment donné du cycle de vie d'un enregistrement. Dans notre cas, vous allez créer un déclencheur qui s'exécute après la mise à jour d'un article.

Il existe deux façons de développer des applications dans Force.com : vous pouvez utiliser l'environnement en ligne, comme vous l'avez fait jusqu'à présent, ou utiliser l'IDE Force.com. Si l'IDE n'est pas installé, son installation va être prendre du temps. Cependant, si vous maîtrisez l'IDE, vous connaissez les fonctionnalités Syntax highlighting (mise en évidence de la syntaxe), Code insight (prédiction de code) et bien d'autres, qui optimisent la productivité du développement, notamment en équipe.



Remarque: Si vous utilisez la version Free Edition, l'ensemble du développement doit être effectué dans votre environnement test sandbox. Si vous n'avez pas encore créé d'environnement test sandbox, reportez-vous à [Création et connexion à un environnement test Sandbox](#) à la page 90.

Prérequis

Connaissances de base

Pour ce didacticiel, il est préférable de maîtriser les notions de base de langages de programmation orientée objet tels que Java ou C#, sans être indispensable. Ce didacticiel peut également être accompli dans l'IDE Force.com, par conséquent, maîtriser Eclipse peut être utile mais n'est pas indispensable.

Configuration logicielle requise

Vous pouvez utiliser l'IDE Force.com pour ce didacticiel. Dans ce cas, Eclipse 3.3 ou 3.4 et le composant IDE Force.com sont requis : wiki.developerforce.com/index.php/Force.com_IDE. Pour créer un projet, reportez-vous à [Création d'un projet dans l'IDE Force.com](#) à la page 90.

Les autorisations « Mode développeur », « Modifier toutes les données » et « Auteur Apex »

Puisque ce didacticiel s'appuie sur le langage Apex, assurez-vous de disposer des autorisations requises pour créer des classes Apex.

Voir aussi :

[Étape 1 : Création d'une définition de déclencheur Apex](#)

[Étape 2 : Définition d'une variable de liste](#)

[Étape 3 : Itération sur une liste et modification du prix](#)

[Étape 4 : Test du déclencheur](#)

[Résumé](#)

Étape 1 : Création d'une définition de déclencheur Apex

Vous devez commencer par créer la définition du déclencheur, qui contient le nom du déclencheur, l'objet affecté et une action qui lance le déclencheur. Vous pouvez créer des déclencheurs dans l'interface Web ou dans l'IDE Force.com. Les instructions sont fournies pour les deux outils.

Pour créer un déclencheur dans l'interface Web :

1. Cliquez sur **Configuration** ► **Créer** ► **Objets** dans le menu latéral.
2. Cliquez sur votre objet personnalisé **Article**.
3. Dans la page de détails Article, accédez à **Déclencheurs**, puis cliquez sur **Nouveau**.
4. Remplacez <name> et <events> afin d'obtenir le code suivant :

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

5. Si vous utilisez l'interface Web, cliquez sur **Enregistrement rapide**, qui enregistre votre travail et vous permet de continuer. L'enregistrement de votre travail à ce stade permet également de vérifier le code saisi, car si la syntaxe est incorrecte, le système n'enregistre pas.

Pour créer le déclencheur dans l'IDE Force.com :

1. Dans le Package Explorer, cliquez avec le bouton droit sur votre projet, puis cliquez sur **New** ► **Apex Trigger**.
2. Dans la boîte de dialogue, saisissez `HandleProductPriceChange` pour le nommer.
3. Dans la liste déroulante **Object**, sélectionnez `Article__c`.



Remarque: Si cet objet ne figure pas dans la liste des objets, cliquez sur **Refresh Objects** (ignorez l'avertissement relatif à la sélection d'une opération).

4. Sélectionnez `after update`.

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. Cliquez sur **Finish**. Le fichier s'ouvre dans l'éditeur.

En savoir plus

Votre déclencheur ne lance aucune opération pour le moment, mais il est prêt pour accepter toute logique que vous souhaitez exécuter lors de la mise à jour d'un enregistrement Article. Avant de passer à la logique du déclencheur, détaillons et examinons chaque partie de la définition du déclencheur.

- `HandleProductPriceChange` : correspond au nom du déclencheur.
- `on Article__c` : correspond à l'objet sur lequel le déclencheur agit, soit l'objet personnalisé `Article__c` dans notre cas.
- `(after update)` : correspond à l'action qui lance le déclencheur. Les déclencheurs Apex sont exécutés en réponse à des actions sur les données, telles que des insertions, des mises à jour et des suppressions, avant ou après l'un de ces événements. Le déclencheur que vous définissez est exécuté après la mise à jour d'un enregistrement.
- `{ }` : le code inséré entre accolades est appelé *body* (corps) et détermine l'action du déclencheur. Vous allez coder le déclencheur à l'étape suivante.

Voir aussi :

[Didacticiel n° 5 : Ajout d'une logique commerciale avec Apex](#)

[Étape 2 : Définition d'une variable de liste](#)

Étape 2 : Définition d'une variable de liste

Vous devez commencer par définir une variable de liste qui contient une liste d'éléments de ligne. Les éléments de ligne ne sont pas tous nécessaires. Sélectionnez uniquement ceux qui figurent dans l'ensemble des enregistrements qui a déclenché l'exécution de ce code et dont le statut est Négociation.

1. Entre les accolades de votre définition de déclencheur, saisissez le commentaire suivant, puis instruisez une variable de liste.

```
// mettre à jour les éléments de ligne associés à des factures en cours
List<El_ment_de_ligne__c> openLineItems = [ ];
```



Remarque: Pour gagner du temps, nous n'allons rien commenter ici, mais il est recommandé de commenter votre code.

2. Entre les parenthèses carrées, saisissez une requête qui récupère les informations depuis votre objet personnalisé Élément de ligne.

```
List<El_ment_de_ligne__c> openLineItems = [SELECT j.Prix_unitaire__c, j.Article__r.Prix__c
FROM El_ment_de_ligne__c j WHERE j.Relev_de_facture__r.Statut__c = 'Négociation' AND
j.Article__r.id IN :Trigger.new FOR UPDATE];
```



Remarque: Comme nous essayons d'ajuster notre exemple de code à la page, il contient des sauts de ligne non indispensables dans le code. Apex ignore les espaces, par conséquent, le format de votre code peut différer. En outre, Apex n'est pas sensible à la casse, par conséquent, vous pouvez saisir par exemple `select` au lieu de `SELECT`.

En savoir plus

La liste `openLineItems` contient une liste d'enregistrements de votre objet personnalisé `El_ment_de_ligne__c`. Le contenu de cette liste est déterminé par la requête insérée entre les accolades, qui est écrite en langage SOQL (Salesforce Object Query Language). Examinons cette requête en détails.

- `SELECT` : détermine quels champs sont récupérés dans l'objet.
- `FROM` : détermine le ou les objets auxquels vous souhaitez accéder. Le « `j` » dans `El_ment_de_ligne__c j` est un alias : il correspond à un raccourci pratique qui permet de se référer à `El_ment_de_ligne__c` avec « `j` ».
- `WHERE` : correspond au début de l'instruction de la condition. Dans notre cas, vous souhaitez renvoyer uniquement les enregistrements dont le statut est `Négociation`.
- `AND` : correspond à la deuxième condition de l'instruction. Elle récupère les ID uniques des nouveaux enregistrements. Le code utilise une variable spéciale, `Trigger.new`, qui est automatiquement initialisée avec les identificateurs des enregistrements mis à jour.
- `FOR UPDATE` : indique à la plate-forme de verrouiller les enregistrements, afin d'empêcher les autres programmes et utilisateurs de les mettre à jour. Le verrou reste en place jusqu'au terme du déclencheur.

Voir aussi :

[Didacticiel n° 5 : Ajout d'une logique commerciale avec Apex](#)

[Étape 1 : Création d'une définition de déclencheur Apex](#)

[Étape 3 : Itération sur une liste et modification du prix](#)

Étape 3 : Itération sur une liste et modification du prix

À l'étape précédente, vous avez créé une liste d'éléments de ligne, puis vous l'avez stockée dans une variable appelée `openLineItems`. Vous pouvez maintenant itérer sur la liste à l'aide d'une boucle `for` et modifier le prix initial d'un article si le nouveau prix est inférieur.

1. Déclarez une boucle `for`.

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

2. Entre les accolades, saisissez une instruction conditionnelle `if`.

```
for (El_ment_de_ligne__c li: openLineItems) { if ( li.Article__r.Prix__c <
li.Prix_unitaire__c ){ } }
```

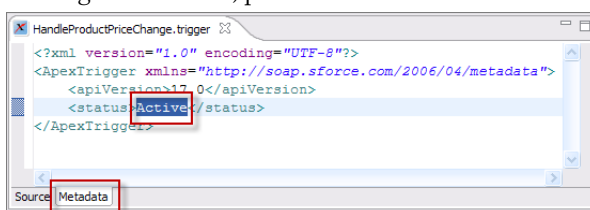
3. Entre les accolades de la boucle `for`, saisissez le code qui met le prix à jour.

```
for (El_ment_de_ligne__c li: openLineItems) { if ( li.Article__r.Prix__c <
li.Prix_unitaire__c ){ li.Prix_unitaire__c = li.Article__r.Prix__c; } }
```

4. La boucle `for` est terminée. Vous allez maintenant mettre à jour les éléments de ligne. Avant la dernière accolade du déclencheur, ajoutez le code suivant :

```
update openLineItems;
```

5. Durant le développement, vous laissez le déclencheur inactif. Pour l'essayer, vous devez l'activer.
- Si vous utilisez l'interface Web, sélectionnez la case **Est actif**, qui est disponible une fois le déclencheur enregistré.
 - Si vous utilisez l'IDE, cliquez sur l'onglet **Metadata**, puis modifiez la valeur de statut en **Active**.



6. Vérifiez que votre code se présente comme dans l'illustration ci-dessous, puis cliquez sur **Enregistrer**.

```
trigger HandleProductPriceChange on Article__c (after update) {
    List<El ment_de_ligne__c> openLineItems = [SELECT j.Prix_unitaire__c,
j.Article__r.Prix__c FROM El ment_de_ligne__c j WHERE j.Relev_de_facture__r.Statut__c =
'Négociation' AND j.Article__r.id IN :Trigger.new FOR UPDATE];

    for (El ment_de_ligne__c li: openLineItems) { if ( li.Article__r.Prix__c <
li.Prix_unitaire__c ) { li.Prix_unitaire__c = li.Article__r.Prix__c; } } update
openLineItems; }
```

En savoir plus

L'instruction finale, `update openLineItems`, met à jour les enregistrements dans la base de données. Cette opération est simple, mais quid de la boucle `for` ?

- `for (El ment_de_ligne__c li: openLineItems) { } :` itère sur la liste pour ouvrir l'élément de ligne. Lors de l'itération, l'élément de ligne actuel est attribué à la variable `li`.
- `if (li.Article__r.Prix__c < li.Prix_unitaire__c) { } :` vérifie si le prix de l'enregistrement de l'article est inférieur au prix actuel. Vous souhaitez exécuter une action uniquement si la condition est vraie.
- `li.Prix_unitaire__c = li.Article__r.Prix__c ;` : cette instruction finale met à jour le prix unitaire dans l'élément de ligne actuel (attribué à la variable `li`) avec le nouveau prix de l'article.

Voir aussi :

[Didacticiel n° 5 : Ajout d'une logique commerciale avec Apex](#)

[Étape 2 : Définition d'une variable de liste](#)

[Étape 4 : Test du déclencheur](#)

Étape 4 : Test du déclencheur

Nous allons maintenant tester le déclencheur dans la sous-application. Vous devez commencer par créer un relevé de facture et commande au moins un produit ou modifier un produit existant. La seule exigence est de modifier le champ de statut du

relevé de facture sur *Négociation*. Vous devez ensuite diminuer le prix unitaire d'un article utilisé dans l'un des éléments de ligne de cette facture, puis vérifier que les valeurs de l'élément de ligne et de la facture sont mises à jour.

1. Cliquez sur l'onglet *Relevés de facture*.
2. Cliquez sur le nom d'un relevé de facture existant.
3. Changez son *Statut* en *Négociation*, puis cliquez sur **Enregistrer**.
4. Notez la valeur totale de la facture. Cliquez ensuite sur un *Élément de ligne* et notez son *Prix unitaire*.
5. Cliquez sur l'onglet *Articles*, puis sélectionnez l'enregistrement *Wee Jet* (ou tout autre article utilisé dans les éléments de ligne).
6. Modifiez l'enregistrement en diminuant le prix unitaire. Pour simuler une réduction extrême, saisissez 0,01.
7. Revenez à la facture et à l'*Élément de ligne*.
8. Notez la modification automatique des valeurs de l'élément de ligne et de la facture.

En savoir plus

Vous avez peut-être remarqué que ce déclencheur est un peu dispendieux : il s'initialise puis fonctionne dans une liste de tous les enregistrements d'éléments de ligne qui appartiennent aux relevés de facture en cours de négociation. Cependant, la mise à jour d'un enregistrement de l'article n'entraîne pas systématiquement la mise à jour de l'enregistrement par le déclencheur. Par exemple, si vous mettez à jour la description de l'enregistrement de l'article ou augmentez son prix, ces conditions n'affectent pas le déclencheur. L'idéal serait que le déclencheur itère sur l'ensemble `Trigger.new` (la liste des enregistrements *Articles* qui ont été mis à jour), créant ainsi un nouvel ensemble d'enregistrements respectant le critère du prix, puis qu'il récupère ensuite les enregistrements d'éléments de ligne associés.

Voir aussi :

[Didacticiel n° 5 : Ajout d'une logique commerciale avec Apex](#)

[Étape 3 : Itération sur une liste et modification du prix](#)

[Résumé](#)

Résumé

Dans ce didacticiel, vous avez développé l'application en ajoutant une logique commerciale qui met à jour toutes les factures en cours lorsque le prix unitaire d'un article diminue. Pour cela, vous avez défini un déclencheur qui s'exécute dès qu'un enregistrement d'article est sauvegardé avec un prix unitaire inférieur. Les déclencheurs sont très utiles dans des scénarios tels que la mise à jour de multiples enregistrements dans des conditions particulières. Plusieurs déclencheurs peuvent être associés à un objet à déclencher en fonction d'événements et de conditions différents.

Lors de la création de déclencheurs Apex, il est important de disposer de tests automatiques qui vérifient leur fonctionnement. Vous allez apprendre comment effectuer un test dans le [Didacticiel n°6 : Ajout de tests à l'application](#) à la page 47.

