



Version 1: Summer '14

Security Workbook



Last updated: June 14, 2014

Table of Contents

About the Security Workbook.....	1
Tutorial 1: Create Users.....	2
Step 1: Create Users Manually through Setup.....	2
Step 2: Create Users Automatically Through the REST API.....	3
Summary.....	6
Tutorial 2: Create Functional Access Controls.....	7
Step 1: Create a Profile.....	7
Step 2: Manage Field-Level Security.....	8
Step 3: Create Permission Sets.....	9
Summary.....	10
Tutorial 3: Create Record-Level Access Controls.....	12
Step 1: Configure Organization-Wide Default Sharing.....	13
Step 2: Create a Role for Your Organizational Hierarchy.....	13
Step 3: Create a Sharing Rule.....	14
Step 4: Create a Public Group.....	15
Summary.....	16
Tutorial 4: Login to Test Authorizations.....	17
Step 1: Edit a User.....	17
Step 2: Create a New Invoice Record.....	17
Step 3: Log In as the New User.....	18
Summary.....	19
Tutorial 5: Security Policies for Authorizing Users.....	20
Step 1: Set Password Policies.....	20
Step 2: Set Session Timeout.....	20
Step 3: Limit Network Access with IP Ranges.....	20
Step 4: Configure Login Access Policies.....	21
Summary.....	22
Tutorial 6: OAuth and Mobile Access.....	23
Step 1: Create an OAuth Application.....	23
Step 2: Create a Connected Application.....	23
Step 3: Finish Your OAuth Application.....	24
Summary.....	25
Tutorial 7: Audit Controls.....	26
Step 1: Using Login History.....	26
Step 2: Using Setup Audit Trail.....	27
Step 3: Using Field History Tracking.....	28
Summary.....	30

Next Steps.....31

About the Security Workbook

Security is of vital importance to an organization. It ensures that users are guaranteed 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 can be applied to grant system administrators and application users no more privilege than what is absolutely necessary to perform their specific jobs.

The Salesforce platform provides a least-privilege, user-centric security model including tools to control user's login and access controls. Because applications are written once and run everywhere, your user's authentication and access controls are consistent regardless of whether they are accessing it from a Web application or mobile 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 takes you through the various policies, rules, and grouping mechanisms that ensure your users are able to log in securely and can do everything that they should and nothing that they shouldn't. Each of the tutorials builds on the previous tutorial 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.

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 (http://wiki.developerforce.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 the tutorials here.

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.

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.

Tell Me More...

Many steps end with a Tell Me More section. If you like to do things quickly, move on to the next step. However, if you're a smell-the-roses type, there's a lot of useful information here and some additional things to try.

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 a different 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 Manually through Setup

In this step, you create a new user and link that user to your current user's account via the Manager field. You use this configuration to make sure that if the new user creates an invoice that meets certain conditions, the invoice is routed to his or her 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 3.

1. From Setup, click **Manage Users > 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 be logging in as him shortly.

The screenshot shows the 'New User' form in Salesforce Setup. The form is titled 'New User' and has a 'User Edit' section with 'Save', 'Save & New', and 'Cancel' buttons. Below is the 'General Information' section with fields for First Name (Bob), Last Name (Smith), Alias (bSmith), Email (your_email@your.org), Username (Bob@wkbk.com), and Community Nickname (Bobby). On the right, there are dropdowns for Role (<None Specified>), User License (Salesforce), and Profile (Standard User), along with checkboxes for Active (checked), Marketing User, and Offline User.

- f. In **Manager**, select the user you created when you signed up for your organization.
 - g. In **User License**, select Salesforce.
 - h. In **Profile**, select Standard User.
4. Click **Save**.

You should now 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.

Tell Me More...

You can also use SAML to create users just-in-time. For more information, see “Just-In-Time Provisioning” in the Salesforce online help, and check out this article on just-in-time user provisioning at <http://blogs.developerforce.com/developer-relations/2011/05/just-in-time-user-provisioning-in-summer-11.html>.

Step 2: Create Users Automatically Through the REST API

In this tutorial you use the Workbench tool to create users through the REST API.

