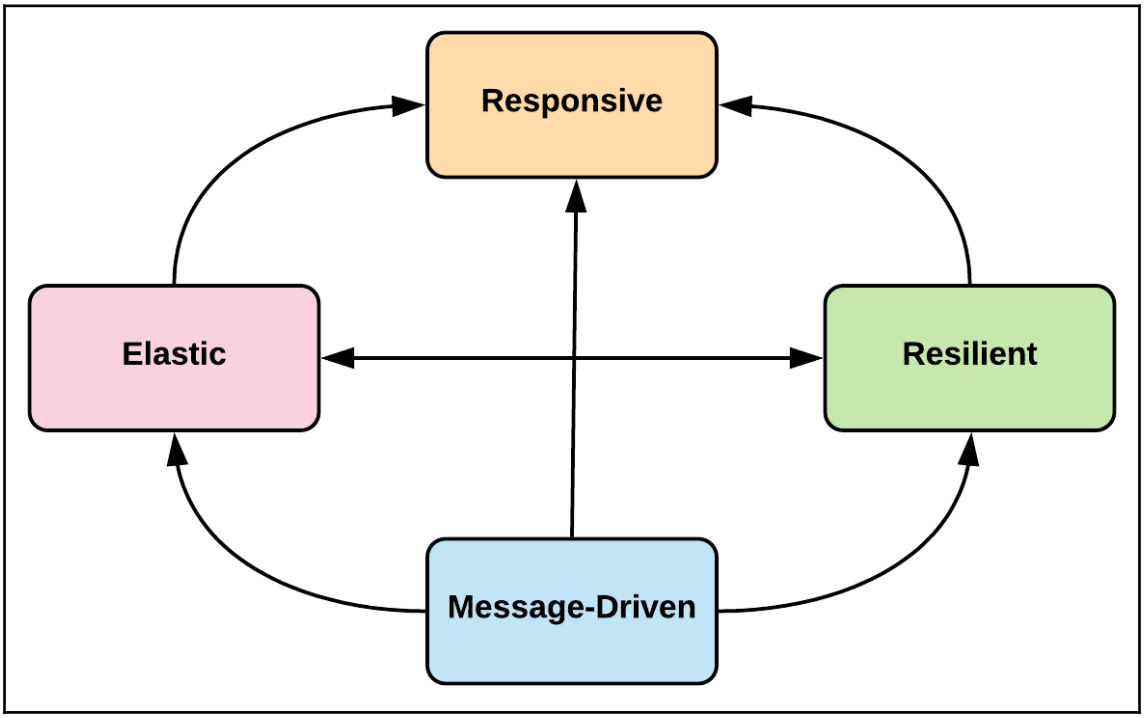
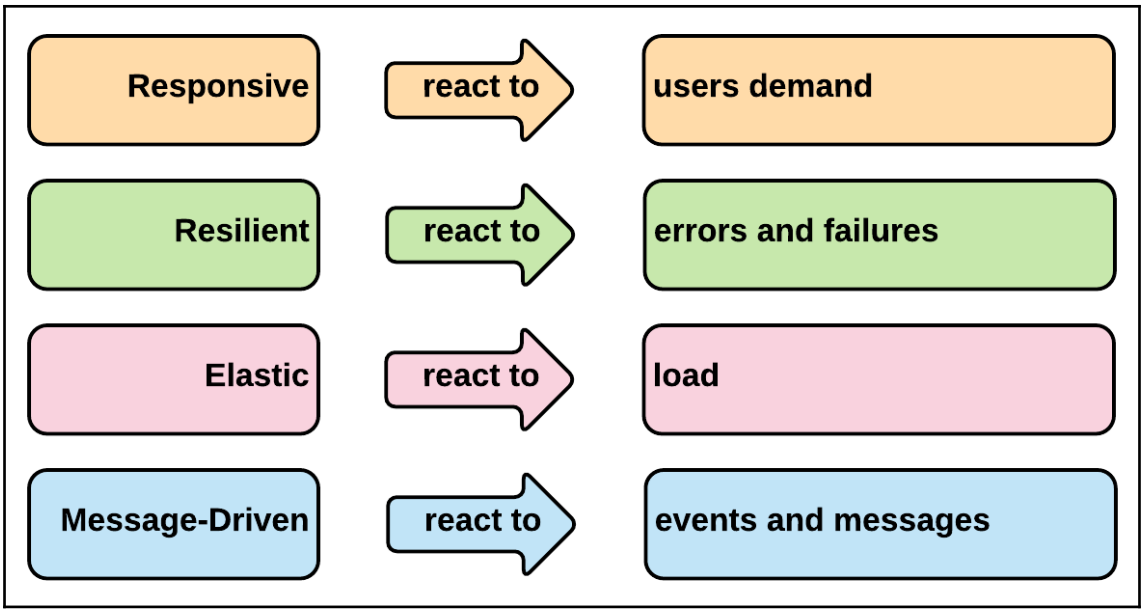
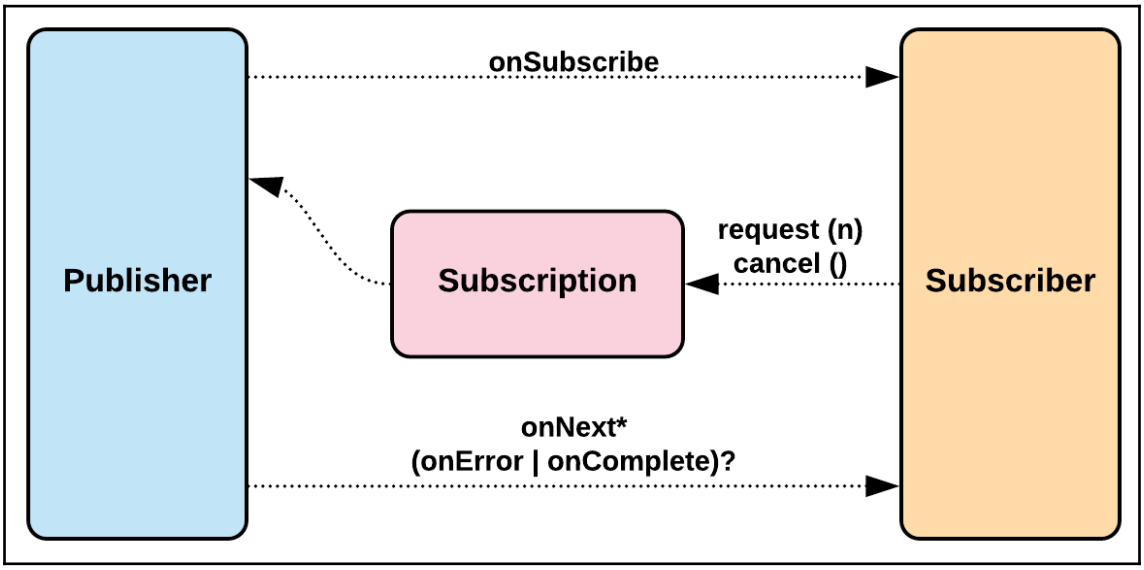
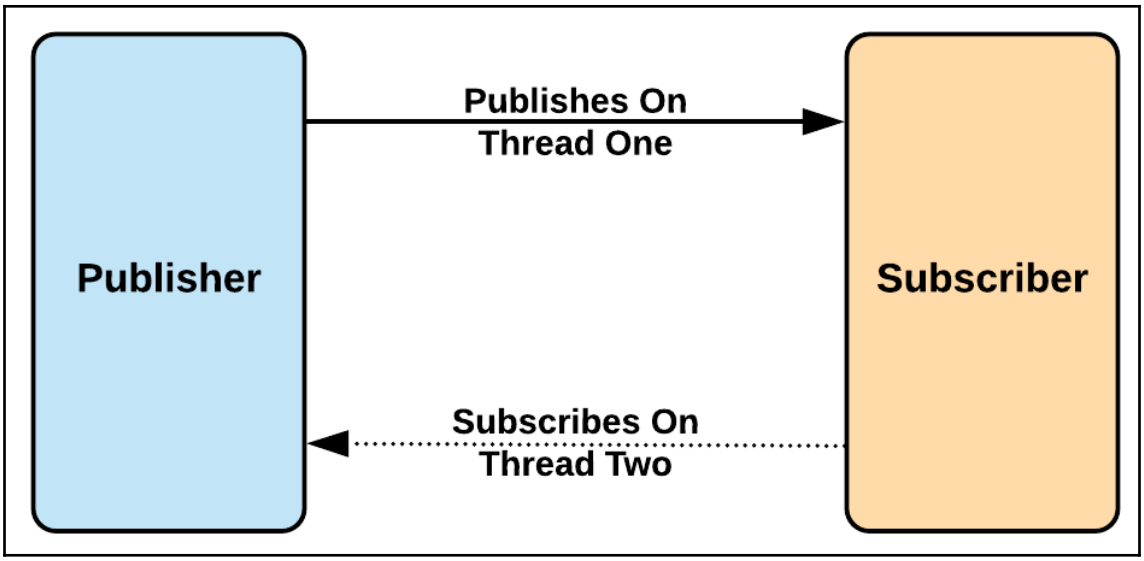
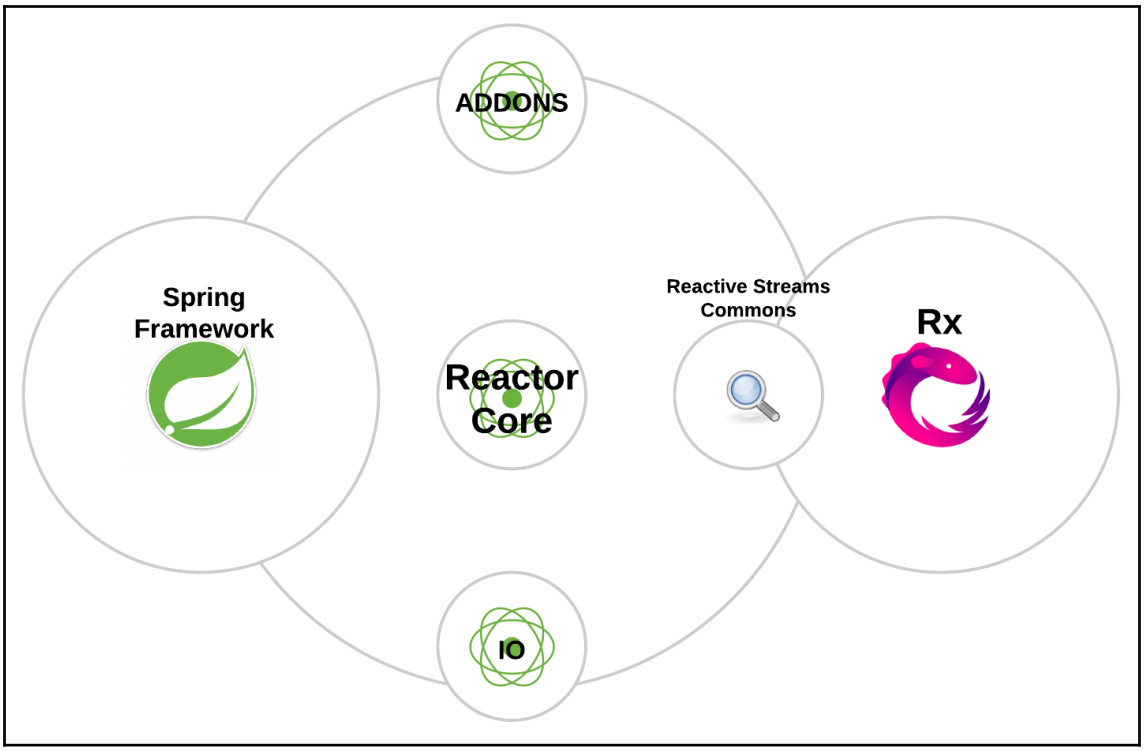
Chapter 1: Overview of Spring 5 and Spring Security 5

