Security Workbook

Version 1, Winter '17

@salesforcedocs

Last updated: September 15, 2016

CONTENTS

ABOUT THE SECURITY WORKBOOK

Security is of vital importance to an organization. It ensures that users have secure authentication and rules-based policies for determining what they can do and which records they can access. It also means that the principle of least privilege is applied so that no administrators or users possess more privileges than needed to perform their jobs.

The Salesforce platform provides a least-privilege, user-centric security model that includes tools to control user's login and access controls. Because applications are written once and run everywhere, users' authentication and access controls are consistent regardless of whether they access the platform from a Web application or smartphone.

Here's a preview of how it's done on the Salesforce platform:

1. Create users – For each person who needs access, create a user.
2. Create functional access controls using profiles and permission sets – Identify the different types of users you need for your application based on the different functions each type needs to access. Use a base level profile for each type of user such that each profile has only the permissions required for that type of user to perform these functions. Then create permission sets to handle exceptions—situations in which a user may need a few more permissions.
3. Create sharing models – For each object, set the organization-wide default record sharing model to determine whether the records that each user owns are public or private.
4. Share private records – Use roles, groups, record sharing rules, and other means to share private records with other users.
5. Use authentication standards like OAuth to log in safely from a variety of platforms – Enable users to easily, and securely log in from any site or device using common authentication standards.
6. Audit everything – Whether it's login history, changes to system configurations, or changes to data for compliance and troubleshooting purposes.

The *Security Workbook* walks you through the policies, rules, and grouping mechanisms that ensure your users can log in securely and do everything that they should and nothing that they shouldn't. Each of the tutorials builds on the previous one to advance the application's development and simultaneously showcase a particular feature of the platform. It might sound like a lot, but it's all quite easy—as you'll soon see.

Workbook Version

This workbook is updated for Summer '14. You should be able to successfully complete all steps using the Summer '14 version of Salesforce.

You can find the latest version of this Workbook at https://developer.salesforce.com/page/Force.com_workbook.

Before You Begin

This workbook is intended for organization administrators and developers new to the Salesforce platform. Although not a requirement, it is helpful to have completed at least the first two tutorials in the Force.com Workbook (`https://developer.salesforce.com/page/Force.com_workbook`) prior to using this workbook. These steps create the environment for the tutorials in this workbook and save you time while working through them.

To save even more time, you can install a package into your new Developer Edition (DE) org that deploys custom objects used in this workbook.

If you don't have a DE org account, sign up for one at http://sforce.co/1uGJt1L.

While you are logged into your DE org:

1. Using the browser window that is logged into your DE org, open a new browser tab and use it to load http://bit.ly/ApexWorkbookPackage1_4.
2. Click **Continue** > **Next** > **Next** > **Install**.
3. Click **View Components**, then take a quick look at the components you just deployed into your org, including three custom objects (Merchandise, Invoice Statement, and Line Item).
4. After you're done, you can close this second browser tab and return to the original tab.

TUTORIAL 1: CREATE USERS

Every new organization is preconfigured with a super-user administrator account that you use to manage everything in the organization, including profiles, permission sets, and users. Throughout this workbook, you'll log in and act as this administrator unless the step specifically tells you to log in as another user.

You can create other user accounts declaratively or programmatically. For example, to get started, you might create other administrative or developer users. But when you build your app, the app can leverage Web services or REST APIs to let users create their own user accounts from a "Sign Up" page.

In this tutorial, you'll create users declaratively and programmatically.

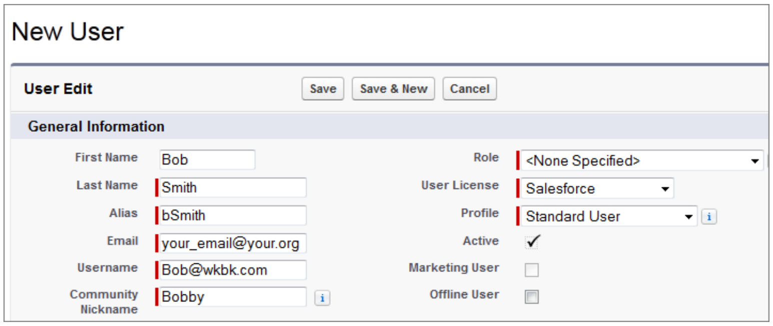
Step 1: Create Users Declaratively through Setup

In this step, you create a user who reports to the user you first created when signing up for the organization. Use this configuration to ensure that when this new user creates an invoice meeting certain conditions, it's routed to the manager.

If you've already completed the Force.com Workbook, this user may already exist and you can skip to the next step on page 4.

1. From Setup, enter `Users` in the `Quick Find` box, then select **Users**.
2. On the All Users page, click **New User**.
3. Enter the following information:
   a. In **First Name**, enter `Bob`.
   b. In **Last Name**, enter `Smith`.
   c. In **Alias**, enter `bSmith`.
   d. In **Email**, enter your own email address, so that you will receive the approval requests routed to Bob Smith.
   e. The **Username** defaults to your email address, but you'll need to create a unique username for Bob, in the form of an imaginary email address.

      Write down Bob's username (his imaginary email address) because you'll log in as him shortly.



First Name	Bob	Role	<None Specified>
Last Name	Smith	User License	Salesforce
Alias	bSmith	Profile	Standard User
Email	your_email@your.org	Active	✓
Username	Bob@wkbk.com	Marketing User	☐
Community Nickname	Bobby	Offline User	☐

   f. In **Manager**, select the user you created when you signed up for your organization. You can use the magnifying glass to search for the user.
   g. In **User License**, select Salesforce.

h. In **Profile**, select `Standard User`.

4. Click **Save**.

   You will receive an email confirming the creation of the new user. You still need to configure authorizations, so don't log in as Bob Smith yet or you'll have to immediately log back in as the administrator.

You can also use SAML to create users just-in-time. For more information, see About Just-in-Time Provisioning for SAML in the Salesforce online help.

Step 2: Create Users Programmatically With the REST API

In this tutorial use the Workbench tool to query for existing users and use the information to create a new user with the REST API.

1. Log into Workbench and run a query to find Bob Smith, the user you created in the previous step.

   a. Type or paste the following URL into your browser: `https://developer.salesforce.com/page/Workbench`.

   b. Leave the default Workbench settings, accept the terms of service, and click **Login with Salesforce**.

   c. Check that the **Logged in as** user in the top right hand corner of the screen is the administrator of your Developer Edition organization. If it isn't, click **(Not you?)** and log in as the administrator of the Developer Edition organization.

   d. Click **Allow** on the "requesting permission" screen.

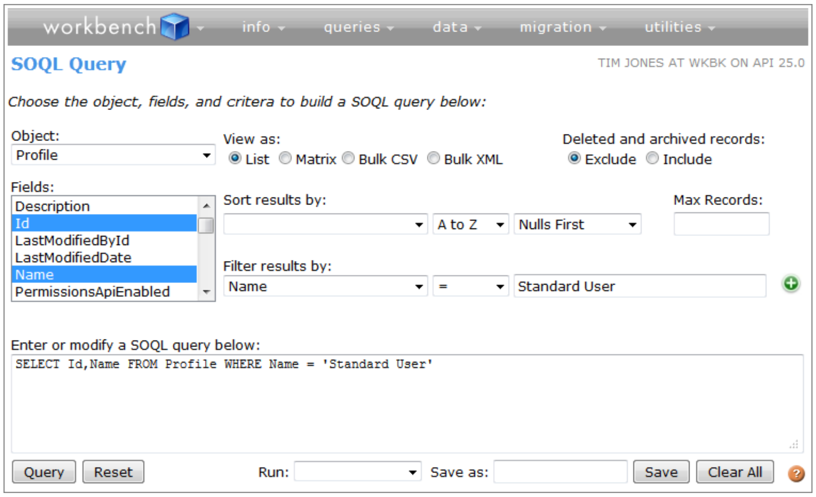
   e. In the workbench menu, select **queries** > **SOQL Query**.

   f. Choose *`Profile`* in **Object**.

   g. Select *`Id`* and *`Name`* in the **Fields** selection box.

      You can select more than one field by holding down the CTRL key and clicking the field names.

   h. Filter results by *`Name = Standard User`*.



SOQL Query

TIM JONES AT WKBK ON API 25.0

Choose the object, fields, and critera to build a SOQL query below:

Object: Profile

View as: ● List ○ Matrix ○ Bulk CSV ○ Bulk XML

Deleted and archived records: ● Exclude ○ Include

Fields: Description, Id, LastModifiedById, LastModifiedDate, Name, PermissionsApiEnabled

Sort results by: A to Z, Nulls First

Max Records:

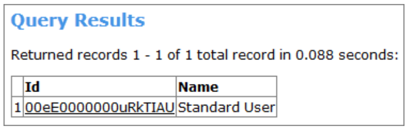
Filter results by: Name = Standard User

Enter or modify a SOQL query below:

```
SELECT Id,Name FROM Profile WHERE Name = 'Standard User'
```

Run: Save as:

   i. Click **Query**.



Query Results

Returned records 1 - 1 of 1 total record in 0.088 seconds:

	Id	Name
1	00eE0000000uRkTIAU	Standard User

j. Copy the Id of the Standard User Profile to an ASCII text editor such as Notepad.

   The Id will be used later when creating the new user.

2. Now use the REST API in Workbench to retrieve Bob Smith’s information.

   a. In the workbench menu, select **utilities** > **REST Explorer**.

   b. Click **Execute** next to `/services/data`.

   c. Click the most recent release.

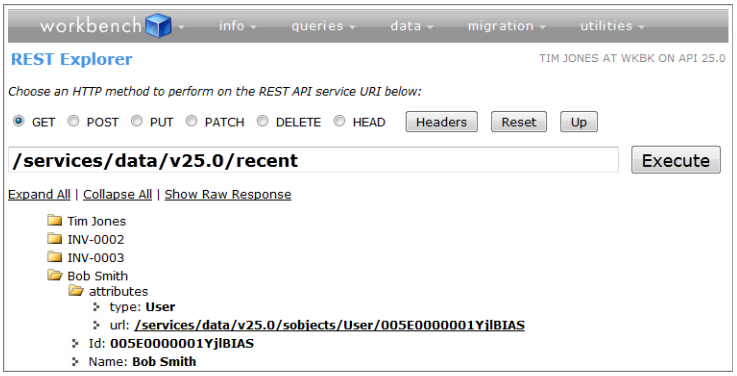
   d. Click **url: /services/data/v*{version#}***.

   e. Click **recent: /services/data/v*{version#}*/recent**.

   f. Click **Bob Smith**.

   g. Click **attributes**.

   h. Click **url: /services/data/v*{version#}*/sobjects/User/*Bob Smith’s user Id***.



