Agenda

1. Salesforce SDK
2. Getting Started
3. Customization

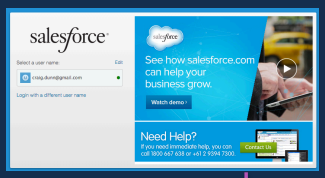
Xamarin University

Salesforce SDK

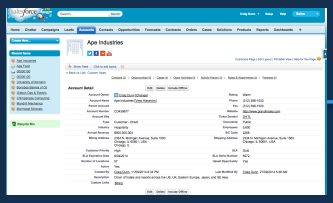
Xamarin University

Salesforce
SaaS Architecture

- Software-as-a-Service
- Hosted by Salesforce
- Web interface



Authenticate



Salesforce

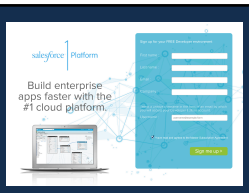
Review & edit data

Xamarin University

Salesforce SDK
Connected App Architecture

- Setup a 'Connected App'
- Embed 'secret key'
- Use OAuth config
- REST communication with server
- Real-time updates on Salesforce.com

Sign-up & set-up



Secret key

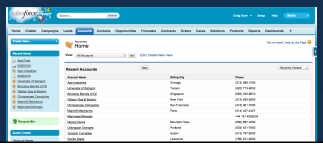
Authenticate

Request the data

Retrieve the data

Salesforce

Review & edit data



Xamarin University

OAuth 2.0
Authentication protocol

- Client ID
- Client Secret
- Redirect URL

Xamarin University

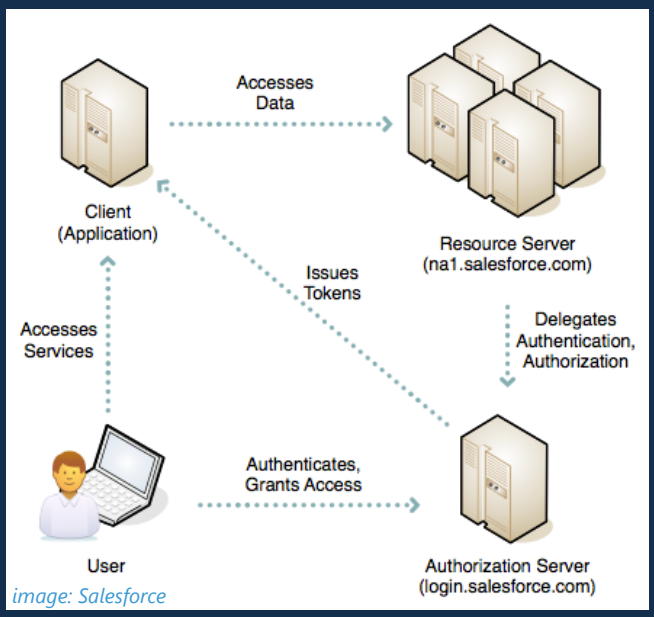


image: Salesforce

OAuth 2.0
Authentication protocol

- Client ID
- Client Secret
- Redirect URL

Xamarin University

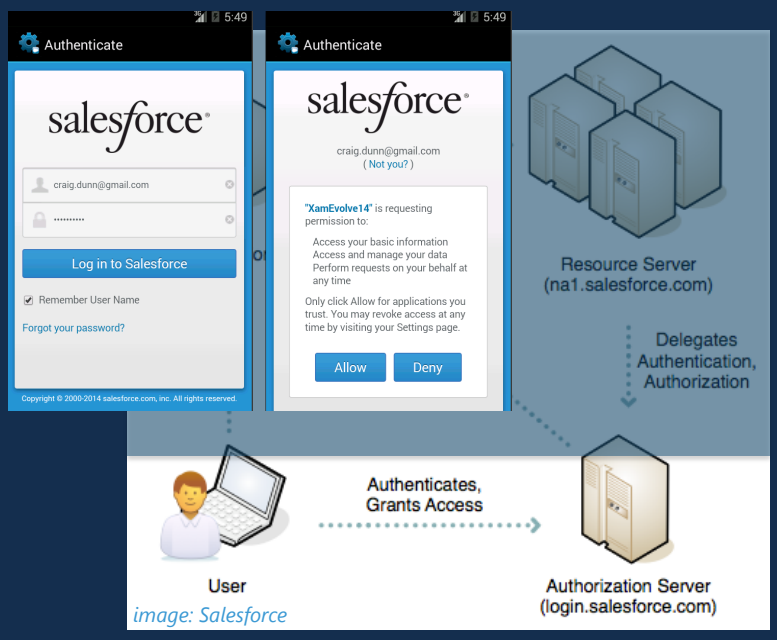


image: Salesforce

REST
Representational State Transfer

- Resources represented by URIs
- HTTP verbs GET POST PUT DELETE (PATCH)
- HTTP status codes