1. Start by logging into Workbench and running a query to find Bob Smith, the user you created in the previous step.
 - a. Type or paste the following URL into your browser: workbench.developerforce.com.
 - 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**.

workbench info queries data migration utilities

SOQL Query TIM JONES AT WKBK ON API 25.0

Choose the object, fields, and criteria 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'

Query Reset Run: Save as: Save Clear All

- i. Click **Query**.

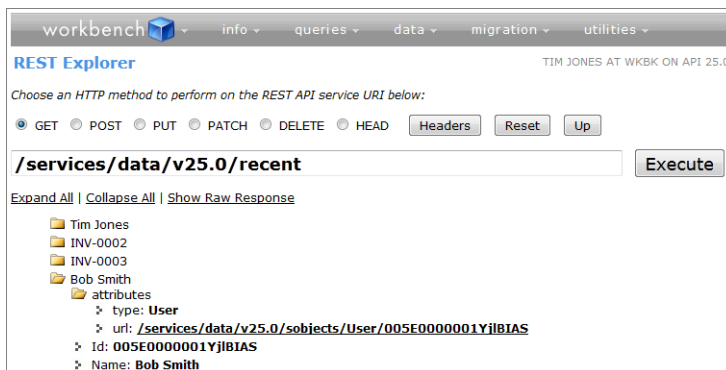
Query Results

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

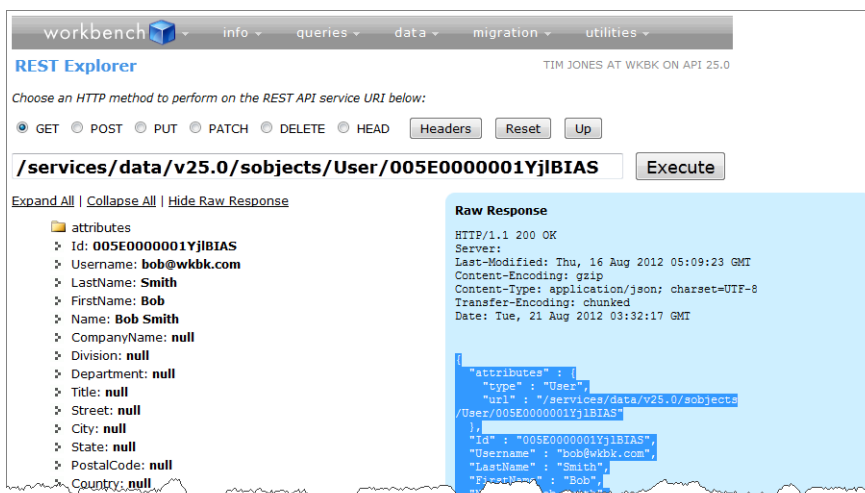
Id	Name
100eE00000000uRkTIAU	Standard User

- 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#}/subjects/User/Bob Smith's user Id`.



- 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.



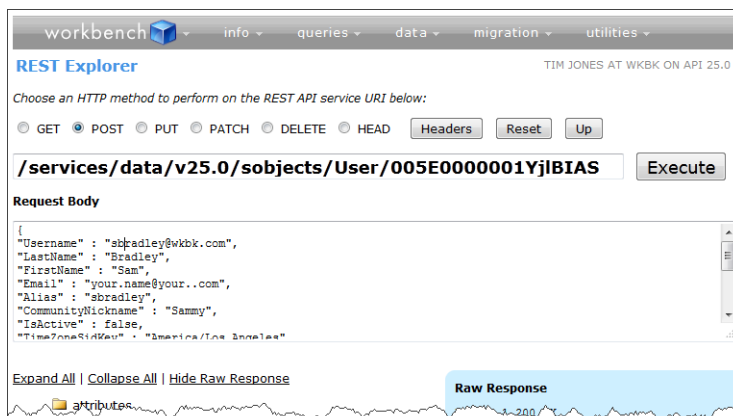
The text between the curly brackets is a JSON object representation of Bob Smith's information.

3. To finish up, create a new user with the REST API.
 - a. Select **POST**.
 - b. Change the REST endpoint to `/services/data/v{version#}/subjects/User/`

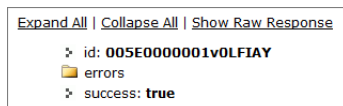
- c. You can either modify Bob Smith's information copied earlier, updating it as needed to have it describe Sam Bradley; or you can copy the following text, making changes to anything in *italics*. Change *yourOrgDomain* and *yourEmailAddress*, and replace *Standard User Profile Obtained from earlier 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.