REST Explorer

TIM JONES AT WKBK ON API 25.0

Choose an HTTP method to perform on the REST API service URI below:

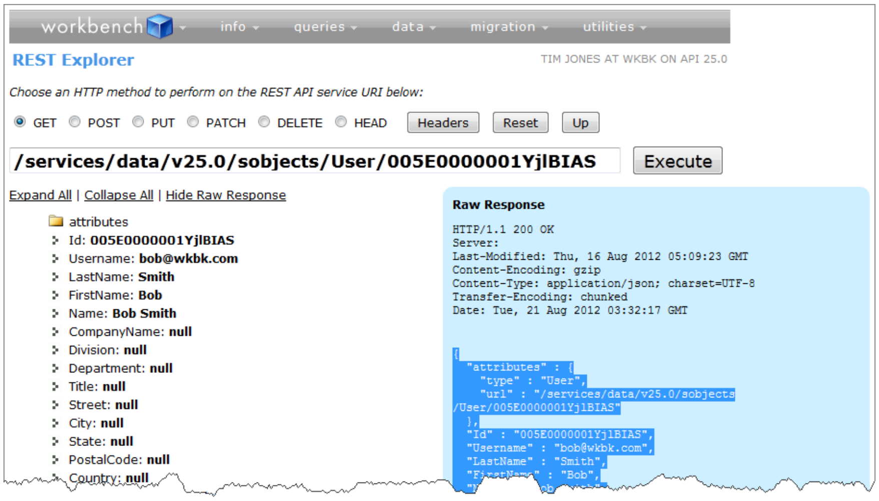
● GET ○ POST ○ PUT ○ PATCH ○ DELETE ○ HEAD

/services/data/v25.0/recent

```
Tim Jones
INV-0002
INV-0003
Bob Smith
    attributes
        type: User
        url: /services/data/v25.0/sobjects/User/005E0000001YjlBIAS
    Id: 005E0000001YjlBIAS
    Name: Bob Smith
```

   i. Click **Show Raw Response**.

   j. Copy everything in **Raw Response** between the curly brackets ({}) and paste it into an ASCII text editor such as Notepad.



Choose an HTTP method to perform on the REST API service URI below:

● GET ○ POST ○ PUT ○ PATCH ○ DELETE ○ HEAD

/services/data/v25.0/sobjects/User/005E0000001YjlBIAS

- attributes
- Id: **005E0000001YjlBIAS**
- Username: **bob@wkbk.com**
- LastName: **Smith**
- FirstName: **Bob**
- Name: **Bob Smith**
- CompanyName: **null**
- Division: **null**
- Department: **null**
- Title: **null**
- Street: **null**
- City: **null**
- State: **null**
- PostalCode: **null**
- Country: **null**

Raw Response

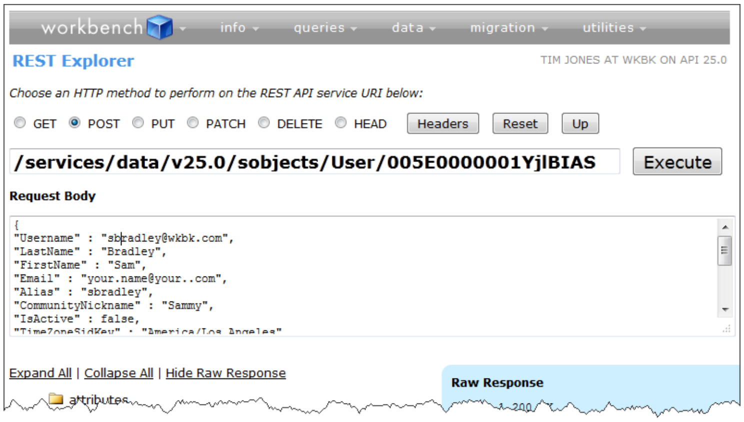
```
HTTP/1.1 200 OK
Server:
Last-Modified: Thu, 16 Aug 2012 05:09:23 GMT
Content-Encoding: gzip
Content-Type: application/json; charset=UTF-8
Transfer-Encoding: chunked
Date: Tue, 21 Aug 2012 03:32:17 GMT


{
  "attributes" : {
    "type" : "User",
    "url" : "/services/data/v25.0/sobjects
/User/005E0000001YjlBIAS"
  },
  "Id" : "005E0000001YjlBIAS",
  "Username" : "bob@wkbk.com",
  "LastName" : "Smith",
  "FirstName" : "Bob",
```

The text between the curly brackets is a JSON object representation of Bob Smith's information, and we'll use it to create a new user.

3. To finish up, create the new user with the REST API using the POST and GET methods.
   a. Select POST.
   b. Change the REST endpoint to `/services/data/v`*`{version#}`*`/sobjects/User/`
   c. You can either modify Bob Smith's information copied earlier, updating it as needed to have it describe Sam Bradley; or you can copy the following text, making changes to anything in italics. Change *yourOrgDomain* and *yourEmailAddress*, and replace *Standard User Profile Obtained from earlier SOQL Query* with the value copied in Step 1.

```
{
"Username" : "sbradley@yourOrgDomain.com",
"LastName" : "Bradley",
"FirstName" : "Sam",
"Email" : "yourEmailAddress",
"Alias" : "sbradley",
"CommunityNickname" : "sbradleyyourorgdomain",
"IsActive" : false,
"TimeZoneSidKey" : "America/Los_Angeles",
"LocaleSidKey" : "en_US",
"EmailEncodingKey" : "ISO-8859-1",
"ProfileId" : "Standard User Profile Obtained from earlier SOQL Query",
"LanguageLocaleKey" : "en_US",
"UserPermissionsMobileUser" : false,
"UserPreferencesDisableAutoSubForFeeds" : false
}
```

   d. Paste the changes into the Request Body.



REST Explorer

TIM JONES AT WKBK ON API 25.0

Choose an HTTP method to perform on the REST API service URI below:

○ GET ● POST ○ PUT ○ PATCH ○ DELETE ○ HEAD

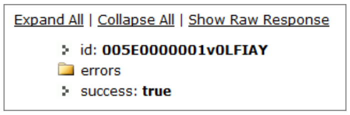
/services/data/v25.0/sobjects/User/005E0000001YjlBIAS

Request Body

```
{
"Username" : "sbradley@wkbk.com",
"LastName" : "Bradley",
"FirstName" : "Sam",
"Email" : "your.name@your..com",
"Alias" : "sbradley",
"CommunityNickname" : "Sammy",
"IsActive" : false,
"TimeZoneSidKey" : "America/Los Angeles"
```

e. Click **Execute**.

f. Verify that an Id is returned and that success is true.



id: **005E0000001v0LFIAY**
errors
success: **true**

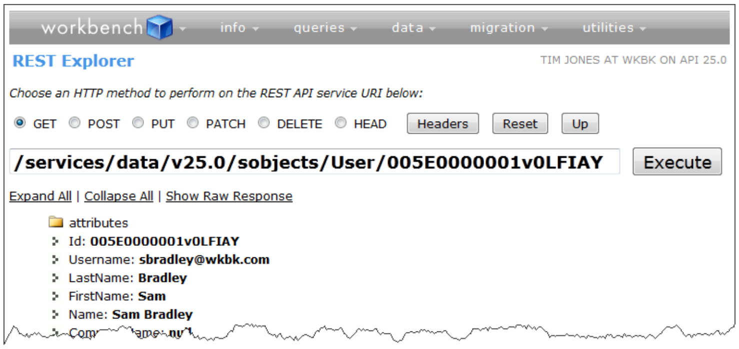
g. Copy the **userid**.

h. Click **GET**.

i. Paste the userId you just created into the REST endpoint `/services/data/v{version#}/sobjects/User/`*`userId you just created`*

j. Click **Execute**.

View the user record details.



REST Explorer

TIM JONES AT WKBK ON API 25.0

Choose an HTTP method to perform on the REST API service URI below:

● GET ○ POST ○ PUT ○ PATCH ○ DELETE ○ HEAD

/services/data/v25.0/sobjects/User/005E0000001v0LFIAY

attributes
Id: **005E0000001v0LFIAY**
Username: **sbradley@wkbk.com**
LastName: **Bradley**
FirstName: **Sam**
Name: **Sam Bradley**

Summary

Creating users is easy, and you can do so using different methods. REST API is becoming the API of choice for managing data like creating users. Workbench is a good tool to simulate REST API requests, but ultimately creating users through the REST API is a great technique for self-provisioning new users from a mobile application using little more than a Web form to capture a couple key user attributes like name and email address.

TUTORIAL 2: CREATE FUNCTIONAL ACCESS CONTROLS

Functional access controls what users can do after they log into an organization. They define which database tables, called *objects*, and which database columns, called *fields*, users can access. They also define other types of access controls that enable users to interact with the organization. In this tutorial, you learn about two types of functional access control containers: profiles and permission sets. You also learn about two specific access controls, object and field permissions.

A profile and a permission set is a collection of functional permissions and settings that control what a user can do within an organization. For example, profiles and permission sets control:

- System-level access
- Different functions within an organization, such as the ability to manage users
- Object-level access, including CRUD permissions for records in each object
- Field-level access, including the ability to read or edit fields for each object
- Access to invoke Apex classes and custom logic
- Access to tabs and apps

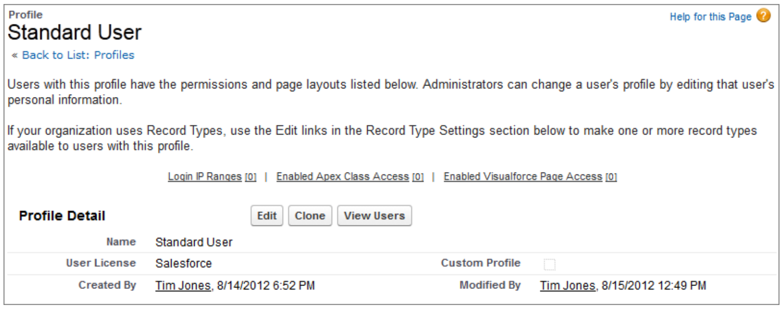
The available permissions you can configure for a profile or permission set depend on the user's license. You can't assign permissions that exceed what a user's license allows. For instance, platform users can't modify all data within an organization or they will violate the terms of their license.