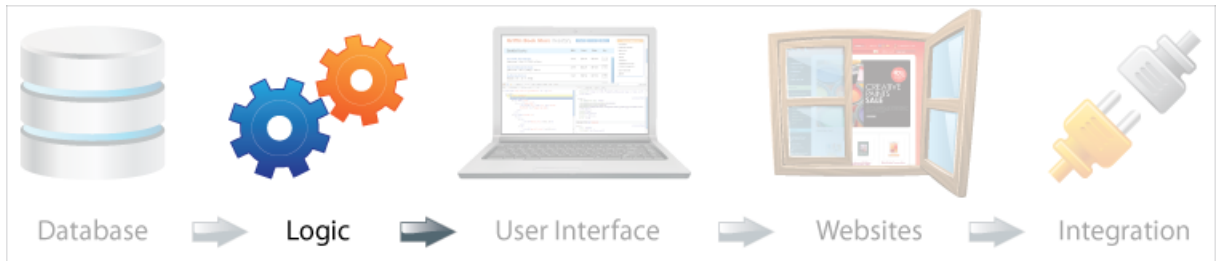
Voir aussi :

[Didacticiel n° 5 : Ajout d'une logique commerciale avec Apex](#)

[Étape 4 : Test du déclencheur](#)

Didacticiel n° 6 : Ajout de tests à l'application

Niveau : Avancé ; Durée : 20 à 30 minutes



Les tests représentent l'une des étapes les plus importantes dans le développement d'une application. Ils permettent de vérifier le comportement de votre code, en s'assurant qu'il n'abuse pas des ressources disponibles. Le test est bénéfique pour vous (il permet d'avoir confiance dans les applications que vous écrivez), mais il est également bénéfique pour la plate-forme. De son côté, la plate-forme nécessite de tester tout code Apex que vous écrivez avant de le déployer dans un environnement de production. Des tests sont également exécutés avant la publication d'une nouvelle version de la plate-forme, afin de vérifier l'absence de tout problème de rétrocompatibilité.

Pour faciliter le test, Apex prend en charge des *tests d'unité*, qui valident la programmation du comportement du code ainsi que les résultats attendus. Tout code Apex inclut un ensemble de limites qui détermine la quantité de ressources consommées. Par exemple, le nombre de requêtes exécutables dans un déclencheur est limité. Les tests de bonne qualité valident le comportement du code, mais vérifient également qu'il ne dépasse pas ces limites. Dans ce didacticiel, vous allez écrire et exécuter un test d'unité, qui essaie le déclencheur Apex que vous venez de créer.

Prérequis

Didacticiel Apex

Vous devez créer un déclencheur Apex en suivant les instructions du [Didacticiel n° 5 : Ajout d'une logique commerciale avec Apex](#) à la page 41.

Configuration logicielle requise

Vous pouvez utiliser l'IDE Force.com pour ce didacticiel. Dans ce cas, Eclipse 3.3 ou 3.4 et le composant IDE Force.com sont requis : wiki.developerforce.com/index.php/Force.com_IDE. Pour créer un projet, reportez-vous à [Création d'un projet dans l'IDE Force.com](#) à la page 90.

Voir aussi :

[Étape 1 : Création d'une classe de test Apex](#)

[Étape 2 : Ajout de méthodes de test à la classe](#)

[Étape 3 : Écriture du code exécutant le déclencheur](#)

[Étape 4 : Exécution du test](#)

[Étape 5 : Affichage de la couverture de code et amélioration des tests](#)

[Résumé](#)

Étape 1 : Création d'une classe de test Apex

Toutes les unités testées sont incluses dans des classes Apex. Dans cette étape, vous allez créer la classe qui comprend les tests d'unités.

Pour créer une classe Apex dans l'interface Web :

1. Cliquez sur **Configuration** ► **Développer** ► **Classes Apex**.
2. Cliquez sur **Nouveau**.
3. Dans le volet de l'éditeur, saisissez le code suivant :

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

4. Cliquez sur **Enregistrement rapide** pour enregistrer et poursuivre l'édition.

Pour créer une classe Apex dans l'IDE Force.com :

1. Dans l'IDE, cliquez avec le bouton droit sur le dossier de votre projet. Cliquez ensuite sur **New** ► **Apex Class**.
2. Dans la page Create Apex Class, saisissez `TestHandleProductPriceChange` pour la nommer.
3. Dans le champ `Template`, sélectionnez `Test Class`.
4. Cliquez sur **Finish** pour créer la classe.

En savoir plus

L'annotation `@isTest` indique à Force.com que tout le code compris dans la classe Apex permet de tester le reste de votre code. Vous allez ensuite créer des méthodes de test dans cette classe, qui exécutent les tests.

Voir aussi :

[Didacticiel n° 6 : Ajout de tests à l'application](#)

[Étape 2 : Ajout de méthodes de test à la classe](#)

Étape 2 : Ajout de méthodes de test à la classe

Vous allez maintenant ajouter à la classe une méthode chargée d'exécuter le test. Le déclencheur que vous avez créé, et que vous souhaitez tester, fonctionne uniquement lorsque les enregistrements sont mis à jour. Par conséquent, commencez par créer des enregistrements de test dans la base de données. Vous souhaitez que la méthode crée un Relevé de facture ainsi que des enregistrements Articles et Élément de ligne, puis les ajouter à la base de données.

1. Commencez par créer la méthode de test qui doit contenir vos trois procédures de test. Entre les accolades, saisissez le code suivant :

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



Remarque: Si vous utilisez l'IDE, remplacez le commentaire et le code générés.

- Entre les accolades, insérez le code ci-dessous, qui crée un relevé de facture et l'insère dans la base de données.

```
Relev_de_facture__c invoice = new Relev_de_facture__c(Statut__c = 'Négociation'); insert invoice;
```

- Sous `insert invoice`, ajoutez le code qui crée les enregistrements d'articles.

```
Article__c[] products = new Article__c[]{ new Article__c(Name = 'item 1', Description__c = 'test product 1', Prix__c = 10, Inventaire_total__c = 10), new Article__c(Name = 'item 2', Description__c = 'test product 2', Prix__c = 11, Inventaire_total__c = 10) }; insert products;
```

- Sous `insert products`, saisissez le code pour ajouter des éléments de ligne.

```
El_ment_de_ligne__c[] lineItems = new El_ment_de_ligne__c[] { new El_ment_de_ligne__c(Relev_de_facture__c = invoice.id, Article__c = products[0].id, Prix_unitaire__c = 10, Unit_s_vendues__c = 3), new El_ment_de_ligne__c(Relev_de_facture__c = invoice.id, Article__c = products[1].id, Prix_unitaire__c = 11, Unit_s_vendues__c = 6) }; insert lineItems;
```

Vous venez de configurer les données. Notez le statut du relevé de facture, que vous avez créé, défini sur Négociation, qui correspond à la condition requise pour lancer le déclencheur. Vous devez maintenant écrire le code qui doit remplir les autres conditions requise pour le déclencheur : vous devez diminuer le prix d'un article. Vous allez le faire à l'étape suivante.

Voir aussi :

[Didacticiel n° 6 : Ajout de tests à l'application](#)

[Étape 1 : Création d'une classe de test Apex](#)

[Étape 3 : Ecriture du code exécutant le déclencheur](#)

Étape 3 : Ecriture du code exécutant le déclencheur

Tel qu'il se présente, le code de votre méthode de test n'est pas très utile : il ne stimule pas le déclencheur, qui agit uniquement lors de la mise à jour du prix d'un enregistrement Article.

- Après `insert lineItems;`, ajoutez les lignes suivantes.

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



Remarque: Ce code modifie le prix d'un enregistrement Article, en l'augmentant. Il appelle ensuite `startTest()`, qui fait partie du framework de test et marque dans le code le début de votre test. Le système ne considère pas le code de configuration et les opérations de base de données qui précèdent comme faisant partie du test. Le test permet également de vérifier que votre code ne dépasse pas certaines limites administratives, qui définissent les ressources que votre code Apex peut utiliser. Lorsque le début et la fin du test sont marqués, le système ne prend pas en compte votre code de configuration dans les limites administratives, ce qui permet de refléter avec plus de précision comment le code se comporterait dans un environnement de production.

2. Vous venez de mettre à jour les produits. Vous devez maintenant vérifier si le prix des éléments de ligne a changé. Il ne devrait pas avoir changé, car le prix a augmenté dans le code de configuration. Ajoutez le code suivant après `Test.stopTest()` ;

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

3. Vérifiez que votre code se présente comme dans l'illustration ci-dessous, puis cliquez sur **Enregistrer**.

```
@isTest private class TestHandleProductPriceChange {
    static testMethod void testPriceChange() { Relev_de_facture__c invoice = new
Relev_de_facture__c (Statut__c = 'Négociation'); insert invoice;

    Article__c[] products = new Article__c[]{ new Article__c(Name = 'item 1',
Description__c = 'test product 1', Prix__c = 10, Inventaire_total__c = 10), new
Article__c(Name = 'item 2', Description__c = 'test product 2', Prix__c = 11,
Inventaire_total__c = 10) }; insert products; El ment_de_ligne__c[] lineItems = new
El ment_de_ligne__c[] { new El ment_de_ligne__c(Relev_de_facture__c = invoice.id,
Article__c = products[0].id, Prix_unitaire__c = 10, Unit_s_vendues__c = 3), new
El ment_de_ligne__c(Relev_de_facture__c = invoice.id, Article__c = products[1].id,
Prix_unitaire__c = 11, Unit_s_vendues__c = 6) }; insert lineItems;

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

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

Voir aussi :

[Didacticiel n° 6 : Ajout de tests à l'application](#)

[Étape 2 : Ajout de méthodes de test à la classe](#)

[Étape 4 : Exécution du test](#)

Étape 4 : Exécution du test

Force.com dispose d'un framework (cadre d'applications) de test qui permet d'exécuter des tests et de vérifier la couverture du code. Vous allez maintenant exécuter les tests et observer la couverture du code qui en résulte.



Remarque: Ces étapes sont présentées dans l'interface Web , mais vous pouvez également exécuter des tests dans l'IDE Force.com en cliquant avec le bouton droit sur le Package Explorer et en sélectionnant **Force.com ► Run Tests**. La suite de ce didacticiel utilise l'interface Web, mais vous pouvez aisément suivre les étapes dans l'IDE.

1. Accédez à votre classe de test en cliquant sur **Configuration ► Développeur ► Classes Apex**, puis sur votre classe de test `TestHandleProductsPriceChange`.
2. Cliquez sur **Exécuter le test**. Un résultat semblable à l'illustration ci-dessous s'affiche.

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

Ce résultat donne plusieurs informations importantes :

- Il indique si vos tests ont réussi ou non. Si la condition booléenne de l'instruction `system.assert`, que vous avez ajoutée au test, avait échoué, ce résultat serait indiqué ici. L'ajout de nombreuses assertions est une méthode efficace pour tester le comportement attendu de votre code.
- Il renvoie des informations sur l'exécution du test. En parcourant le journal de débogage, vous pouvez par exemple constater que votre règle de validation `Commande_en_stock` a été déclenchée. Vous pouvez observer les enregistrements créés, le nombre de requêtes exécutées et davantage.
- Il indique la couverture du code : le nombre de lignes de code exécutées dans d'autres classes ou déclencheurs.

Notez également que le résultat indique une couverture de 80 % pour le déclencheur `HandleProductPriceChange`. Cette couverture est suffisante pour le déploiement, mais nous recherchons la perfection. À l'étape suivante, nous allons afficher la couverture du code afin de détecter les emplacements auxquels vous devez ajouter des tests pour obtenir un taux de 100 %.

Voir aussi :

[Didacticiel n° 6 : Ajout de tests à l'application](#)

[Étape 3 : Écriture du code exécutant le déclencheur](#)

[Étape 5 : Affichage de la couverture de code et amélioration des tests](#)

Étape 5 : Affichage de la couverture de code et amélioration des tests

Vous avez écrit deux séries de code. Vous avez un déclencheur, que nous allons appeler *code de production*, et un code dans la classe de test, que nous allons appeler *code de test*. Le terme *couverture de code* se rapporte à la proportion du code de production couverte par votre code de test. En d'autres termes, lorsqu'un utilisateur exécute votre code de test, exécute-t-il le code de production dans son intégralité ou en partie seulement ? S'il exécute une partie du code seulement, cela peut indiquer la présence de bogues dans la partie non testée de votre code de production. Pour illustrer ces concepts, vous pouvez afficher un graphique de couverture de code.



Remarque: Si vous utilisez l'IDE Force.com pour ce didacticiel, continuez avec l'IDE. Le développement simultané dans l'IDE et dans l'interface Web nécessite une synchronisation avec le serveur à chaque passage de l'une à l'autre. La synchronisation est aisée, il suffit de cliquer avec le bouton droit sur votre classe et de sélectionner **Force.com** ► **Synchronize with Server**, mais il est encore plus facile de s'en passer !

1. Dans la section Couverture de code, cliquez sur **80**.

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

2. La page Couverture de code s'ouvre. Le surlignage bleu marque les lignes de code qui *ont* été couvertes (exécutées) selon notre méthode de test. Le surlignage rouge marque les lignes qui *n'ont pas* été exécutées. Dans notre cas, le ligne 12 n'a pas été exécutée car nous ne diminuons pas le prix d'un article, mais nous l'augmentons.

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. Cette ligne rouge indique clairement que votre test n'est pas complet. Modifions maintenant la méthode de test afin d'améliorer la couverture de code. Fermez le fenêtre Couverture de code, puis cliquez sur **Configuration** ► **Développer** ► **Classes Apex** et sélectionnez votre classe de test **TestHandleProductPriceChange**.
4. Cliquez sur **Modifier**.
5. Remplacez le code compris entre `products[0].prix__c = 20;` et l'accolade suivante par le code ci-dessous :