- e. Click **Execute**.
f. Verify that an Id is returned and that success is true.



- g. Copy the **userid**.
h. Click **GET**.
i. Paste the `userId` you just created into the REST endpoint
`/services/data/v{version#}/subjects/User/{userId you just created}`
j. Click **Execute**.
View the user record details.



Summary

REST API is fast 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 are about what a user can do after they've logged into an organization. They define which database tables, called *objects*, and which database columns, called *fields*, they can access. They also define other types of access controls that enable users to functionally 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 for 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

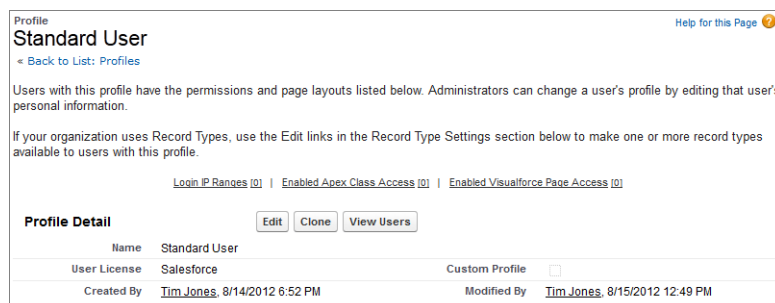
The available permissions you can configure for a profile or permission set depend on the user's license. You cannot assign permissions that would exceed what a user's license allows. For instance, platform users may not have the right to modify all data within an organization or they would violate the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

Step 1: Create a Profile

Use profiles to assign the same permission to many users. A profile is a baseline 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 only explicitly granted to users. You can change object permissions in custom profiles, but not standard profiles.

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

1. From Setup, click **Manage Users > Profiles**.
2. Click **Standard User**.
3. Select **Clone** next to Profile Detail.



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.

The screenshot shows the 'Profile Edit' page for a user named 'Warehouse User'. The page is divided into several sections:

- Profile Edit:** Includes fields for Name (Warehouse User), User License (Salesforce), and Description. There are 'Save' and 'Cancel' buttons.
- Custom App Settings:** A table with columns for 'Visible' and 'Default' for various apps. The 'Warehouse' app is highlighted with a blue selection circle.
- Custom Tab Settings:** Two dropdown menus for 'Invoice Statements' and 'Merchandise', both set to 'Default On'.
- Custom Object Permissions:** A table with columns for 'Basic Access' (Read, Create, Edit, Delete) and 'Data Administration' (View, Modify) for 'Invoice Statements', 'Line Items', and 'Merchandise'. All permissions are checked.

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

10. Click **Save**.

Tell Me More...

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 be configuring object and field permissions. In this use case, invoices need to be marked as approved based on a field in the invoice, but not everyone should have 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, click **Create > Objects > 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. For **Field Name**, type Approved.

Invoice Statement
New Custom Field

Step 2. Enter the details Step 2 of 4

Field Label

Default Value ☐ Checked ☒ Unchecked

Field Name

Previous Next Cancel

6. Click **Next**.
7. On the Establish field-level security page, select **Read-Only** on the Field-Level Security for Profile row at the top of the Read-Only column.

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
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Previous Next Cancel

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

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

Tell Me More...

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 online help.

Step 3: Create Permission Sets

Permission sets make salesforce.com administrator's and developer's lives easier by assigning permissions to users with more granularity than what a profile already provides. While a user must have a profile and may have only one assigned, a user may have zero, one, or many permission sets. This provides a level of flexibility when creating multiple apps, having users in multiple regions, or working within multiple industries. Because all permissions layer in a positive way between a profile and a permission set, there can never be a conflict. Regardless of whether a user has a permission granted on their profile or any one of their permission sets, they have the access that the permission 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 user. Field-level security could have just been added to the user's profile, but because users from different departments may have different profiles, it's easier and more efficient to create just one permission set for these different users rather than modify many different profiles.

1. From Setup, click **Manage Users > 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.

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

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**.

Tell Me More...

Permission sets can be used for lots of different scenarios. Check out the following blog posting for some great ideas:
<http://blogs.salesforce.com/product/2012/01/permission-set-best-practice-you-should-try-this-out-at-home.html>.