Step 1: Create a Profile

Use profiles to assign the same permission to many users. A profile is a collection of permissions and other settings associated with a user or a group of users. Your organization has a number of standard profiles already defined. If you create an app, the permissions and settings to access the app and associated objects are disabled for most profiles. This security setting ensures that access to the app and its data is explicitly granted to users. You can change object permissions in custom profiles, but not standard profiles.

In this step, you create a custom profile that you can assign to users who need to access the Warehouse app and its custom objects.

1. From Setup, enter `Profiles` in the `Quick Find` box, then select **Profiles**.
2. Click **Standard User**.
3. Select **Clone** next to Profile Detail.



Profile
Help for this Page
« Back to List: Profiles
Login IP Ranges [0] | Enabled Apex Class Access [0] | Enabled Visualforce Page Access [0]
Edit
Clone
View Users

Standard User

Users with this profile have the permissions and page layouts listed below. Administrators can change a user's profile by editing that user's personal information.

If your organization uses Record Types, use the Edit links in the Record Type Settings section below to make one or more record types available to users with this profile.

Profile Detail

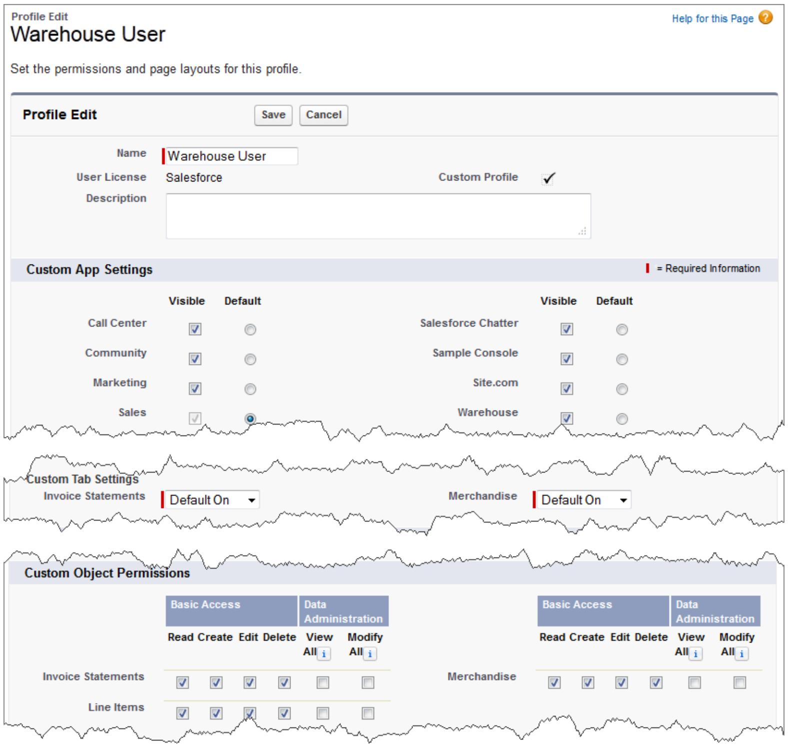
Name	Standard User		
User License	Salesforce	Custom Profile	□
Created By	Tim Jones, 8/14/2012 6:52 PM	Modified By	Tim Jones, 8/15/2012 12:49 PM

4. In the Profile Name field, type `Warehouse User`.
5. Click **Save**.

6. On the detail page for your new profile, click **Edit**.
7. Under Custom App Settings, select **Visible** for all settings.
8. Under Tab Settings, select **Default On** for Merchandise.

   If the Invoice Statements are not also set to **Default On**, change the value to **Default On**.

9. At the bottom of the profile edit page, under Custom Object Permissions, select **Read**, **Create**, **Edit**, and **Delete** for the Invoice Statements, Line Items, and Merchandise objects.



Profile Edit

Warehouse User

Set the permissions and page layouts for this profile.

Name: Warehouse User

User License: Salesforce

Custom Profile: ✓

Description:

Custom App Settings (▌ = Required Information)

	Visible	Default		Visible	Default
Call Center	☑	○	Salesforce Chatter	☑	○
Community	☑	○	Sample Console	☑	○
Marketing	☑	○	Site.com	☑	○
Sales	☑	●	Warehouse	☑	○

Custom Tab Settings

Invoice Statements: Default On

Merchandise: Default On

Custom Object Permissions

	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Invoice Statements	☑	☑	☑	☑	□	□
Line Items	☑	☑	☑	☑	□	□

	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Merchandise	☑	☑	☑	☑	□	□

If you don't see the objects, you can create them by following the steps in the Force.com workbook.

10. Click **Save**.

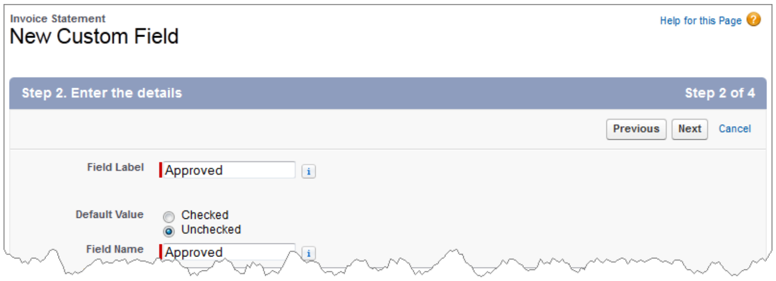
You've just seen how easy it is to create and edit a custom profile. If you need to edit many profiles, you can use enhanced profile list views to create a custom list view of your profiles and then edit the profiles from the list. For more information, see Editing Multiple Profiles with Profile List Views in the Salesforce online help.

Step 2: Manage Field-Level Security

Profiles and permission sets control many different types of granular access. Most of the time you will probably configure object and field permissions. In this case, invoices must be marked as approved based on a field in the invoice, but not everyone needs this access. In each department, an individual is chosen to approve invoices. For this, you'll create a new field and set some base-level field access.

1. From Setup, enter *`Objects`* in the `Quick Find` box, select **Objects**, then select **Invoice Statement**.
2. Under Custom Fields & Relationships, click **New**.
3. On the Choose the field type page, select **Checkbox** and click **Next**.

4. On the Enter the details page, in **Field Label**, type `Approved`.
5. **Field Name**, should be auto-populated with `Approved`.



Invoice Statement

New Custom Field

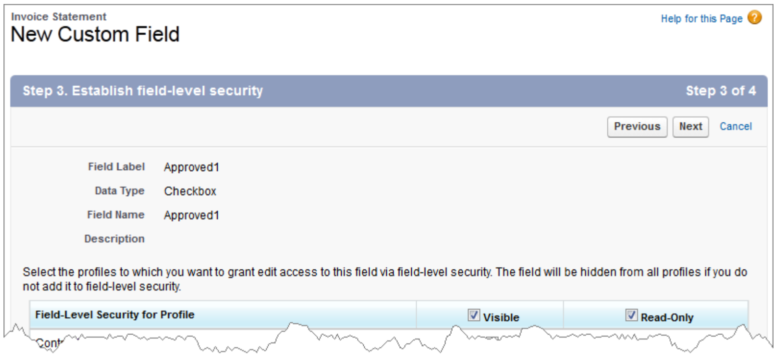
Step 2. Enter the details Step 2 of 4

Field Label: Approved

Default Value: ○ Checked ● Unchecked

Field Name: Approved

6. Click **Next**.
7. On the Establish field-level security page, select **Read-Only** for all profiles. (Look at the top of the Field-Level Security for Profile box for this checkbox.)



Invoice Statement

New Custom Field

Step 3. Establish field-level security Step 3 of 4

Field Label: Approved1

Data Type: Checkbox

Field Name: Approved1

Description:

Select the profiles to which you want to grant edit access to this field via field-level security. The field will be hidden from all profiles if you do not add it to field-level security.

Field-Level Security for Profile	☑ Visible	☑ Read-Only

Cont

This makes the Approved field read-only for all profiles.

8. Click **Next**.
9. On Add to Page Layouts, click **Save**.

Once you set up field-level security, you can use it to let administrators restrict users' access to view and edit specific fields in reports, lists, templates, and many other places. For more information, see Field-Level Security Overview in the Salesforce Help.

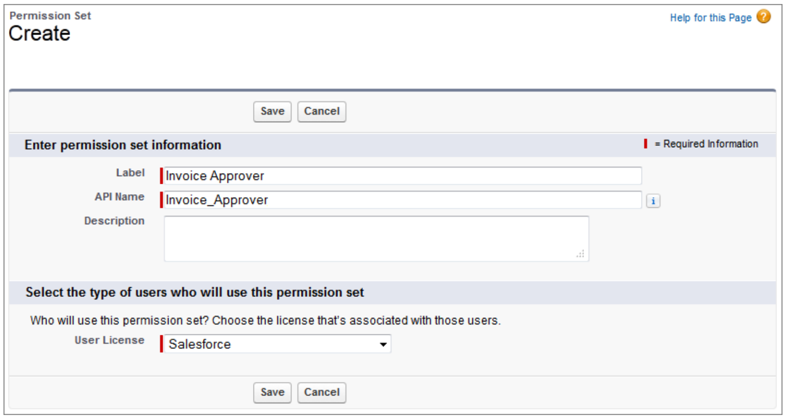
Step 3: Create Permission Sets

Permission sets make Salesforce administrator's and developer's lives easier by assigning permissions to users with more granularity than what profiles offer. While a user must have a profile and can have only one assigned, a user can have zero, one, or many permission sets. This provides flexibility when creating multiple apps, having users in multiple regions, or working within multiple industries. All permissions layer in a positive way between a profile and a permission set, so there are no conflicts. Regardless of whether users have permissions granted on their profiles or any one of their permission sets, they have the access that the permissions grants.