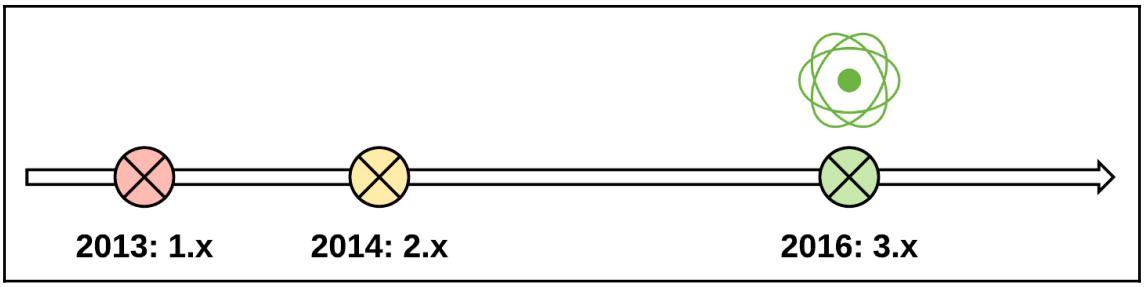


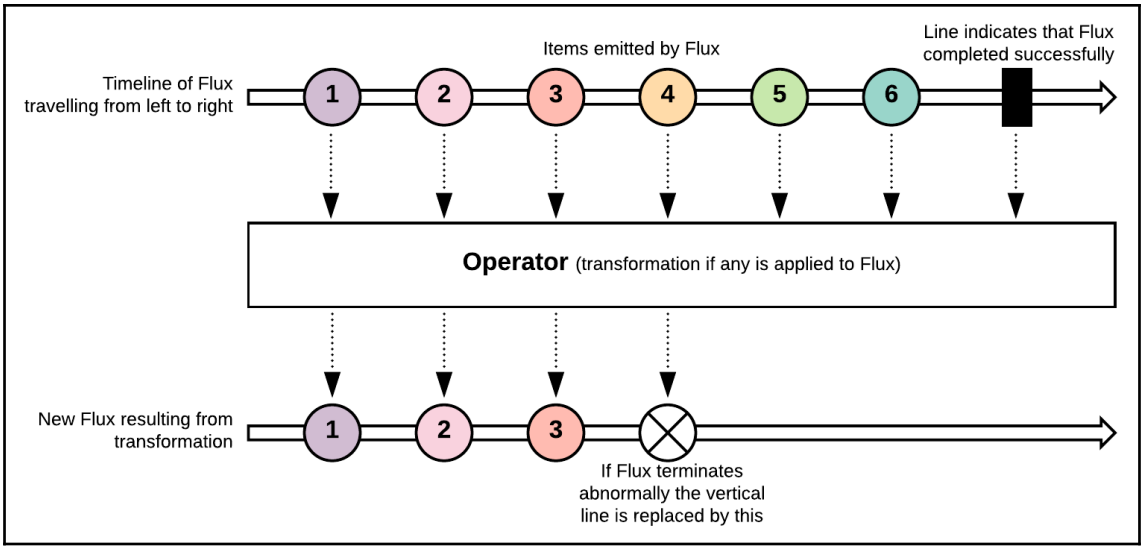

Responsive
react to
users demand
Resilient
react to
errors and failures
Elastic
react to
load
Message-Driven
react to
events and messages

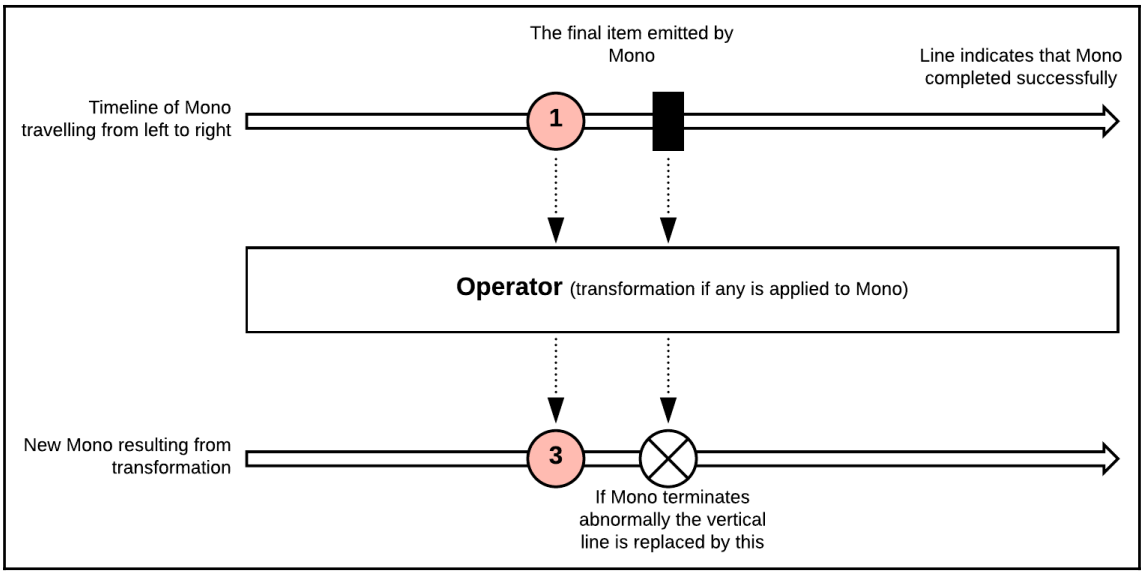

Publisher
onSubscribe
Subscription
request (n)
cancel ()
Subscriber
onNext*
(onError | onComplete)?