Summary

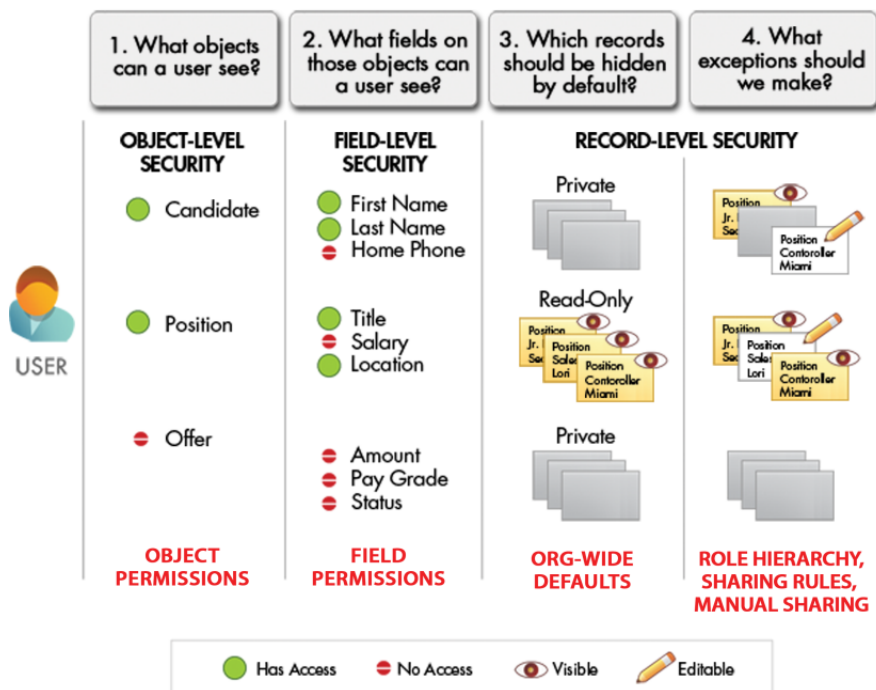
You've seen how to use profiles, permission sets, and field-level security. Profiles and permission sets provide help you control access. Profiles contain many different types of access controls. One example is Field-Level Security. Field-Level Security provides a highly granular and scalable form of 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 assigned. 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 for the most strict user requirement, 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 inherited up a role hierarchy, through additional public groups assigned to a user or nested into one another, and by sharing rules.

The first place to configure your sharing settings is organization-wide defaults. Organization-wide defaults allow you 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, click **Security Controls > 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	✓

Tell Me More...

When you set an object's organization-wide defaults to Private, you limit access to records, and 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 the record. 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, click **Manage Users > 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

Save Save & New Cancel

5. Click **Save**.

Tell Me More...

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, wait until it's all put together on the user detail page.

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 have access to 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 access that rolls up the role hierarchy.

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 will modify it after it's closed.

1. From Setup, click **Security Controls > 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.

- For **Access Level**, choose **Read/Write**.

Setup Help for this Page ?

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

Rule Name ⓘ

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--		AND

[Add Filter Logic...](#)

Step 4: Select the users to share these records with

Share with

Step 5: Select the level of access for the users

Access Level

- Click **Save**.
- Click **Okay** at the popup.

Tell Me More...

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 also need access to invoices during tax time. Where roles automatically can contain another role, public groups may either contain users, roles, or other public groups. Whereas a user may have one role, a user may be a member of zero, one, or many different groups. This makes public groups similar to permission sets where flexibility is needed granting additional access. Public groups specifically 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 any other business segmentation of users you need to group together for record access. Public groups are typically used with sharing rules to open up access.

- From Setup, click **Manage Users > Public Groups**.
- Click **New**.
- In the **Label** field, type **Contractors**.
- In the **Group Name** field, type **Contractors**.
- Select **Grant Access Using Hierarchies**.
- In **Search:**, select **Users**.

7. Move **User: Bob Smith** from Available Members to Selected Members.

Group Membership
New Group

Group Information

Save Cancel

New Public Group ⓘ = Required Information

Label Contractors

Group Name Contractors ⓘ

Grant Access Using Hierarchies ☒ ⓘ

Search: Users for: Find

Available Members

User: Tim Jones

Add

Remove

Selected Members

User: Bob Smith

Save Cancel

8. Click **Save**.

Tell Me More...