```
products[0].prix__c = 20; // augmenter le prix products[1].prix__c = 5; // diminuer le
prix Test.startTest(); update products; Test.stopTest();

lineItems = [SELECT id, prix_unitaire__c FROM El_ement_de_ligne__c WHERE id IN :lineItems];
System.assert(lineItems[0].prix_unitaire__c == 10); // inchangé
System.assert(lineItems[1].prix_unitaire__c == 5); // changé !
```

6. Cliquez sur **Enregistrer**.
7. Cliquez ensuite sur **Exécuter le test**. Vous obtenez une couverture de code de 100 %.

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

Voir aussi :

[Didacticiel n° 6 : Ajout de tests à l'application](#)

[Étape 4 : Exécution du test](#)

[Résumé](#)

Résumé

Dans ce didacticiel, vous avez créé des tests pour le déclencheur Apex et observé comment les outils de test intégrés peuvent aider à obtenir une couverture de code de 100 %. La création de tests d'unités lors du développement est nécessaire pour le déploiement et représente aussi l'une des clés de la réussite à long terme d'un déploiement.

Il est toutefois important de noter que la couverture de code n'est pas le seul objectif d'un test. Vous devez également valider tous les scénarios possibles et il en manque un au test que vous venez de créer. En l'état, le test vérifie uniquement les relevés de facture dont le statut est Négociation. Il devrait également contrôler les autres valeurs de statut. Par exemple, si le statut d'un relevé de facture *n'est pas* Négociation, l'augmentation ou la diminution du prix ne devrait avoir aucun effet.

Voir aussi :

[Didacticiel n° 6 : Ajout de tests à l'application](#)

[Étape 5 : Affichage de la couverture de code et amélioration des tests](#)

Didacticiel n° 7 : Élaboration d'une interface utilisateur personnalisée en utilisant Visualforce

Niveau : Intermédiaire ; **Durée :** 30 à 45 minutes



Visualforce est un framework (cadre d'applications) d'interface utilisateur basée sur des composants pour la plate-forme Force.com. Dans les didacticiels précédents, nous avons élaboré et étendu votre application en utilisant une interface utilisateur générée automatiquement. Visualforce offre beaucoup plus de contrôle sur l'interface utilisateur grâce à un cadre d'affichage qui comprend un langage de balisage semblable à l'HTML, une bibliothèque de composants réutilisables, qui peuvent être étendue, et un modèle de contrôleur basé sur un Apex. Visualforce prend en charge le style Model-View-Controller (MVC) pour la conception de l'interface utilisateur, qui est très flexible.

Dans ce didacticiel, vous allez utiliser Visualforce afin de créer une nouvelle interface utilisateur pour l'application Entrepôt, qui affiche une feuille d'inventaire permettant de répertorier chaque article en stock et de mettre à jour les quantités. L'objectif de la feuille d'inventaire est de mettre à jour le système informatique avec les décomptes physiques des articles, s'ils changent.

Prérequis

Connaissances de base

Pour ce didacticiel, il est préférable de maîtriser les langages de balisage tels que HTML et XML, sans être indispensable.

Voir aussi :

[Étape 1 : Activation du Mode de développement de Visualforce](#)

[Étape 2 : Création d'une page Visualforce](#)

[Étape 3 : Ajout d'une ressource statique Feuille de style](#)

[Étape 4 : Ajout d'un contrôleur à la page](#)

[Étape 5 : Affichage de la Feuille d'inventaire en tant que page Visualforce](#)

[Résumé](#)

Étape 1 : Activation du Mode de développement de Visualforce

Le Mode de développement incorpore un éditeur de page Visualforce dans votre navigateur. Il permet d'afficher en même temps le code et un aperçu de la page. Le Mode de développement ajoute également un éditeur Apex pour l'édition des contrôleurs et des extensions.

1. Cliquez sur **Configuration** ► **Mes informations personnelles** ► **Informations personnelles**.
2. Cliquez sur **Modifier**.
3. Sélectionnez la case Mode de développement, puis cliquez sur **Enregistrer**.

User Detail		Edit	Change Password
Name	Admin User	Role	
Alias	AUser	User License	Salesforce
Email	mkorf5@salesforce.com	Profile	System Administrator
Username	workbook2009@de.com	Active	<input checked="" type="checkbox"/>
Community Nickname	workbook20091.2547864026304668E12	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	

Voir aussi :

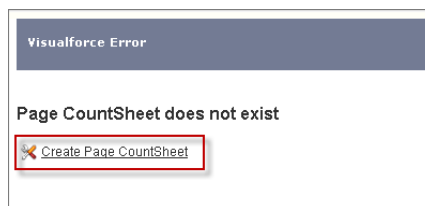
[Didacticiel n° 7 : Élaboration d'une interface utilisateur personnalisée en utilisant Visualforce](#)

[Étape 2 : Création d'une page Visualforce](#)

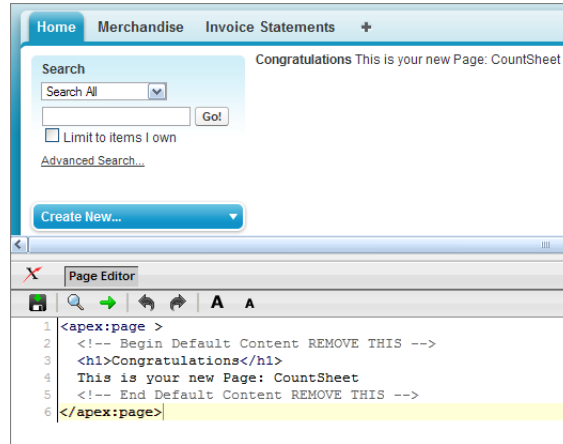
Étape 2 : Création d'une page Visualforce

Dans cette étape, vous allez créer une page Visualforce qui servira de feuille d'inventaire.

1. Dans votre navigateur, ajoutez /apex/Feuille_Inventaire à l'URL de votre instance Salesforce.com. Par exemple, si votre instance Salesforce.com est `https://na1.salesforce.com`, la nouvelle URL devient `https://na1.salesforce.com/apex/Feuille_Inventaire`. Le message d'erreur suivant s'affiche : La page Apex Feuille_Inventaire n'existe pas.



2. Cliquez sur le lien **Create Page Feuille_Inventaire** pour créer la page.
3. Cliquez sur le lien Page Editor dans le coin inférieur gauche de la page. L'onglet Page Editor affiche le code et un aperçu de la nouvelle page (qui contient un texte par défaut), comme dans l'illustration ci-dessous.



- Changez le terme « Congratulations » de la balise `<h1>` en Feuille d'inventaire et supprimez les commentaires. Le code de la page doit se présenter comme suit :

```
<apex:page>
<h1>Feuille d'inventaire</h1>
</apex:page>
```

- Cliquez sur l'icône **Enregistrer** en haut de la fenêtre Page Editor. La page est rechargée pour appliquer vos modifications.

En savoir plus

Notez que le code de la page est très semblable à un code HTML. Une page Visualforce combine des balises HTML, telles que `<h1>`, à des balises Visualforce spécifiques commençant par `<apex:>`

Voir aussi :

[Didacticiel n° 7 : Élaboration d'une interface utilisateur personnalisée en utilisant Visualforce](#)

[Étape 1 : Activation du Mode de développement de Visualforce](#)

[Étape 3 : Ajout d'une ressource statique Feuille de style](#)

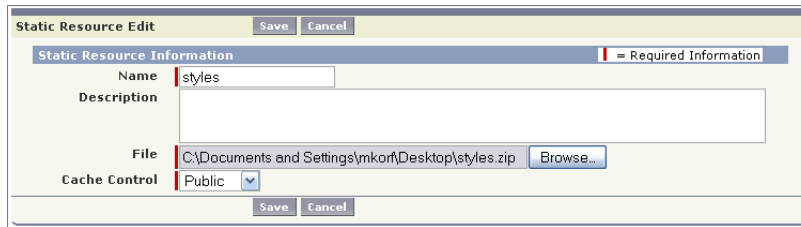
Étape 3 : Ajout d'une ressource statique Feuille de style

Afin d'améliorer la présentation de votre application Entrepôt, vous allez utiliser une feuille de style (fichier CSS) personnalisée qui spécifie la couleur, la police et la mise en page du texte sur la page. La plupart des pages Web et des concepteurs de pages Web utilisent la norme Web CSS. Nous avons créé une feuille CSS pour vous. Pour que vos pages référencent une feuille de style, vous devez la charger en tant que *ressource statique*. Une ressource statique est un fichier ou un ensemble de fichiers stocké dans Force.com. Une fois ajoutée en tant que ressource statique, une feuille de style peut être référencée par n'importe quelle page Visualforce.

Pour ajouter une feuille de style en tant que ressource statique :

- Dans votre navigateur, accédez à developer.force.com/workbook/styles. Téléchargez le fichier et enregistrez-le sur votre Bureau.
- Depuis la sous-application, cliquez sur **Configuration** ► **Développer** ► **Ressources statiques**, puis cliquez sur **Nouveau**.
- Dans le champ Nom, saisissez `Styles`.

4. Cliquez sur **Parcourir**, puis recherchez le fichier `styles.zip` que vous venez de télécharger.
5. Dans la liste de sélection **Contrôle du cache**, sélectionnez **Public**.



6. Cliquez sur **Enregistrer**.



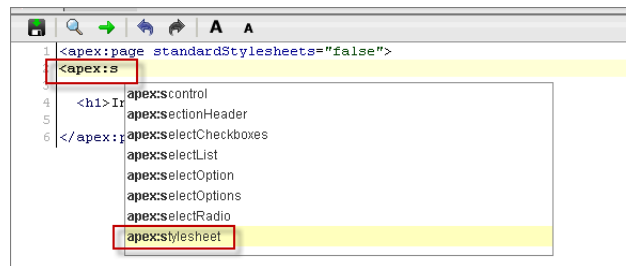
Remarque: Si vos pages Visualforce sont publiées dans un site Web public, Force.com utilise un réseau global de distribution de contenu pour conserver des copies de vos fichiers statiques dans des serveurs cache.

Vous devez maintenant modifier votre page Visualforce pour référencer la feuille de style.

1. Comme vous l'avez fait pour créer la page, ajoutez le texte `/apex/Feuille_Inventaire` à l'URL de votre instance Salesforce.com.
2. Modifiez les attributs de la balise `<apex:page>` et saisissez le code suivant pour supprimer la feuille de style standard, l'en-tête et le menu latéral.

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

3. Vous devez maintenant indiquer à la page l'emplacement de la feuille de style. Insérez une ligne sous la première balise `<apex:page>`, puis tapez `<apex:`
4. L'éditeur est doté d'une fonction de mise en évidence de la syntaxe, qui affiche une liste déroulante d'éléments disponibles dans ce contexte. Commencez à taper `stylesheet`, puis sélectionnez `apex:stylesheet` lorsque le code s'affiche dans la liste déroulante.



5. Spécifiez maintenant l'emplacement de la feuille de style comme indiqué ci-dessous.

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

6. Vérifiez que votre code se présente comme suit :

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

7. Cliquez sur l'icône **Enregistrer** en haut de la fenêtre Page Editor.

Notez la nouvelle présentation de la page avec une police et un emplacement différents pour le titre, et l'absence d'en-tête et de menu latéral.

En savoir plus

Examinons plus en détails le code de la feuille de style.

- `$Resources` est une variable globale à laquelle Visualforce a accès. `$Resource.styles` permet de référencer la ressource intitulée « Styles » que vous avez créée précédemment.
- La fonction `URLFOR()` indique l'emplacement de la ressource statique, ainsi qu'un fichier dans cette ressource, et calcule l'URL qui doit être générée dans votre page finale. Si vous avez reconnu la syntaxe, c'est que vous l'avez déjà rencontrée en évaluant dynamiquement les valeurs lors du rendu de la page Visualforce.

Voir aussi :

[Didacticiel n° 7 : Élaboration d'une interface utilisateur personnalisée en utilisant Visualforce](#)

[Étape 2 : Création d'une page Visualforce](#)

[Étape 4 : Ajout d'un contrôleur à la page](#)

Étape 4 : Ajout d'un contrôleur à la page

Grâce au modèle de conception Contrôleur-Vue-Modèle de Visualforce, il est aisé de séparer la vue et son style de la base de données et de la logique sous-jacentes. Avec ce modèle, la vue (la page Visualforce) interagit avec un contrôleur. Dans le cas présent, le contrôleur est généralement une classe Apex, qui présente quelques fonctionnalités de la page. Par exemple, le contrôleur peut contenir une logique qui doit être exécutée lors d'un clic sur un bouton. En général, un contrôleur interagit également avec le modèle (la base de données), en présentant les données que la vue souhaite afficher.

Tous les objets Force.com comprennent des contrôleurs standard par défaut que vous pouvez utiliser pour interagir avec les données associées à l'objet. Par conséquent, il est rarement nécessaire d'écrire soi-même le code du contrôleur. Vous pouvez étendre les contrôleurs standard afin d'ajouter de nouvelles fonctionnalités ou de créer de tout nouveaux contrôleurs personnalisés. Dans ce didacticiel, vous allez utiliser le contrôleur par défaut.

1. Si Page Editor n'est pas ouvert dans votre page Visualforce, cliquez sur **Page Editor** pour modifier la page.
2. Modifiez votre code afin d'activer le contrôleur standard `Article__c` standard en changeant la première balise `<apex:page>`. L'éditeur ignore les espaces entre les balises, vous pouvez saisir le texte sur une nouvelle ligne.

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

3. Ajoutez ensuite la définition du contrôleur de liste standard.

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

4. Cliquez sur l'icône **Enregistrer** en haut de la fenêtre Page Editor. La page n'affiche aucune modification. Cependant, vous avez indiqué que la page doit utiliser un contrôleur et défini la variable `products`. Par conséquent, la variable sera disponible dans le corps de la page et représentera une liste d'enregistrements Article.

En savoir plus

L'attribut `recordSetVar` active un contrôleur de liste standard qui fournit une prise en charge de contrôleur supplémentaire pour répertorier un nombre d'enregistrements avec la pagination. Sa valeur, que vous avez définie sur « products », permet d'inclure à une nouvelle variable, `products`, l'ensemble des enregistrements à afficher dans la page Visualforce.

Voir aussi :

[Didacticiel n° 7 : Élaboration d'une interface utilisateur personnalisée en utilisant Visualforce](#)

[Étape 3 : Ajout d'une ressource statique Feuille de style](#)

[Étape 5 : Affichage de la Feuille d'inventaire en tant que page Visualforce](#)

Étape 5 : Affichage de la Feuille d'inventaire en tant que page Visualforce

Toutes les fonctionnalités sont désormais en place pour étoffer la page Visualforce. Elle affiche un tableau de tous les enregistrements d'articles, avec un champ de saisie qui permet de mettre à jour l'inventaire.

1. Sur une nouvelle ligne, sous la balise `</h1>`, commencez à taper `<apex:f`, puis sélectionnez la balise `<apex:form>` lorsqu'elle s'affiche dans la liste déroulante. Le formulaire permet de mettre à jour le tableau.
2. Appuyez sur Entrée et notez que le système génère automatiquement les balises ouvrante et fermante.
3. Placez le curseur entre les balises, puis créez un tableau de données. Commencez à taper `<apex:d`, puis sélectionnez `dataTable` dans la liste déroulante.
4. Vous devez maintenant ajouter des attributs à la balise `dataTable`. L'attribut de valeur indique les éléments de la liste sur lesquels le composant `dataTable` doit effectuer une itération. L'attribut `var` attribue chaque élément de cette liste, pour une seule itération, à la variable `pittem`. L'attribut `rowClasses` affecte des noms de feuille de style CSS à des lignes alternatives. Sur une ou plusieurs lignes de la balise, saisissez :

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

5. Vous allez maintenant définir chaque colonne, puis déterminer l'origine des données en recherchant le champ approprié dans la variable `pittem`. Ajoutez le code suivant entre les balises `dataTable` ouvrante et fermante.

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

6. Cliquez sur **Enregistrer** ; le tableau s'affiche.

Inventory Count Sheet	
Product	
Wee Jet	

L'attribut `headerValue` a simplement fourni un titre d'en-tête pour la colonne, qui est suivi d'une liste de lignes : une ligne pour chaque enregistrement d'article. L'expression `{!pittem.name}` indique que vous souhaitez afficher le champ de nom sur la ligne actuelle.

7. Ajoutez maintenant deux colonnes supplémentaires après la balise fermante de la première colonne.

```
<apex:column headerValue="Inventaire"> <apex:outputField
value="{!pitem.Inventaire_total_c}"/> </apex:column> <apex:column headerValue="Décompte
physique"> <apex:inputField value="{!pitem.Inventaire_total_c}"/> </apex:column>
```



Remarque: La deuxième colonne est un champ `inputField`, pas `outputField`. Le champ `inputField` affiche une valeur, mais il permet également de la modifier.

8. Cliquez sur **Enregistrer** pour générer une feuille d'inventaire ! Elle répertorie les enregistrements d'articles, affiche l'inventaire actuel et fournit un champ de saisie pour le décompte physique.
9. Pour parfaire la présentation, ajoutez un bouton permettant de modifier le décompte physique de n'importe quelle ligne, puis actualisez les valeurs de la page. Pour cela, saisissez le code suivant directement au-dessus de la ligne `</apex:form>`.

