



salesforce

THE CUSTOMER SUCCESS PLATFORM

SALES SERVICE MARKETING COMMUNITY ANALYTICS APPS

Secure Coding: Field-Level Security, CRUD, and Sharing

Kyle Tobener

Product Security Engineer

@KyleKyle

Maxwell Feldman

Product Security Engineer

dreamforce®

Safe Harbor

Safe harbor statement under the Private Securities Litigation Reform Act of 1995:

This presentation may contain forward-looking statements that involve risks, uncertainties, and assumptions. If any such uncertainties materialize or if any of the assumptions proves incorrect, the results of salesforce.com, inc. could differ materially from the results expressed or implied by the forward-looking statements we make. All statements other than statements of historical fact could be deemed forward-looking, including any projections of product or service availability, subscriber growth, earnings, revenues, or other financial items and any statements regarding strategies or plans of management for future operations, statements of belief, any statements concerning new, planned, or upgraded services or technology developments and customer contracts or use of our services.

The risks and uncertainties referred to above include – but are not limited to – risks associated with developing and delivering new functionality for our service, new products and services, our new business model, our past operating losses, possible fluctuations in our operating results and rate of growth, interruptions or delays in our Web hosting, breach of our security measures, the outcome of any litigation, risks associated with completed and any possible mergers and acquisitions, the immature market in which we operate, our relatively limited operating history, our ability to expand, retain, and motivate our employees and manage our growth, new releases of our service and successful customer deployment, our limited history reselling non-salesforce.com products, and utilization and selling to larger enterprise customers. Further information on potential factors that could affect the financial results of salesforce.com, inc. is included in our annual report on Form 10-K for the most recent fiscal year and in our quarterly report on Form 10-Q for the most recent fiscal quarter. These documents and others containing important disclosures are available on the SEC Filings section of the Investor Information section of our Web site.

Any unreleased services or features referenced in this or other presentations, press releases or public statements are not currently available and may not be delivered on time or at all. Customers who purchase our services should make the purchase decisions based upon features that are currently available. Salesforce.com, inc. assumes no obligation and does not intend to update these forward-looking statements.

No Photos Required....



Slides and demos will be made available after the talk!

Primary Topic Today: Authorization

- We will be covering developer-oriented authorization topics on the Salesforce platform.
- Specific features to cover include:
 - FLS
 - CRUD
 - Sharing
- Useful for anyone in the following areas:
 - Salesforce Developers
 - Salesforce Partners
 - Salesforce Administrators

What is Authorization?

“Authorization dictates what a user is permitted to access.”

Guiding Principle: Least Privilege

“A person should only have access to the minimum amount of information required to accomplish their duties, ensuring that their ability to take advantage of excess privilege purposefully or accidentally is minimized.”

A Note: Salesforce Contexts

- User Context - Current user's authorization respected
- System Context - Current user's authorization ignored
 - This is done on purpose to allow more extensible and flexible coding, but needs to be done properly!

CRUD

CRUD

	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Accounts	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Assets	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Campaigns	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Cases	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

What is CRUD? **C**reate **R**ead **U**ppdate **D**elete!

- » Controlled on the profile
- » Dictates user abilities object by object

CRUD for Developers

- Apex Classes do not enforce CRUD
 - Why? System Context
- Visualforce Pages do enforce CRUD
 - Why? User Context

Enforcing CRUD in Apex

```
<sObject>.sObjectType.getDescribe()
```

- isCreateable()
- isAccessible()
- isUpdateable()
- isDeletable()

```
1 Public Class MyController {  
2     Public String getmyAccount {  
3         if (!Account.sObjectType.getDescribe().isAccessible())  
4             return '';  
5     }  
6 }
```

Demo: CRUD

Trivia!

“Which of the following Visualforce code patterns respect the R (read) in CRUD?”

1. `<apex:outputField value="{!sObject.Field__c}"/>`
2. `<apex:outputText value="{!sObject.Field__c}"/>`
3. `{!sObject.Field__c}` *Note : (Naked merge Field)*
4. `<apex:outputText value="{!Object.String}"/>`

Trivia (answered)!

“Which of the following Visualforce code patterns respect the R (read) in CRUD?”