Just as sharing rules can share access to roles, they can also share to public groups. Just as roles nest within one another, 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 user you just created and his access levels. First you need to make some changes to his user details.

1. From Setup, click **Manage Users** > **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 \(0+\)](#)

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	✓

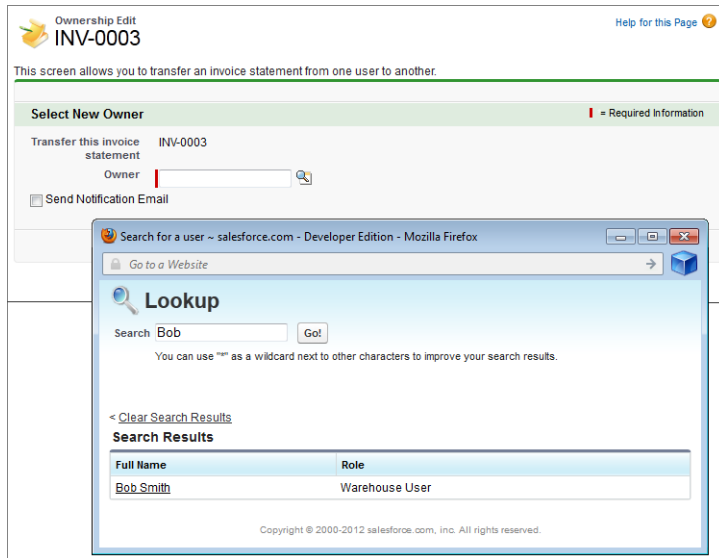
5. Click **Save**.
6. Scroll down to Permission Set Assignments.
7. Click **Edit Assignments**.
8. Move Invoice Approver from **Available Permission Sets** to **Enabled Permission Sets**.
9. Click **Save**.

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.



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. 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 it has an open status according to the sharing rule you created. 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**.

Tell Me More...

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 the [Granting Login Access](#) topic in the online help.

Summary

You've seen how to add roles, profiles, and permission sets to a user. After making access changes, it's always a good idea to test those changes. In this tutorial you saw how to make sure your access levels were correct for the appropriate 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.

1. Log in as the administrator if you aren't already.
2. From Setup, click **Security Controls > 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 you want, although when people step away from their computers, you want to make sure unauthorized people do not get access to data you don't want them to see.

1. From Setup, click **Security Controls > Session Settings**.
2. In **Timeout value**, select 30 minutes.
3. Click **Save**.

Tell Me More...

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

Step 3: Limit Network Access with IP Ranges

A great way to control where users access your organization is to control the network. By applying IP range restrictions, you can allow all users to only 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, click **Manage Users > Profiles**.
To control this by group of users, select Bob Smith's Warehouse User profile.
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**.

Tell Me More...

If a user tries to log in from an unrecognized IP range, they will be forced to send themselves an authorization token that will allow them to log in from that new IP address.

Step 4: Configure Login Access Policies

Login-as is a powerful tool for administrators and support reps at Salesforce or their partners to help troubleshoot issues by logging in as another user. You may need to control whether any user can grant access to Salesforce or any other support. You may also want to allow an administrator to login as any user automatically. Here you'll make this more restrictive by only allowing administrators to grant access to Salesforce support to help resolve issues that may arise.

1. From Setup, click **Security Controls > Login Access Policies**.
2. Click **Available to Administrators Only**.
3. Click **Save**.
4. Log out as the Administrator.
5. Log in as Bob Smith.
6. At the top of any Salesforce page, click the down arrow next to your name. From the menu under your name, select **Setup** or **My Settings**—whichever one appears.
7. From the left pane, select one of the following:
 - If you clicked **Setup**, select **My Personal Information > Grant Login Access**.
 - If you clicked **My Settings**, select **Personal > Grant Account Login Access**.
8. Grant Access to Your Company's Administrator with an Access Duration of **1 Year**.