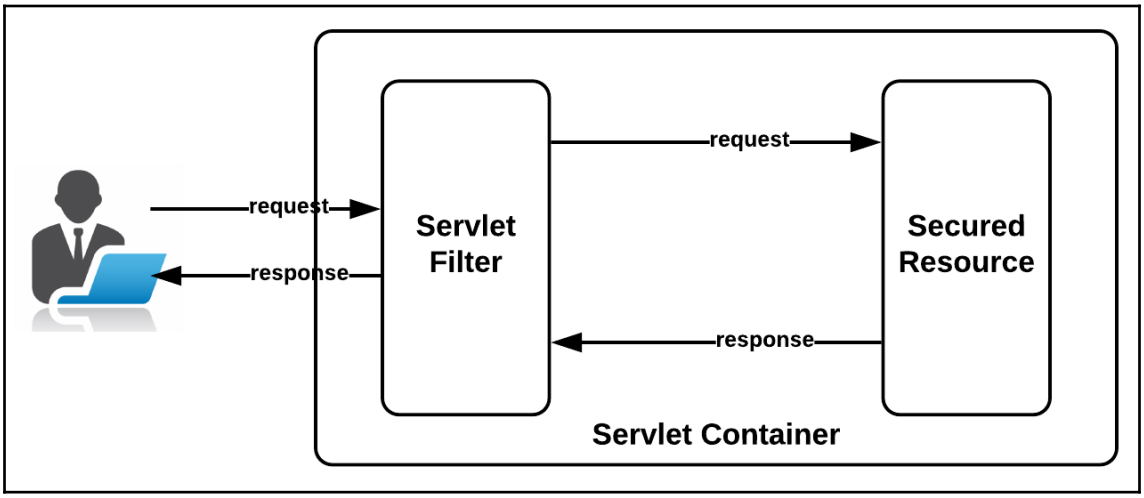

Publisher
Publishes On
Thread One
Subscriber
Subscribes On
Thread Two

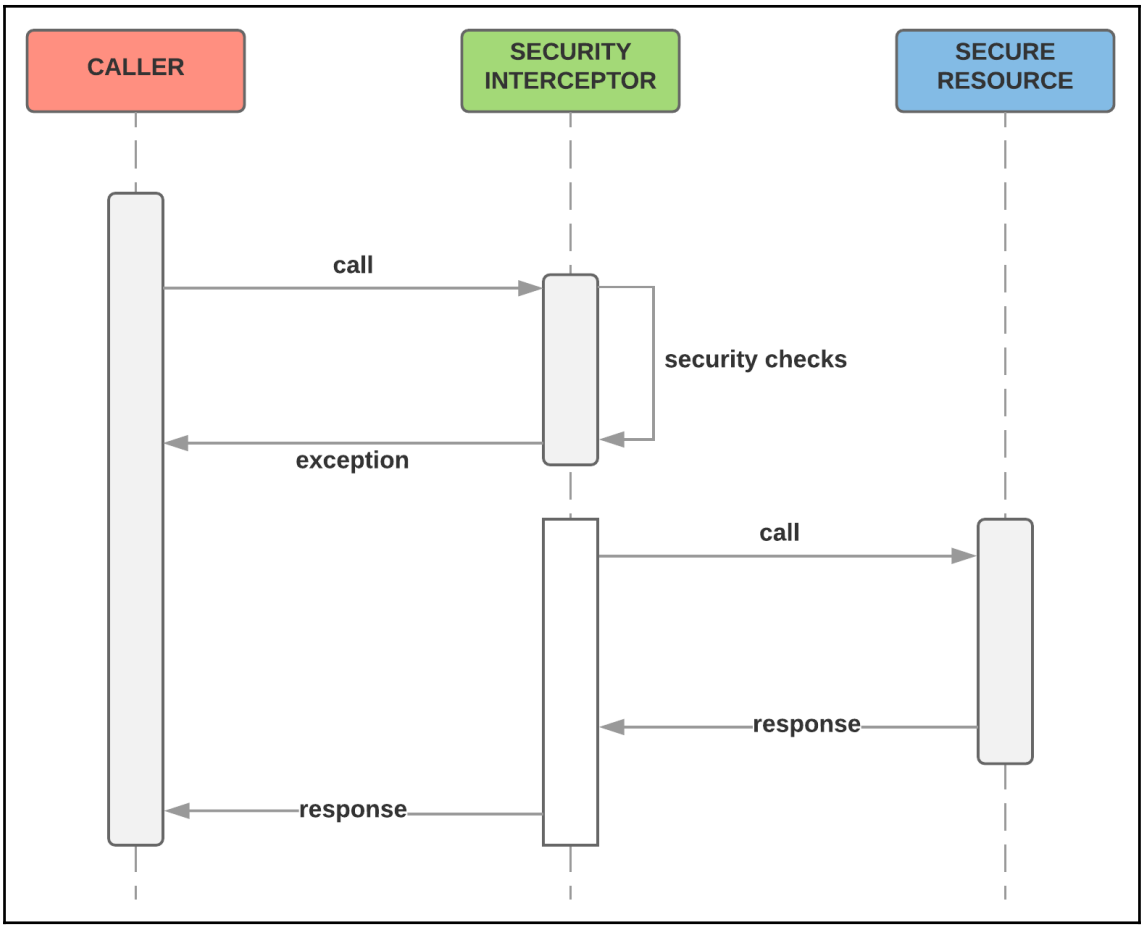

ADDONS
Spring
Framework
Reactor
Core
Reactive Streams
Commons
Rx
IO

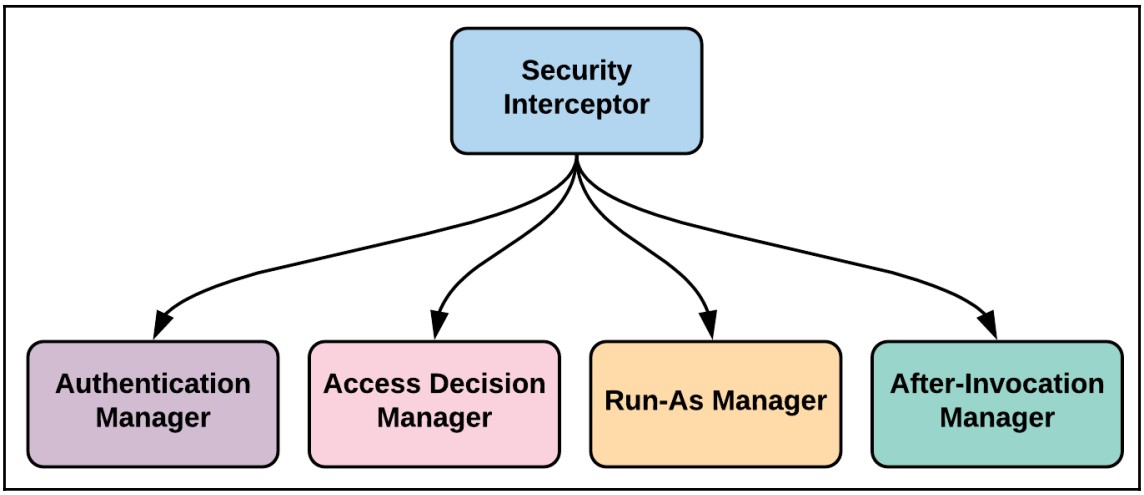

2013: 1.x
2014: 2.x
2016: 3.x

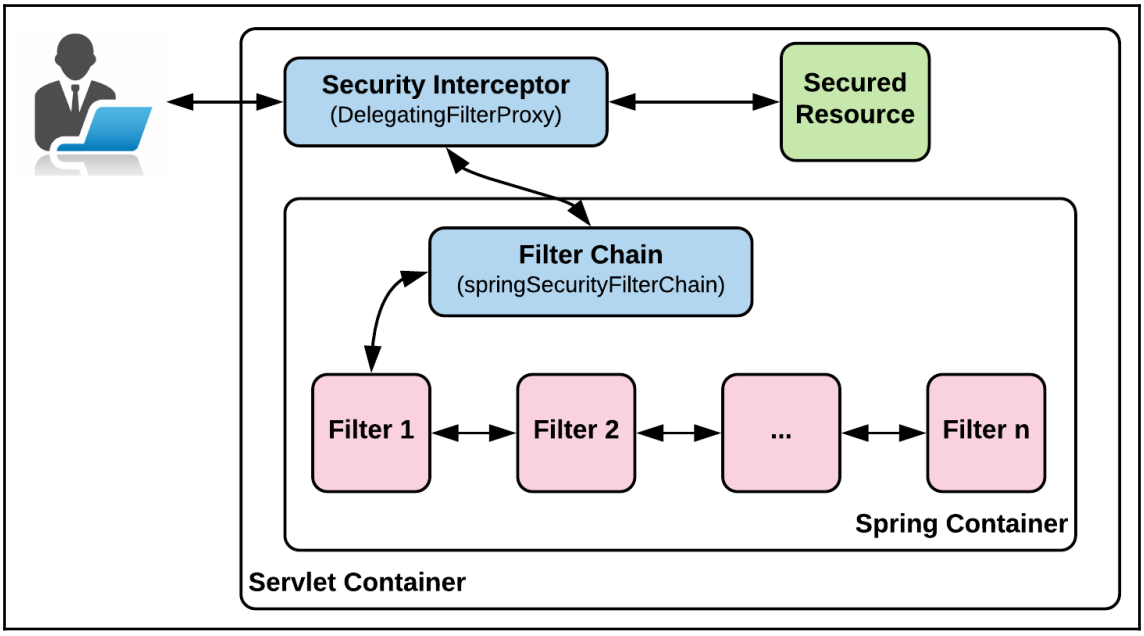

Items emitted by Flux
Line indicates that Flux completed successfully
Timeline of Flux travelling from left to right
1
2
3
4
5
6
Operator (transformation if any is applied to Flux)
New Flux resulting from transformation
1
2
3
If Flux terminates abnormally the vertical line is replaced by this

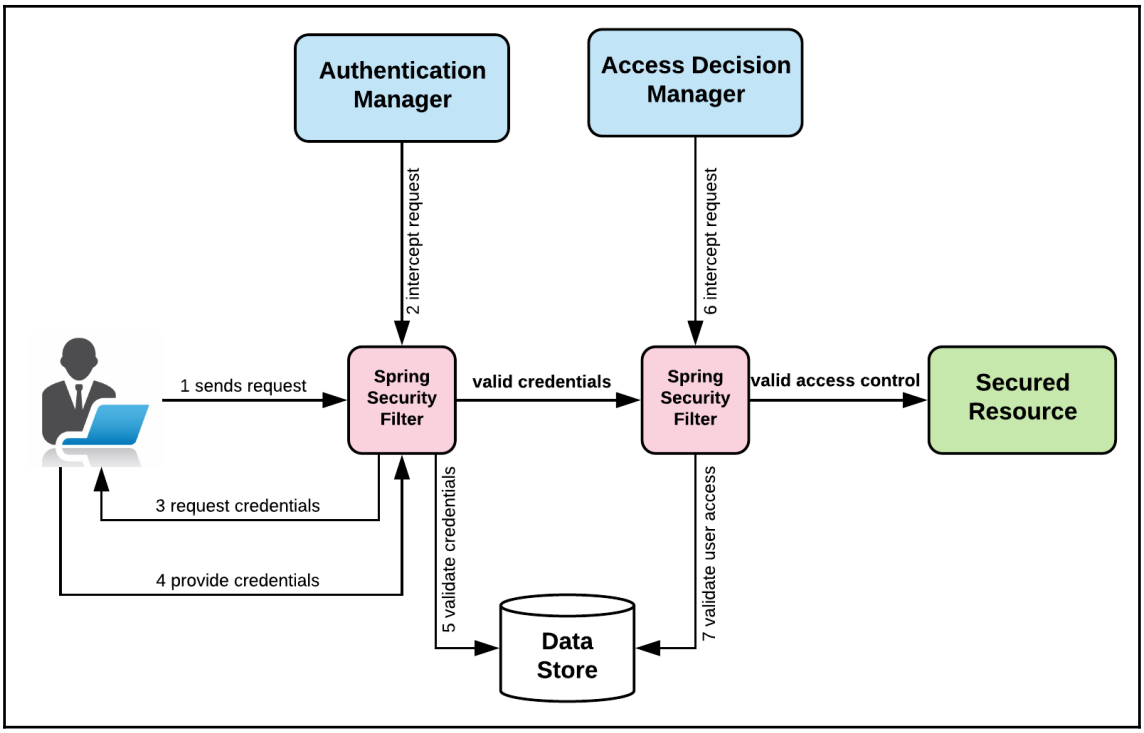

The final item emitted by Mono
Line indicates that Mono completed successfully
Timeline of Mono travelling from left to right
1
Operator (transformation if any is applied to Mono)
New Mono resulting from transformation
3
If Mono terminates abnormally the vertical line is replaced by this


request
response
Servlet Filter
request
response
Secured Resource
Servlet Container

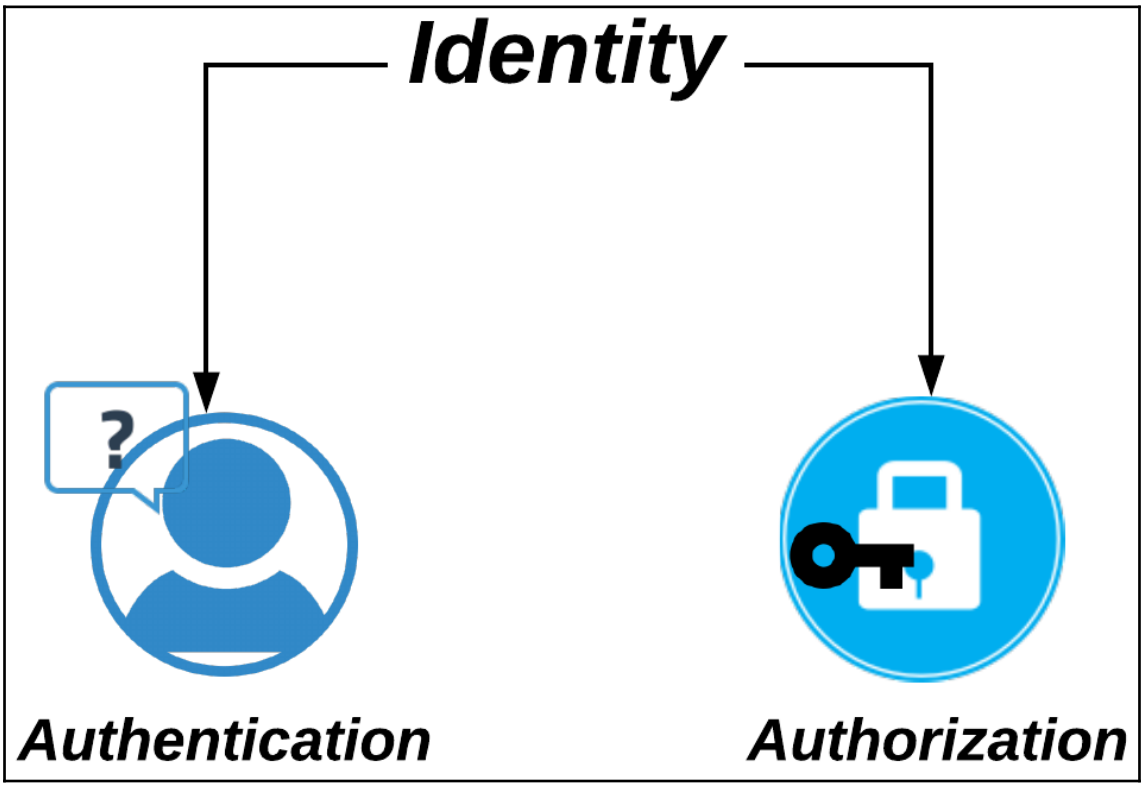

CALLER
SECURITY INTERCEPTOR
SECURE RESOURCE
call
security checks
exception
call
response
response

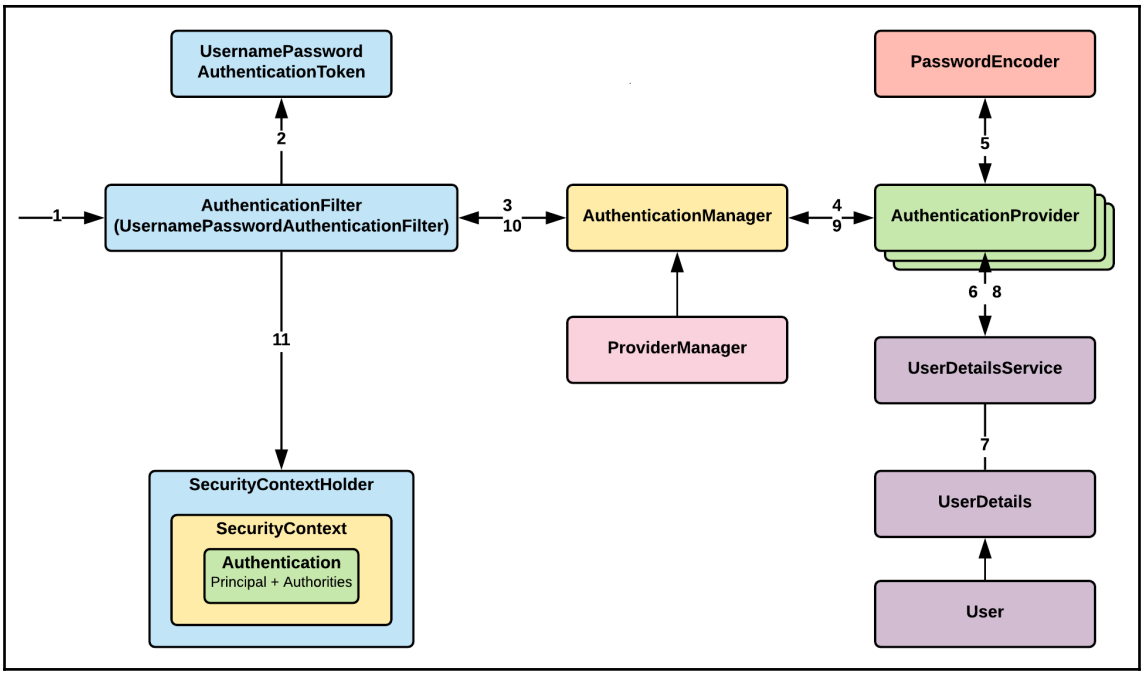

Security Interceptor
Authentication Manager
Access Decision Manager
Run-As Manager
After-Invocation Manager

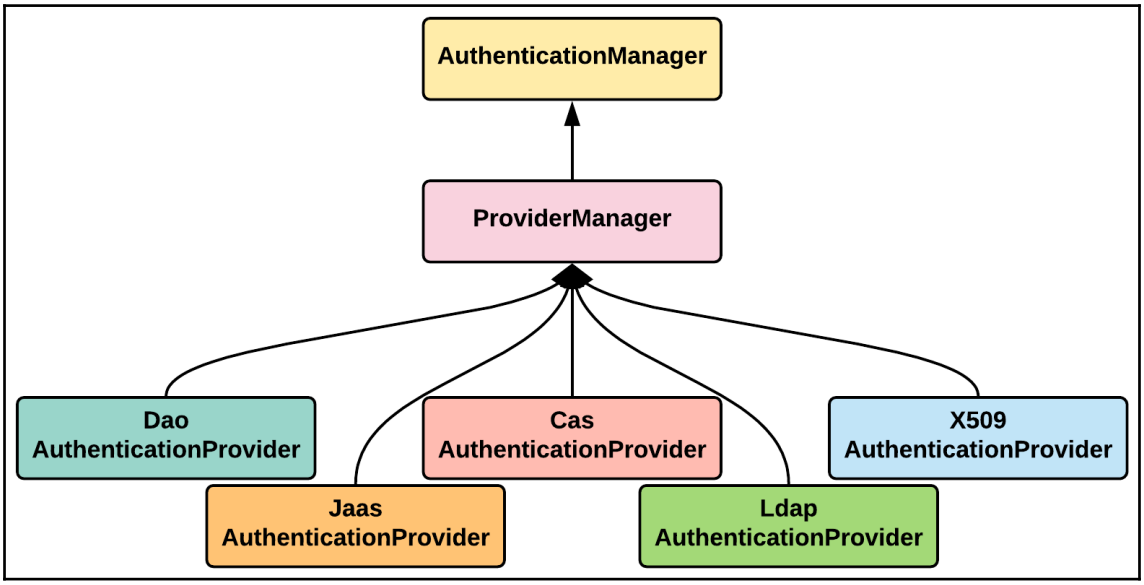

Security Interceptor
(DelegatingFilterProxy)
Secured Resource
Filter Chain
(springSecurityFilterChain)
Filter 1
Filter 2
...
Filter n
Spring Container
Servlet Container

