



heroku



Accessing Real-World APIs from Clojure

Pat Patterson
@metadaddy
Principal Developer Evangelist
salesforce.com

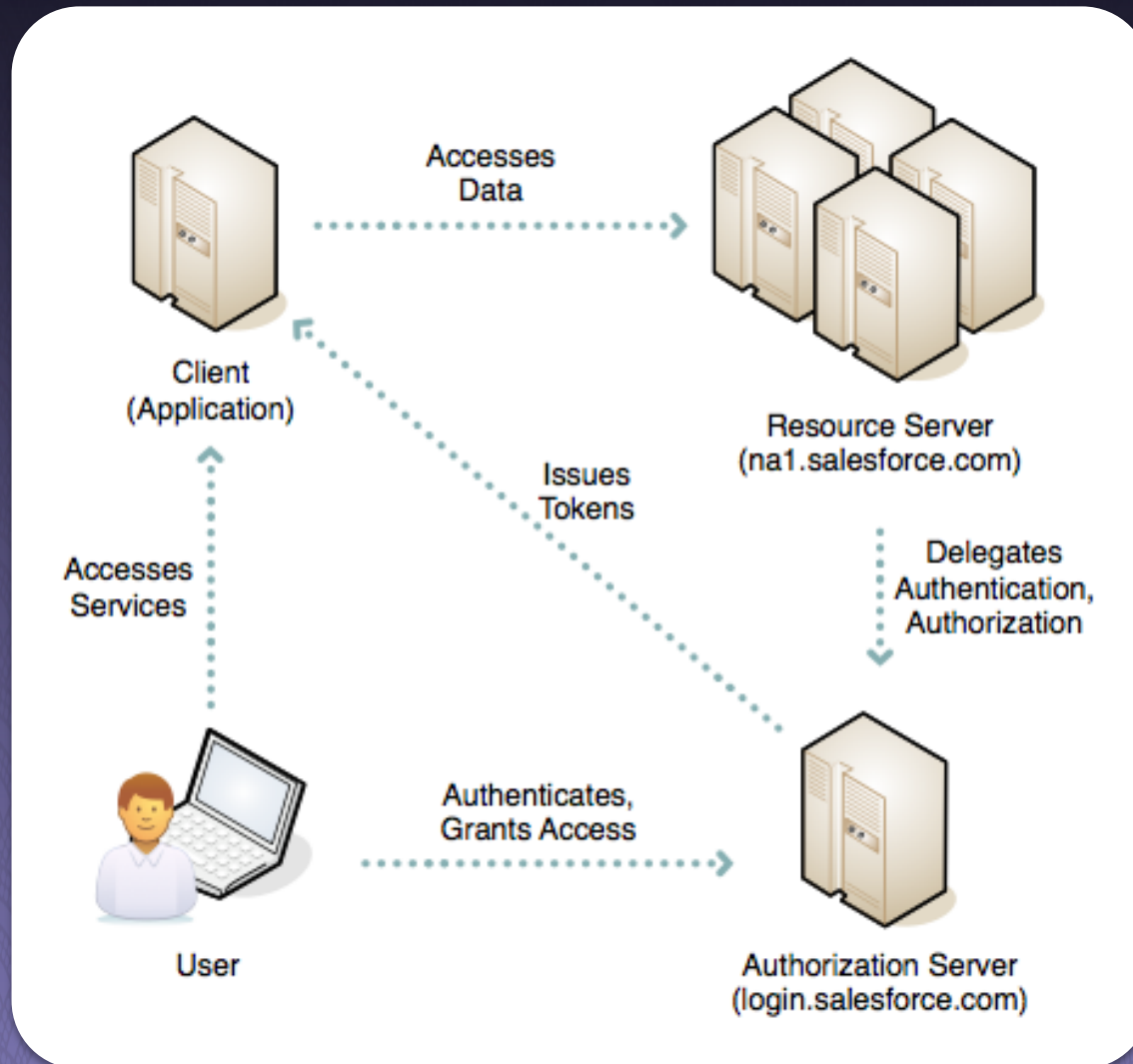
The Problem

- I'm writing a sales analysis tool in Clojure
- I want to retrieve CRM data from Salesforce via their REST API
- I don't want to handle usernames and passwords!