- HTTP headers for authentication
- JSON payloads

Xamarin University

http://en.wikipedia.org/wiki/Representational_state_transfer

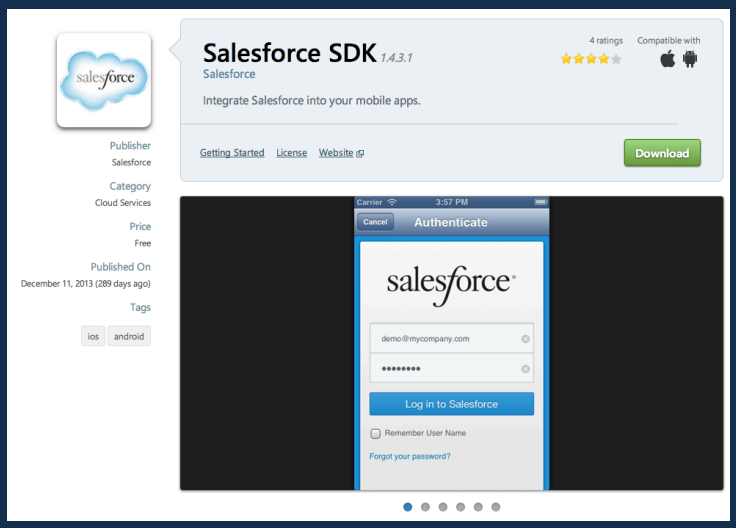
REST
When to use REST?

API Name	What It's For	When to Use It	Protocol	Data Format
REST API	Accessing objects in your organization using REST.	You want to leverage the REST architecture to integrate with your organization. No WSDL requirement. Well-suited for browser-based applications, mobile apps, and highly-interactive social applications.	REST	JSON, XML

Xamarin University

Client Libraries
Salesforce SDK Component

- Xamarin Component Store
  - iOS
  - Android



Xamarin University

Client Libraries
Salesforce SDK Component

- Create, Update, and Delete `SObjects`
- Query data with SOQL
- Search with SOSL
- Consistent C# API across iOS & Android

Xamarin University

Getting Started

Xamarin University

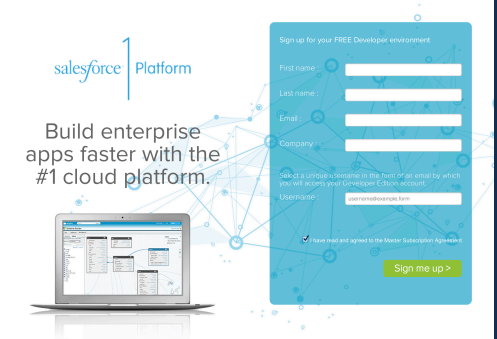
developer.salesforce.com

- Sign-up for a free account
- Configure a 'connected app'
- Run Salesforce SDK sample

Xamarin University

Sign-up

- It's easy for anyone to sign up as a developer
- If your organization already uses Salesforce, use a different email address for playing around