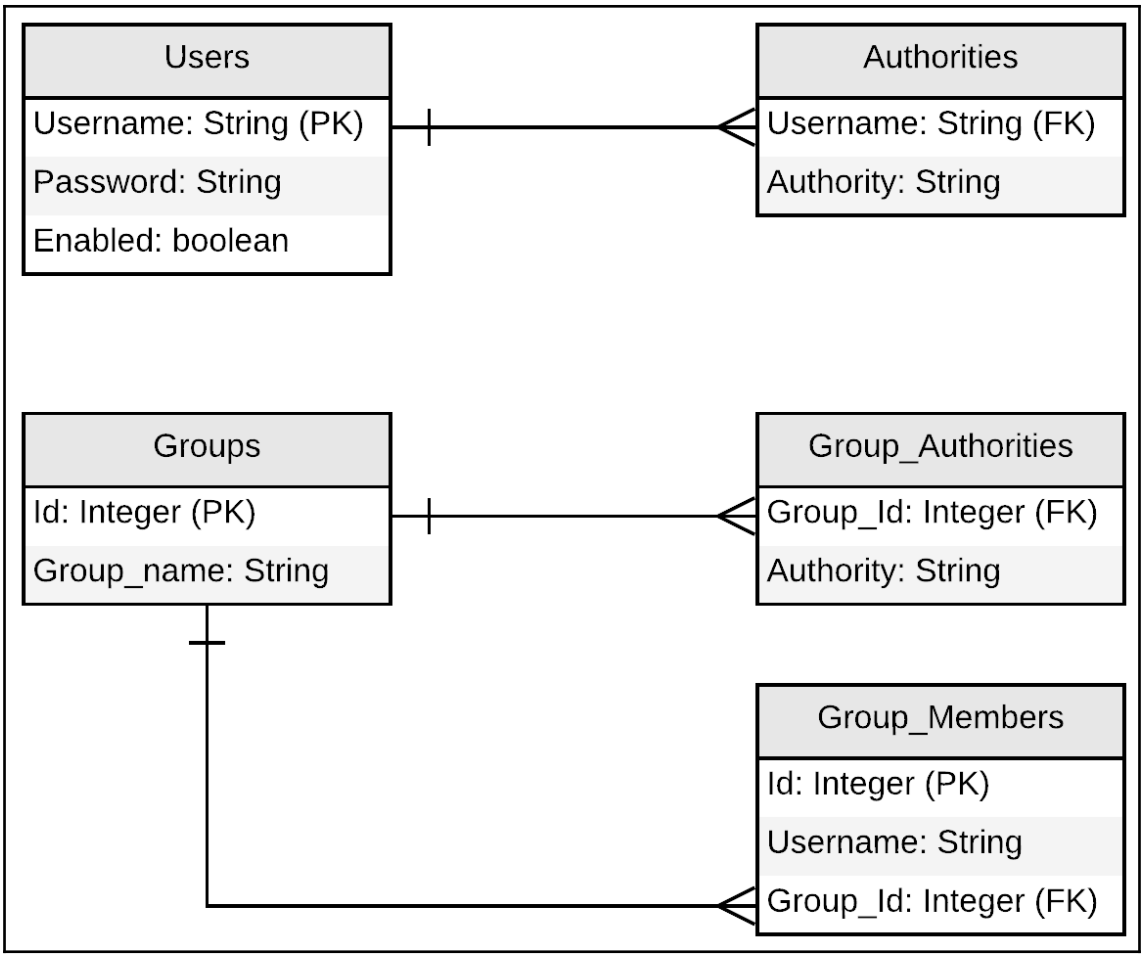

Authentication Manager
Access Decision Manager
2 intercept request
6 intercept request
1 sends request
Spring Security Filter
valid credentials
Spring Security Filter
valid access control
Secured Resource
3 request credentials
4 provide credentials
5 validate credentials
7 validate user access
Data Store

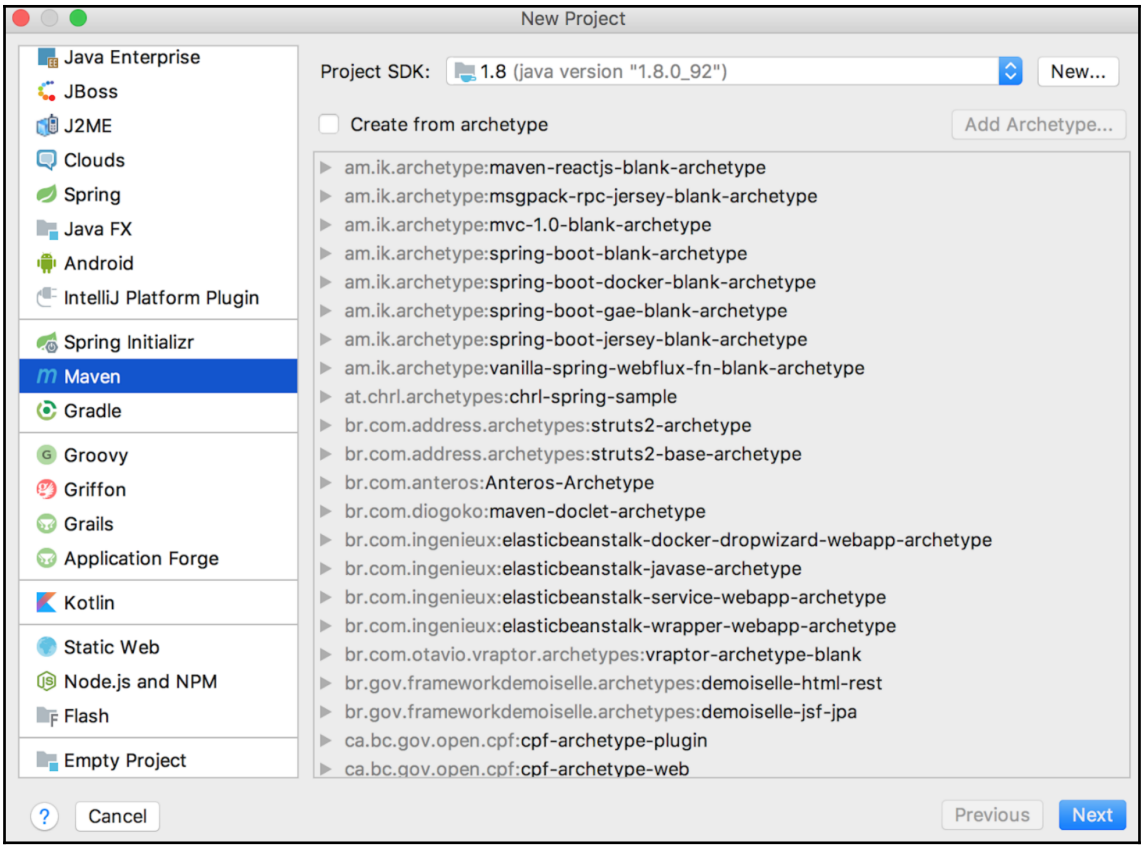
Chapter 2: Deep Diving into Spring Security

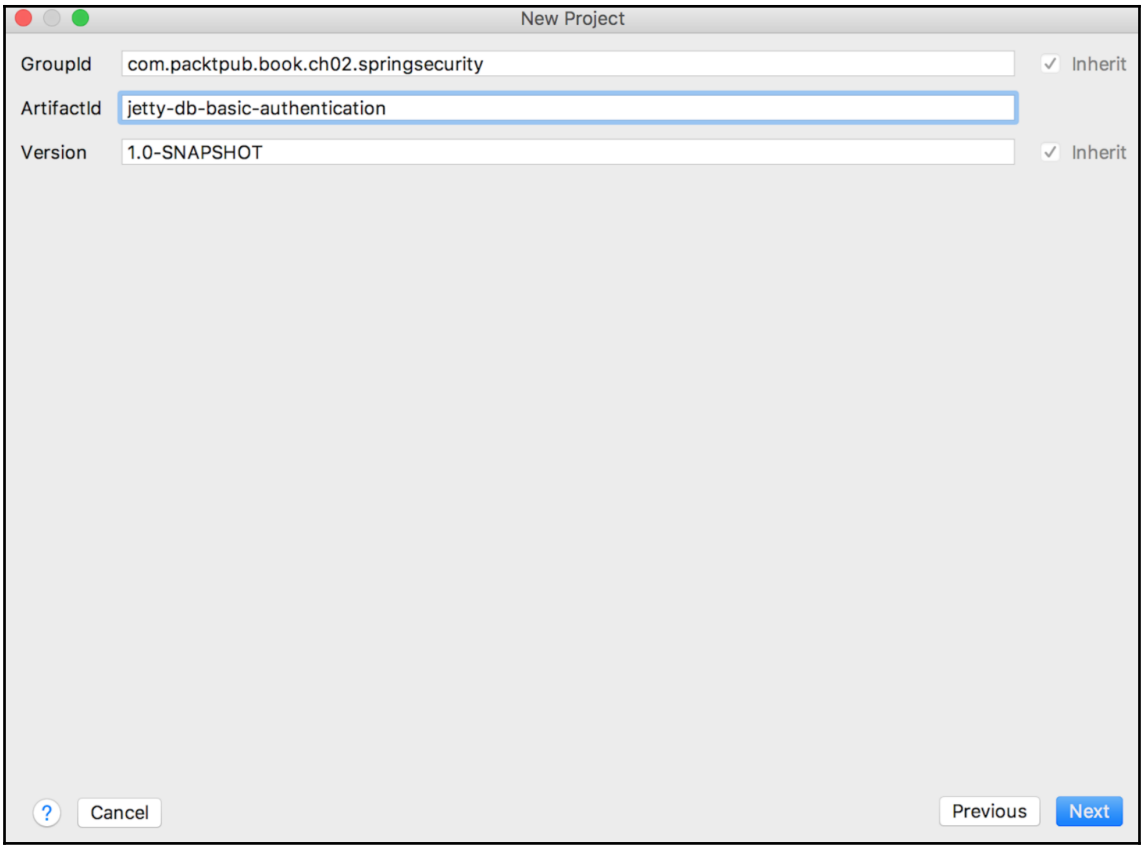


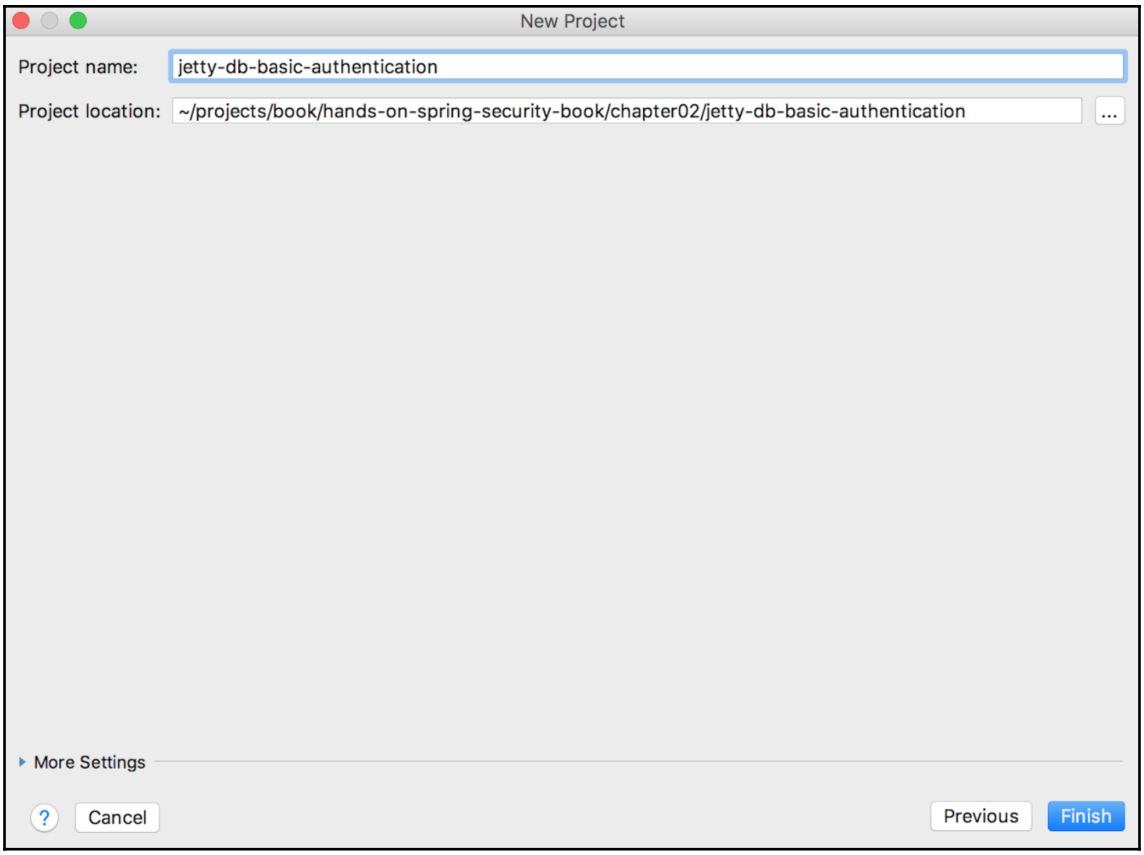

UsernamePassword
AuthenticationToken
2
1
AuthenticationFilter
(UsernamePasswordAuthenticationFilter)
3
10
AuthenticationManager
4
9
PasswordEncoder
5
AuthenticationProvider
6
8
ProviderManager
UserDetailsService
7
11
SecurityContextHolder
SecurityContext
Authentication
Principal + Authorities
UserDetails
User