Now that there's a field that allows some users to mark Invoices approved, create a permission set to assign it to the designated users. Field-level security could have been added to the user's profile, but because users from different departments may have different profiles, it's easier to create just one permission set for these different users rather than modify many different profiles.

1. From Setup, enter `Permission Sets` in the `Quick Find` box, then select **Permission Sets**.

2. Click **New**.
3. For the permission set label, type `Invoice Approver`.
4. For the API name, type `Invoice_Approver`.
5. Select Salesforce for the **User License** type.



Permission Set
Create

Help for this Page

Save Cancel

Enter permission set information = Required Information

Label: Invoice Approver
API Name: Invoice_Approver
Description:

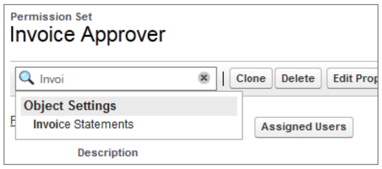
Select the type of users who will use this permission set

Who will use this permission set? Choose the license that's associated with those users.

User License: Salesforce

Save Cancel

6. Click **Save**.
7. In **Find Settings...**, type `Invoice Statements`.



Permission Set
Invoice Approver

Invoi

Clone Delete Edit Prop

Object Settings
Invoice Statements

Assigned Users

Description

8. Click **Edit**.
9. In Field Permissions, select **Edit** to the right of **Approved**.
10. Select **Edit** to the right of **Status**, and then click **Save**.

Permission sets can be used for lots of different scenarios. For more information, see Permission Sets Overview in the Salesforce online help as well as this video about permission sets.

Summary

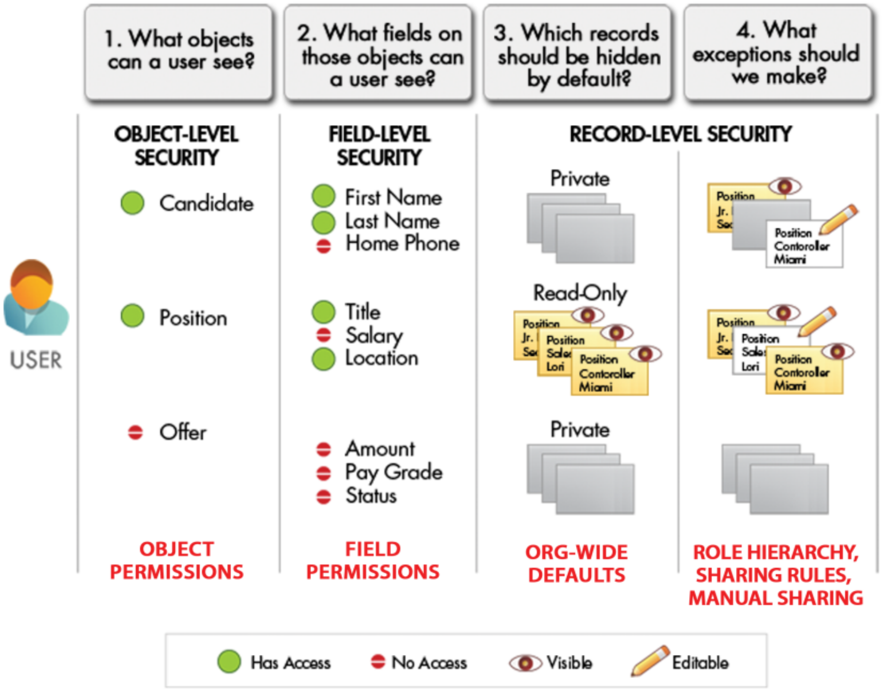
You've seen how to use profiles, permission sets, and field-level security. Profiles and permission sets help you control access to data. Profiles contain many different types of access controls. One example is Field-Level Security. Field-Level Security provides a highly granular and scalable form of control over what columns or fields a user can access. It's possible to manage access to millions of fields for millions of users using this access control.

Unlike profiles, which are limited to one per user, a user can have many permission sets. Permission sets give users access to various tools and functions with fine granularity.

TUTORIAL 3: CREATE RECORD-LEVEL ACCESS CONTROLS

Inherent in the design of the Salesforce platform security model is the notion of record ownership, which helps to simplify the implementation of row-level least privilege data security policies. The creator of a record owns the record after creation and has full access—the owner can read, update, delete, and transfer ownership for the record. Once they transfer the record to another user, they lose access to it unless it's granted back to them through a variety of methods discussed here.

Various data access controls determine whether organization users can access records they don't own. These controls include an object's sharing model, role hierarchies, groups, and sharing rules.



Each object has a sharing model, also known as an organization-wide default (OWD), which governs the default organization-wide access levels users have to each other's records in the object.

- With an object that uses a private sharing model, the record owner can read and manage a record, assuming that the user's profile provides object-level access. Other users can work with records they don't own only by other record sharing means.
- With an object that uses a public read-only sharing model, any user can read all records on the object, assuming that the user's profile provides the Read permission and field-level access for the object.
- With an object that uses a public read/write sharing model, any user can read and write all records in the object, as permitted by the object- and field-level permissions in each user's profile.

An object can have different sharing requirements based on the user context, so it's very important to consider this fact when setting its OWD. A good rule of thumb is to set each object's OWD to be as strict as necessary and then use sharing rules to open up access, as required.

Step 1: Configure Organization-Wide Default Sharing

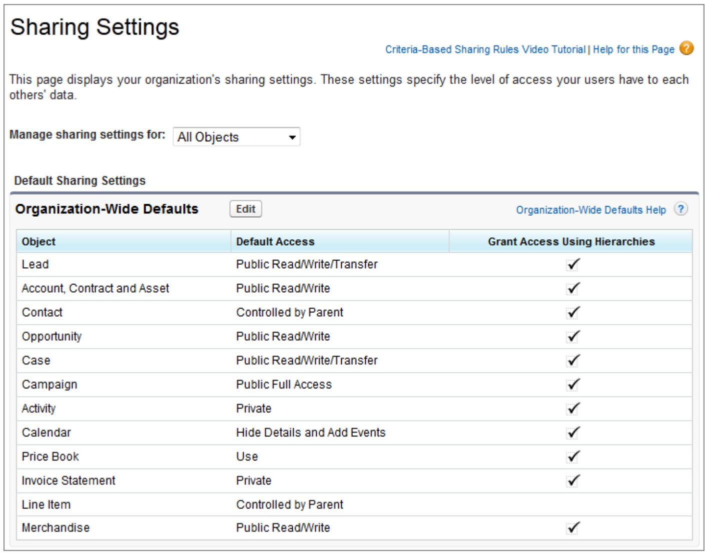
Force.com apps have scalable and granular record level access controls. All users have full record access to records they own. Access can be obtained through

- inheritance up a role hierarchy.
- public groups assigned to a user or nested into one another.
- sharing rules.

Begin by first configuring your organization-wide defaults. Use organization-wide defaults to choose the lowest common level of access for your entire organization.

Because not all invoices should be shared publicly in the company, you set the default access to private.

1. From Setup, enter `Sharing Settings` in the `Quick Find` box, then select **Sharing Settings**.
2. Under Organization Wide-Defaults, click **Edit**.
3. Change the picklist next to the Invoice Statement object to **Private**.
4. Click **Save**.



Sharing Settings

Criteria-Based Sharing Rules Video Tutorial | Help for this Page

This page displays your organization's sharing settings. These settings specify the level of access your users have to each others' data.

Manage sharing settings for: All Objects

Default Sharing Settings

Organization-Wide Defaults

Edit
Organization-Wide Defaults Help

Object	Default Access	Grant Access Using Hierarchies
Lead	Public Read/Write/Transfer	✓
Account, Contract and Asset	Public Read/Write	✓
Contact	Controlled by Parent	✓
Opportunity	Public Read/Write	✓
Case	Public Read/Write/Transfer	✓
Campaign	Public Full Access	✓
Activity	Private	✓
Calendar	Hide Details and Add Events	✓
Price Book	Use	✓
Invoice Statement	Private	✓
Line Item	Controlled by Parent	
Merchandise	Public Read/Write	✓

When you set an object's organization-wide defaults to Private, you limit access to records. You can then open access using a variety of means, such as roles, groups, and sharing rules.

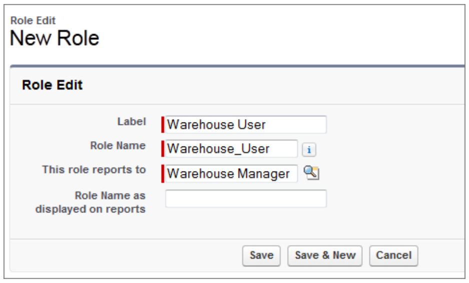
Step 2: Create a Role for Your Organizational Hierarchy

Records can be owned, and ownership has its privileges. An owner has full access to a record including the ability to read, write, transfer, share, and delete it. Users are assigned a role to simplify access. A role is a group of users who all live at the same level of an organizational hierarchy in terms of record or row level access privilege requirements. Access to records can roll up a role hierarchy so that everyone above the owner in the organization has the same level of access as the owner.

1. From Setup, enter `Roles` in the `Quick Find` box, then select **Roles**.
2. If the Sample Role Hierarchy appears, click **Set Up Roles**.

3. Under the root role, which is the name of your company, click **Add Role**.
   a. In the `Label` field, type *`Warehouse Manager`*.
   b. In the `Role Name` field, type *`Warehouse_Manager`*.
   c. In the `This role reports to` field, make sure your company's name or root role show.

4. Click **Save & New**.
   a. In the `Label` field, type *`Warehouse User`*.
   b. In the `Role Name` field, type *`Warehouse_User`*.
   c. In the `This role reports to` field, click the lookup magnifying glass.
   d. Select *`Warehouse Manager`*.



Role Edit
New Role

Role Edit

Label: Warehouse User
Role Name: Warehouse_User
This role reports to: Warehouse Manager
Role Name as displayed on reports:

5. Click **Save**.

You now have a two-role hierarchy. You can use this hierarchy to enable private record sharing, such as Invoice Statements, from Warehouse Users to Warehouse Managers. While you can assign your users here, we will finish a few more steps before assigning users to roles.