Expand All | Collapse All

Quick Find

Personal Setup

- My Personal Information
 - Personal Information
 - Change My Password
 - Reset My Security Token
 - My Groups
 - Change My Display
 - Grant Login Access**
 - Calendar Sharing

Grant Login Access [Help for this Page](#)

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

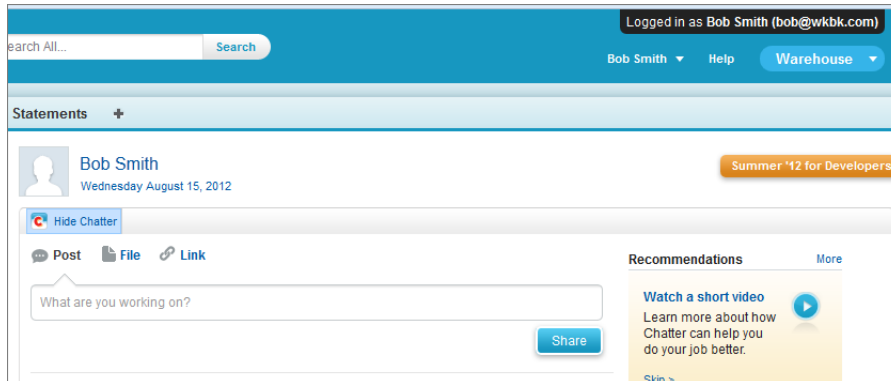
Grant Access To	Access Duration
Your Company's Administrator	1 Year (exp. 8/12/2013)

Save Cancel

Note that you can only grant login access to the company administrator. Only administrators with Manage Users permission may grant access to Salesforce Support.

9. Click **Save**.
10. Log out as Bob Smith by clicking **Bob Smith > Logout**.
11. Log in as the Administrator.
12. From Setup, click **Manage Users > Users**.
13. Click the **Login** link next to Bob Smith's name.

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



14. Log out as Bob Smith, which returns you to the Users setup page as the System Administrator

Tell Me More...

If your organization is configured for it, you can also control whether users have to first grant access to an administrator or whether you can log in as any standard user automatically. For more information, check out <http://blogs.salesforce.com/product/2012/06/log-in-as-any-user-without-first-having-access-granted-new-feature-in-summer-12.html>.

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

You don't want just anyone accessing your data, so there needs to be authentication on the mobile device. 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. Thereafter, as long as a valid access token accompanies all API interactions, you don't need to worry about authentication.

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 different platforms.

Step 1: Create an OAuth Application

Before an application can use OAuth, you have to configure your 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

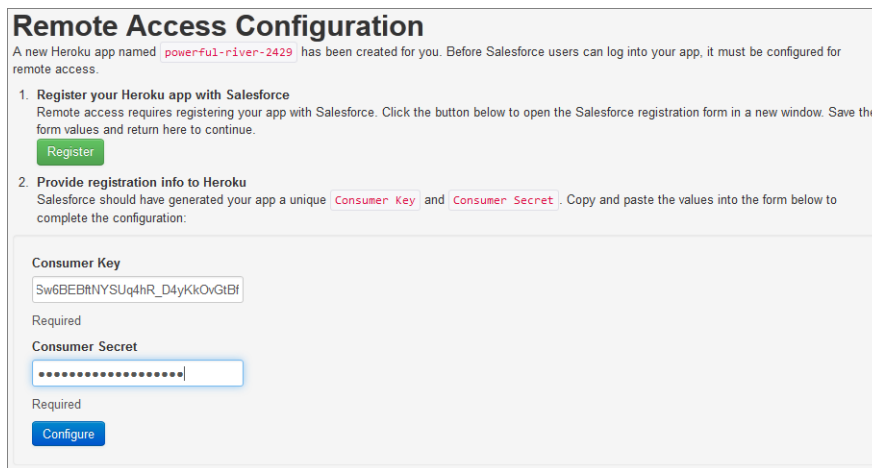
Add the application from Heroku to your list of connected apps.

1. On the Apps page, scroll down to find the Connected Apps related list and click **New**.
2. **Connected App Name** should be the name of your Heroku app.
3. **API Name** should be 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. **Contact Email** should be your administrator's email address.
5. Select **Enable OAuth Settings**.
6. **Callback URL** should be the URL to your Heroku app including `/_auth`.
For example, `https://arcane-crag-2451.herokuapp.com/_auth`
7. For **Selected OAuth Scopes**, add the following.
 - a. Full access
 - b. Perform requests on your behalf at any time (refresh_token)
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.
[Register](#)
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

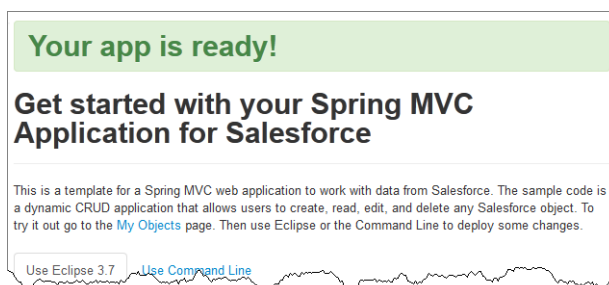
Required

Consumer Secret

Required

[Configure](#)

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!