1. `<apex:outputField value="{!sObject.Field__c}"/>`

2. `<apex:outputText value="{!sObject.Field__c}"/>`

3. `{!sObject.Field__c}` **Note : (Naked merge Field)**

4. `<apex:outputText value="{!Object.String}"/>`

FLS

FLS

Field Name	Field Type	Visible	Read-Only
Account Name	Name	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Account Number	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Account Owner	Lookup	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Account Site	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Account Source	Picklist	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Active	Picklist	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Annual Revenue	Currency	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Billing Address	Address	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Created By	Lookup	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

What is FLS? **Field Level Security!**

- » Controlled on the profile
- » Dictates which fields are visible to a user on a given object

FLS For Developers

- Apex classes do not enforce FLS
 - Why? System Context
- Visualforce pages do enforce FLS
 - User mode
 - Exception: de-referenced fields
 - `{!Contact.Email}` = yes
 - `{!contactEmail}` = NO

Enforcing FLS in Apex

`Schema.sObjectType.<sObject>.fields.<field>`

- `isAccessible()`
- `isUpdateable()`

```
1 Public Class MyController {  
2     Public String getmyAccount {  
3         if (!Schema.sObjectType.Account.fields.Name.isAccessible()) {  
4             return '';  
5         }  
6         ...  
7     }
```

Demo: FLS

When does the Platform stop respecting FLS?

When you assign from an sObject to a primitive!

Apex:

```
Random_Sensitive_Object_1__c r;  
wRandom_Sensitive_Object_1 wR;
```

```
    wR.Sensitive_Number = r.Sensitive_Number__c;
```

Visualforce:

```
<apex:OutputText value="{!r.Sensitive_Number__c}" /> <!-- FLS RESPECTED -->  
<apex:OutputText value="{!wR.Sensitive_Number}" /> <!-- FLS IGNORED -->
```

Trivia!

“We showed you how to respect FLS read permissions in Apex. Which one of the following would allow you to respect the FLS read permission in Visualforce?”

1. `Rendered="{!$ObjectType.CustomObject__c.fields.CustomField__c.isAccessible}"`
2. `Rendered="{!$ObjectType.CustomObject__c.CustomField__c.isAccessible()}"`
3. `Rendered="{!$ObjectType.CustomObject__c.fields.CustomField__c.Accessible}"`
4. `Rendered="{!$ObjectType.CustomObject__c.CustomField__c}"`

Trivia (answered)!

“We showed you how to respect FLS read permissions in Apex. Which one of the following would allow you to respect the FLS read permission in Visualforce?”

1. `Rendered="{ !$ObjectType.CustomObject__c.fields.CustomField__c.isAccessible}"`
2. `Rendered="{ !$ObjectType.CustomObject__c.CustomField__c.isAccessible()}"`
3. **`Rendered="{ !$ObjectType.CustomObject__c.fields.CustomField__c.Accessible}"`**
4. `Rendered="{ !$ObjectType.CustomObject__c.CustomField__c}"`

Sharing

Sharing

Object	Default Access	Grant Access Using Hierarchies
Lead	Private	✓
Account, Contract and Asset	Public Read Only	✓
Contact	Public Read Only	✓
Opportunity	Private	✓
Case	Private	✓
Campaign	Private	✓

What is Sharing? Record Level Access!

- » Controlled outside the profile via Org-Defaults, Roles, Ownership, and sharing rules
- » Dictates which records of an object a user can see

Sharing for Developers

- Apex classes do not enforce sharing (by default)
 - Why? System Context
- Visualforce pages do not enforce sharing
 - Rely on controller for record access
- Exception: standard controllers enforce sharing

Enforcing Sharing in Apex

Use the “With Sharing” keywords.

- Default is without sharing
- Invoked classes respect defined sharing. If no sharing is defined, they inherit sharing from the invoking parent

```
1  Public with sharing Class MyController {  
2    //... With Sharing is Applied ...  
3    Public without sharing Class MyInnerClass {  
4      // ... Sharing is not applied to this class ...  
5    }  
6  }
```

Demo: Sharing

Sharing Behavior Recap

	No sharing	Without sharing	With sharing
Inner method (no sharing)	All	All	Shared
Inner class (no sharing)	All	All	Shared
Inner class without sharing	All	All	All
Inner class with sharing	Shared	Shared	Shared
External class (no sharing)	All	All	Shared
External class without sharing	All	All	All
External class with sharing	Shared	Shared	Shared

Trivia!

In the code snippet below the class is defined without sharing and it queries the private account object. Assume the running user has no visibility to any account records. When invoking this class via the developer console, does the running user see any accounts? Explain why!

```
1 public without sharing Class queryPrivate {  
2     public List<account> a_list;  
3     a_list =[select name from account limit 1];  
4     system.debug(a_list);  
5 }
```

Trivia (answered)!