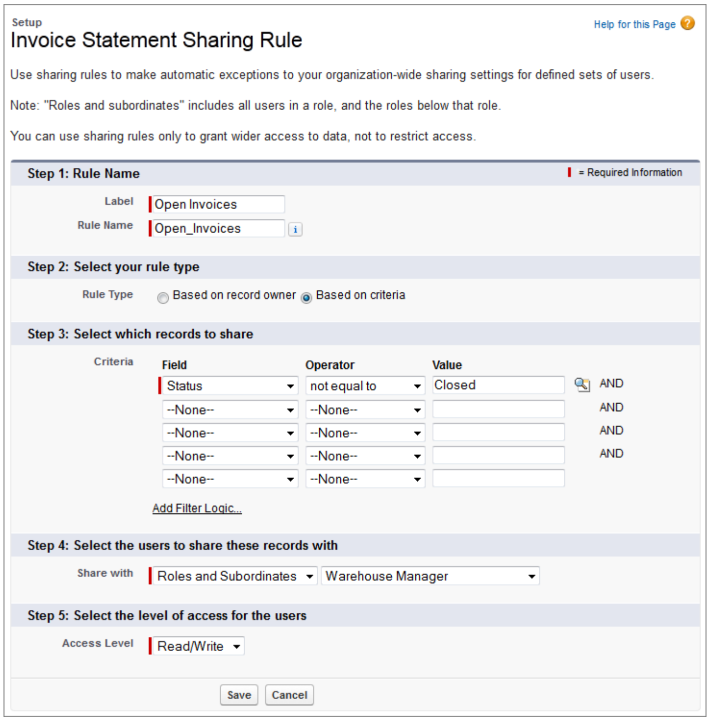
Step 3: Create a Sharing Rule

Sometimes record access isn't determined strictly by a user's position in the role hierarchy. You may want to enable users from one branch of the hierarchy to access another branch's records. Or you may want to create access based on some criteria on the record. These policies determine how you extend access beyond the role hierarchy structure.

In this case, all open invoices should be available to warehouse users; however, when the invoice closes it should no longer be accessible so that no one modifies it after it's closed.

1. From Setup, enter *`Sharing Settings`* in the `Quick Find` box, then select **Sharing Settings**.
2. Under `Invoice Statement Sharing Rules`, click **New**.
3. In the `Label` field, type *`Open Invoices`*.
4. In the `Rule Name` field, type *`Open_Invoices`*.
5. For `Rule Type`, select **Based on Criteria**.
6. In Criteria, select the following values:
   a. For `Field`, choose **Status**.
   b. For `Operator`, choose **Not equal to**.
   c. For the `Value` type, choose **Closed**.

7. For `Share With`, choose `Roles and Subordinates` and select `Warehouse Manager` from the picklist.
8. For `Access Level`, choose `Read/Write`.



Setup

Invoice Statement Sharing Rule

Use sharing rules to make automatic exceptions to your organization-wide sharing settings for defined sets of users.

Note: "Roles and subordinates" includes all users in a role, and the roles below that role.

You can use sharing rules only to grant wider access to data, not to restrict access.

Step 1: Rule Name

= Required Information

Label: Open Invoices

Rule Name: Open_Invoices

Step 2: Select your rule type

Rule Type: ○ Based on record owner ● Based on criteria

Step 3: Select which records to share

Criteria

Field	Operator	Value	
Status	not equal to	Closed	AND
--None--	--None--		AND
--None--	--None--		AND
--None--	--None--		AND
--None--	--None--		

Add Filter Logic...

Step 4: Select the users to share these records with

Share with: Roles and Subordinates, Warehouse Manager

Step 5: Select the level of access for the users

Access Level: Read/Write

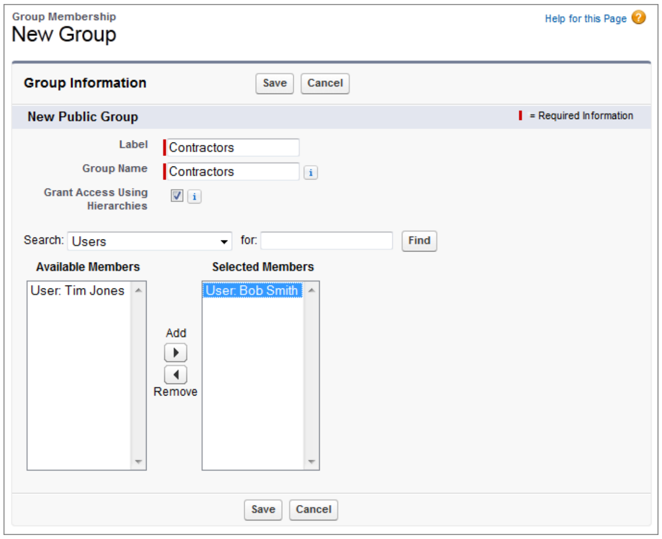
9. Click **Save**.
10. Click **Okay** at the popup.

Sharing rules are a powerful way to set security policies that open access to groups of users that normally wouldn't have access.

Step 4: Create a Public Group

Much like roles, public groups consist of users and are designed to share access outside of your organizational hierarchy. An example is a group of auditors who need access to invoices during tax time. Where roles can contain another role, public groups may either contain users, roles, or other public groups. And where users may have one role, users may be a member of zero, one, or many different groups. This makes public groups similar to permission sets. Public groups give you flexibility when sharing records to more than one logical grouping of users, regardless of whether the group is based on a region, industry, skill, or other segmentation. Public groups are typically used with sharing rules to open up access.

1. From Setup, enter *`Public Groups`* in the `Quick Find` box, then select **Public Groups**.
2. Click **New**.
3. In the `Label` field, type *`Contractors`*.
4. In the `Group Name` field, type *`Contractors`*.
5. Select `Grant Access Using Hierarchies`.
6. In `Search:`, select **Users**.
7. Move **User: Bob Smith** from `Available Members` to `Selected Members`.



Group Membership

New Group

Group Information

New Public Group

Label: Contractors

Group Name: Contractors

Grant Access Using Hierarchies: ☑

Search: Users for:

Available Members: User: Tim Jones

Selected Members: User: Bob Smith

8. Click **Save**.

Sharing rules can share access to roles and they can also share to public groups. Additionally, roles nest within one another, but because public groups may contain either users or other groups, it's possible to nest groups in a simple hierarchical structure to help roll up access from one group of users to another.

Summary

You've learned how to use organization-wide default sharing to limit access, and roles, sharing, and groups to provide access on an as-needed basis. You've also seen how to nest multiple roles and groups for flexibility when specifying access. Use these tools to control record access in your organization.

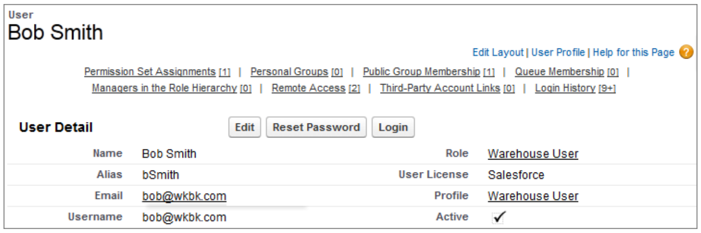
TUTORIAL 4: LOGIN TO TEST AUTHORIZATIONS

The user access levels must be tested to make sure everything functions correctly. By logging in as an end-user, you can test your authorization controls.

Step 1: Edit a User

Now test the access levels of the user you just created. Before starting, though, first make some changes to the user details.

1. From Setup, enter `Users` in the `Quick Find` box, then select **Users**.
2. On the All Users page, click **Smith, Bob**.
3. Click **Edit** on Bob Smith's detail page.
4. Enter the following information:
   - For **Role**, select Warehouse User.
   - For **Profile**, select Warehouse User.


User
Bob Smith
Edit Layout | User Profile | Help for this Page
Permission Set Assignments [1] | Personal Groups [0] | Public Group Membership [1] | Queue Membership [0] |
Managers in the Role Hierarchy [0] | Remote Access [2] | Third-Party Account Links [0] | Login History [9+]
User Detail
Edit
Reset Password
Login

Name	Bob Smith	Role	Warehouse User
Alias	bSmith	User License	Salesforce
Email	bob@wkbk.com	Profile	Warehouse User
Username	bob@wkbk.com	Active	✓

5. Click **Save**.
6. Scroll down to Permission Set Assignments.
7. Click **Edit Assignments**.
8. Move Invoice Approver from **Available Permission Sets** to **Enabled Permission Sets**.
9. Click **Save**.

Step 2: Create a New Invoice Record

Now confirm that you can make Bob Smith the owner of an invoice.

1. Click the **Invoice Statements** tab.
2. Click **New**.
3. Click **Save**.

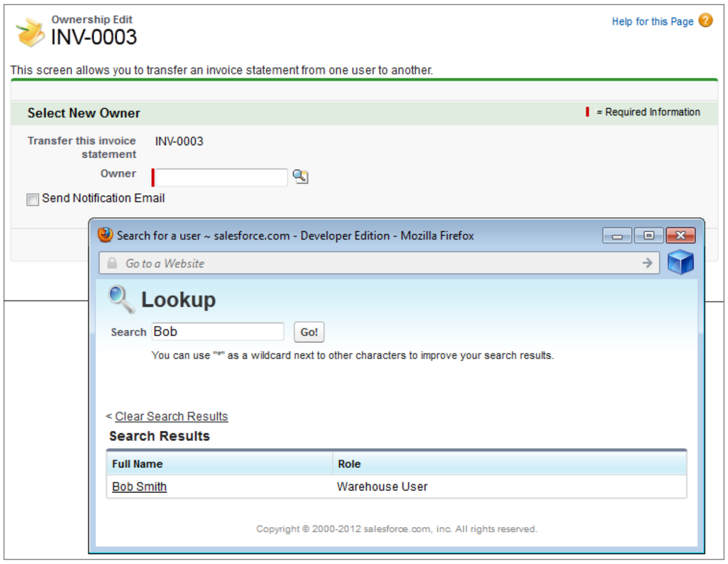
   Note the invoice number.
4. Click the **Invoice Statements** tab.
5. Click **New**.

