Creating Lightweight Integrations with the Force.com REST API

Richard Seroter
VICE PRESIDENT OF PRODUCT, CENTURYLINK

@rseroter www.seroter.com

Overview

- Anatomy of a REST API call
- Authenticating users
- Explaining REST objects, actions
- HTTP response codes
- Using conditional requests
- Defining composite calls
- Building custom REST services
- Summary

REST API

Simple access to Salesforce data and functionality via RESTful endpoints.

```
POST
  /services/data/v35.0/sobjects/Voter__c
  HTTP/1.1
Host: na11.salesforce.com
Authorization: Bearer 00DA0...
Content-Type: application/json


{
  "Name": "Jean-Ralphio Saperstein",
  "Precinct__c": "a06G000000jvq9z“
}
```

HTTP verbs

Resource identified in URI

Secured via OAuth Bearer Token

JSON or XML payloads

When Do You Use It?

Integrating with mobile apps

Want lightweight JSON interactions

Hypermedia is useful to your app

Authenticating REST API Users

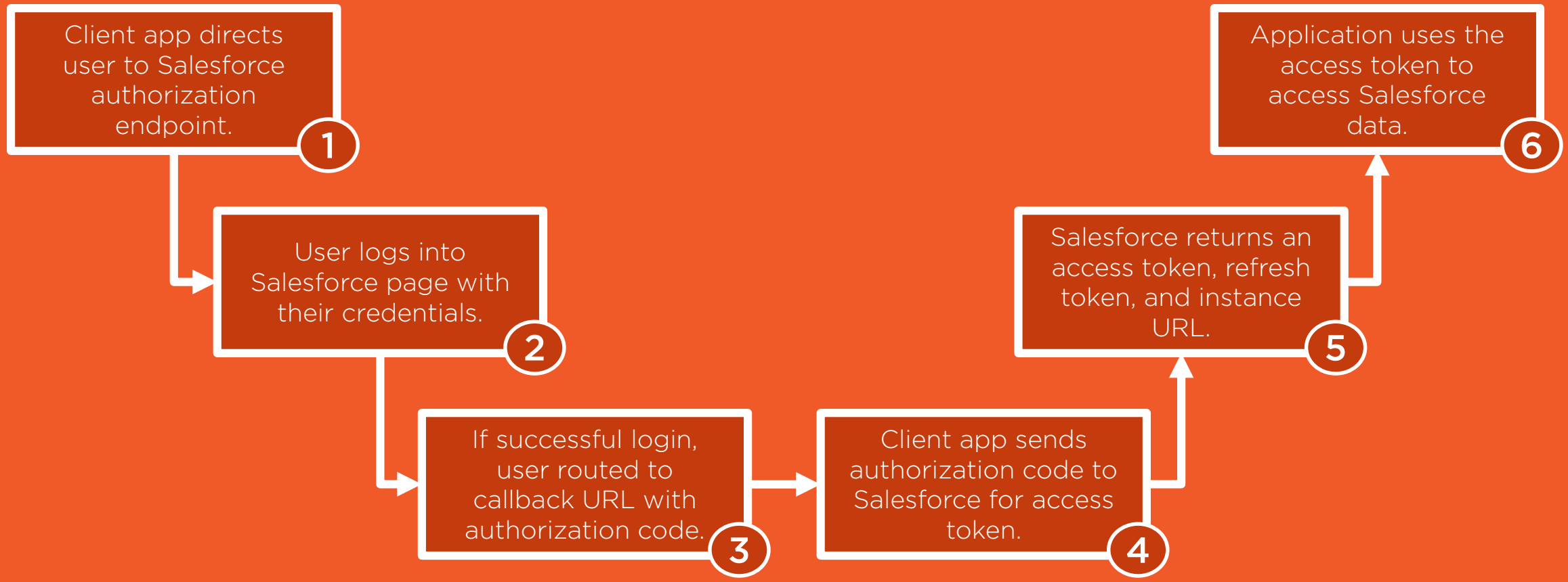
Define Connected App

User, object, field security applies

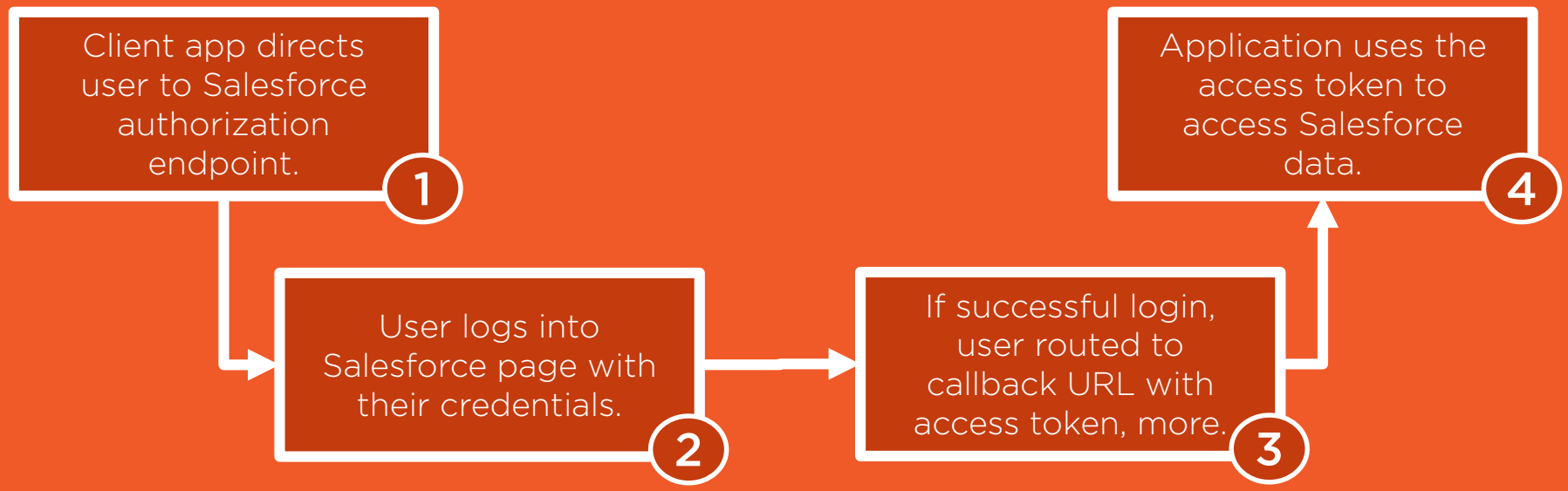
OAuth security model

Six authentication flows available

OAuth Flow: Web Server Flow

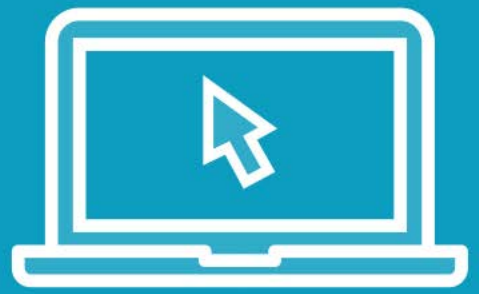


OAuth Flow: User-agent Authentication Flow



OAuth Flow: Username-Password Flow

Demo



Create a Connected App

View OAuth settings

API login with username/password flow

Review API results

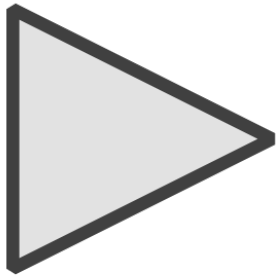
What Are REST Objects?

Same data model and objects as in SOAP API

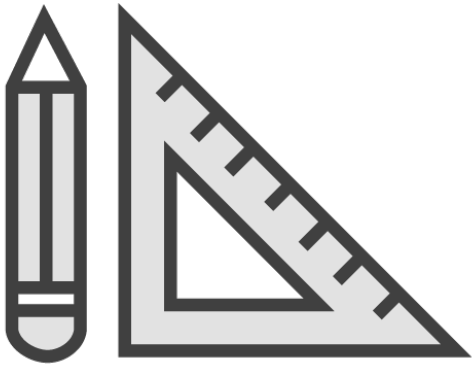
Records are like database rows

Standard, custom, external objects supported

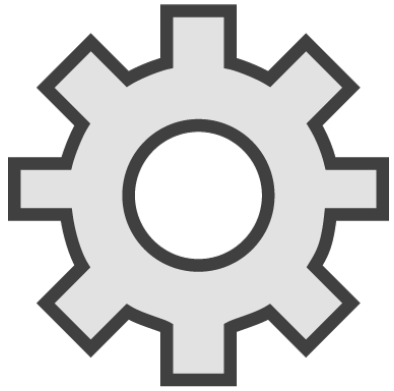
Types of REST API Calls via Standard Interface