```
<br/> <apex:commandButton action="{!quicksave}" value="Actualiser le décompte" />
```

Inventory Count Sheet

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

En savoir plus

- Le composant `dataTable` produit un tableau avec des lignes, chaque ligne étant trouvée en effectuant une itération sur la liste. Le contrôleur standard que vous utilisez pour cette page a été défini sur `Article__c` et `recordSetVar` a été défini sur `products`. Le contrôleur renseigne ainsi automatiquement la variable de la liste de produits avec les enregistrements d'articles récupérés dans la base de données. Le composant `dataTable` utilise cette liste.
- Vous devez référencer la ligne actuelle pendant que vous effectuez une itération sur la liste. Cette instruction `var="pitem"` attribue une variable appelée `pitem` qui contient la ligne actuelle.
- Les attributs `rowClasses` et `styleClass` utilisent simplement certains styles de la feuille CSS que vous avez chargée dans la ressource statique. Vous pouvez les retirer sans risque, seule la présentation en souffrira !
- Chaque contrôleur standard dispose de diverses méthodes pour tous les objets Force.com. Le composant `commandButton` affiche le bouton et invoque une méthode appelée `quicksave` dans le contrôleur standard, qui met à jour les valeurs dans les enregistrements. Ici, vous mettez à jour le décompte physique du produit et procédez à un enregistrement rapide, qui actualise le produit avec le nouveau décompte.
- La mise en page n'est pas présentée dans cet exemple, mais la fonctionnalité existe. Si vous disposez d'un nombre d'enregistrements à parcourir suffisant, ajoutez le code suivant sous `commandButton` pour l'action de page-flipping (tournage de page).

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

Voir aussi :

[Didacticiel n° 7 : Élaboration d'une interface utilisateur personnalisée en utilisant Visualforce](#)

[Étape 4 : Ajout d'un contrôleur à la page](#)

[Résumé](#)

Résumé

Félicitations ! Vous avez créé une nouvelle interface pour votre application Entrepôt en créant une page Visualforce qui utilise un contrôleur standard. Votre page est hautement configurable. Par exemple, vous pouvez aisément changer les données affichées dans chaque ligne en modifiant les composants de la colonne. La page utilise également de nombreuses fonctionnalités fournies en arrière-plan par le contrôleur standard. Par exemple, le contrôleur interroge automatiquement la base de données pour rechercher tous les enregistrements d'articles et les attribuer à la variable `products`. Il fournit également un outil de sauvegarde des données via l'enregistrement rapide.

Vous avez créé quelques composants Visualforce. Vous pouvez désormais publier ces fonctionnalités dans un site Web. Vous allez découvrir cette étape dans le [Didacticiel n° 8 : Création d'une page Web publique en utilisant les Sites](#).

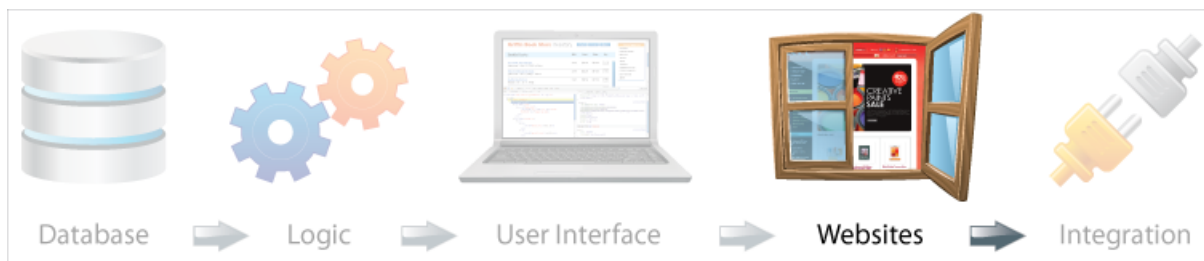
Voir aussi :

[Didacticiel n° 7 : Élaboration d'une interface utilisateur personnalisée en utilisant Visualforce](#)

[Étape 5 : Affichage de la Feuille d'inventaire en tant que page Visualforce](#)

Didacticiel n° 8 : Création d'une page Web publique en utilisant les Sites

Niveau : Intermédiaire ; **Durée :** 20 à 30 minutes



L'application que vous venez de créer nécessite que vous vous connectiez pour l'utiliser. Bien que Force.com prenne en charge votre application dans un environnement multi-utilisateurs, vous devez parfois éliminer la connexion, par exemple sur un site Web public.

Les Sites permettent de créer des sites Web publics et des applications directement intégrés à votre organisation sans que les utilisateurs se connectent avec un nom d'utilisateur et un mot de passe. Vous pouvez publier des informations stockées dans votre organisation via l'URL de marque de votre choix. Vous pouvez également adapter la présentation des pages du site à la marque de votre organisation. Les sites étant hébergés sur des serveurs Force.com, aucun problème d'intégration ne se pose. Puisque les sites sont créés sur des pages natives, la validation des données dans les informations collectées est automatiquement exécutée. Vous pouvez également permettre aux utilisateurs de s'inscrire, ou de se connecter, de façon transparente à un portail associé depuis votre site public.

Dans ce didacticiel, vous allez créer une page Visualforce, activer des Sites pour votre organisation, enregistrer votre nom de domaine Force.com et publier la page Visualforce que vous avez créée en tant que catalogue de produits public en ligne.

Prérequis

Didacticiel Visualforce

Vous devez créer la page Visualforce en suivant les instructions du [Didacticiel n° 7 : Élaboration d'une interface utilisateur personnalisée en utilisant Visualforce](#) à la page 54.

Voir aussi :

- [Étape 1 : Création d'une page de catalogue de produits](#)
- [Étape 2 : Enregistrement d'un nom de domaine Force.com](#)
- [Étape 3 : Création d'un site Force.com](#)
- [Étape 4 : Configuration et test du site](#)
- [Résumé](#)

Étape 1 : Création d'une page de catalogue de produits

Dans cette étape, vous allez cloner la page d'inventaire que vous avez créée précédemment. La nouvelle page Catalogue de produits affichera la description et le prix au lieu de l'inventaire et du décompte.

1. Cliquez sur **Configuration** ► **Développer** ► **Pages**. Si l'éditeur Visualforce est toujours ouvert, cliquez sur le bouton **Retour** de votre navigateur jusqu'à la page Configuration.
2. Sélectionnez la page **Feuille_Inventaire** que vous avez créée.
3. Cliquez sur **Cloner**.
4. Dans l'éditeur Page Editor, changez les champs **Étiquette** et **Nom** en **Catalogue**.
5. Toujours dans l'éditeur Page Editor, modifiez le contenu de la balise `<h1>` en **Catalogue de produits**.
6. Recherchez les lignes suivantes :

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

7. Changez **Inventaire** en **Description** et **Inventaire_total__c** en **Description__c**. Vous conservez le même tableau, mais en changeant l'en-tête et les données. Le code doit se présenter comme suit :

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

8. Vous allez maintenant effectuer des modifications similaires dans la colonne suivante pour afficher le prix. Changez **Décompte physique** en **Prix**. Changez **inputfield** en **outputfield**. Changez **Inventaire_total__c** en **Prix__c**, comme indiqué ci-dessous.

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

9. Pour terminer, supprimez les balises `<apex:form>` ouvrante et fermante, et `<apex:commandButton>`, car vous ne souhaitez pas accepter de saisie dans cette page.
10. Vérifiez que votre code se présente comme indiqué ci-dessous, puis cliquez sur **Enregistrer**.

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

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

  <h1>Catalogue de produits</h1>

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

</apex:page>
```

En savoir plus

À ce stade, notez plusieurs points importants.

- De nombreux composants de la plate-forme Force.com peuvent être clonés et, comme vous l'avez constaté, le clonage d'une page Visualforce est aisé.
- Les contrôleurs standard de la page Visualforce rendent toutes vos données aisément accessibles. Il suffit de modifier deux valeurs dans le tableau pour afficher les données de champs différents.

- Avez-vous remarqué qu'il est facile de modifier un champ d'entrée en champ de sortie ? Le contrôleur standard Visualforce est puissant, mais vous pouvez l'étendre pour créer des fonctionnalités personnalisées, et vous pouvez même créer votre propre contrôleur.

Voir aussi :

[Didacticiel n° 8 : Création d'une page Web publique en utilisant les Sites](#)

[Étape 2 : Enregistrement d'un nom de domaine Force.com](#)

Étape 2 : Enregistrement d'un nom de domaine Force.com

Votre domaine Force.com unique, qui héberge votre site, est formé à partir du *préfixe de domaine* unique vous enregistrez, plus de `force.com`. Par exemple, si vous choisissez « masociete » comme préfixe de domaine, votre nom de domaine est `http://www.masociete.force.com`.

Pour commencer, enregistrez le domaine Force.com de votre société en procédant comme suit.

1. Cliquez sur **Configuration** ► **Développer** ► **Sites**.
2. Saisissez un nom unique pour votre domaine Force.com. Ce nom ne peut contenir que des caractères de soulignement et alphanumériques, et il doit être unique dans votre organisation. Il doit commencer par une lettre, ne pas inclure d'espace, ne pas se terminer par un trait de soulignement et ne doit pas contenir deux traits de soulignement consécutifs. Salesforce.com recommande d'utiliser le nom de votre société ou une variante, par exemple `masociete`.



ATTENTION: Une fois votre nom de domaine Force.com enregistré, vous ne pouvez pas le modifier.

Les noms de domaine de Free Edition sont attribués automatiquement et ne peuvent pas être mis à jour. Créez une adresse Web personnalisée si vous souhaitez utiliser un nom de domaine personnalisé.

3. Cliquez sur **Vérifier la disponibilité** pour confirmer que le nom de domaine saisi est unique. S'il est déjà utilisé, vous êtes invité à le modifier.
4. Lisez, puis acceptez les Conditions d'utilisation des Sites en cochant la case.
5. Cliquez sur **Enregistrer mon domaine Force.com**. Une fois les Conditions d'utilisation acceptées et votre domaine Force.com enregistré, les modifications liées à la création du site sont consignées dans le journal d'audit et la liste associée Historique du site de votre organisation. L'entrée en vigueur de votre enregistrement peut prendre jusqu'à 48 heures.

Félicitations ! Vous êtes prêt à créer votre premier site Force.com.



Voir aussi :

[Didacticiel n° 8 : Création d'une page Web publique en utilisant les Sites](#)

[Étape 1 : Création d'une page de catalogue de produits](#)

[Étape 3 : Création d'un site Force.com](#)

Étape 3 : Création d'un site Force.com

Comme votre domaine est enregistré, vous pouvez sélectionner la page Visualforce Catalogue de produits que vous venez de créer comme page d'accueil de votre site.

1. Accédez à la page Sites en cliquant sur **Configuration** ► **Développer** ► **Sites**.
2. Cliquez sur **Nouveau**. La page Modifier le site s'affiche.
3. Dans la page Modifier le site, renseignez les informations relatives au site :
 - a. Dans les champs Étiquette du site and Nom du site, saisissez Catalogue.
 - b. Dans le champ Page d'accueil du site actif, saisissez Catalogue.
 - c. Sélectionnez la case Actif.

4. Cliquez sur **Enregistrer**.

Voir aussi :

- [Didacticiel n° 8 : Création d'une page Web publique en utilisant les Sites](#)
- [Étape 2 : Enregistrement d'un nom de domaine Force.com](#)
- [Étape 4 : Configuration et test du site](#)

Étape 4 : Configuration et test du site

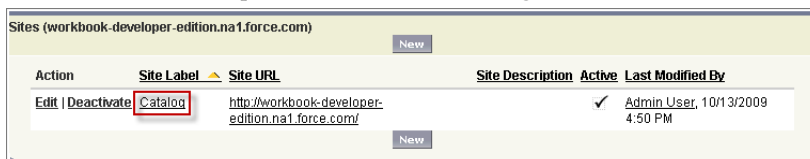
Vous avez créé votre site et désigné une page par défaut. Vous allez bientôt pouvoir l'essayer.

La plate-forme comprend plusieurs dispositifs de contrôle pour la sécurité des données. L'un des contrôles empêche l'affichage des données, même dans les pages publiques, tant que vous n'activez pas explicitement leur affichage. Dans cette étape, vous allons basculer ce paramètre de sécurité pour l'objet Article, puis visiter votre nouveau site.

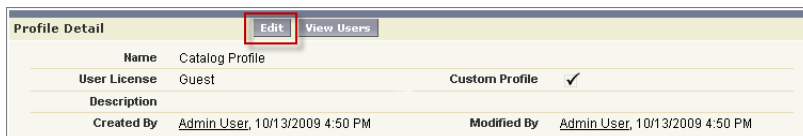
1. Accédez aux Sites en cliquant sur **Configuration** ► **Développer** ► **Sites**.
2. Cliquez sur le lien URL de site du site Catalogue de produits. Cette action ouvre un nouvel onglet ou une nouvelle fenêtre (selon votre navigateur). Cependant, au lieu de la page Catalogue de produits, une grande page Autorisation requise

s'affiche. Le visiteur anonyme du site Web n'est pas encore autorisé à visualiser les données présentées dans la page. Corrigeons ce problème. Revenez à la page de configuration.

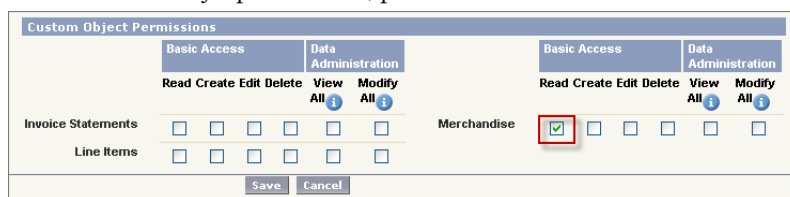
3. Cliquez sur le lien **Étiquette** du site, qui doit se nommer **Catalogue**.



4. Cliquez sur **Paramètres d'accès public**, puis cliquez sur **Modifier** dans la section Détails du profil.

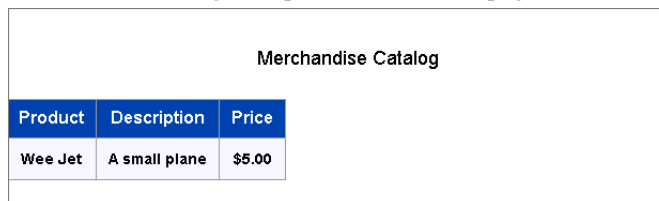


5. Accédez à la section Autorisations de l'objet personnalisé, puis sélectionnez l'autorisation **Lecture** pour l'objet **Articles**.



6. Cliquez sur **Enregistrer**.

7. Revenez à votre site Web et actualisez votre navigateur pour afficher votre page.



Remarque: Si un message d'autorisation requise s'affiche, vous n'avez probablement pas activé votre Page d'accueil du site actif sur Catalogue à l'étape précédente.

En savoir plus

- Vous avez attribué une page Visualforce unique au site et vous l'avez définie comme page d'accueil de ce site. Vous pouvez maintenant créer des pages Visualforce supplémentaires et les attribuer également au site. Par exemple, si vous ajoutez la page Meme au site, vous pouvez accéder à cette page avec une URL semblable à `http://workbook-developer-edition.na1.force.com/Meme` (selon votre nom domaine bien entendu).
- Force.com enregistre le nombre de vues de page d'un site Developer Edition et lui impose une limite de bande passante quotidienne.