6. Change the status to **Closed**.
7. Click **Save**.

   Note the invoice number.

8. Change the owner of the record to `Bob Smith` by clicking **Change** at **Owner**.

   You can use the Lookup feature to find Bob Smith.



Ownership Edit

INV-0003

Help for this Page

This screen allows you to transfer an invoice statement from one user to another.

Select New Owner

= Required Information

Transfer this invoice statement: INV-0003

Owner:

□ Send Notification Email

Search for a user ~ salesforce.com - Developer Edition - Mozilla Firefox

Go to a Website

Lookup

Search: Bob

Go!

You can use "*" as a wildcard next to other characters to improve your search results.

< Clear Search Results

Search Results

Full Name	Role
Bob Smith	Warehouse User

Copyright © 2000-2012 salesforce.com, inc. All rights reserved.

Step 3: Log In as the New User

When Bob Smith was created, you received an email with his login information. Now log in as Bob Smith to test his access to invoices.

1. Click *Your Name* > **Logout**.
2. Now log in as Bob Smith.

   If this is your first time logging in as Bob, you have to change the password.

3. From the app menu, click **Sales** > **Warehouse** to select the Warehouse app.
4. Click the **Invoice Statements** tab.

   You can access the tab because of Bob's profile.

5. Click **Go**.
6. Click the next-to-last invoice in the list.
7. Click **Edit**.

   You can edit the invoice because, based on the sharing rule you created, it has an open status. You can edit both the status and the Approved field because of Bob's permission set.

8. Click **Save**.
9. Click **Back to List: Invoice Statements**.
10. Click the last invoice in the list.

11. Click **Edit**.

    You can edit the invoice because, as Bob Smith, you own it.

12. Click `Your Name` > **Logout**.

In this case, you set Bob Smith's email address to your own and reset the password. Bob Smith can also grant login access, which does not involve sharing a password. To learn more, read Granting Login Access in the online help.

Summary

You've practiced adding roles, profiles, and permission sets to a user. After making access changes, it's always a good idea to test the updates you made. In this tutorial you saw how to make sure your access levels are correct for a user.

TUTORIAL 5: SECURITY POLICIES FOR AUTHORIZING USERS

As an administrator, you have a lot of control over how users log into your organization. The next set of tutorials provide ways of configuring security that help conform to security best practices and your corporate security policies.

Step 1: Set Password Policies

Password policies enable you to set controls around passwords. For example, you might want users to have longer passwords that contain a variety of characters rather than shorter ones that might be less secure.

1. Log in as the administrator if you aren't already.
2. From Setup, enter *`Password Policies`* in the `Quick Find` box, then select **Password Policies**.
3. In **User passwords expire in**, select *`90 days`*.
4. In **Enforce password history**, select *`8 passwords remembered`*.
5. In **Minimum password length**, select *`8 characters`*.
6. In **Password complexity requirement**, select *`Must mix alpha and numeric characters`*.
7. In **Password question requirement**, select *`Cannot contain password`*.
8. In **Maximum invalid login attempts**, select *`5`*.
9. In **Message**, type *`Contact your company's administrator for assistance by dialing your phone number`* where *your phone number* is any number.
10. Click **Save**.

Step 2: Set Session Timeout

Sessions can last as long as needed. However, when people step away from their computers, it's wise to force their sessions to timeout after a specified period. This can help to ensure that unauthorized people do not gain access to data they should not.

1. From Setup, enter *`Session Settings`* in the `Quick Find` box, then select **Session Settings**.
2. In **Timeout value**, select *`30 minutes`*.
3. Click **Save**.

Session settings are a good way to ensure that idle users don't expose unauthorized access to your organization's data.

Step 3: Limit Network Access with IP Ranges

A great way to limit access to your organization is to set IP range requirements. By applying IP range restrictions, you can require users to log into corporate-controlled IP ranges, or you can choose by groups of users whether they can log in through a controlled IP range.

1. From Setup, enter *`Profiles`* in the `Quick Find` box, then select **Profiles**.
2. Click **Warehouse User**.
3. Scroll to the bottom of the profile and click New on **Login IP Ranges**.

4. In **Start IP Address**, type the beginning of your network's range for this group of users.
5. In **End IP Address**, type the end of your network's range for this group of users.
6. Click **Save**.

If users try to log in from an unrecognized IP range, they are forced to use an authorization token that allows them to log in from that new IP address.

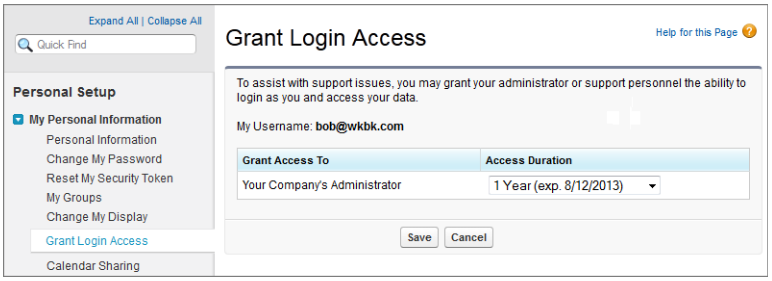
Step 4: Configure Login Access Policies

Login as is a powerful tool used by administrators and support representatives at Salesforce or their partners to help troubleshoot issues. In Summer '15, Salesforce enabled this feature by default. When the "Administrators Can Log in as Any User" setting is enabled, administrators with "Modify All Data" permission and delegated administrators with "View Setup and Configuration" permission can log in as any user. Users don't have to grant access first.

If the "Administrators Can Log in as Any User" setting is not enabled, users must give permission for an administrator to log in as them. Administrators can limit who can grant this permission to others outside your organization.

Here you configure your organization's login access policies. For the purposes of this tutorial, you allow only administrators (not other users) to grant access to Salesforce Support. You don't allow administrators to log in as another user unless the user grants access.

1. From Setup, enter `Login Access Policies` in the `Quick Find` box, then select **Login Access Policies**.
2. In the list of support organizations, click **Available to Administrators Only** for Salesforce Support. If the "Administrators Can Log in as Any User" setting is enabled, disable it.
3. Click **Save**.
4. Log out as the Administrator.
5. Log in as Bob Smith.
6. From your personal settings, enter `Login Access` in the `Quick Find` box, then select the option to grant login access.
7. Grant access to `Your Company's Administrator` with an access duration of **1 Year**.



Grant Login Access

To assist with support issues, you may grant your administrator or support personnel the ability to login as you and access your data.

My Username: **bob@wkbk.com**

Grant Access To	Access Duration
Your Company's Administrator	1 Year (exp. 8/12/2013)

You can only grant login access to the company administrator. Only administrators with the "Manage Users" permission can grant access to Salesforce Support.

8. Click **Save**.
9. Log out as Bob Smith by clicking **Bob Smith** > **Logout**.
10. Log in as the Administrator.
11. From Setup, enter `Users` in the `Quick Find` box, then select **Users**.
12. Click the **Login** link next to Bob Smith's name.

You see a notification at the top of the screen that you are logged in as Bob Smith. You can now see and do everything that Bob Smith can. Your login and logout as Bob Smith are recorded in the Setup Audit Trail for audit purposes.

13. Log out as Bob Smith.

Go back to the Login Access Policies page and enable the “Administrators Can Log in as Any User” setting. Now you can log in as any standard user. Users don’t have to grant access first.

Summary

There are several tools available to help you implement your security policies:

- Password requirements
- Session settings
- Network access
- Login access

These tools let you fine-tune your security policies. For more security tools, see Security Overview in the Salesforce online help.

TUTORIAL 6: OAUTH AND MOBILE ACCESS

Mobile devices are more popular than ever, so authentication on them is necessary. However, the anti-pattern of storing a username and password is an insecure way of providing authentication. Using OAuth 2.0, the client application delegates the authentication to a provider (in this case Force.com), which in turn issues an access token if the user successfully authenticates.

This tutorial uses Heroku to demonstrate how an app built outside of Force.com can easily and securely log into an app built on the Salesforce platform. This same authentication pattern may be applied to any external app, including mobile apps built on other platforms.

Step 1: Create an OAuth Application

Before an application can use OAuth, you have to configure the environment.

1. In a new browser tab, go to the following website: https://securityworkbook.herokuapp.com/.
2. Click **Get Started with Spring MVC**.

   You might be prompted to allow access for the "AGI" app. If so, continue with this tutorial by clicking **Allow**.
3. Enter your Heroku credentials. If you don't have any, click **Sign Up** to create your Heroku account and then restart this procedure.
4. Note the name of your new Heroku application.
5. Click **Register**.

   A new tab will open to the Salesforce login screen.
6. Login to your Developer Edition organization using your administrator credentials.

   You might briefly see the Remote Access page, which then redirects you to the Apps page. Remote access apps have been replaced by connected apps and any existing Remote Access applications were automatically migrated to connected apps with the Summer '13 release.

Step 2: Create a Connected Application

Add the application from Heroku to your list of connected apps.

1. On the Apps page, scroll down to the Connected Apps related list and click **New**.
2. For Connected App Name, enter the name of your Heroku app.
3. For API Name, enter the name of your Heroku app, but replace the dashes with underscore characters or remove the dashes. Heroku requires dashes for the app name, but Salesforce doesn't allow dashes in API names.
4. For Contact Email, enter your admin's email address.
5. Select **Enable OAuth Settings**.
6. For Callback URL, enter the URL to your Heroku app including `/_auth`.