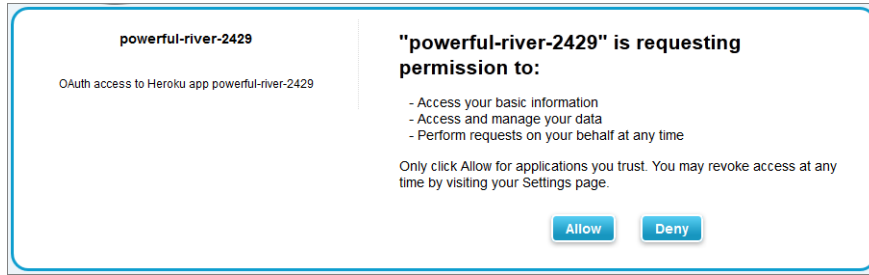
Get started with your Spring MVC Application for Salesforce

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.

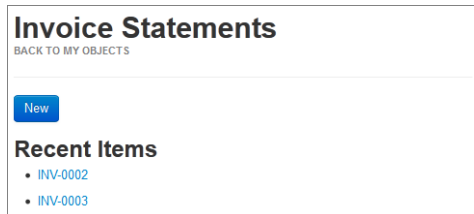
[Use Eclipse 3.7](#) [Use Command Line](#)

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**.



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.



Summary

This Heroku application is yours to modify. There are instructions on the application's home page to help you take this the next step further, including modifying the OAuth code. If you would like to dig into OAuth more, check out the following blog article: http://wiki.developerforce.com/page/Digging_Deeper_into_OAuth_2.0_on_Force.com.

Tutorial 7: Audit Controls

Auditing is critical to troubleshooting and providing compliance documentation as users interact with an organization. The Salesforce platform has several types of audit controls to address these needs. You'll learn about three specific audit controls:

1. Login History—Tracks details about all logins regardless of how the user logs into the organization
2. Setup Audit Trail—Tracks configuration changes made to the organization
3. 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 logged into your organization. You can use it for troubleshooting failed authentications by displaying information such as where, when, and how a user tried to log in. Because it's accessible in the API, it's easy to use SOQL to query for information about a specific user, time frame, or login type.

1. From Setup, click **Manage Users > 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, click **Manage Users > Users** and click your admin username. Then, copy the ID from the browser address bar.
For example, 005E00000001YjRT.
5. Enter the following URL in your browser: <https://workbench.developerforce.com>.
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, click **(Not you?)** and log in as the administrator of the 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**.

The screenshot shows the Workbench SOQL Query interface. The 'Object' is set to 'LoginHistory'. The 'View as' options are 'List', 'Matrix', 'Bulk CSV', and 'Bulk XML', with 'List' selected. The 'Deleted and archived records' options are 'Exclude' and 'Include', with 'Exclude' selected. The 'Fields' list includes 'count()', 'ApiType', 'ApiVersion', 'Application', 'Browser', and 'ClientVersion', with 'Application' and 'Browser' selected. The 'Sort results by' dropdown is set to 'A to Z' and 'Nulls First'. The 'Filter results by' dropdown is set to 'UserId' with the value '005E0000001YjRT'. The SQL query is: `SELECT Application,Browser,LoginTime,LoginType FROM LoginHistory WHERE UserId = '005E0000001YjRT'`. The 'Query Results' section shows 71 records returned in 0.132 seconds. The table below shows the first four rows of the results.

	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

Tell Me More...

Login history is an important way of tracking who is logging into your organization. It allows you to see where they are logging in from, when they logged in, and even how they logged into the organization. Because of the volume of the data, login history is automatically removed after six months. If you want to keep it for 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 for longer storage.