- Les Sites permettent d'utiliser un serveur de distribution de contenu global afin d'accéder et de mettre en cache rapidement les pages de votre site dans des environnements de production. Il vous suffit de modifier le composant `page` pour insérer une instruction cache.

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

Voir aussi :

[Didacticiel n° 8 : Création d'une page Web publique en utilisant les Sites](#)

[Étape 3 : Création d'un site Force.com](#)

[Résumé](#)

Résumé

Félicitations, vous avez créé un site Web public ! Tout internaute dans le monde peut accéder au site que vous venez de créer, et parcourir votre page d'accueil. Pour créer le site, vous avez cloné une page Visualforce, puis activé votre organisation Developer Edition pour la fonctionnalité Sites. Vous avez ensuite enregistré un nom de domaine Force.com unique, enregistré la page comme page d'accueil, modifié la sécurité et testé le site pour vérifier son fonctionnement.

Vous disposez d'un site avec un catalogue de produits. L'étape logique suivante est de permettre aux personnes de commander des articles. Vous allez le faire dans le [Didacticiel n° 9 : Création d'une boutique](#) à la page 68.

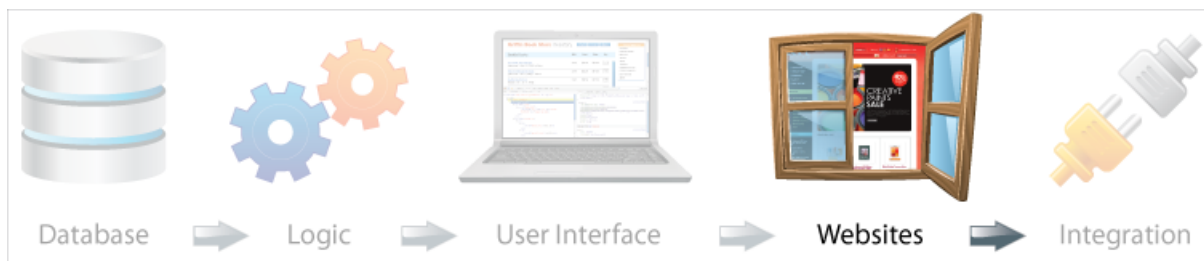
Voir aussi :

[Didacticiel n° 8 : Création d'une page Web publique en utilisant les Sites](#)

[Étape 4 : Configuration et test du site](#)

Didacticiel n° 9 : Création d'une boutique

Niveau : Intermédiaire ; Durée : 30 minutes



Apex est un langage de programmation orienté objet. Apex permet de créer des classes et des méthodes, de faire des appels à la base de données, de créer de services Web, d'envoyer des e-mails et davantage. Dans ce didacticiel, vous allez créer une simple page d'une boutique Visualforce utilisant un contrôleur que vous allez écrire en Apex. Vous allez également créer une classe Apex, découvrir une autre syntaxe Apex et apprendre comment transmettre des valeurs entre une page Visualforce et son contrôleur.

Prérequis

Sous-application Entrepôt

Vous devez commencer par créer une application Entrepôt de base en suivant les instructions du [Didacticiel n° 2 : Ajout de relations](#) à la page 13.

Configuration logicielle requise

Vous pouvez utiliser l'IDE Force.com pour ce didacticiel. Dans ce cas, Eclipse 3.3 ou 3.4 et le composant IDE Force.com sont requis : wiki.developerforce.com/index.php/Force.com_IDE. Pour créer un projet, reportez-vous à [Création d'un projet dans l'IDE Force.com](#) à la page 90.

Les autorisations « Mode développeur », « Modifier toutes les données » et « Auteur Apex »

Puisque ce didacticiel s'appuie sur le langage Apex, assurez-vous de disposer des autorisations requises pour créer des classes Apex.

Voir aussi :

[Étape 1 : Création d'un contrôleur](#)

[Étape 2 : Ajout de méthodes au contrôleur](#)

[Étape 3 : Création de la boutique](#)

[Étape 4 : Bonus - Mise à jour de la page avec AJAX](#)

[Résumé](#)

Étape 1 : Création d'un contrôleur

Au lieu d'utiliser le contrôleur par défaut, comme vous l'avez fait dans le didacticiel précédent, vous allez écrire vous-même le code du contrôleur. Généralement, les contrôleurs récupèrent les données à afficher dans une page Visualforce et contiennent un code qui est exécuté en réponse à des actions de page, par exemple un clic sur un bouton.

Votre boutique rudimentaire va inclure deux méthodes : `getProducts()`, qui renvoie les produits affichés dans votre boutique, et `shop()`, qui s'exécute lorsque les produits ont été sélectionnés.

Pour créer une classe dans l'interface Web :

1. Cliquez sur **Configuration** ► **Développer** ► **Classes Apex**.
2. Cliquez sur **Nouveau**.
3. Ajoutez le code suivant comme définition de la classe, puis cliquez sur **Enregistrement rapide**.

```
public class Boutique {
}
```

Pour créer la classe dans l'IDE Force.com :

1. Dans l'IDE, cliquez avec le bouton droit sur le dossier du projet, puis sélectionnez **New** ► **Apex Class**.
2. Dans la page Create Apex Class, saisissez `Boutique` pour la nommer.
3. Cliquez sur **Finish**.

Vous disposez désormais d'une classe rudimentaire pour votre contrôleur. Elle ne comprend ni méthode ni champ, vous les ajouterez à l'étape suivante.

Voir aussi :

[Didacticiel n° 9 : Création d'une boutique](#)

[Étape 2 : Ajout de méthodes au contrôleur](#)

Étape 2 : Ajout de méthodes au contrôleur

À l'étape précédente, vous avez créé un contrôleur personnalisé. Vous allez maintenant lui ajouter des méthodes. Pour commencer, vous allez ajouter la méthode `shop()`. Cette méthode n'a aucun effet pour le moment, mais elle sera utile comme espace réservé pour l'action exécutée lors d'un clic sur le bouton Boutique. Vous souhaitez également que la boutique affiche des articles, avec un décompte, pour permettre à l'utilisateur final de sélectionner une quantité pour chaque article qu'il souhaite acheter.

1. Ajoutez le code suivant à la classe (juste après la ligne `public class Boutique {}`).

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

2. Dans la classe actuelle (à la ligne suivante), définissez un champ qui contient une liste de `DisplayMerchandise`, ainsi qu'une classe interne appelée `DisplayMerchandise` pour contenir les données.

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

3. À la ligne suivante, définissez la méthode `getProducts()`, qui va initialiser les produits.

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

```
FROM Article__c WHERE Inventaire_total__c > 0]) { products.add(new
DisplayMerchandise(item)); } } return products; }
```

4. Vérifiez que votre code se présente comme indiqué ci-dessous, puis cliquez sur **Enregistrer**.

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

    DisplayMerchandise[] products;

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

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

Vous avez désormais terminé la définition d'un simple contrôleur. Il utilise des classes et des méthodes Apex standard pour accéder à la base de données.

En savoir plus

La classe `DisplayMerchandise` « enveloppe » le type `Article` que vous avez déjà dans la base de données, en ajoutant un nouveau champ décimal. Le constructeur vous permet de créer une instance `DisplayMerchandise` en passant un enregistrement `Article` existant. La variable d'instance `products` est définie en tant que liste d'instances `DisplayMerchandise`.

La méthode `getProducts()` exécute une requête (le texte entre crochets, également appelée requête SOQL) qui renvoie tous les enregistrements `Articles` dont l'inventaire total est positif. Elle itère ensuite sur les enregistrements renvoyés par la requête, en les ajoutant à une liste de produits `DisplayMerchandise` qui est ensuite renvoyée.

Voir aussi :

[Didacticiel n° 9 : Création d'une boutique](#)

[Étape 1 : Création d'un contrôleur](#)

[Étape 3 : Création de la boutique](#)

Étape 3 : Création de la boutique

Votre boutique utilise une autre page Visualforce, vous allez la créer maintenant.

1. Dans la barre d'adresse de votre navigateur, ajoutez `/apex/Boutique` à la fin de votre instance. Par exemple : `https://na1.salesforce.com/apex/Boutique`. Un message d'erreur que vous connaissez s'affiche.
2. Cliquez sur le lien **Create Page Boutique**.
3. Cliquez sur l'onglet Page Editor en bas de la fenêtre.
4. Remplacez le texte de l'éditeur par le code suivant :

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

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



Remarque: L'attribut `controller="Boutique"` indique à la page Visualforce que vous créez d'utiliser la classe Apex Boutique pour son contrôleur.

- Vous êtes prêt à insérer le contenu du formulaire entre les balises « form » ouvrante et fermante. Vous allez créer un tableau qui itère sur les produits et un bouton qui appelle la méthode `shop`. Saisissez le code ci-dessous à la ligne suivante, entre les balises `form` ouvrante et fermante :

```
<apex:dataTable value="{!products}" var="pitem" rowClasses="odd,even"> <apex:column
headerValue="Produit"> <apex:outputText value="{!pitem.merchandise.name}"/> </apex:column>
<apex:column headerValue="Prix"> <apex:outputText value="{!pitem.merchandise.prix__c}"/>
</apex:column> <apex:column headerValue="Quantité"> <apex:inputText
value="{!pitem.count}"/> </apex:column> </apex:dataTable> <br /> <apex:commandButton
action="{!shop}" value="Acheter" />
```

- Cliquez sur **Enregistrer**. Votre Boutique doit se présenter comme dans l'illustration ci-dessous.

The screenshot shows a Visualforce page titled "Store Front". At the top, there is a header "Store Front". Below it is a table with three columns: "Product", "Price", and "#Items". The first row of the table contains "Wee Jet", "5.0", and an empty input field. Below the table is a "Buy" button. The screenshot also shows the Page Editor interface with the Visualforce code for the page, which matches the code provided in the previous block.

En savoir plus

Nous avons utilisé ici de nombreux éléments. Examinons-les plus en détails.

- L'attribut de valeur `dataTable` est défini sur « products », qui indique que le tableau doit itérer sur une liste appelée `products`. Puisque vous utilisez un contrôleur personnalisé, Visualforce recherche automatiquement une méthode appelée `getProducts()` dans votre contrôleur Apex.
- La méthode `getProducts()` de votre contrôleur renvoie une grappe d'objets `DisplayMerchandise`. Chaque grappe forme une nouvelle ligne et est attribuée à la variable `pitem` de la page Visualforce durant l'itération de `dataTable`.

- La page Visualforce contient une action, `{!shop}`. Comme il s'agit d'une action, une méthode portant un nom identique est utilisée dans le contrôleur Apex.

Voir aussi :

[Didacticiel n° 9 : Création d'une boutique](#)

[Étape 2 : Ajout de méthodes au contrôleur](#)

[Étape 4 : Bonus - Mise à jour de la page avec AJAX](#)

Étape 4 : Bonus - Mise à jour de la page avec AJAX

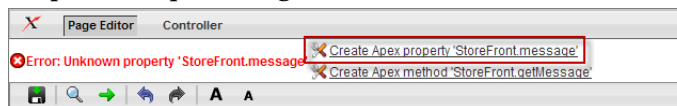
Cette étape est facultative. Elle montre comment Visualforce renvoie de façon transparente des données à votre contrôleur, dans lesquelles elles peuvent être traitées. Par exemple, la méthode `shop()` que vous avez écrite dans le contrôleur a accès aux quantités d'articles saisies par l'utilisateur final dans la page Visualforce. Cette étape présente comment accéder à ces données dans le contrôleur. Vous allez simplement afficher de nouveau les données sur la page (à l'aide d'une mise à jour de page AJAX agréable). Dans un scénario réel, vous pouvez par exemple envoyer la commande par e-mail, appeler un service Web, ajouter les articles à un panier d'achat, et ainsi de suite.

1. Accédez à votre page Visualforce en utilisant son URL, par exemple `https://na1.salesforce.com/apex/Boutique`.
2. Cliquez sur l'onglet Page Editor.
3. Modifiez votre page en ajoutant le code ci-dessous après la ligne `</apex:form>`.

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

Vous venez de créer un volet de sortie identifié par `msg`, qui affiche un article à partir du contrôleur appelé "message", qui n'est pas encore créé.

4. Cliquez sur **Enregistrer**. L'éditeur indique que vous n'avez aucune méthode ou propriété appelée « message » et vous invite à en créer une.
5. Cliquez sur **Créer une propriété Apex 'Boutique.message'**.



6. Modifiez maintenant la balise `commandButton` pour inclure un attribut `reRender`.

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

Vous avez modifié la page Visualforce afin d'utiliser une mise à jour AJAX existante. Elle met à jour le volet identifié par "msg" après avoir appelé la méthode `shop()` du contrôleur. Modifiez maintenant la méthode `shop()` afin de mettre à jour la propriété du message affichée dans le volet avec une liste d'éléments sélectionnés.

1. Cliquez sur l'onglet Contrôleur en regard de l'onglet Page Editor.
2. Recherchez la méthode `shop()` (lignes 4 à 6) et remplacez-la avec le code suivant :

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



Remarque: Notez que ce code utilise simplement la variable `products`. Visualforce s'assure automatiquement que les données modifiées dans l'interface utilisateur sont reflétées par les données de la variable `products`.

3. Cliquez sur l'icône **Enregistrer**.
4. Testez maintenant le panier d'achat. Ajoutez une quantité à un article, puis cliquez sur **Acheter**. Notez le champ de texte qui s'affiche sous la liste des articles, indiquant la quantité de produits achetés.

Store Front

Product	Price	#Items
Wee Jet	1.0	<input style="width: 50px;" type="text" value="3.0"/>

You bought: Wee Jet (3)

En savoir plus

Ce simple ajout présente de nombreuses fonctionnalités puissantes.

- Comme vous l'avez vu dans cette étape, Visualforce a automatiquement reflété les modifications apportées aux données du formulaire dans la variable `products`. Cette fonctionnalité est très puissante et permet d'élaborer rapidement des formulaires et d'autres pages de saisie complexes.
- Lorsque vous cliquez sur le bouton **Acheter**, le volet est mis à jour sans actualisation complète de l'écran. L'effet AJAX, qui nécessite généralement une manipulation JavaScript complexe, a été réalisé avec un simple attribut `reRender`.

Voir aussi :

[Didacticiel n° 9 : Création d'une boutique](#)

[Étape 3 : Création de la boutique](#)

[Résumé](#)

Résumé

Apex est un langage puissant de programmation orientée objet, qui comprend de nombreuses fonctionnalités habituellement disponibles dans des langages similaires tels que Java. Dans ce didacticiel, vous avez créé une classe Apex et utilisé quelques fonctionnalités du langage, telles que les grappes, l'itération et l'interrogation de la base de données. Si vous avez effectué l'étape bonus, vous avez modifié la page Visualforce pour utiliser les données renvoyées au contrôleur afin d'afficher un message en utilisant une mise à jour AJAX, sans actualisation de la page entière.

Vous pouvez ensuite créer un portail authentifié pour permettre aux personnes de se connecter à votre site, de créer un nom d'utilisateur et un mot de passe, et d'acheter des articles. Pour cela, vous devez créer un portail, que nous ne présentons pas dans ce manuel. Nous avons toutefois inclus des étapes détaillées pour présenter la création d'un portail fonctionnant avec la sous-application que vous avez créée. Visitez la page developer.force.com/workbook pour consulter le didacticiel consacré au portail et d'autres compléments à ce manuel.