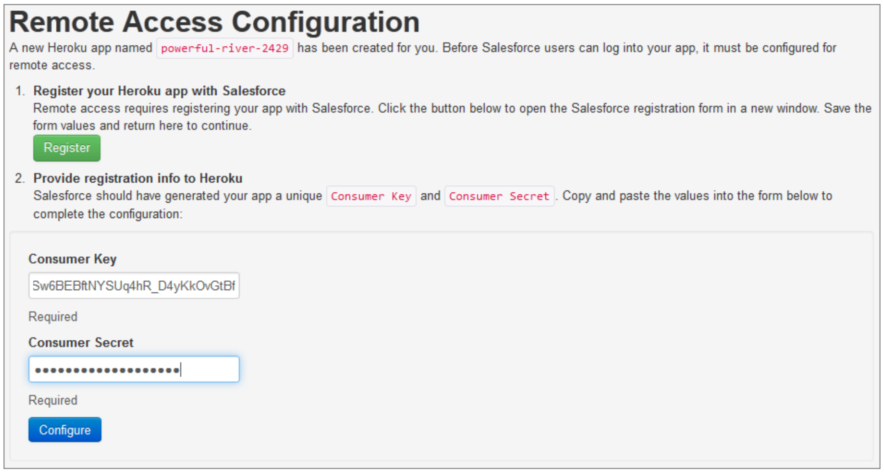
   For example, if your app is glacial-temple-2472, the callback URL is `https://glacial-temple-2472.herokuapp.com/_auth`.
7. For Selected OAuth Scopes, select the **Full access** and **Perform requests on your behalf at any time (refresh_token, offline_access)** options.

8. Click **Save**.

Step 3: Finish Your OAuth Application

Now connect up the Heroku application with the Salesforce OAuth provider.

1. On the Connected App detail page, copy the **Consumer Key** value.
2. Go back to the Heroku tab in your browser and paste in the **Consumer Key**.
3. Go back to the Salesforce tab in your browser.
4. Click to reveal your **Consumer Secret**.
5. Copy your **Consumer Secret**.
6. Go back to the Heroku tab in your browser and paste in the **Consumer Secret**.



Remote Access Configuration

A new Heroku app named `powerful-river-2429` has been created for you. Before Salesforce users can log into your app, it must be configured for remote access.

1. **Register your Heroku app with Salesforce**
   Remote access requires registering your app with Salesforce. Click the button below to open the Salesforce registration form in a new window. Save the form values and return here to continue.
2. **Provide registration info to Heroku**
   Salesforce should have generated your app a unique `Consumer Key` and `Consumer Secret`. Copy and paste the values into the form below to complete the configuration:

Consumer Key

Sw6BEBftNYSUq4hR_D4yKkOvGtBf

Required

Consumer Secret

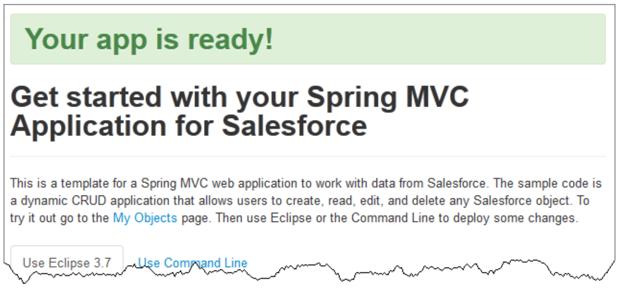
•••••••••••••••••••

Required

7. Click **Configure**.

   This may take several minutes.

8. Click on the **My Objects** link in the first paragraph of the page.



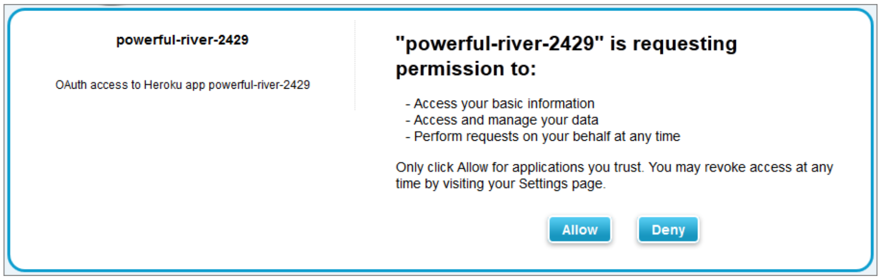
Your app is ready!

Get started with your Spring MVC Application for Salesforce

This is a template for a Spring MVC web application to work with data from Salesforce. The sample code is a dynamic CRUD application that allows users to create, read, edit, and delete any Salesforce object. To try it out go to the My Objects page. Then use Eclipse or the Command Line to deploy some changes.

You're redirected to the Salesforce OAuth screen. Make sure you are logged in as your Developer Edition org administrator in the top right corner of the page.

9. Click **Allow**.



powerful-river-2429

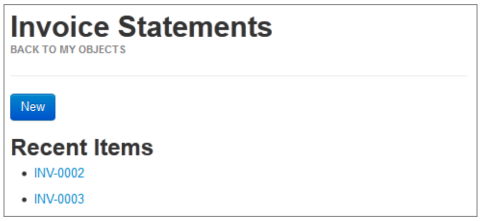
OAuth access to Heroku app powerful-river-2429

"powerful-river-2429" is requesting permission to:

- Access your basic information
- Access and manage your data
- Perform requests on your behalf at any time

Only click Allow for applications you trust. You may revoke access at any time by visiting your Settings page.

By clicking on any object, you can now view any records you have access to through your profile and role configurations. For example, clicking **Invoice Statement** shows you your invoice objects.



Invoice Statements

BACK TO MY OBJECTS

Recent Items

- INV-0002
- INV-0003

Summary

This Heroku application is yours to modify. There are instructions on the application's home page to help you do even more, including modifying the OAuth code.

TUTORIAL 7: AUDIT CONTROLS

Auditing is critical to troubleshooting and providing compliance documentation. The Salesforce platform has several types of audit controls to address both troubleshooting and documentation needs. You'll learn about the following controls:

- Login History: Tracks details about all logins, regardless of how the user logs into the organization
- Setup Audit Trail: Tracks configuration changes made to the organization
- Field History Tracking: Tracks changes to field or column data in the organization

Step 1: Using Login History

Login history helps you to determine when a user successfully or unsuccessfully logs into your organization. Use it for troubleshooting failed authentications by displaying information such as where, when, and how a user tries to log in. Because this information is accessible in the API, it's easy to use SOQL to query for detail about a specific user, the time frame, or the login type.

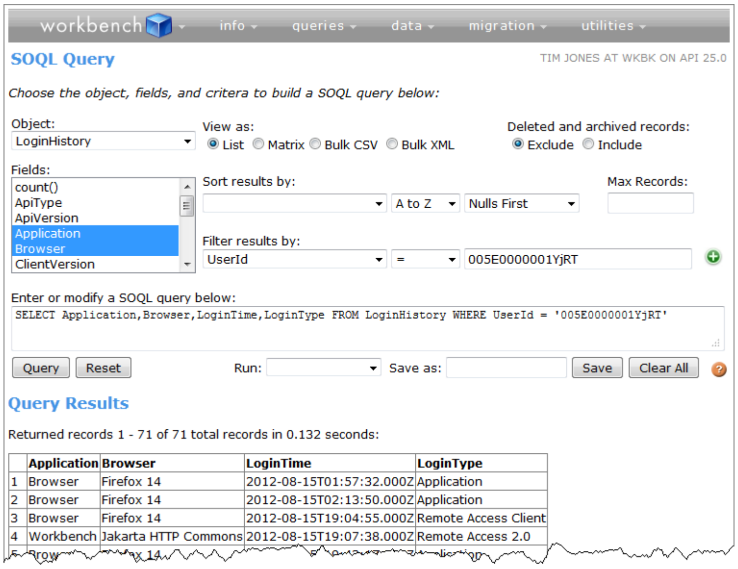
1. From Setup, enter *`Login History`* in the `Quick Find` box, then select **Login History**.
2. View the last 20 entries or create a list view to display up to the last 20,000 entries.
3. Click **Download Now** to view the last six months.
4. To view your login history as the administrator, from Setup, enter *`Users`* in the `Quick Find` box, then select **Users** and click your admin username. Then, copy the ID from the browser address bar.

   For example, *`005E0000001YjRT`*.

5. Enter the following URL in your browser: `https://developer.salesforce.com/page/Workbench`.
6. Leave the default settings and select **I agree to the terms of service** before clicking **Login with Salesforce**.

   Check the username in the top right hand corner of the screen. This is the user you're logged in as and must be the administrator of your Developer Edition organization. If it isn't, log out and log in as the administrator of your Developer Edition organization.

7. Click **Allow** on the OAuth authentication screen if it is displayed.
8. In the Workbench menu, select **queries** > **SOQL Query**.
9. Choose `LoginHistory` from Object.
10. Control-click to choose `Application, Browser, LoginType`, and `LoginTime` in Fields.
11. Filter results by: UserId = *`your UserId copied in Step 4`*.
12. Click **Query**.



SOQL Query

TIM JONES AT WKBK ON API 25.0

Choose the object, fields, and critera to build a SOQL query below:

Object: LoginHistory

View as: ● List ○ Matrix ○ Bulk CSV ○ Bulk XML

Deleted and archived records: ● Exclude ○ Include

Fields: count() ApiType ApiVersion **Application** **Browser** ClientVersion

Sort results by: A to Z Nulls First

Max Records:

Filter results by: UserId = 005E0000001YjRT

Enter or modify a SOQL query below:

```
SELECT Application,Browser,LoginTime,LoginType FROM LoginHistory WHERE UserId = '005E0000001YjRT'
```

Query Results

Returned records 1 - 71 of 71 total records in 0.132 seconds:

	Application	Browser	LoginTime	LoginType
1	Browser	Firefox 14	2012-08-15T01:57:32.000Z	Application
2	Browser	Firefox 14	2012-08-15T02:13:50.000Z	Application
3	Browser	Firefox 14	2012-08-15T19:04:55.000Z	Remote Access Client
4	Workbench	Jakarta HTTP Commons	2012-08-15T19:07:38.000Z	Remote Access 2.0

Login history is an important way of tracking who logs into your organization. It allows you to see where users log in from, when they log in, and how they log in. Because of the volume of data, login history is automatically removed after six months. If you want to keep it longer, such as for compliance regulations, consider using one of the Salesforce Web services APIs to copy the history records to a custom object or external data store.

Step 2: Using Setup Audit Trail

Setup Audit Trail provides the last six months of changes made by administrators. This can be critical for troubleshooting configuration issues. It can also be important for compliance purposes and provides an audit trail of changes.

1. From Setup, enter `View Setup Audit Trail` in the `Quick Find` box, then select **View Setup Audit Trail**.
2. View the last 20 entries.