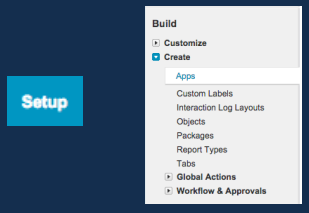


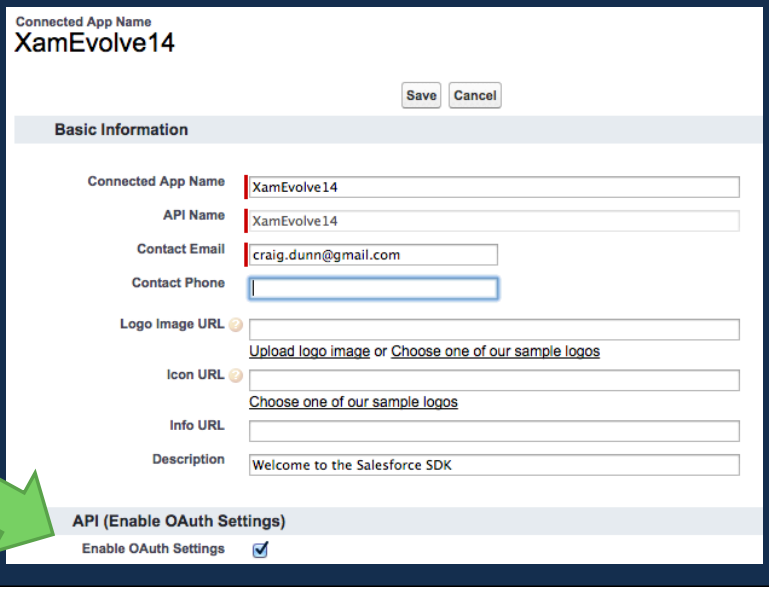
http://developer.salesforce.com/

Xamarin University

Connected App

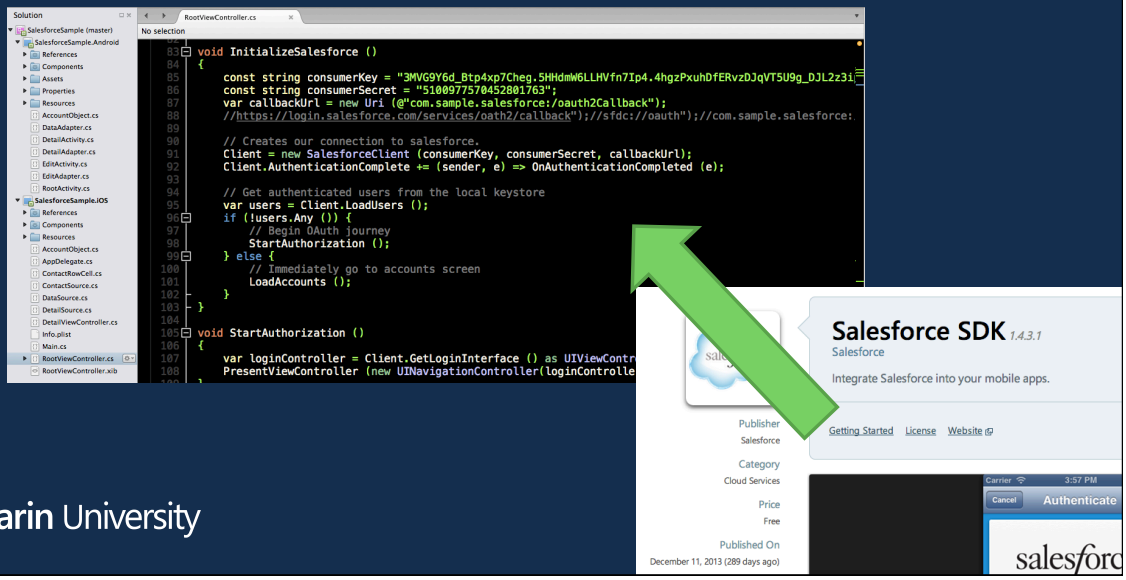
- Create new Connected App





Xamarin University

Getting Started



Xamarin University

Getting Started
Connected App settings

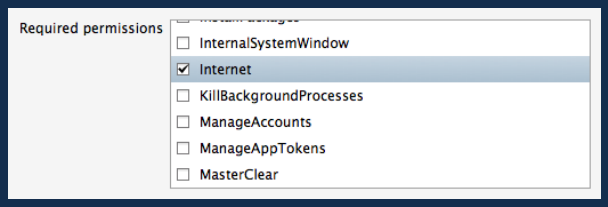
API (Enable OAuth Settings)