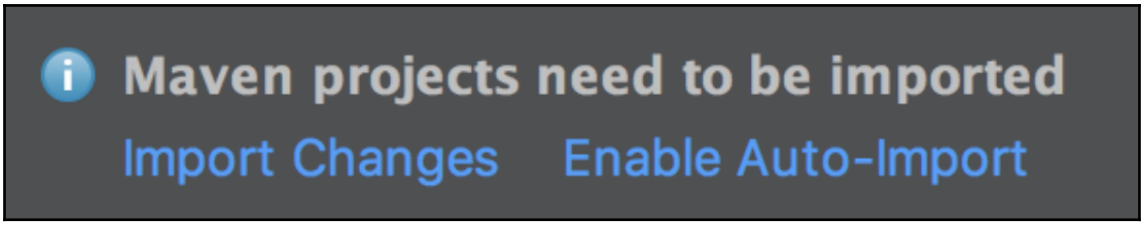

AuthenticationManager
ProviderManager
Dao
AuthenticationProvider
Cas
AuthenticationProvider
X509
AuthenticationProvider
Jaas
AuthenticationProvider
Ldap
AuthenticationProvider


Users
Username: String (PK)
Password: String
Enabled: boolean
Authorities
Username: String (FK)
Authority: String
Groups
Id: Integer (PK)
Group_name: String
Group_Authorities
Group_Id: Integer (FK)
Authority: String
Group_Members
Id: Integer (PK)
Username: String
Group_Id: Integer (FK)

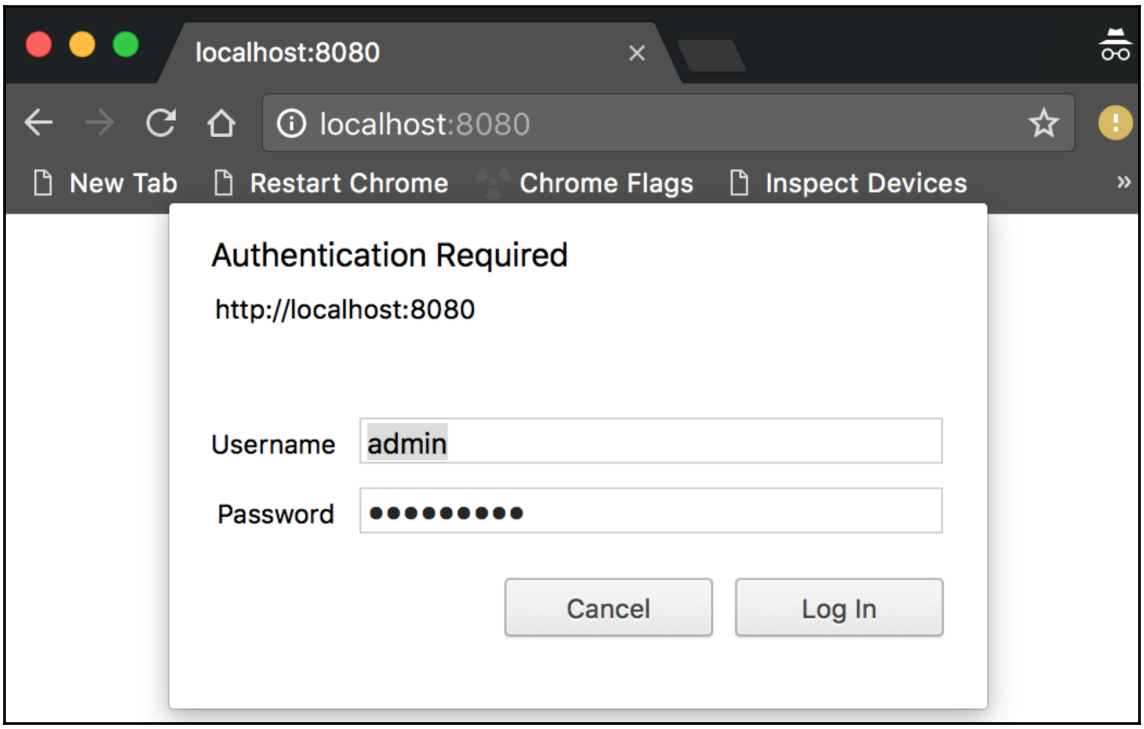

New Project
Java Enterprise
JBoss
J2ME
Clouds
Spring
Java FX
Android
IntelliJ Platform Plugin
Spring Initializr
Maven
Gradle
Groovy
Griffon
Grails
Application Forge
Kotlin
Static Web
Node.js and NPM
Flash
Empty Project
Project SDK:
1.8 (java version "1.8.0_92")
New...
Create from archetype
Add Archetype...
am.ik.archetype:maven-reactjs-blank-archetype
am.ik.archetype:msgpack-rpc-jersey-blank-archetype
am.ik.archetype:mvc-1.0-blank-archetype
am.ik.archetype:spring-boot-blank-archetype
am.ik.archetype:spring-boot-docker-blank-archetype
am.ik.archetype:spring-boot-gae-blank-archetype
am.ik.archetype:spring-boot-jersey-blank-archetype
am.ik.archetype:vanilla-spring-webflux-fn-blank-archetype
at.chrl.archetypes:chrl-spring-sample
br.com.address.archetypes:struts2-archetype
br.com.address.archetypes:struts2-base-archetype
br.com.anteros:Anteros-Archetype
br.com.diogoko:maven-doclet-archetype
br.com.ingenieux:elasticbeanstalk-docker-dropwizard-webapp-archetype
br.com.ingenieux:elasticbeanstalk-javase-archetype
br.com.ingenieux:elasticbeanstalk-service-webapp-archetype
br.com.ingenieux:elasticbeanstalk-wrapper-webapp-archetype
br.com.otavio.vraptor.archetypes:vraptor-archetype-blank
br.gov.frameworkdemoiselle.archetypes:demoiselle-html-rest
br.gov.frameworkdemoiselle.archetypes:demoiselle-jsf-jpa
ca.bc.gov.open.cpf:cpf-archetype-plugin
ca.bc.gov.open.cpf:cpf-archetype-web
Cancel
Previous
Next

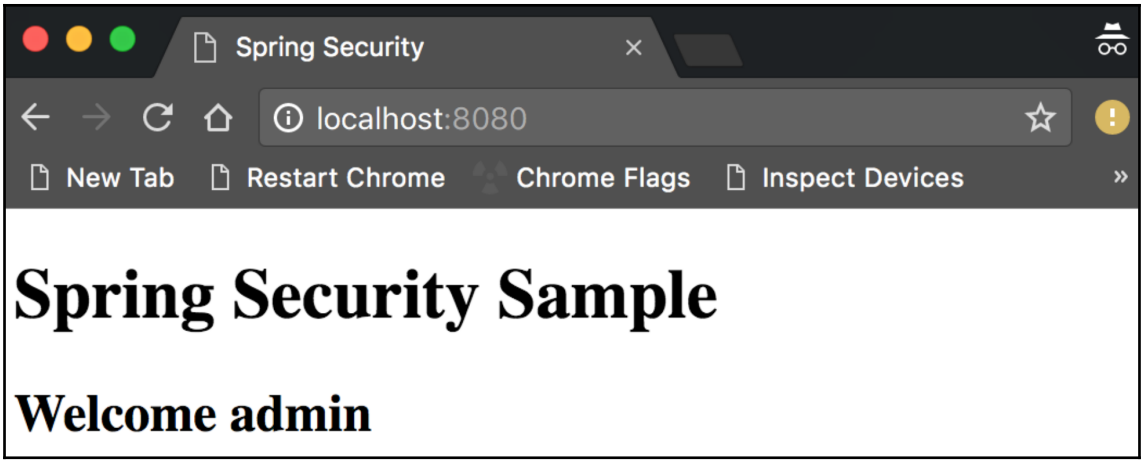

New Project
GroupId
com.packtpub.book.ch02.springsecurity
Inherit
ArtifactId
jetty-db-basic-authentication
Version
1.0-SNAPSHOT
Inherit
Cancel
Previous
Next

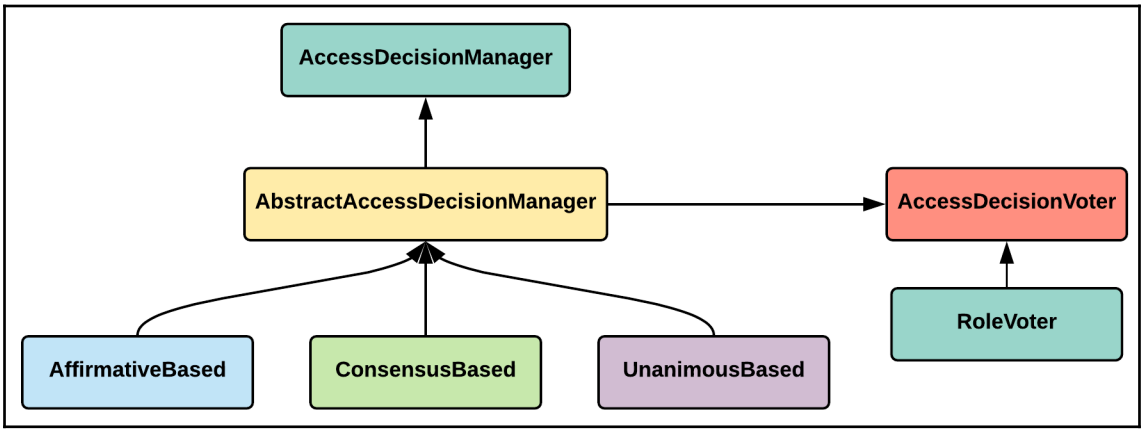

New Project
Project name:
jetty-db-basic-authentication
Project location:
~/projects/book/hands-on-spring-security-book/chapter02/jetty-db-basic-authentication
...
More Settings
Cancel
Previous
Finish