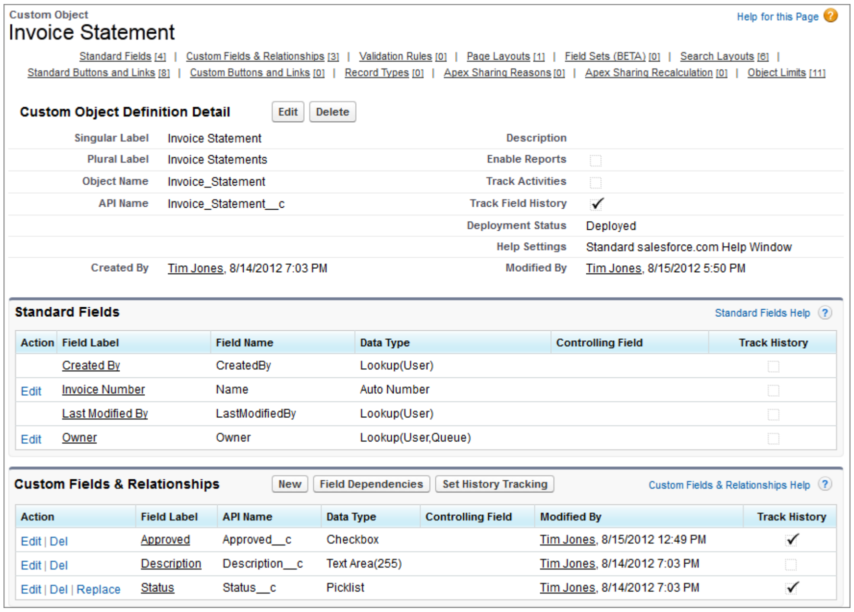
   Note that when an administrator logs in as another user, Force.com audits both the user and the delegate user login and logout events, as well as any changes made by the administrator in setup while logged in as another user.
3. Click the **download** link for the last six months of audit trail entries.

Setup Audit Trail not only logs changes made to the configurations of the organization, it also logs who made the change even if a user is a delegate user using the Login As feature.

Step 3: Using Field History Tracking

Field History Tracking determines who changed a value on a field, when it was changed, and what the old and new values are. This can be helpful when troubleshooting data changes within a record. You can also use it for compliance purposes to track the changes made over time. The last eighteen months of field history are kept for each organization.

1. From Setup, enter `Objects` in the `Quick Find` box, select **Objects**, then select **Invoice Statement**.
2. Enable field history tracking for invoice statements.
   a. Make sure **Track Field History** is checked in. If it's not checked, edit the custom object and select **Track Field History**.
   b. Go to Custom Fields & Relationships and click **Set History Tracking**. Select Approved and Status.


Custom Object
Help for this Page
Standard Fields [4] | Custom Fields & Relationships [3] | Validation Rules [0] | Page Layouts [1] | Field Sets (BETA) [0] | Search Layouts [6] |
Standard Buttons and Links [8] | Custom Buttons and Links [0] | Record Types [0] | Apex Sharing Reasons [0] | Apex Sharing Recalculation [0] | Object Limits [11]
Edit
Delete
Standard Fields Help
New
Field Dependencies
Set History Tracking
Custom Fields & Relationships Help

Invoice Statement

Custom Object Definition Detail

Singular Label	Invoice Statement	Description	
Plural Label	Invoice Statements	Enable Reports	□
Object Name	Invoice_Statement	Track Activities	□
API Name	Invoice_Statement__c	Track Field History	✓
		Deployment Status	Deployed
		Help Settings	Standard salesforce.com Help Window
Created By	Tim Jones, 8/14/2012 7:03 PM	Modified By	Tim Jones, 8/15/2012 5:50 PM

Standard Fields

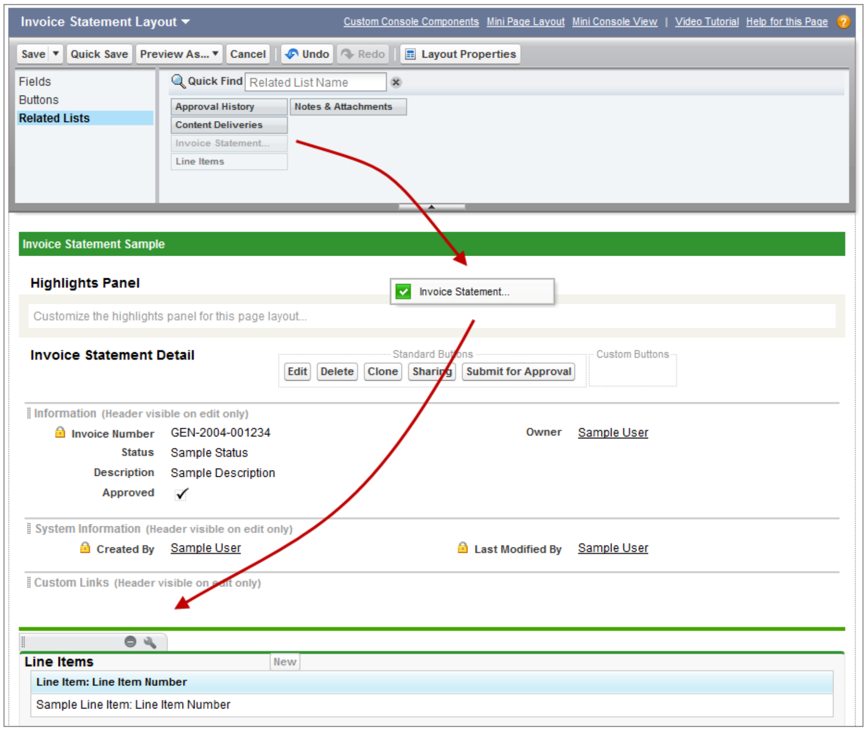
Action	Field Label	Field Name	Data Type	Controlling Field	Track History
	Created By	CreatedBy	Lookup(User)		□
Edit	Invoice Number	Name	Auto Number		□
	Last Modified By	LastModifiedBy	Lookup(User)		□
Edit	Owner	Owner	Lookup(User,Queue)		□

Custom Fields & Relationships

Action	Field Label	API Name	Data Type	Controlling Field	Modified By	Track History
Edit | Del	Approved	Approved__c	Checkbox		Tim Jones, 8/15/2012 12:49 PM	✓
Edit | Del	Description	Description__c	Text Area(255)		Tim Jones, 8/14/2012 7:03 PM	□
Edit | Del | Replace	Status	Status__c	Picklist		Tim Jones, 8/14/2012 7:03 PM	✓

c. Click **Save**.

3. Add the Invoice Statement History related list to the invoice statements page layout.
    a. Go to Page Layouts, and click **Edit** for the Invoice Statement Layout.
    b. Click **Related Lists** in the left window of the toolbar.
    c. Drag and drop the Invoice Statement History related list below the first section of the layout.

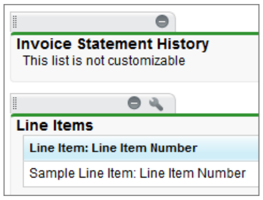


Invoice Number	GEN-2004-001234	Owner	Sample User
Status	Sample Status		
Description	Sample Description		
Approved	✓		
Created By	Sample User	Last Modified By	Sample User

Line Items

Line Item: Line Item Number
Sample Line Item: Line Item Number

The Invoice Statement Sample should look like the following.



Invoice Statement History

This list is not customizable

Line Items

Line Item: Line Item Number
Sample Line Item: Line Item Number

d. Click **Save**.

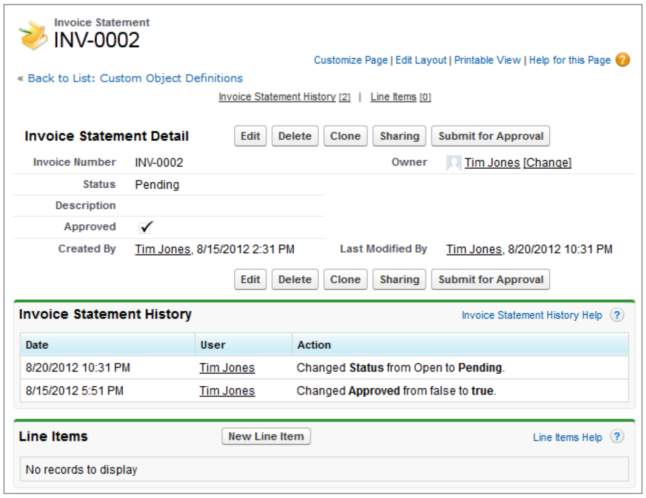
e. Click **Yes** to overwrite users' related list customizations.

4. Click the **Invoice Statements** tab.

   a. Click into any invoice or create a new one.

   b. Edit the record and change the Approved or Status field that you are tracking.

   c. View the Invoice Statement History related list to see the changes that were made.



Invoice Statement
Customize Page | Edit Layout | Printable View | Help for this Page
« Back to List: Custom Object Definitions
Invoice Statement History [2] | Line Items [0]
Edit
Delete
Clone
Sharing
Submit for Approval
Invoice Statement History Help
New Line Item
Line Items Help

INV-0002

Invoice Statement Detail

Invoice Number	INV-0002	Owner	Tim Jones [Change]
Status	Pending		
Description			
Approved	✓		
Created By	Tim Jones, 8/15/2012 2:31 PM	Last Modified By	Tim Jones, 8/20/2012 10:31 PM

Invoice Statement History

Date	User	Action
8/20/2012 10:31 PM	Tim Jones	Changed **Status** from Open to **Pending**.
8/15/2012 5:51 PM	Tim Jones	Changed **Approved** from false to **true**.

Line Items

No records to display

Access Field History Tracking from each record's related list, through reports, or through a read-only SOQL query using the API. This allows you to query across users and records to get the bigger picture of changes made to your records.

Summary

You've seen how to use the Salesforce auditing tools to both troubleshoot problems and monitor changes to your organization and to your data. In addition, these tools can also help you meet data compliance requirements by providing an audit trail for all changes.

NEXT STEPS

To continue exploring, visit https://developer.salesforce.com/docs. View the official documentation for Salesforce and its APIs, read the articles and tutorials, and check out the other resources to help you build your applications.