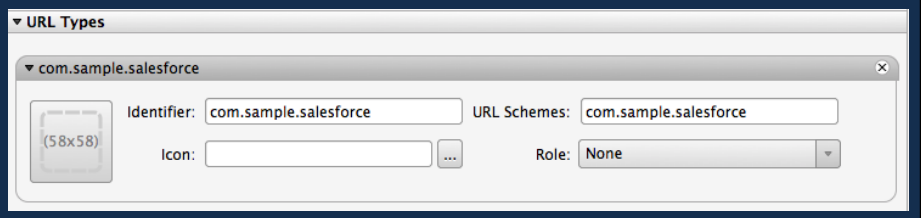
Consumer Key	3MVG9Y6d_Btp4xp7Cheg.5HHdmW6LLHVfn7Ip4.4hgzPxuhDfERvzDJqVT5U9g_DJL2z3ipo9YXNIxdIwmEhb	Consumer Secret	5100977570452801763
Selected OAuth Scopes	Access your basic information (id, profile, email, address, phone) Access and manage your data (api) Perform requests on your behalf at any time (refresh_token, offline_access)	Callback URL	com.sample.salesforce:/oauth2Callback

```
const string consumerKey = "3MVG9Y6d_Btp4xp7Cheg.5HHdmW6L
const string consumerSecret = "5100977570452801763";
var callbackUrl =
    new Uri (@"com.sample.salesforce:/oauth2Callback");
```

Xamarin University

Connected App
Android and iOS configuration





Xamarin University

Getting Started
Connect and authenticate

```
// Creates our connection to salesforce.
Client = new SalesforceClient (consumerKey, consumerSecret, callba

// Get authenticated users from the local keystore
var users = Client.LoadUsers ();

if (!users.Any ())
{
    client.AuthenticationComplete += (sender, e) => OnAuthenticati

    // Starts the Salesforce login process.
    var loginUI = client.GetLoginInterface ();
    DisplayThe (loginUI); // PLATFORM SPECIFIC !
```

Xamarin University

Getting Started
Load Account data

```
}
else
{   // We're ready to fetch some data!
    // Let's grab some sales accounts to display.
    IEnumerable<SObject> results =
        await client.ReadAsync (
                "SELECT Id, Name, AccountNumber FROM Account");

    DoSomethingAmazingWith (results); // SObjects
}
```