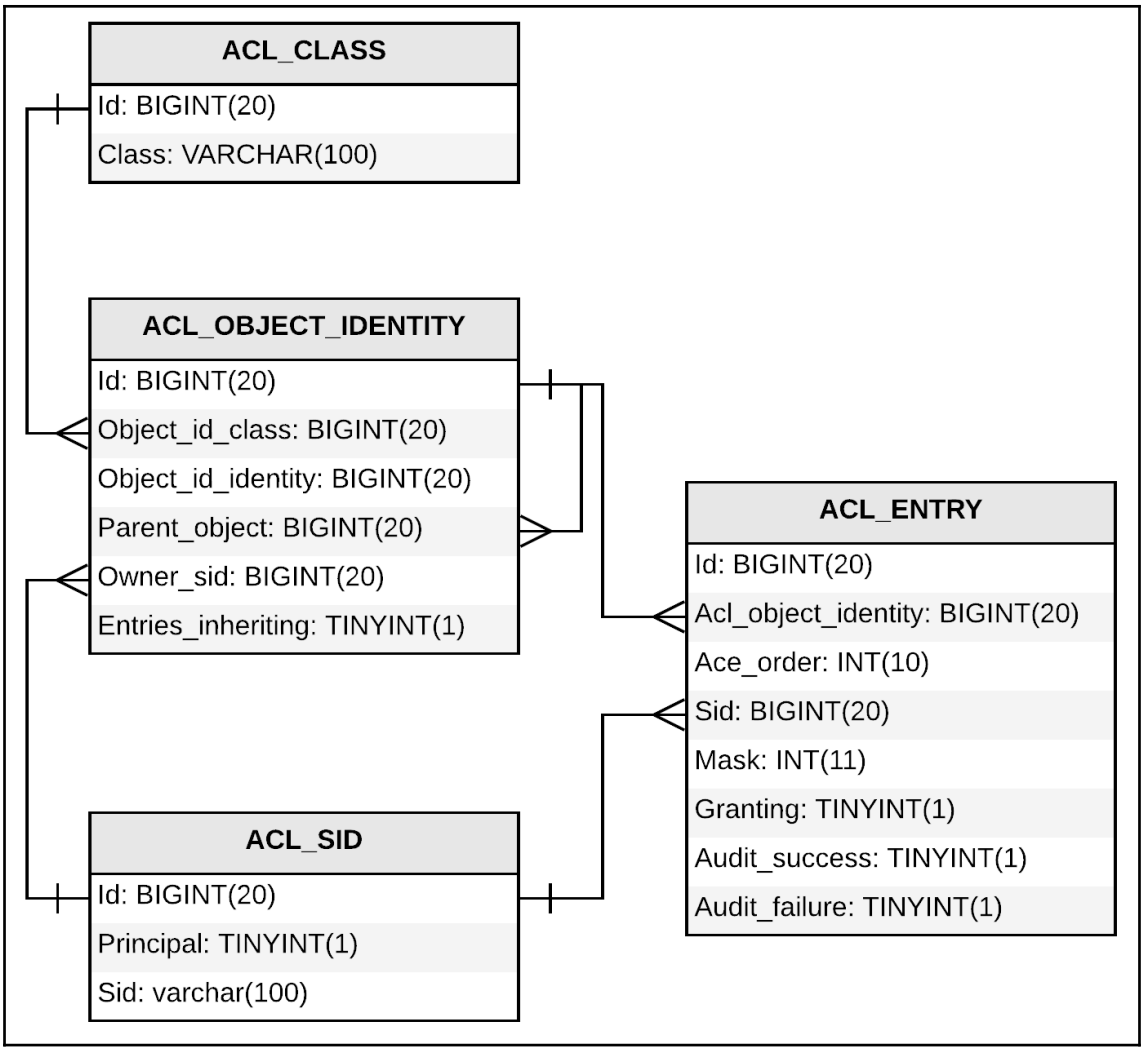

Maven projects need to be imported
Import Changes
Enable Auto-Import


Terminal
INFO: Mapped "{[/],methods=[GET]}" onto public java.lang.String com.packtpub.book.ch02.springsecurity.cor
Apr 16, 2018 9:49:25 PM org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapte
INFO: Looking for @ControllerAdvice: WebApplicationContext for namespace 'dispatcher-servlet': startup da
Apr 16, 2018 9:49:25 PM org.springframework.web.servlet.DispatcherServlet initServletBean
INFO: FrameworkServlet 'dispatcher': initialization completed in 487 ms
[INFO] Started o.e.j.m.p.JettyWebAppContext@622fdb81{/,file:///Users/tjohn/projects/book/spring-security-
Users/tjohn/projects/book/spring-security-book/chapter02/spring-security-basic-authentication/src/main/we
[INFO] Started ServerConnector@4cc12db2{HTTP/1.1,[http/1.1]}{0.0.0.0:8080}
[INFO] Started @7440ms
[INFO] Started Jetty Server

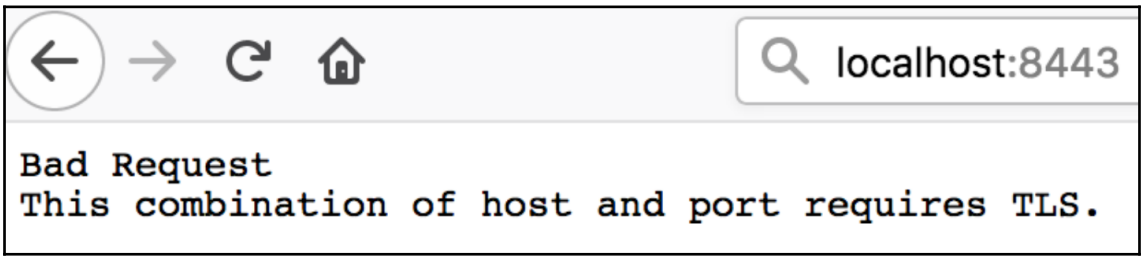

localhost:8080
localhost:8080
New Tab
Restart Chrome
Chrome Flags
Inspect Devices
Authentication Required
http://localhost:8080
Username
admin
Password
Cancel
Log In

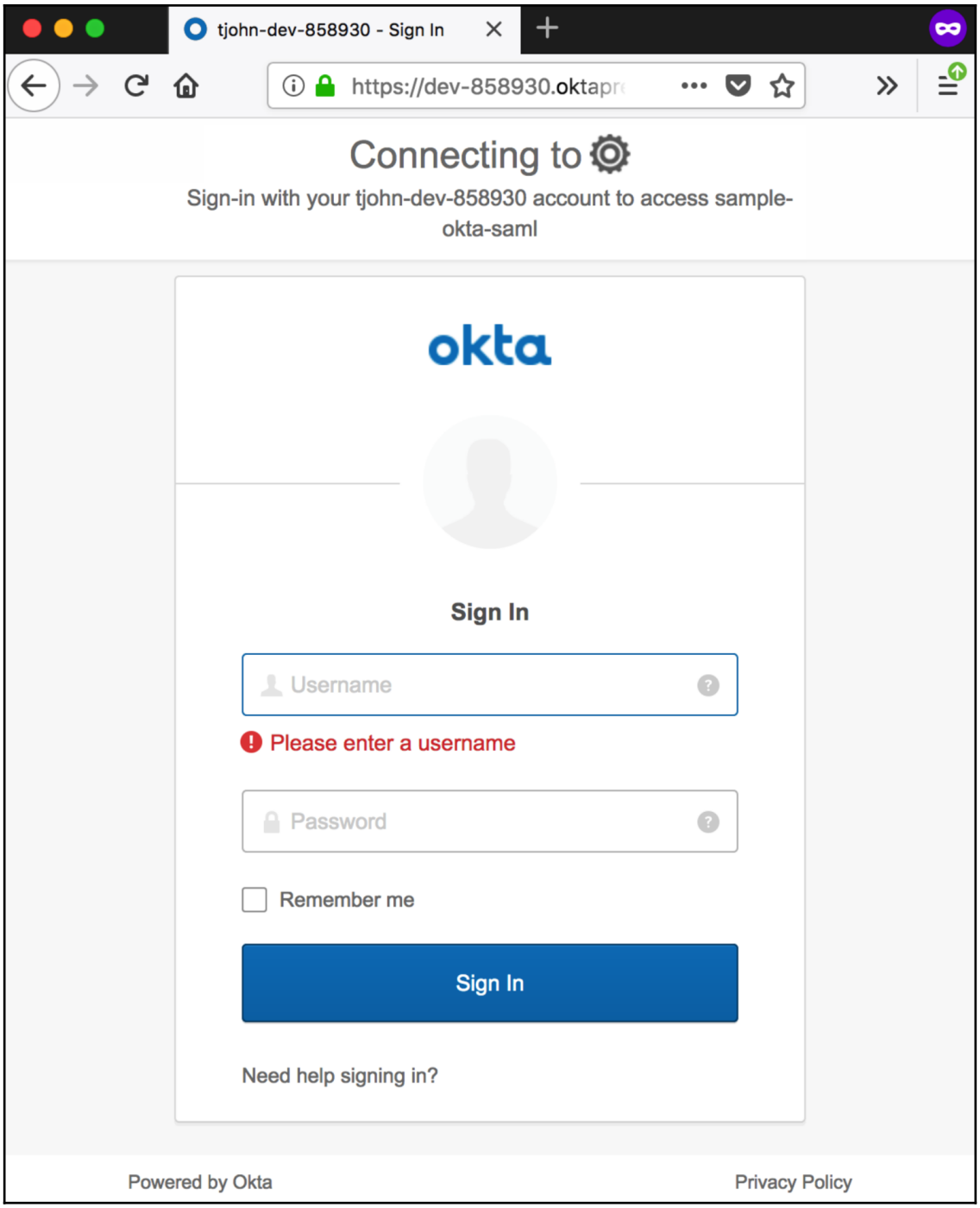

Spring Security
localhost:8080
New Tab
Restart Chrome
Chrome Flags
Inspect Devices
Spring Security Sample
Welcome admin

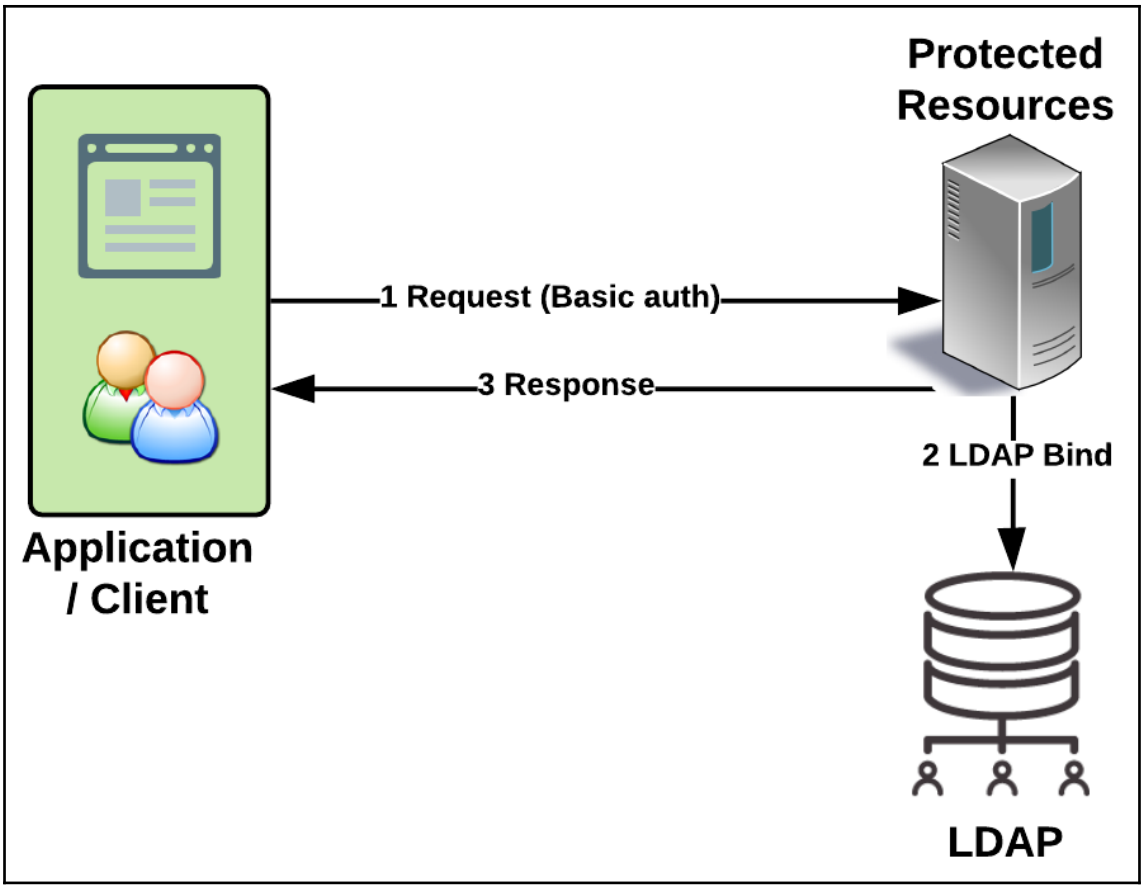

AccessDecisionManager
AbstractAccessDecisionManager
AccessDecisionVoter
RoleVoter
AffirmativeBased
ConsensusBased
UnanimousBased