The developer console runs in user context, so sharing will be respected even if you call a class that is explicitly defined as without sharing. Fun!

```
1 public without sharing Class queryPrivate {  
2     public List<account> a_list;  
3     a_list =[select name from account limit 1];  
4     system.debug(a_list);  
5 }
```

Recap - Basics

FLS

Sharing

	A	B	C	D	E
	First Name	Last Name	Title	Mailing Street	Mailing City
1					
2	Lenny	Howe	Lawyer	2320 Sand Hill Road	Palo Alto
3	Aidan	Plante	Sr. Sales Director	12 Spear Street	San Francisco
4	Laurie	Darby		22 5th Ave	New York
5	Jim	Steele	VP Sales	421 Main Street	Palo Alto
6	Gavin	Fontana		P.O. Box 420	Minneapolis
7	Andy	Rother		1132 Westchester Ave	White Plains
8	Jason	Price	Sr. Sales Director	1445 Lawton Lane	
9	Felix	Frye		1111 Westrun Blvd	White Plains
10	Sophie	Kostos	Purchasing Rep	1515 Broadway	New York
11	Paul	Huxtable	District Manager	P.O. Box B-740	Shamburg
12	Francis	Buchner	Account Manager	P.O. Box A-455	Shamburg
13	Mandy	Hall	Account Executive	122 Chestnut Street	San Francisco
14	Jay	Price	Sales Supervisor	2455 Paces Ferry Road	Atlanta
15	Jack	Fallon		Forest Road	Middlesex
16	Rick	Lykor		Reisholzer Werftstrasse 38-42	Duesseldorf
17	George	Fiss	Sr. Sales Director	1111 Westchester Ave	White Plains
18	Robert	Stamps	Account Executive	Postfach 2103	Weisbaden
19	Zoe	Kramer	VP of Sales	100 Abbott Park Rd.	Abboot Park
20	Edward	Stamos	President and CEO	10 Main Rd.	New York City
21	Leanne	Tomlin	VP Customer Support	10 Main Rd.	New York
22	Jen	Jacobs		101 California Street	San Francisco

CRUD

Recap – Developer Tools

Here are the developer methods we covered for respecting authorization:

1. CRUD

- Apex does not respect CRUD. Visualforce with a standard controller does respect CRUD
- Use `Account.sObjectType.getDescribe().isAccessible()` to enforce CRUD in Apex

2. FLS

- Visualforce respect FLS for sObjects, Apex does not
- Use `Schema.sObjectType.Account.fields.Name.isAccessible()` to enforce FLS in Apex
- Use `rendered="{!$ObjectType.CustomObject__c.fields.CustomField__c.Accessible}"` to enforce in VF

3. Sharing

- By default, Apex does not respect sharing
- Use “with sharing” in the class definition to enforce sharing in Apex
- Best practice: Make all classes with sharing, make exceptions inner methods defined as without sharing

Additional Resources

- Secure Coding Guidelines - https://developer.salesforce.com/page/Testing_CRUD_and_FLS_Enforcement
- CRUD & FLS Enforcement Guide - https://developer.salesforce.com/page/Enforcing_CRUD_and_FLS
- Salesforce StackExchange - <http://salesforce.stackexchange.com/questions/tagged/security>
- Developer.Salesforce.com Security Forum - <https://developer.salesforce.com/forums> (full link hidden)
- Security Office Hours (Partners) - <http://security.force.com/security/contact/ohours>
- Security Implementation Guide - <https://developer.salesforce.com/././securityImplGuide/> (full link hidden)

Slides + Demo

- Get Slides Here:
 - DF Chatter Group – [Link Here](#)
 - @kylekyle Twitter – <https://www.twitter.com/kylekyle>
- Want to play with our demo code?
 - Dreamforce Demo Trial Signup: <https://security.secure.force.com/DFtrialsignup>

Secure Development Sessions

Secure Coding: Field-level Security, CRUD, and Sharing

Monday, October 13 @ 11:00 a.m. - 11:40 a.m.

Secure Coding: Storing Secrets in Your Salesforce Instance

Monday, October 13 @ 2:00 p.m. - 2:40 p.m.

Building Secure Mobile Apps

Monday, October 13 @ 5:00 p.m. - 5:40 p.m.

Protect Your Data Against Malicious Scripts

Tuesday, October 14 @ 11:00 a.m. - 11:40 a.m.

Secure Coding: External App Integration

Wednesday, October 15 @ 9:00 a.m. - 9:40 a.m.

Secure Coding: SSL, SOAP, and REST

Thursday, October 16 @ 10:30 a.m. - 11:10 a.m.

dreamforce



Announcements:

Force.com Code Scanner now supports Salesforce1 and JavaScript! Try it here:
<http://bit.ly/SF1Scanner>

Chimera Web App Scanner alpha nominations are open. Partners apply at:
<http://bit.ly/SFChimera>

Live security office hours are available in the Partner Zone.

salesforce

Q&A



Thank You