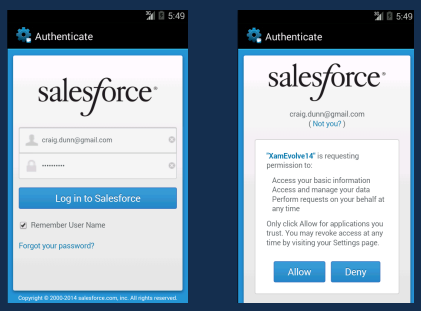
Xamarin University

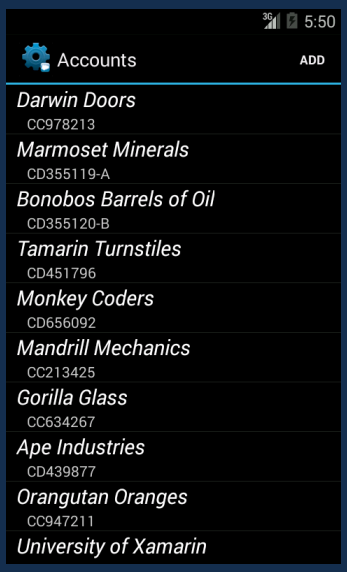
Demo

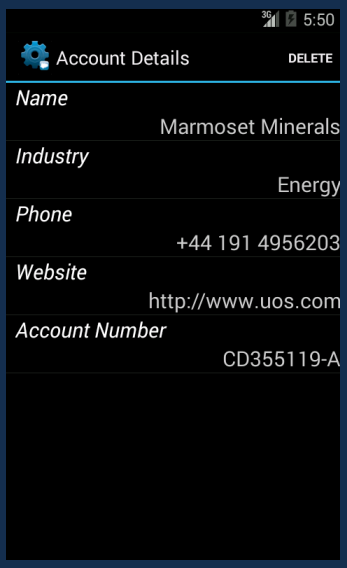
Xamarin University

Getting Started

Works on Android too ;)







Xamarin University

Customization

Xamarin University

Customization

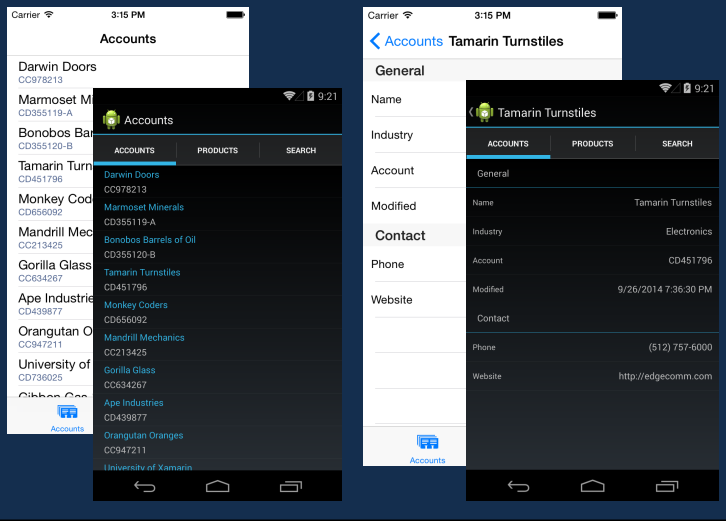
- Xamarin.Forms
- Custom SObjects & Queries
- Editing
- Search

Xamarin University

Xamarin.Forms

Using a component with Xamarin.Forms

- Xamarin.Forms
  - iOS & Android
- Shared Project
  - #if directives for OAuth
- Use the same credentials as the previous sample



Xamarin University

Queries
Getting data out of Salesforce with SOQL

- SELECT `Id` in queries

```
"SELECT Id, Name, ... "
```

- FROM objects in Salesforce

```
"SELECT Id, Name, ... FROM Account"
```

- WHERE, IN, LIKE '%' for filtering

```
"SELECT Id, Name, ... FROM Account
 WHERE Id = 'asdf' "
```

- ORDER BY, GROUP BY, HAVING
- LIMIT, OFFSET for paging

Xamarin University

Queries
Getting data out of Salesforce

- `Client.QueryAsync`

```
IEnumerable<SObject> response = await Client.QueryAsync
          (@"SELECT Id, Name, AccountNumber, Phone, Website,
              Industry, LastModifiedDate FROM Account");

response.Select (s => s.As<AccountObject> ()).ToList();
```

Xamarin University

Custom SObjects

Creating SObject classes

- Subclass `SObject`
  - ResourceName getter
  - PreparingForUpdate event
  - GetOption and SetOption

```
public class ProductObject : SObject
{
    public ProductObject ()
    {
        this.PreparingUpdateRequest += Prep
    }
    public override string ResourceName {
        get { return "Product2"; }
    }
    public string Name
    {
        get { return GetOption ("Name"); }
        set { SetOption ("Name", value); }
    }
    // remaining fields...
```