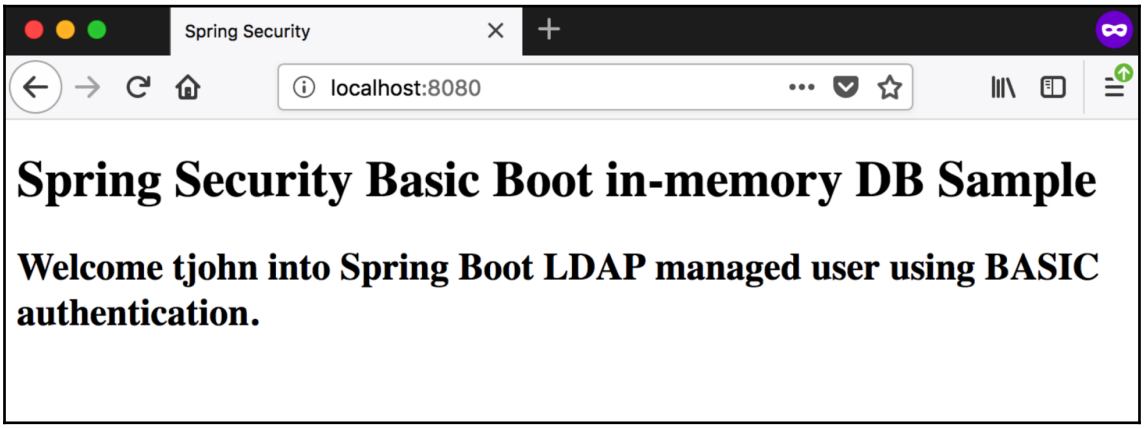

ACL_CLASS
Id: BIGINT(20)
Class: VARCHAR(100)
ACL_OBJECT_IDENTITY
Id: BIGINT(20)
Object_id_class: BIGINT(20)
Object_id_identity: BIGINT(20)
Parent_object: BIGINT(20)
Owner_sid: BIGINT(20)
Entries_inheriting: TINYINT(1)
ACL_ENTRY
Id: BIGINT(20)
Acl_object_identity: BIGINT(20)
Ace_order: INT(10)
Sid: BIGINT(20)
Mask: INT(11)
Granting: TINYINT(1)
Audit_success: TINYINT(1)
Audit_failure: TINYINT(1)
ACL_SID
Id: BIGINT(20)
Principal: TINYINT(1)
Sid: varchar(100)

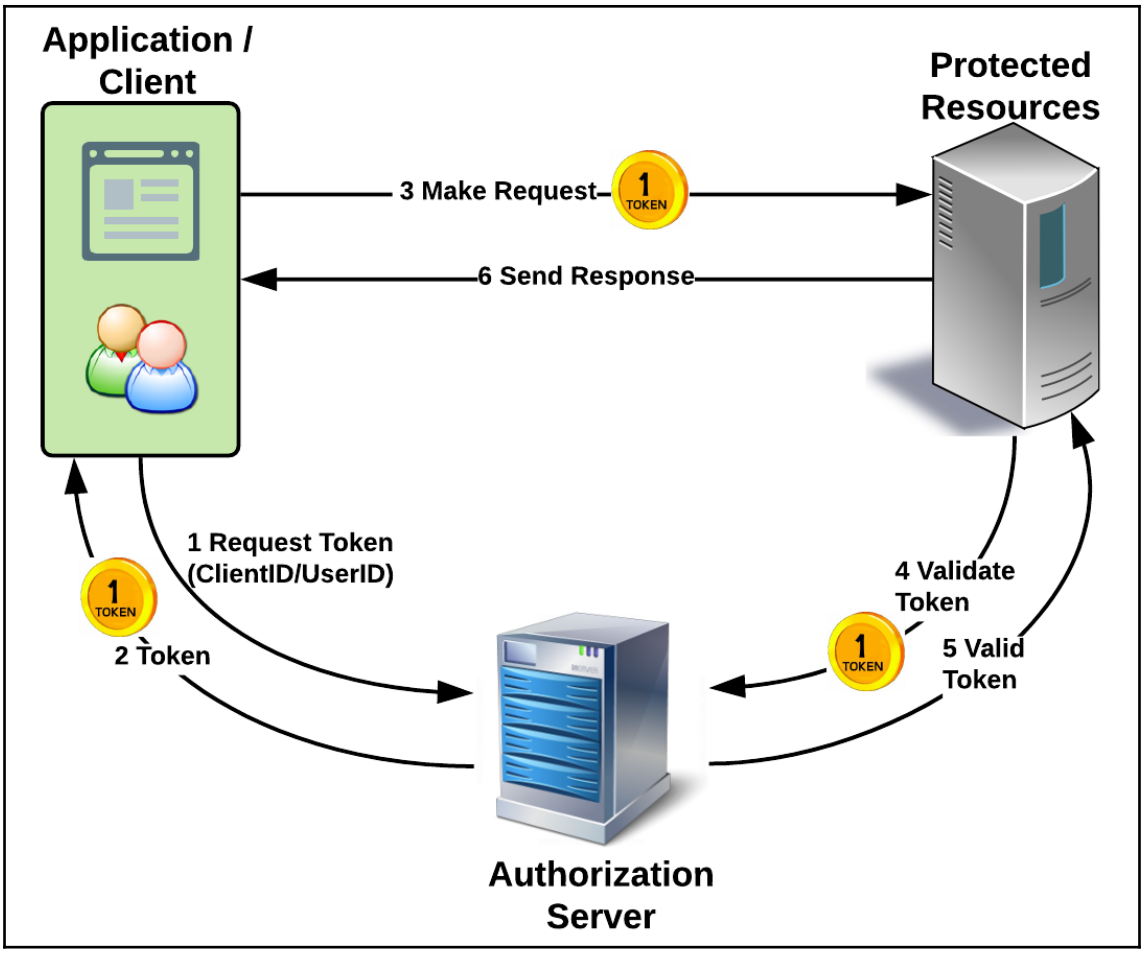
Chapter 3: Authentication Using SAML, LDAP, and OAuth/OIDC

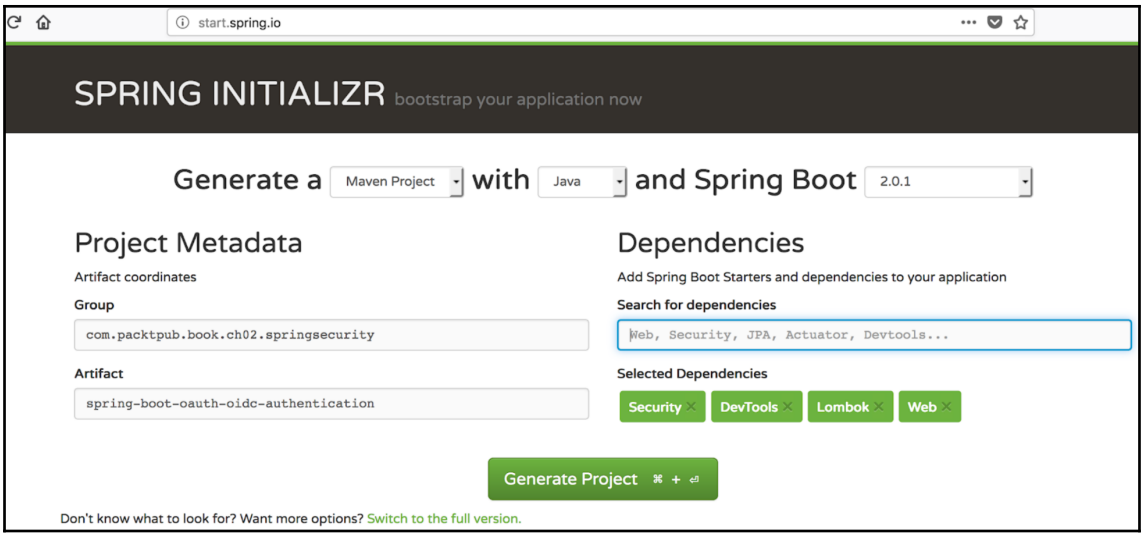


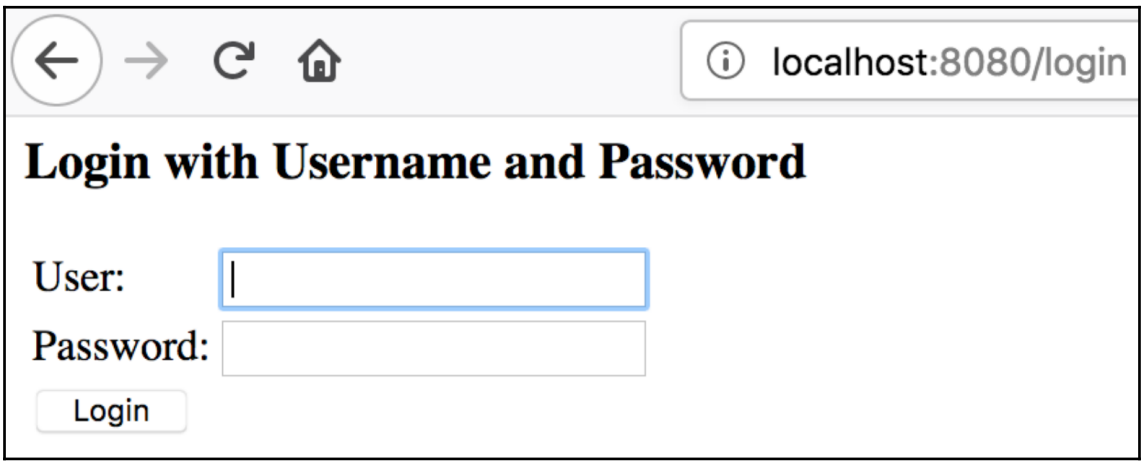

tjohn-dev-858930 - Sign In
https://dev-858930.oktapr
Connecting to
Sign-in with your tjohn-dev-858930 account to access sample-okta-saml
okta
Sign In
Username
Please enter a username
Password
Remember me
Sign In
Need help signing in?
Powered by Okta
Privacy Policy

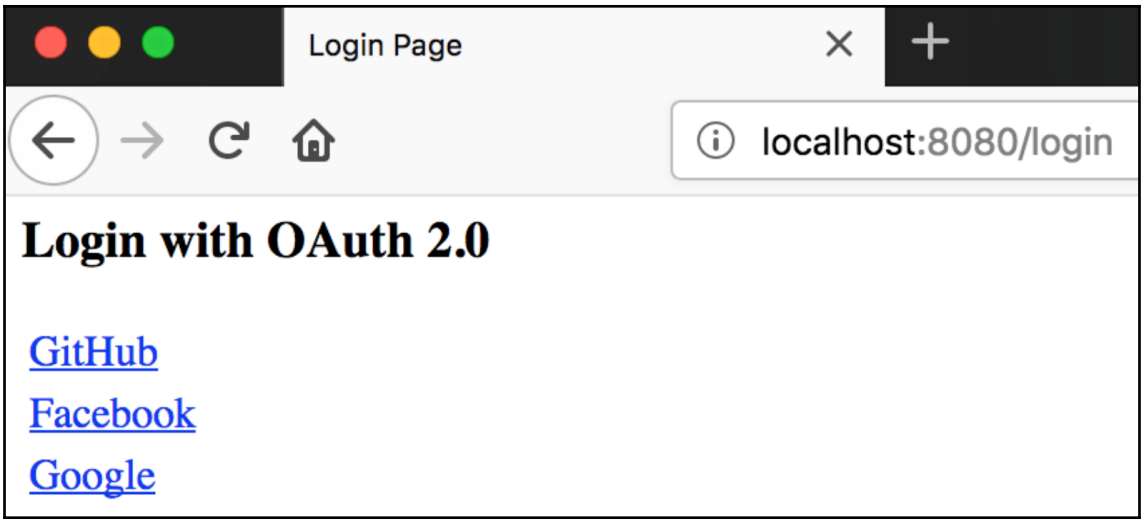

Protected
Resources
1 Request (Basic auth)
3 Response
2 LDAP Bind
Application
/ Client
LDAP

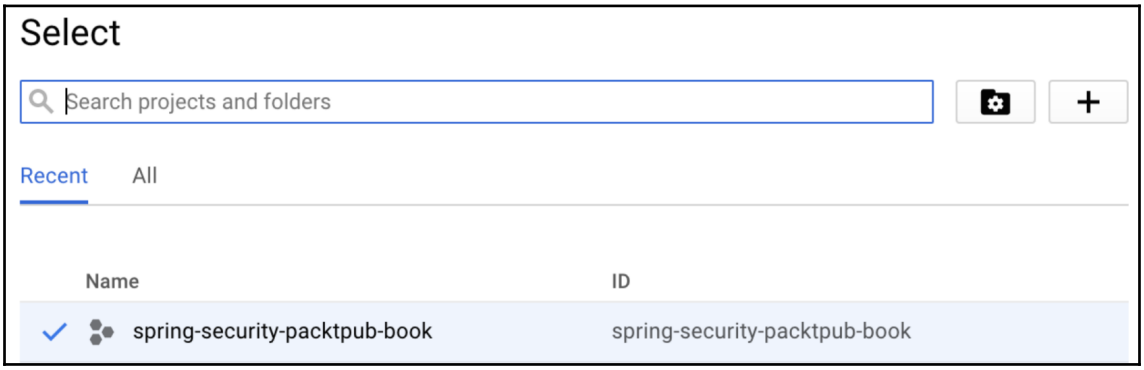

Spring Security
localhost:8080
Spring Security Basic Boot in-memory DB Sample
Welcome tjohn into Spring Boot LDAP managed user using BASIC authentication.