Vous pouvez également distribuer votre sous-application en l'intégrant à un package à publier ou en la déployant dans une organisation de production. Pour plus d'informations, reportez-vous à [Distribution de votre application](#) à la page 91.

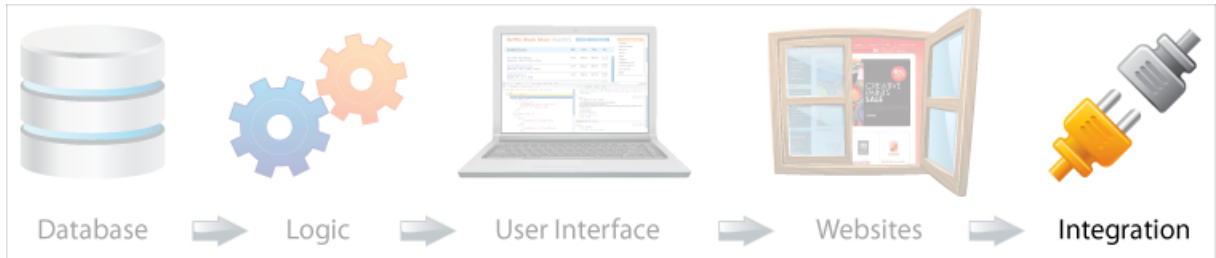
Voir aussi :

[Didacticiel n° 9 : Création d'une boutique](#)

[Étape 4 : Bonus - Mise à jour de la page avec AJAX](#)

Didacticiel n° 10 : Création d'une sous-application de bureau avec Adobe Flash Builder for Force.com.

Niveau : Avancé ; Durée : 30 minutes



Adobe Flash Builder for Force.com est un ensemble d'outils qui permet de développer des sous-applications Force.com Stratus, applications de bureau fonctionnant sous l'environnement Adobe Integrated Runtime (AIR®), qui exploitent la logique et les capacités de base de données Force.com. Les sous-applications Force.com Stratus peuvent fonctionner *en ligne* et *hors ligne*. Elles sont par conséquent idéales pour les utilisateurs qui ne disposent pas toujours d'une connexion Internet, mais qui doivent accéder aux données de vos sous-applications Force.com.

Ce didacticiel montre comment utiliser Adobe Flash Builder for Force.com afin de créer une sous-application Force.com Stratus de base appelée Suivi inventaire, qui fournit un accès hors ligne aux données des articles. La sous-application doit permettre aux responsables d'effectuer un suivi d'inventaire sur des Tablet PC lorsqu'ils se déplacent dans un entrepôt équipé d'une connexion Wi-Fi irrégulière. Elle inclura les fonctionnalités de sous-application Force.com Stratus standard ci-dessous.

- Un écran de connexion qui demande aux utilisateurs leurs identifiants et établit une connexion avec Force.com lorsqu'ils sont en ligne.
- Une interface de résolution des conflits permettant aux utilisateurs de gérer aisément les conflits qui se produisent lorsque les valeurs de la sous-application Force.com Stratus violent les règles de validation ou sont en conflit avec les modifications effectuées par d'autres utilisateurs.
- Une barre d'état qui informe les utilisateurs de la connexion à Internet de la sous-application Force.com Stratus et indique le nombre de conflits de données et d'erreurs non résolus.

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

Pour créer la sous-application, nous allons utiliser les technologies Adobe Flash, telles que MXML et les composants ActionScript, avec les classes Force.com Stratus générées par Adobe Flash Builder for Force.com. Il n'est cependant pas nécessaire de connaître MXML et ActionScript pour créer la sous-application Inventaire Stratus. Ce didacticiel contient le code dont vous avez besoin pour créer et utiliser votre première sous-application Force.com Stratus.



ATTENTION: Adobe Flash Builder for Force.com est actuellement disponible uniquement dans les organisations Developer Edition en tant qu'aperçu pour développeur. Les fonctionnalités peuvent changer profondément avant sa disponibilité complète. Dans l'aperçu du développeur, n'utilisez pas Adobe Flash Builder for Force.com pour créer des sous-applications destinées à des environnements de production.

Voir aussi :

- [Étape 1 : Configuration des informations personnelles Salesforce.com](#)
- [Étape 2 : Création d'une Configuration de porte-documents hors-ligne](#)
- [Étape 3 : Génération de votre WDSL Entreprise et téléchargement d'un certificat client](#)
- [Étape 4 : Installation et lancement de Adobe Flash Builder for Force.com](#)
- [Étape 5 : Création d'un projet Force.com Stratus](#)
- [Étape 6 : Définition des attributs du composant Stratus Application](#)
- [Étape 7 : Création d'un composant Window Suivi Inventaire](#)
- [Étape 8 : Test de la sous-application Suivi Inventaire](#)
- [Résumé](#)

Étape 1 : Configuration des informations personnelles Salesforce.com

Les sous-applications Force.com Stratus exploitent la technologie de Connect Offline dans Force.com. Pour développer ou utiliser une sous-application Force.com Stratus, vous devez sélectionner l'option `Utilisateur hors ligne` dans vos informations personnelles Salesforce.com.

1. Dans Salesforce.com, accédez à **Configuration** ► **Gestion des utilisateurs** ► **Utilisateurs**.

2. Cliquez sur **Modifier** en regard de votre nom.
3. Activez la case `Utilisateur hors ligne`.

4. Cliquez sur **Enregistrer**.

Voir aussi :

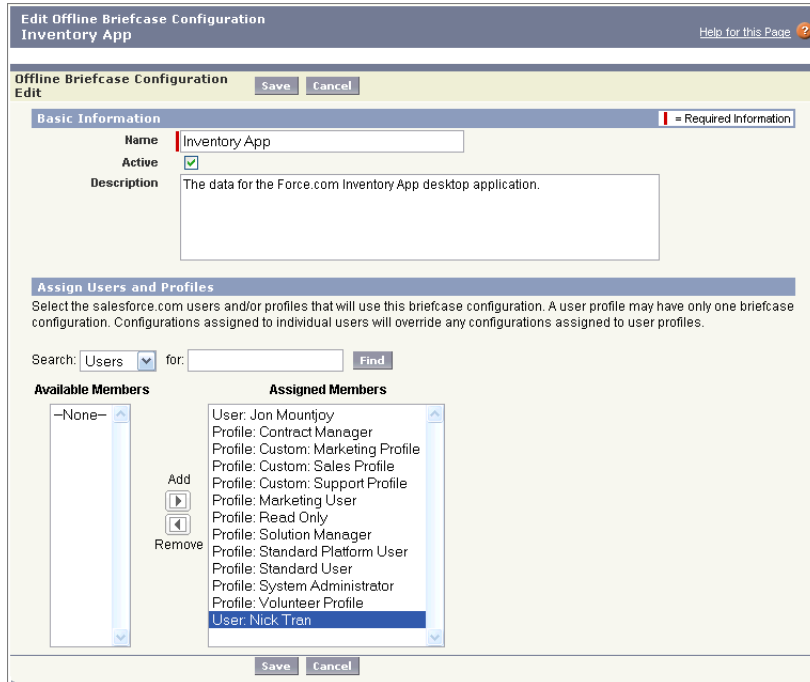
[Didacticiel n° 10 : Création d'une sous-application de bureau avec Adobe Flash Builder for Force.com.](#)

[Étape 2 : Création d'une Configuration de porte-documents hors-ligne](#)

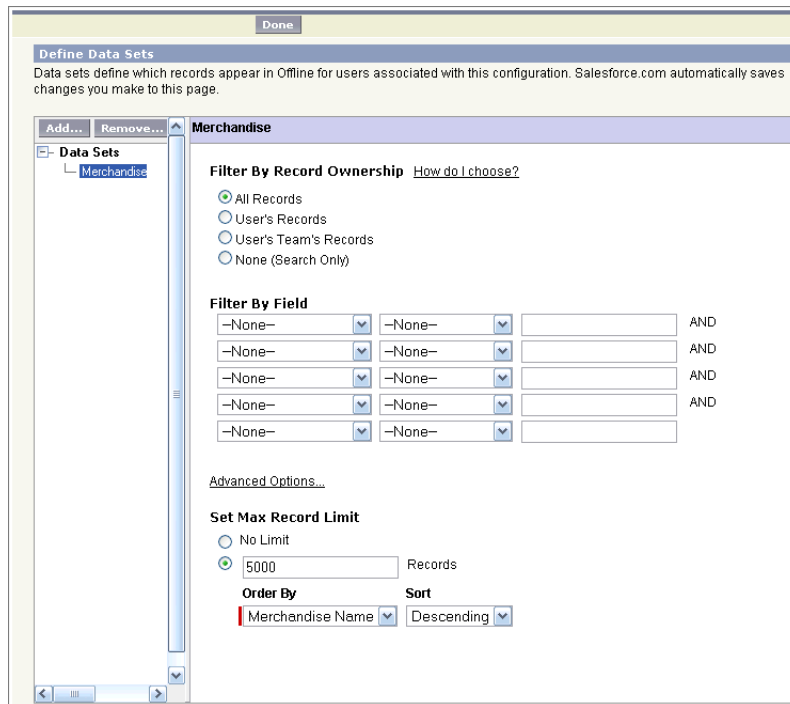
Étape 2 : Création d'une Configuration de porte-documents hors-ligne

Votre sous-application de bureau Force.com accède uniquement aux données Force.com spécifiées dans la configuration de porte-documents hors-ligne qui vous a été attribuée dans Salesforce.com. La sous-application Suivi Inventaire doit accéder aux données des articles. Par conséquent, vous devez créer une configuration de porte-documents hors-ligne qui inclut l'objet Article.

1. Dans Salesforce.com, cliquez sur **Configuration** ► **Administration d'ordinateur de bureau** ► **Configurations des porte-documents hors-ligne**.
2. Cliquez sur **Nouvelle configuration du porte-documents hors connexion**.
3. Saisissez un nom pour votre configuration du porte-documents hors ligne, tel que `Données de la sous-application Suivi Inventaire`.
4. Sélectionnez la case `Actif`.
5. Sélectionnez votre nom dans la liste `Membres disponibles`, puis cliquez sur **Ajouter** pour le déplacer vers la liste `Membres attribués`.



6. Cliquez sur **Enregistrer**. La page de détails Configuration des porte-documents hors-ligne s'affiche.
7. Cliquez sur **Modifier** dans la liste associée Ensembles de données.
8. Cliquez sur **Ajouter**.
9. Sélectionnez **Article**, puis cliquez sur **OK**.
10. Sélectionnez **Nom** de l'article dans la liste de sélection Organiser par.



11. Cliquez sur **Terminé**.

En savoir plus

Les configurations de porte-documents hors-ligne sont des ensembles de paramètres qui déterminent les enregistrements disponibles dans les sous-applications Force.com Stratus et dans Connect Offline Force.com. Connect Offline est une application cliente qui permet d'accéder à un sous-ensemble de données Force.com à l'aide de la même interface de navigateur que le système en ligne, mais sans connexion Internet. Connect Offline permet d'afficher, de modifier, de créer et de supprimer des comptes, des activités, des contacts, des opportunités, des pistes et des enregistrements d'objet personnalisé (groupes de relations incluses). Vous pouvez également ajouter et mettre à jour des produits et des planifications dans les opportunités.

Vous pouvez créer plusieurs configurations de porte-documents et associer chacune d'elles à un utilisateur et à un profil spécifique afin de répondre simultanément aux besoins de différents types d'utilisateurs hors ligne. Par exemple, une configuration peut inclure des pistes et des opportunités pour les commerciaux, et une autre inclure des comptes et les opportunités associées pour les responsables de compte.

Voir aussi :

[Didacticiel n° 10 : Création d'une sous-application de bureau avec Adobe Flash Builder for Force.com.](#)

[Étape 1 : Configuration des informations personnelles Salesforce.com](#)

[Étape 3 : Génération de votre WSDL Entreprise et téléchargement d'un certificat client](#)

Étape 3 : Génération de votre WSDL Entreprise et téléchargement d'un certificat client

Les services Web sont des formes d'intégration courantes et Force.com fournit un ensemble puissant de fonctionnalités qui prennent en charge les appels de services Web entrants et sortants. Force.com génère également automatiquement un point de terminaison de service Web qui permet d'accéder aux données de votre application. *WSDL Entreprise* fait référence au langage de définition de services Web, qui décrit vos objets Force.com. Il correspond généralement à tout élément requis lors de l'intégration à un autre système, car il décrit tous les objets et le service Web.

Adobe Flash Builder for Force.com permet d'importer votre WSDL Entreprise dans des projets de développement. Lors de l'importation de votre WSDL, Adobe Flash Builder for Force.com génère des classes ActionScript pour chaque objet de votre WSDL Entreprise, ce qui permet de référencer des objets Force.com dans votre code ActionScript.

Nous allons importer plus tard notre WSDL Entreprise dans notre projet de développement afin de créer une classe pour l'objet Article. Pour le moment, enregistrez une copie locale de votre WSDL Entreprise actuel. Téléchargez également un certificat client Salesforce.com, qui autorise les applications clientes à accéder à Force.com.

1. Dans Salesforce.com, cliquez sur **Configuration** ► **Développer** ► **API**.
2. Cliquez sur **Générer WSDL Entreprise**.
3. Dans votre navigateur, cliquez sur **Fichier** ► **Enregistrer sous**.
4. Nommez le fichier `entreprise.wsdl`, puis cliquez sur **Enregistrer**.
5. Cliquez sur **Générer Certificat client** pour générer et enregistrer le certificat client pour votre ordinateur.

Voir aussi :

[Didacticiel n° 10 : Création d'une sous-application de bureau avec Adobe Flash Builder for Force.com.](#)

[Étape 2 : Création d'une Configuration de porte-documents hors-ligne](#)

[Étape 4 : Installation et lancement de Adobe Flash Builder for Force.com](#)

Étape 4 : Installation et lancement de Adobe Flash Builder for Force.com

Adobe Flash Builder for Force.com s'installe en tant que version autonome d'Adobe Flash Builder, l'environnement de développement intégré d'Adobe (IDE) basé sur la plate-forme Eclipse.

1. Téléchargez le fichier zip Adobe Flash Builder for Force.com depuis le site developer.force.com/flashbuilder.
2. Décompressez le contenu du fichier zip.
3. Double-cliquez sur le programme d'installation.
4. Une fois l'installation terminée double-cliquez sur le raccourci Adobe Flash Builder for Force.com de votre Bureau.