Core calls



Describe calls



Utility calls

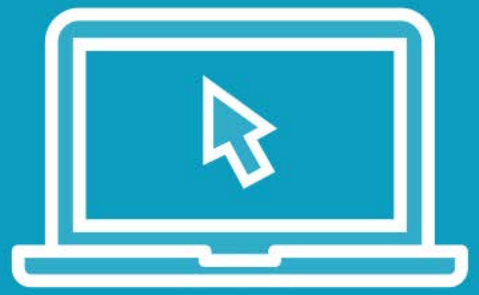
```
GET
/services/data/v35.0/sobjects/Account/001A000000YB
X3CIAX.xml

 Host: na11.salesforce.com

Authorization: Bearer 00D...

----------

GET
/services/data/v35.0/sobjects/Account/001A000000YB
X3CIAX

Host: na11.salesforce.com

Authorization: Bearer 00D...

Accept: application/xml
```

- Append .xml or .json to URI
- Add “accept” header

REST API Response Codes

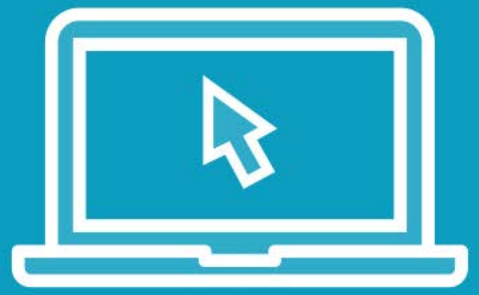
HTTP code	Industry response	Salesforce meaning
200 – OK	"The request has succeeded"	Success for GET or HEAD requests
201 – Created	"The request has been fulfilled and resulted in a new resource being created"	Success for POST requests
204 – No Content	"The server has fulfilled the request but does not need to return an entity-body"	Success for DELETE requests
300 – Multiple Choices	"The requested resource corresponds to any one of a set of representations"	ID exists in more than one record, so all matching records are returned
304 – Not Modified	"a conditional GET request and access is allowed, but the document has not been modified"	Content hasn't changed since a specific time and date
400 – Bad Request	"The request could not be understood by the server due to malformed syntax"	Request not understood, usually because JSON/XML body has an error
401 – Unauthorized	"The request requires user authentication"	Session token has expired or in invalid
403 – Forbidden	"The server understood the request, but is refusing to fulfill it"	Logged in user likely doesn't have necessary permissions
404 – Not Found	"The server has not found anything matching the Request-URI"	Record or object not found
500 – Internal Server Error	"The server encountered an unexpected condition which prevented it from fulfilling the request"	Error occurred within Force.com itself

Demo



- Call login API to get access token
- Retrieve records for a standard object
- Retrieve records for a custom object
- Perform SOQL query
- Switch results between XML and JSON

Demo



Set credentials in the sample app

Review precinct.js and its code

Test application

Client-side Caching with Conditional Requests

If using “Account” object …

If-Match header

If-None-Match header

Requires use of an Etag(s)

If using other objects …

If-Modified-Since header

If-Unmodified-Since header

Works against individual records

Maximize Round Trips with Composite Calls

Batch up a set of operations

Create nested records

Create set of unrelated records

```
POST /services/data/v35.0/composite/batch/ HTTP/1.1
Host: na11.salesforce.com
Content-Type: application/json
Authorization: Bearer 00DA...


{
 "batchRequests" : [
    {
          "method" : "GET",
          "url" :
"v35.0/sobjects/Voter__c/a00G000000JdWIAV"
    },{
          “method" : "GET",
          "url" : "v35.0/sobjects/Precinct__c/a06G000q9zIAA"
    },{
          "method" : "GET",
          "url" : "v35.0/sobjects/Voter_Donation__c/
a077JIAW?fields=Name,Candidate_Name__c"
    }]
}
```