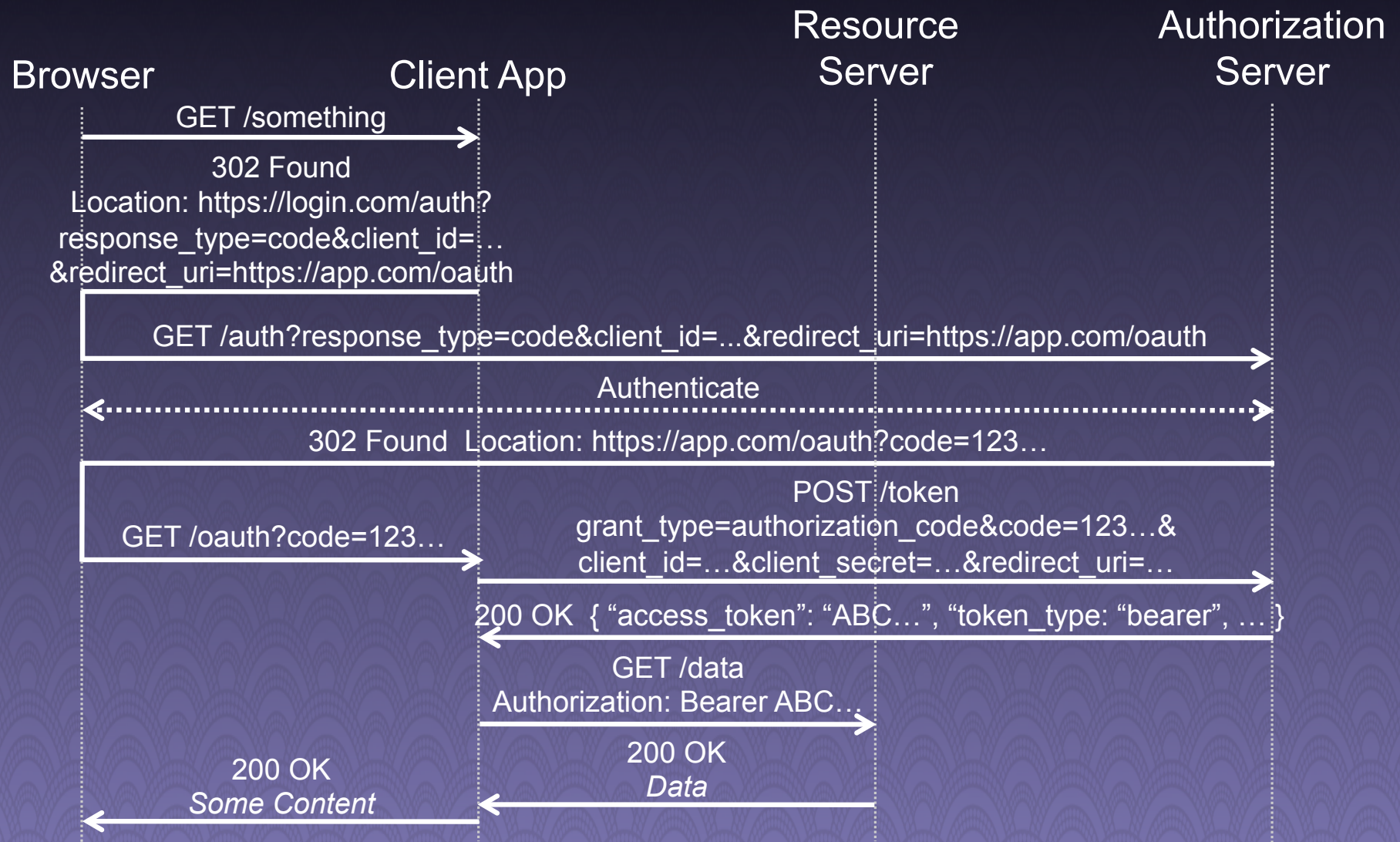
OAuth 2.0

- Open Authorization
- Allows an app to securely access an API on behalf of a user
- Currently an Internet Draft (draft 23 – Jan 21, 2012)
- Widely deployed
 - Facebook (draft 20)
 - Google (draft 10)
 - salesforce.com (draft 10)