Voir aussi :

[Didacticiel n° 10 : Création d'une sous-application de bureau avec Adobe Flash Builder for Force.com.](#)

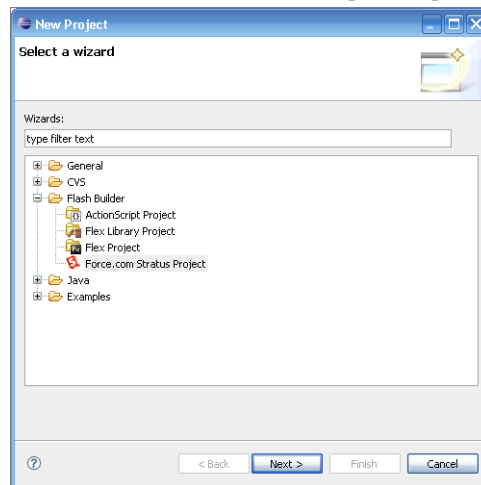
[Étape 3 : Génération de votre WSDL Entreprise et téléchargement d'un certificat client](#)

[Étape 5 : Création d'un projet Force.com Stratus](#)

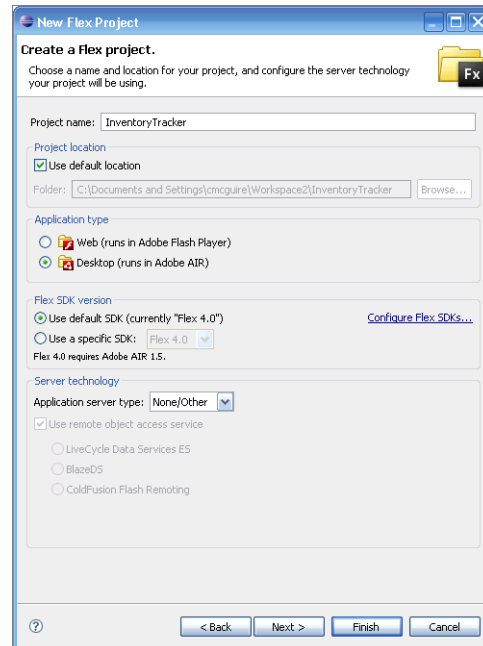
Étape 5 : Création d'un projet Force.com Stratus

Une fois Adobe Flash Builder for Force.com installé, un nouveau type de projet, appelé Force.com Stratus Project est disponible dans Flash Builder. Les projets Force.com Stratus permettent d'importer votre WSDL Entreprise et de générer automatiquement des classes ActionScript pour vos objets Salesforce.com. Flash Builder fournit également des conseils de code et des avertissements du compilateur basés sur votre WSDL.

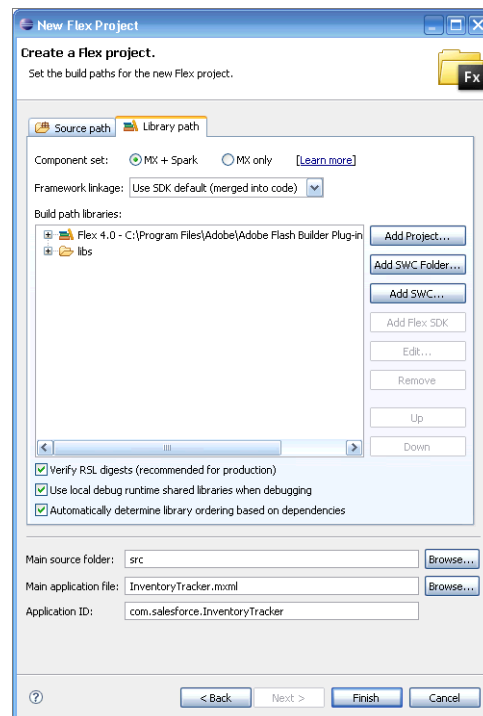
1. Dans Flash Builder, sélectionnez **Fichier** ► **Nouveau projet**. L'assistant Nouveau projet s'ouvre.
2. Sélectionnez Force.com Stratus Project dans le dossier Flash Builder, puis cliquez sur **Suivant**.



3. Saisissez `Suivi inventaire` dans Nom du projet, puis sélectionnez `Application de bureau` (exécution dans Adobe AIR).
4. Cliquez sur **Suivant**.

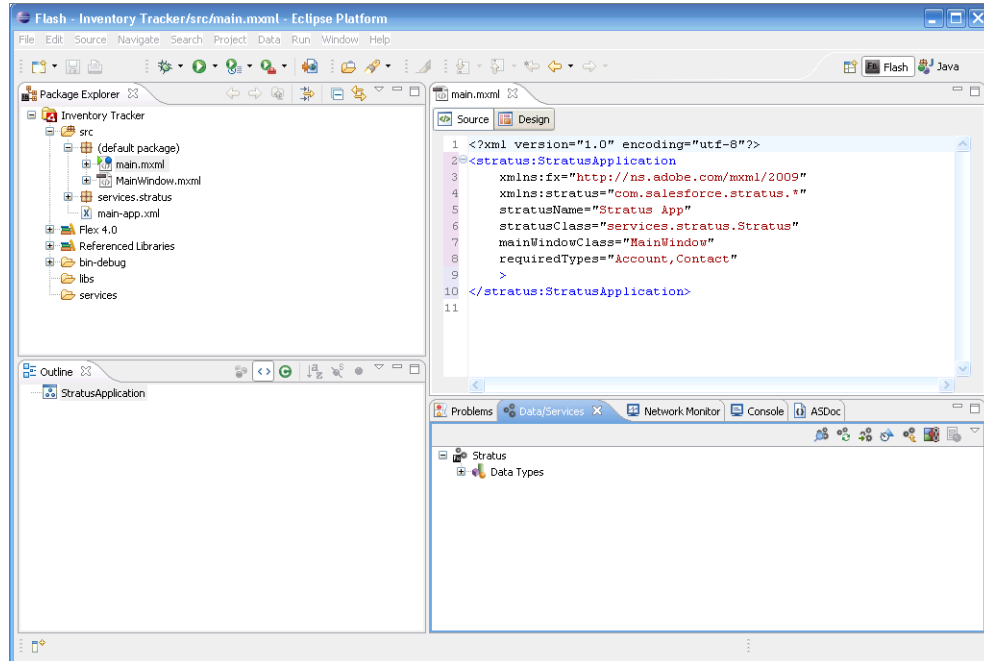


5. Acceptez les paramètres par défaut pour le dossier de sortie, puis cliquez sur **Suivant**.
6. Saisissez `com.salesforce.SuiviInventaire` dans le champ ID d'application. L'environnement d'exécution AIR et le système d'exploitation identifient l'application. Pour garantir un ID d'application unique, utilisez la notation inverse du nom de domaine.
7. Cliquez sur **Terminer** pour continuer.



8. Sélectionnez votre WSDL Entreprise, puis cliquez sur **Suivant**.

Une fois le projet Force.com Stratus créé, l'arborescence suivante s'affiche dans l'Explorateur de pack de Flash Builder.



Voir aussi :

[Didacticiel n° 10 : Création d'une sous-application de bureau avec Adobe Flash Builder for Force.com.](#)

[Étape 4 : Installation et lancement de Adobe Flash Builder for Force.com](#)

[Étape 6 : Définition des attributs du composant Stratus Application](#)

Étape 6 : Définition des attributs du composant Stratus Application

Chaque sous-application Force.com Stratus est contenue dans un composant Stratus Application. Le composant Stratus Application connecte la sous-application Force.com Stratus à Force.com, fournit la connexion Force.com et la fonctionnalité d'authentification, et gère la synchronisation initiale des données entre Force.com et la sous-application Force.com Stratus.

Les projets Force.com Stratus ont un fichier `main.mxml` par défaut qui contient un modèle Stratus Application que vous pouvez modifier pour votre sous-application Suivi Inventaire. Modifiez les attributs de l'élément `stratus:StratusApplication` dans ce fichier.

1. Ouvrez le fichier `main.mxml` situé dans **src** ► **(package par défaut)** s'il n'est pas déjà ouvert.
2. Définissez l'attribut `stratusName` sur `Suivi Inventaire`. Il contrôle le nom qui s'affiche dans la barre de titre de l'application.
3. Définissez l'attribut `requiredTypes` sur `Article__c`. Cet attribut détermine les objets dans votre sous-application Force.com que la sous-application Force.com Stratus référence.
4. Votre code final doit se présenter comme suit :

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

5. Enregistrez le fichier.

Voir aussi :

[Didacticiel n° 10 : Création d'une sous-application de bureau avec Adobe Flash Builder for Force.com.](#)

[Étape 5 : Création d'un projet Force.com Stratus](#)

[Étape 7 : Création d'un composant Window Suivi Inventaire](#)

Étape 7 : Création d'un composant Window Suivi Inventaire

L'interface utilisateur de Suivi Inventaire est composée de deux parties. La partie inférieure contient une liste d'enregistrements d'articles. Lorsque vous sélectionnez un enregistrement dans cette liste, ses détails s'affichent dans la partie supérieure. Vous pouvez cliquer sur les détails d'un enregistrement pour les modifier. Un bouton **Enregistrer**, sous les détails de l'enregistrement, permet de valider ces modifications dans Force.com.

La logique de la fonctionnalité principale de la sous-application Suivi Inventaire repose sur ActionScript et MXML qui utilisent deux ensembles de composants Adobe, Spark et Halo, ainsi que des composants Stratus. Les composants Stratus sont générés par Adobe Flash Builder for Force.com. Ils simplifient le processus d'accès à la logique et aux données Force.com dans votre code.

Vous allez saisir la logique de l'application dans le composant par défaut Window qui est généré par Adobe Flash Builder for Force.com lorsque vous créez un projet Force.com Stratus. Le composant par défaut Window se trouve dans le fichier `MainWindow.mxml`. Le code par défaut est :

```
<?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="Emplacement de votre contenu"/> </s:VGroup> <fx:Script> <![CDATA[ import
com.salesforce.stratus.StratusMessage; private function onWindowComplete():void {

    }new StratusMessage(StratusMessage.STATUS_INFO, "Application initialisée").showAsStatus();
// emplacement de votre code    }    ]]> </fx:Script> <stratus:StatusBar/> </s:Window>
```

Le code par défaut comprend :

- Un composant Spark Window qui spécifie les espaces de noms, la couleur de l'arrière-plan et les dimensions de la sous-application Force.com Stratus. Le composant comprend également un attribut `windowComplete` qui détermine la fonction exécutée lorsqu'un utilisateur se connecte. L'attribut `showStatusBar`, qui détermine si l'application restitue la barre d'état Flash Builder standard, est défini sur `false` pour permettre à l'application d'afficher la barre d'état Stratus à la place.
- Un composant Spark Layout qui organise les éléments de mise en page selon leurs paramètres individuels.
- Un composant Spark VerticalLayout qui organise les éléments de mise en page dans une séquence verticale, de haut en bas, avec des espaces en option entre les éléments et un remplissage en option autour de la séquence d'éléments.
- Un composant Spark VGroup qui crée un regroupement vertical des éléments de l'interface utilisateur.
- Un composant Spark Label qui crée une étiquette pour le regroupement vertical.

- Un composant XML Script dans lequel vous pouvez saisir l'ActionScript. Ce composant contient l'ActionScript par défaut qui initialise les données lorsque la sous-application Force.com Stratus s'ouvre.
- Un composant Stratus StatusBar qui restitue une barre d'état en bas de la sous-application Force.com Stratus.

Modifiez le code par défaut afin de mettre en oeuvre la logique de la sous-application Suivi Inventaire comme suit :

1. Supprimez le composant Label.

```
<s:Label text="Emplacement de votre contenu"/>
```

2. Dans le composant VGroup, ajoutez un composant Stratus FieldContainer. Les composants Stratus FieldContainer regroupent les données et simplifient le processus de manipulation de plusieurs champs dans une seule opération. Le composant FieldContainer que vous créez ici restitue également les détails de l'enregistrement sélectionné en haut de l'écran.

Définissez la largeur du composant FieldContainer sur 100% et saisissez l'ID _editFieldContainer. Vous allez référencer l'ID FieldContainer plus loin dans le code.

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

3. Dans le composant FieldContainer, ajoutez un composant LabelAndField pour chaque champ de l'objet Article que vous souhaitez afficher, puis référez chaque champ par son nom d'API Force.com interne. Un composant LabelAndField est un composant Stratus qui restitue la valeur du champ Salesforce.com avec son étiquette. Les champs que vous créez avec le composant FieldAndLabel fonctionnent automatiquement dans les sous-applications Force.com Stratus de la même façon que dans l'interface utilisateur Salesforce.com. Par exemple, les champs de date affichent un calendrier sur un simple clic. Les champs respectent également les dépendances de champ et fournissent automatiquement l'édition en ligne, les détails de survol, les notifications d'erreur ainsi que le texte d'aide de l'icône Info.

```
<stratus:LabelAndField field="Article__c.Name" /> <stratus:LabelAndField  
field="Article__c.Description__c" /> <stratus:LabelAndField field="Article__c.Prix__c"  
/> <stratus:LabelAndField field="Article__c.Inventory_total__c" />
```

4. Une fois les composants LabelAndField ajoutés, assurez-vous d'insérer l'élément fermant FieldContainer si vous ne l'avez pas déjà fait.

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

5. Ajoutez ensuite un bouton Spark standard, étiquetez-le **Enregistrer** et configurez-le pour exécuter une fonction appelée onEditSaveClick. que vous allez écrire plus loin.

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

6. Sous le bouton **Enregistrer**, dans l'interface utilisateur Suivi Inventaire, vous souhaitez restituer une liste d'enregistrements d'articles. Vous utilisez pour cela un composant ActionScript DataGrid. Saisissez le code comme suit :

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

Notez que l'attribut `dataProvider` référence une variable que vous allez créer plus loin et l'attribut `itemClick` référence une fonction que vous allez également créer plus loin.

- Le composant final dans l'élément `VGroup` est le composant `Stratus StatusBar`, qui fait partie du code par défaut. Laissez ce composant inchangé et assurez-vous d'ajouter l'élément fermant `VGroup`.

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

- L'interface utilisateur de la sous-application Suivi Inventaire est en place. Vous devez toutefois ajouter l'ActionScript qui récupère les données Salesforce.com et les enregistre dans la sous-application Force.com `Stratus`. Commencez l'ActionScript en important les classes dont vous avez besoin pour l'ActionScript dans la section `<![CDATA[` de l'élément `<fx:Script>`. Le code se présente comme suit :

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

Les classes sont les suivantes :

- `ArrayCollection` : classe wrapper ActionScript qui expose une grappe en tant que collection accessible et qui peut être manipulée.
- `IManaged` : interface ActionScript qui fournit le contrat pour un objet géré.
- `StratusWrapper` : composant Stratus qui enveloppe la fonctionnalité Data Management Service (DMS) d'Adobe, un composant clé de la plate-forme Adobe AIR qui fournit un modèle de gestion de la synchronisation des données client-serveur.
- `StratusResponder` : composant Stratus qui s'assure que les fonctions `responder NULL` ne génèrent aucun problème interne.
- `Article__c` : classe ActionScript de l'objet Article générée par Adobe Flash Builder for Force.com.
- `StratusMessage` : composant Stratus qui fournit des messages d'erreur et d'information contextuels et standard, notamment lors de conflit de données.

- Créez une variable qui contient les données d'articles. Le code utilisera cette variable pour afficher les données dans la liste.

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

- Créez une variable que votre code peut utiliser pour accéder à la fonctionnalité DMS.

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

- La fonction `onWindowComplete()` dans le code par défaut interroge la base de données lors du chargement de l'application. Modifiez cette fonction pour renvoyer de façon spécifique les données des articles. La fonction complète se présente comme suit :

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

Vous avez référencé une fonction appelée attribut `WindowComplete` dans l'élément `Window` au début de cette étape du didacticiel.