Batch resource URI

Independent operations

Use SObjects, query, search

No context between requests

Executed serially

```
POST /services/data/v35.0/composite/tree/Voter__c HTTP/1.1
Host: na11.salesforce.com
Content-Type: application/json
Authorization: Bearer 00DA...

{
 "records": [{
          "attributes": {"type": "Voter__c", "referenceId":
"ref1"},
          "Name": "Leslie Knope",
          "Precinct__c": "a06G0000vq9zIAA",
          "Voter_Donations__r": {
            "records": [{
                    "attributes": {"type": "Voter_Donation__c",
"referenceId": "ref2"},
                    "Amount__c": 500
             }]
           }
 }]
}
```

Tree and SObject URI

Root record matches URI SObject

Nested relationship or master/detail record type

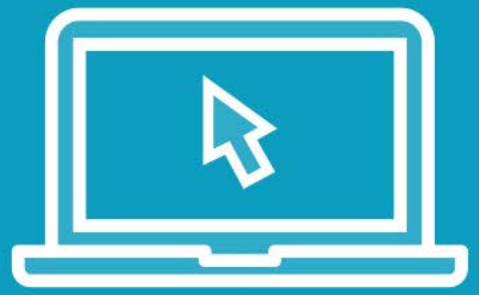
```
POST /services/data/v35.0/composite/tree/Voter__c HTTP/1.1
Host: na11.salesforce.com
Content-Type: application/json
Authorization: Bearer 00DA...

{
 "records": [{
          "attributes": {"type": "Voter__c", "referenceId":
"ref1"},
     "Name": “Leslie Knope",
          "Precinct__c": "a06G0000009zIAA"
   },
   {
          "attributes": {"type": "Voter__c", "referenceId":
"ref2"},
          "Name": “Ron Swanson",
          "Precinct__c": "a06G000jvq9zIAA"
    }
  ]
}
```

Tree and SObject URI

Array of same record type

“All or nothing” transaction

Demo



Issue batch query

Create multiple records via SObject tree

Create nested records

Create multiple records via Batch

```
GET /services/apexrest/VotersWithDonation
  /a00G00T9Jdg HTTP/1.1
Host: na11.salesforce.com
Authorization: Bearer 00DA...
Accept: application/json

----------------

@RestResource(urlMapping='/VotersWithDonation/*')
global class CustomVoterDonationService {

    global class VoterDonation {

    String VoterName {get; set;}
    String VoterParty {get; set;}
     List<Voter_Donation__c> Donations {get; set;}
    }


    @HttpGet
    global static VoterDonation GetVoter()
    {
        //get voter ID from request
        RestRequest request = RestContext.Request;
        String voterId = request.requestURI.substring(
request.RequestURI.lastIndexOf('/')+1);
        Voter__c voter = [SELECT ID, Name FROM Voter__c
WHERE ID = :voterId];

        VoterDonation vdon = new VoterDonation();
        vdon.VoterName = voter.Name;

        vdon.Donations = [SELECT Candidate_Name__c, Amount__c, Donation_Dat
e__c FROM Voter_Donation__c
WHERE Voter__r.ID = :voterId ];

        return vdon;
    }

}
```

Special URL

Session ID or OAuth security

JSON or XML representation

Wildcards supported

Global class required

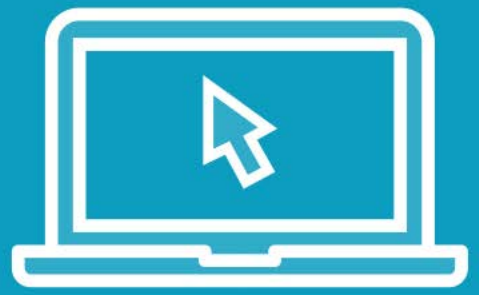
Can use user defined classes

GET, PATCH, POST, PUT, DELETE annotations

RestContext, request, response

Operations use system context

Demo



- Create custom class
- Define operation that returns aggregate data
- Decorate with REST annotations
- Test from Postman
- Consume from Node.js application

Summary

- Overview
- Anatomy of a REST API call
- Authenticating users
- Explaining REST objects, actions
- HTTP response codes
- Using conditional requests
- Defining composite calls
- Building custom REST services