Step 2: Using Setup Audit Trail

Setup Audit Trail provides the last six months of changes made by administrators. This can be critical when troubleshooting when a configuration changed and who changed it so that you can track down why they made the change. It can also be important for compliance purposes, and provides an auditor an audit trail of changes made to your system.

1. From Setup, click **Security Controls > 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.

Tell Me More...

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. It can also be used 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, click **Create > Objects**.
2. Click **Invoice Statement**.
 - a. Make sure **Track Field History** is checked in Custom Object Definition Detail. If it's not checked, edit the custom object and select **Track Field History**.
 - b. Click **Set History Tracking** for Approved and Status in the Custom Fields & Relationships related list.

Custom Object [Help for this Page](#)

Invoice Statement

[Standard Fields \(4\)](#) | [Custom Fields & Relationships \(3\)](#) | [Validation Rules \(0\)](#) | [Page Layouts \(1\)](#) | [Field Sets \(BETA\) \(0\)](#) | [Search Layouts \(0\)](#) | [Standard Buttons and Links \(8\)](#) | [Custom Buttons and Links \(0\)](#) | [Record Types \(0\)](#) | [Apex Sharing Reasons \(0\)](#) | [Apex Sharing Recalculation \(0\)](#) | [Object Limits \(11\)](#)

Custom Object Definition Detail Edit Delete

Singular Label	Invoice Statement	Description	
Plural Label	Invoice Statements	Enable Reports	<input type="checkbox"/>
Object Name	Invoice_Statement	Track Activities	<input type="checkbox"/>
API Name	Invoice_Statement__c	Track Field History	<input checked="" type="checkbox"/>
		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 [Standard Fields Help](#)

Action	Field Label	Field Name	Data Type	Controlling Field	Track History
Created By	CreatedBy	CreatedBy	Lookup(User)		<input type="checkbox"/>
Edit Invoice Number	Name	Name	Auto Number		<input type="checkbox"/>
Last Modified By	LastModifiedBy	LastModifiedBy	Lookup(User)		<input type="checkbox"/>
Edit Owner	Owner	Owner	Lookup(User,Queue)		<input type="checkbox"/>

Custom Fields & Relationships New Field Dependencies Set History Tracking [Custom Fields & Relationships Help](#)

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	<input checked="" type="checkbox"/>
Edit Del	Description	Description__c	Text Area(255)		Tim Jones, 8/14/2012 7:03 PM	<input type="checkbox"/>
Edit Del Replace	Status	Status__c	Picklist		Tim Jones, 8/14/2012 7:03 PM	<input checked="" type="checkbox"/>

- c. Click **Save**.
3. In the **Page Layouts** related list, click **Edit** on the Invoice Statement Layout.
 - a. Click **Related Lists** in the left window of the toolbar.
 - b. Drag and drop the Invoice Statement History related list below the first section of the layout.

Invoice Statement Layout

Custom Console Components Mini Page Layout Mini Console View Video Tutorial Help for this Page

Save Quick Save Preview As... Cancel Undo Redo Layout Properties

Fields Buttons Related Lists

Quick Find Related List Name

Approval History Notes & Attachments

Content Deliveries

Invoice Statement...

Line Items

Invoice Statement Sample

Highlights Panel

Customize the highlights panel for this page layout...

Invoice Statement Detail

Standard Buttons Custom Buttons

Edit Delete Clone Sharing Submit for Approval

Information (Header visible on edit only)

Invoice Number GEN-2004-001234 Owner Sample User

Status Sample Status

Description Sample Description

Approved ✓

System Information (Header visible on edit only)

Created By Sample User Last Modified By Sample User

Custom Links (Header visible on edit only)

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

- c. Click **Save**.
 - d. Click **Yes** to overwrite users' related list customizations.
4. Click **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
INV-0002

Customize Page | Edit Layout | Printable View | Help for this Page

« Back to List: Custom Object Definitions

Invoice Statement History (2) | Line Items (0)

Invoice Statement Detail [Edit] [Delete] [Clone] [Sharing] [Submit for Approval]

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

[Edit] [Delete] [Clone] [Sharing] [Submit for Approval]

Invoice Statement History Invoice Statement History Help

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 [New Line Item] Line Items Help

No records to display

Tell Me More...

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 the Dev Center at www.salesforce.com and view the official documentation for Salesforce and the APIs, read the articles and tutorials, and check out the other resources there to help you build awesome applications.