Xamarin University

Editing SObjects

Modifying SObjects on the server

```
// Create
string newId = await Client.CreateAsync (mySObject);

// Update
await Client.UpdateAsync (mySObject);

// Delete
bool wasDeleted = await Client.DeleteAsync (mySObject);
```

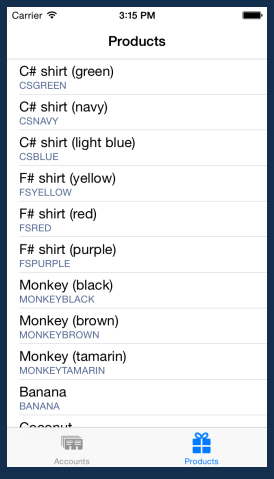
Xamarin University

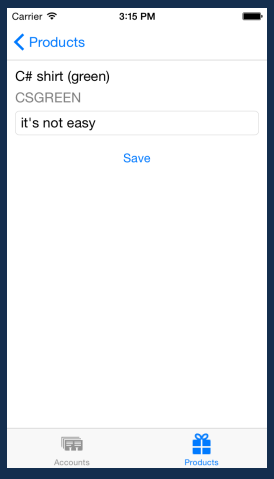
Demo

Xamarin University

Editing SObjects

Modifying SObjects on the server





Xamarin University

Search
Search using SOSL

- FIND using {} to contain the search term
- IN Salesforce-defined groups
- RETURNING specifies objects and optionally fields
- * and ? wildcards

```
"FIND {query}"
```

```
"FIND {query} IN NAME FIELDS"
```

```
"FIND {query}
IN NAME FIELDS
RETURNING Account (Id, Name, AccountNu
Website, Industry, LastModifiedDate)"
```

Xamarin University

Search
Search using SOSL

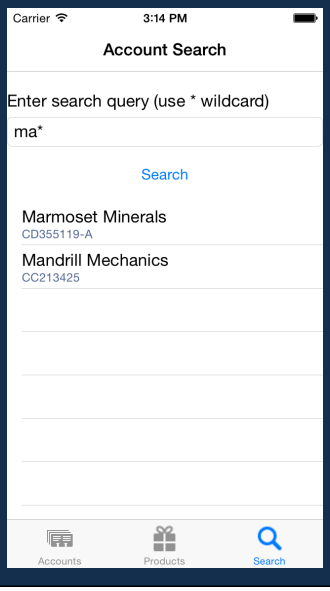
```
var sosl =
    " FIND {" + query + "} "+
    " IN NAME FIELDS "+
    " RETURNING Account ";
IEnumerable<SearchResult> searchResponse =
                await Client.SearchAsync (sosl);
// HACK: parse out all the Id values into a string

// now pull the 'found' SObjects
var soql =
    "SELECT Id, Name, AccountNumber, Phone, Website,
    "FROM Account "+
    "WHERE Id IN (" + ids + ")";

IEnumerable<SObject> queryResponse =
                await Client.QueryAsync (soql);
```

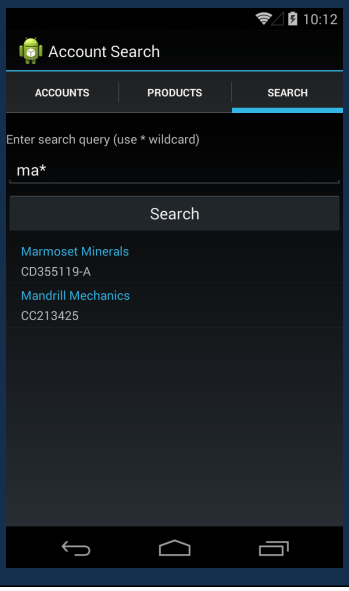
Xamarin University

Demo

Xamarin University

Search
Search using SOSL





Xamarin University

Summary

- Salesforce is easily integrated into Xamarin apps
- Custom SObjects are easy to create, edit, save and search
- Works in Xamarin.Forms!

Xamarin University

Xamarin Evolve 2014

Integrating with Salesforce

Craig Dunn
craig@xamarin.com

Xamarin University