OAuth 2.0 Roles



OAuth 2.0 Web Server Flow



clj-oauth2

- Written by Moritz Heidkamp (doo.net)
- `github.com/DerGuteMoritz/clj-oauth2`
- Initial implementation worked with Facebook (OAuth 2.0 Draft 20)

Configuration

```
(:require [clj-oauth2.client :as oauth2])

(def login-uri
  (get (System/getenv) "LOGIN_URI" "https://login.salesforce.com"))

(def force-com-oauth2
  {:authorization-uri (str login-uri "/services/oauth2/authorize")
   :access-token-uri (str login-uri "/services/oauth2/token")
   :redirect-uri (System/getenv "REDIRECT_URI")
   :client-id (System/getenv "CLIENT_ID")
   :client-secret (System/getenv "CLIENT_SECRET")
   :grant-type "authorization_code"})
```

Redirect for Auth

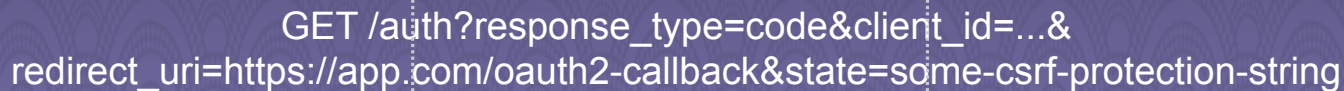
```
(def auth-req
  (oauth2/make-auth-request force-com-oauth2 "some-csrf-protection-string"))

;;;

(redirect (:uri auth-req))
```

302 Found
Location: https://login.com/auth?
response_type=code&client_id=..
&redirect_uri=https://app.com/
oauth2-callback

GET /auth?response_type=code&client_id=...&
redirect_uri=https://app.com/oauth2-callback&state=some-csrf-protection-string



Get an Access Token

```
(GET "/oauth2-callback"  
  {params :params}  
  (let [access-token  
        (oauth2/get-access-token force-com-oauth2 params auth-req)]  
    {:body (...)})))
```



Get a Resource

```
(oauth2/get  
  "https://na1.salesforce.com/services/data/v24.0/subjects/Account/0015000000VALDxAAP"  
  {:oauth2 access-token})
```

Diagram illustrating an HTTP GET request and response:

- Request:** GET /services/data/...
Authorization: Bearer ABC...
- Response:** 200 OK
Data

The diagram shows a horizontal arrow pointing right for the request and a horizontal arrow pointing left for the response, with vertical dashed lines indicating the communication boundaries.

**Can we abstract
away the
redirections here?**

Ring Middleware

```
(defroutes handler
  (GET "/test"
    {oauth2 :oauth2}
    (:body (oauth2/get
      (str
        (:instance_url (:params oauth2))
        "/services/data/v24.0/subjects/Account/0015000000VALDxAAP")
        {:oauth2 oauth2}))))))

(def app
  (-> handler
    (wrap-oauth2 force-com-oauth2)
    wrap-session
    wrap-keyword-params
    wrap-params))
```


**Can we provide a
Clojure API for the
Force.com REST
API?**

Clojure Toolkit for Force.com

```
(query "SELECT Name, Id FROM Account ORDER BY Name LIMIT 20")  
  
(create "Account" {:Name "Initech"})  
  
(retrieve "Account" (:id params))  
  
(update "Account" (:id params) {:Website "http://www.initech.com/"})  
  
(delete "Account" (:id params))
```


Conclusion

- If you're working with RESTful APIs, you need to grok OAuth 2.0
- Don't code from scratch!
 - Use `clj-oauth2`
 - Contribute to `clj-oauth2`
- Try out Heroku's Clojure support

Resources

`github.com/metadaddy-sfdc`

`/clj-oauth2`

`/Force.com-Toolkit-for-Clojure`

`developer.force.com/REST`

`www.heroku.com`