12. Créez une fonction nommée `onEditSaveClick` qui copie en mémoire, depuis l'interface utilisateur, les modifications qu'un utilisateur effectue dans les données.

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

Cette fonctionnalité est déclenchée par un utilisateur en cliquant sur le bouton **Enregistrer**. Cette fonction exécute également une simple validation des données saisies par l'utilisateur. Si les données sont valides, la fonction appelle la fonction `commitToDB`.

13. Créez la fonction `commitToDB`, qui valide les données dans la mémoire de la base de données et les efface du composant `FieldContainer`.

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

14. Créez une fonction qui restitue les données `FieldContainer` dans la sous-application Force.com Stratus. Référez le composant `FieldContainer` avec l'ID que vous avez défini plus haut (`_editFieldContainer`).

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

15. Enregistrez le fichier.

Votre code final doit se présenter comme suit :

```
<?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="Article__c.Name" /> <stratus:LabelAndField
field="Article__c.Description__c" /> <stratus:LabelAndField field="Article__c.Prix__c" />
<stratus:LabelAndField field="Article__c.Inventaire_total__c" /> </stratus:FieldContainer>
    <s:Button label="Enregistrer" click="onEditSaveClick()" /> <mx:DataGrid id="_dataGrid"
width="100%" height="100%" dataProvider="{_gridDataProvider}"
itemClick="onDataGridItemClick()"> <mx:columns> <mx:DataGridColumn dataField="Nom"/>
<mx:DataGridColumn dataField="Description__c" headerText="Description"/> <mx:DataGridColumn
dataField="Prix__c" headerText="Prix"/> <mx:DataGridColumn dataField="Inventaire_total__c"
headerText="Inventaire total"/> </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.Article__c;

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

    public function onWindowComplete():void { new StratusMessage(StratusMessage.STATUS_INFO,
    "Application initialisée").showAsStatus(); _stratusWrapper = StratusWrapper.getInstance();
    _stratusWrapper.query("select * from Article__c", new StratusResponder(
    function(rows:ArrayCollection):void { _gridDataProvider = rows; }, null )); }
]]></fx:Script>
</s:Window>
```

```

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:Article__c =
_dataGrid.selectedItem as Article__c; _editFieldContainer.fieldCollection.render(merchandise);
} ]]> </fx:Script> </s:Window>

```

Voir aussi :

[Didacticiel n° 10 : Création d'une sous-application de bureau avec Adobe Flash Builder for Force.com.](#)

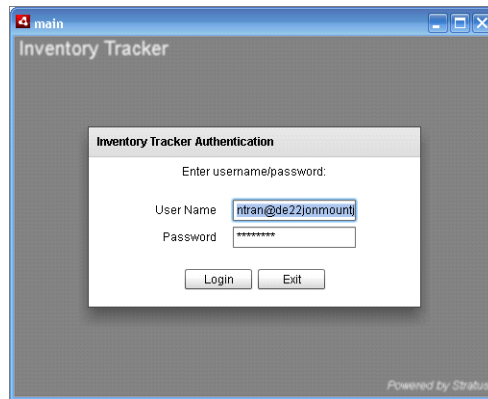
[Étape 6 : Définition des attributs du composant Stratus Application](#)

[Étape 8 : Test de la sous-application Suivi Inventaire](#)

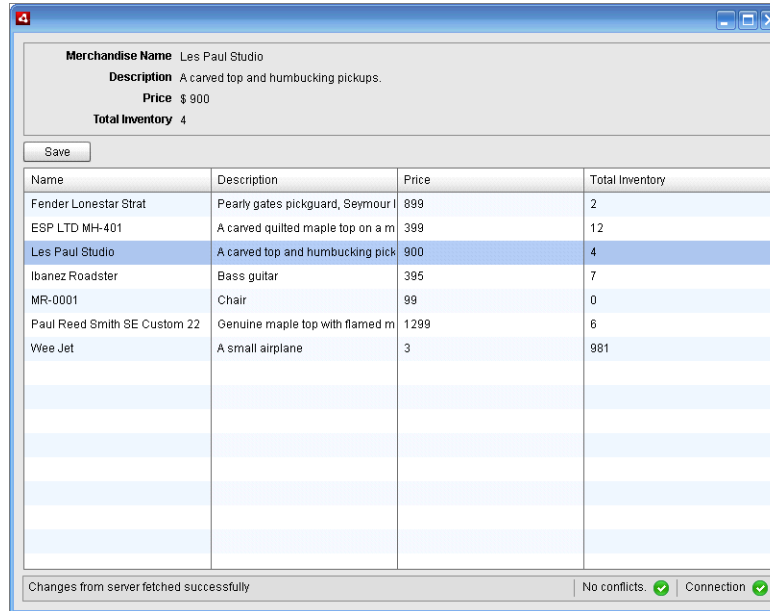
Étape 8 : Test de la sous-application Suivi Inventaire

La sous-application Suivi Inventaire est prête et peut être testée.

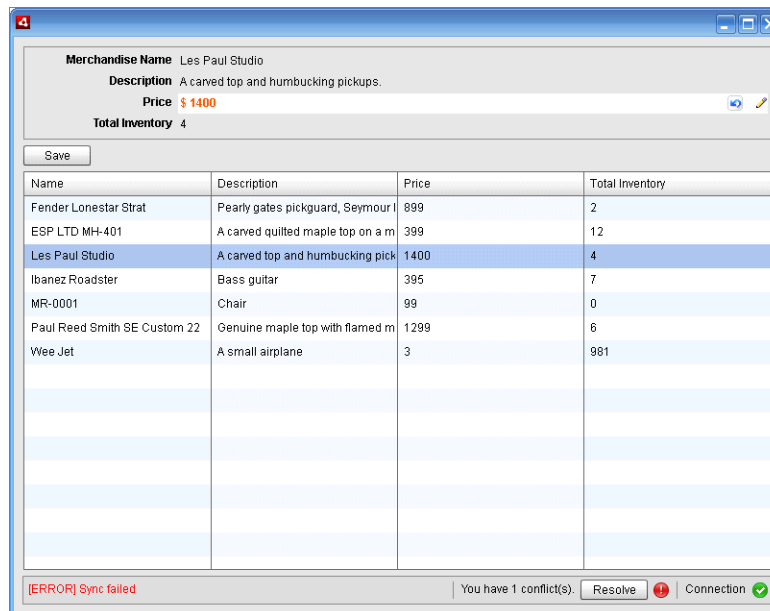
1. Assurez-vous de disposer d'une connexion Internet.
2. Dans Flash Builder, cliquez avec le bouton droit sur le projet Suivi Inventaire, puis sélectionnez **Exécuter en tant que ► Application de bureau**. L'écran de connexion de la sous-application Suivi Inventaire s'affiche.



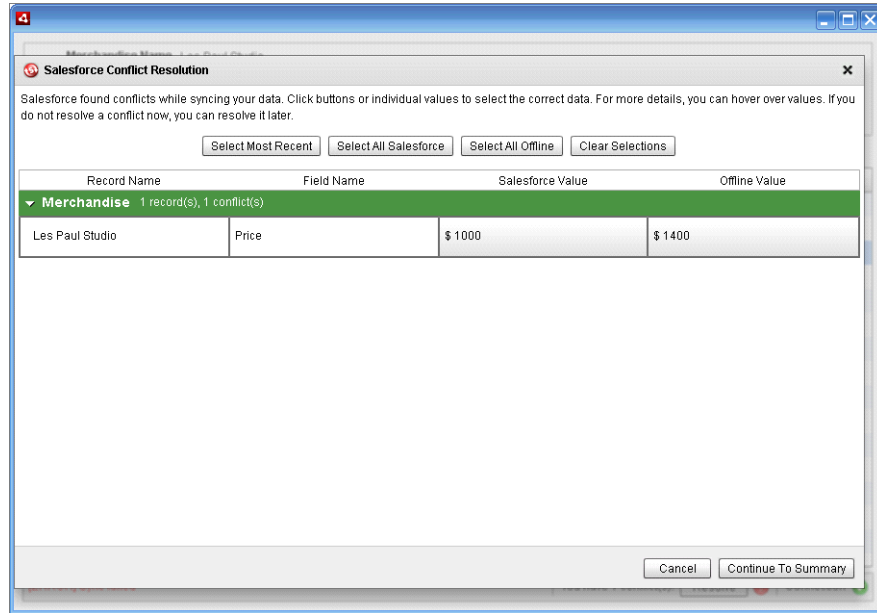
3. Dans l'écran de connexion, saisissez le nom d'utilisateur et le mot de passe de la sous-application Force.com, puis cliquez sur **Connexion**. L'interface utilisateur Suivi Inventaire s'affiche.



- Sélectionnez un enregistrement dans la liste et modifiez ses données, puis cliquez sur **Enregistrer**.
- Connectez-vous à la sous-application Force.com et vérifiez que vos modifications sont affichées.
- Pour tester les capacités de résolution des conflits de données, modifiez un champ dans votre sous-application Force.com, cliquez sur **Enregistrer**, puis modifiez le même champ dans la sous-application Suivi Inventaire en utilisant une valeur différente. Lorsque vous cliquez sur **Enregistrer** dans la sous-application Suivi Inventaire, notez l'alerte de la barre d'état qui signale un conflit.



- Cliquez sur **Résoudre** dans la barre d'état pour lancer l'interface de résolution des conflits.



L'interface de résolution des conflits permet de visualiser les valeurs dans Salesforce.com et dans la sous-application Suivi Inventaire. Sélectionnez la valeur que vous souhaitez conserver, puis cliquez sur **Continuer vers le récapitulatif**.

Voir aussi :

[Didacticiel n° 10 : Création d'une sous-application de bureau avec Adobe Flash Builder for Force.com.](#)

[Étape 7 : Création d'un composant Window Suivi Inventaire](#)

[Résumé](#)

Résumé

Dans ce didacticiel, vous avez créé une sous-application Force.com Stratus qui permet aux utilisateurs de visualiser et de modifier les données Force.com *avec* et *sans* connexion Internet. Ce didacticiel présente les notions fondamentales de Adobe Flash Builder for Force.com avec peu de code. Vous pouvez évidemment utiliser Adobe Flash Builder for Force.com pour créer des sous-applications Force.com Stratus beaucoup plus puissantes afin d'optimiser l'interaction entre les utilisateurs et vos sous-applications Force.com.

Voir aussi :

[Didacticiel n° 10 : Création d'une sous-application de bureau avec Adobe Flash Builder for Force.com.](#)

[Étape 8 : Test de la sous-application Suivi Inventaire](#)

Annexe

Création et connexion à un environnement test sandbox

Si vous utilisez la version Free Edition, vous devez effectuer l'ensemble du développement dans un environnement test sandbox. Pour créer cet environnement :

1. Connectez-vous à votre organisation Free Edition.
2. Cliquez sur **Configuration** ► **Gestion des données** ► **Sandbox**.
3. Cliquez sur **Nouveau**.
4. Saisissez un **Nom**, par exemple **Dév.**
5. Cliquez sur **Lancer la copie**.

La création d'une copie d'organisation de production prend un peu de temps. Pour suivre la progression, actualisez votre navigateur et observez le champ **Statut**. Si vous ne souhaitez pas attendre, poursuivez vos activités et nous vous enverrons un e-mail une fois la copie terminée.

Lorsque l'environnement test sandbox est disponible, connectez-vous comme suit :

1. Si vous êtes passé à une autre page durant la copie de l'environnement sandbox, cliquez sur **Configuration** ► **Gestion des données** ► **Sandbox**.
2. En regard de l'environnement sandbox, cliquez sur **Connexion**.
3. Votre navigateur ouvre une page de connexion. Saisissez votre mot de passe, puis cliquez sur **Connexion**.

En savoir plus

Une fois connecté, notez que le champ **Nom d'utilisateur** est différent de votre nom d'utilisateur habituel. Force.com ajoute automatiquement votre nom sandbox pour vous permettre de vous connecter à plusieurs environnements test sandbox. Notez également que l'URL dans la barre d'adresse de votre navigateur a changé. Chaque environnement test sandbox possède sa propre instance, qui est différente de celle utilisée par votre organisation habituelle.

Création d'un projet dans l'IDE Force.com

Pour créer un projet dans l'IDE Force.com :

1. Sélectionnez **File** ► **New** ► **Force.com Project**.
2. Saisissez les identifiants de votre organisation.
 - a. Saisissez le nom d'un projet. Une organisation peut avoir plusieurs noms de projet. Il est préférable d'identifier chaque projet avec un nom explicite. Appelez notre projet `Manuel utilisation`.
 - b. Saisissez le nom d'utilisateur permettant de se connecter en tant qu'administrateur. Il se présente sous la forme d'une adresse de messagerie.
 - c. Saisissez votre mot de passe.
 - d. Si vous utilisez un environnement test sandbox pour vos activités de développement, sélectionnez **Sandbox** dans la liste déroulante du champ **Environnement**.
 - e. Cliquez sur **Suivant**.



Remarque: Si vous obtenez une erreur `LOGIN_MUST_USE_SECURITY_TOKEN`, cela signifie que votre adresse IP actuelle n'est pas reconnue dans une plage réseau de confiance. Vous devez réinitialiser votre jeton de sécurité. Dans l'interface utilisateur de Salesforce.com, cliquez sur **Configuration** ► **Mes informations personnelles** ► **Réinitialiser mon jeton de sécurité**, puis suivez les instructions affichées.

3. Dans la page Project Content, cliquez sur **Finish** pour récupérer les composants Apex et Visualforce.

Distribution de votre application

Votre application étant terminée, nous allons vous présenter différentes méthodes de distribution. Si vous développez cette application pour l'installer dans plusieurs organisations, vous pouvez l'intégrer à un *package*. Si vous la développez pour l'installer dans une seule organisation, vous pouvez la *déployer*.

À propos des packages

Les packages correspondent à des porte-documents qui contiennent vos composants, votre code ou vos applications. Ils permettent de publier une application dans plusieurs organisations. La plate-forme Force.com offre deux types de packages, *non gérés* et *gérés* :

- Les packages non gérés permettent de distribuer des copies de composants intégrés, qui comprennent l'ensemble du code source. Ces packages sont généralement gratuits, sans option de mise à niveau. Une fois le package non géré installé dans une autre organisation, en tant qu'éditeur, vous n'avez aucun contrôle sur l'utilisation des composants intégrés.
- Les packages gérés permettent de distribuer des copies de composants intégrés en gardant le contrôle de la propriété intellectuelle et des licences. Vous pouvez également distribuer aisément des mises à niveau pour les clients qui ont installé d'anciennes versions de l'application. Les packages gérés sont plus efficaces pour distribuer des applications dans Force.com AppExchange.

Pour plus d'informations sur les packages, reportez-vous à « Présentation des packages » dans l'aide en ligne de Force.com.

À propos du déploiement

Pour déployer votre code dans une organisation de production dont vous possédez les identifiants de connexion, vous disposez des deux options suivantes :

- Si vous maîtrisez l'interface à ligne de commande, utilisez l'Outil de migration Force.com pour le déploiement. Pour plus d'informations, reportez-vous au guide wiki.developerforce.com/index.php/Migration_Tool_Guide.

- Si vous préférez une interface graphique, utilisez l'IDE Force.com. Pour plus d'informations, reportez-vous à la page wiki.developerforce.com/index.php/Force.com_IDE.