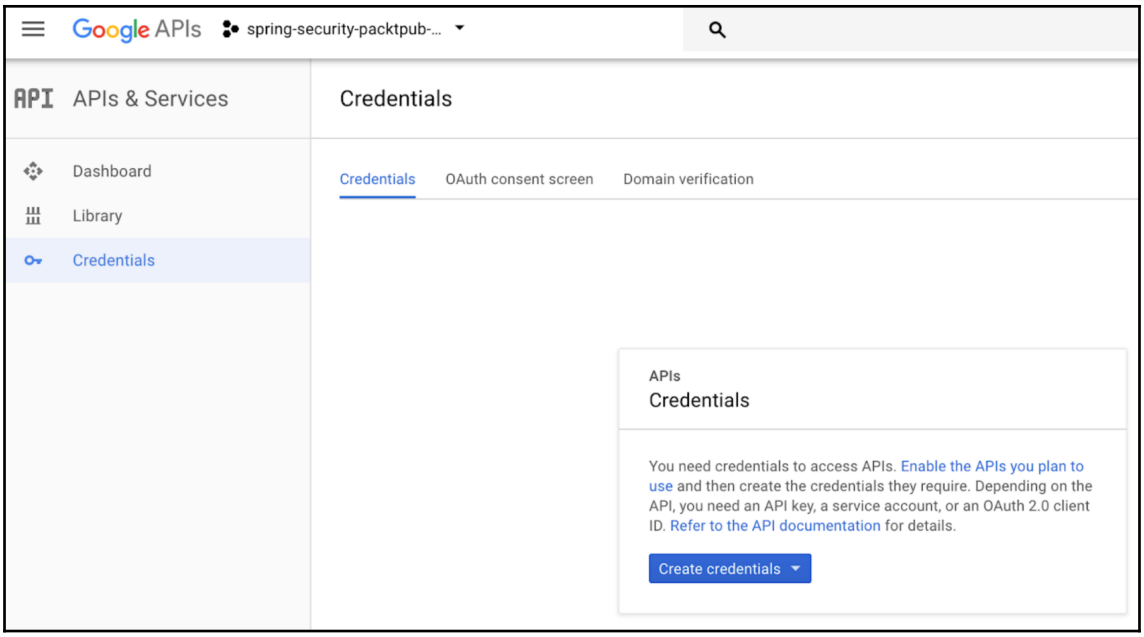

Application / Client
Protected Resources
3 Make Request
1 TOKEN
6 Send Response
1 Request Token (ClientID/UserID)
1 TOKEN
2 Token
4 Validate Token
1 TOKEN
5 Valid Token
Authorization Server

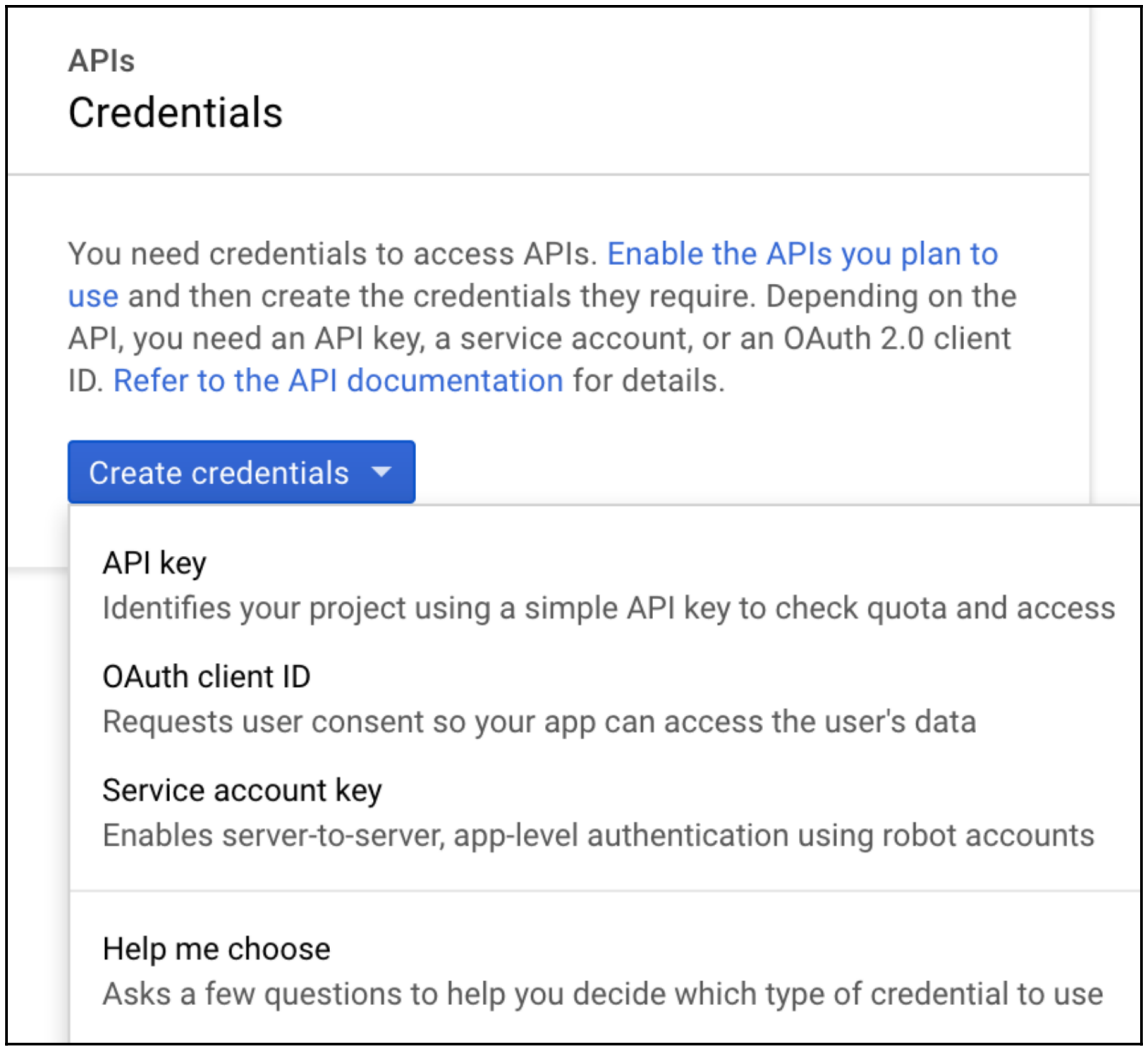

start.spring.io
SPRING INITIALIZR bootstrap your application now
Generate a Maven Project with Java and Spring Boot 2.0.1
Project Metadata
Artifact coordinates
Group
com.packtpub.book.ch02.springsecurity
Artifact
spring-boot-oauth-oidc-authentication
Dependencies
Add Spring Boot Starters and dependencies to your application
Search for dependencies
Web, Security, JPA, Actuator, Devtools...
Selected Dependencies
Security
DevTools
Lombok
Web
Generate Project ⌘ + ⏎
Don't know what to look for? Want more options? Switch to the full version.

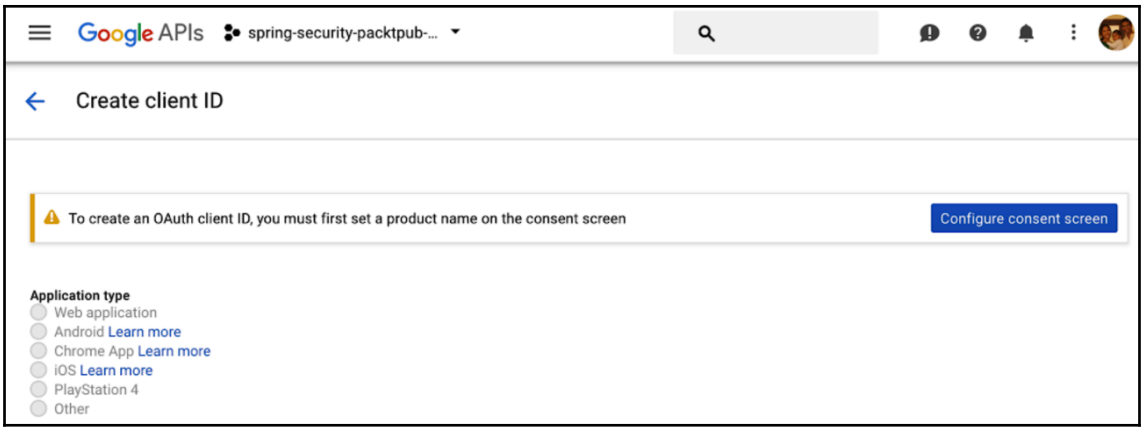

localhost:8080/login
Login with Username and Password
User:
Password:
Login

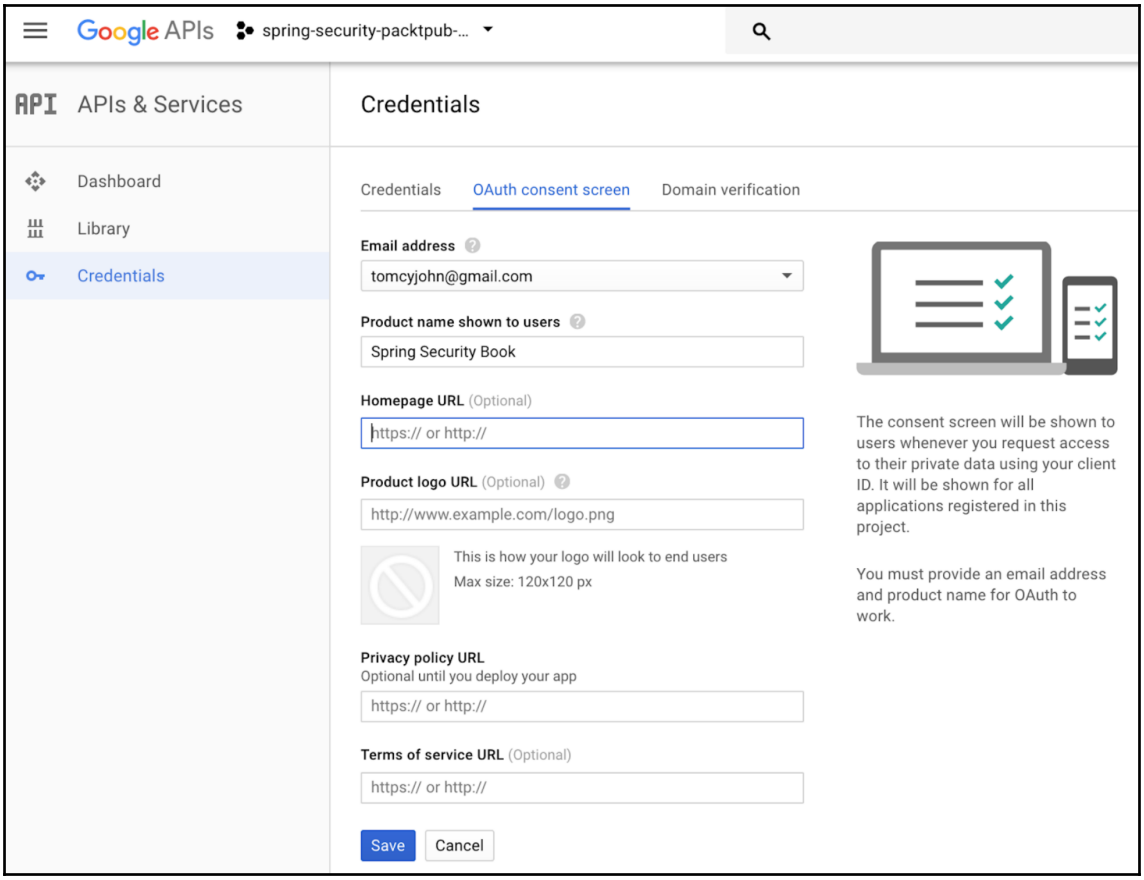

Login Page
localhost:8080/login
Login with OAuth 2.0
GitHub
Facebook
Google

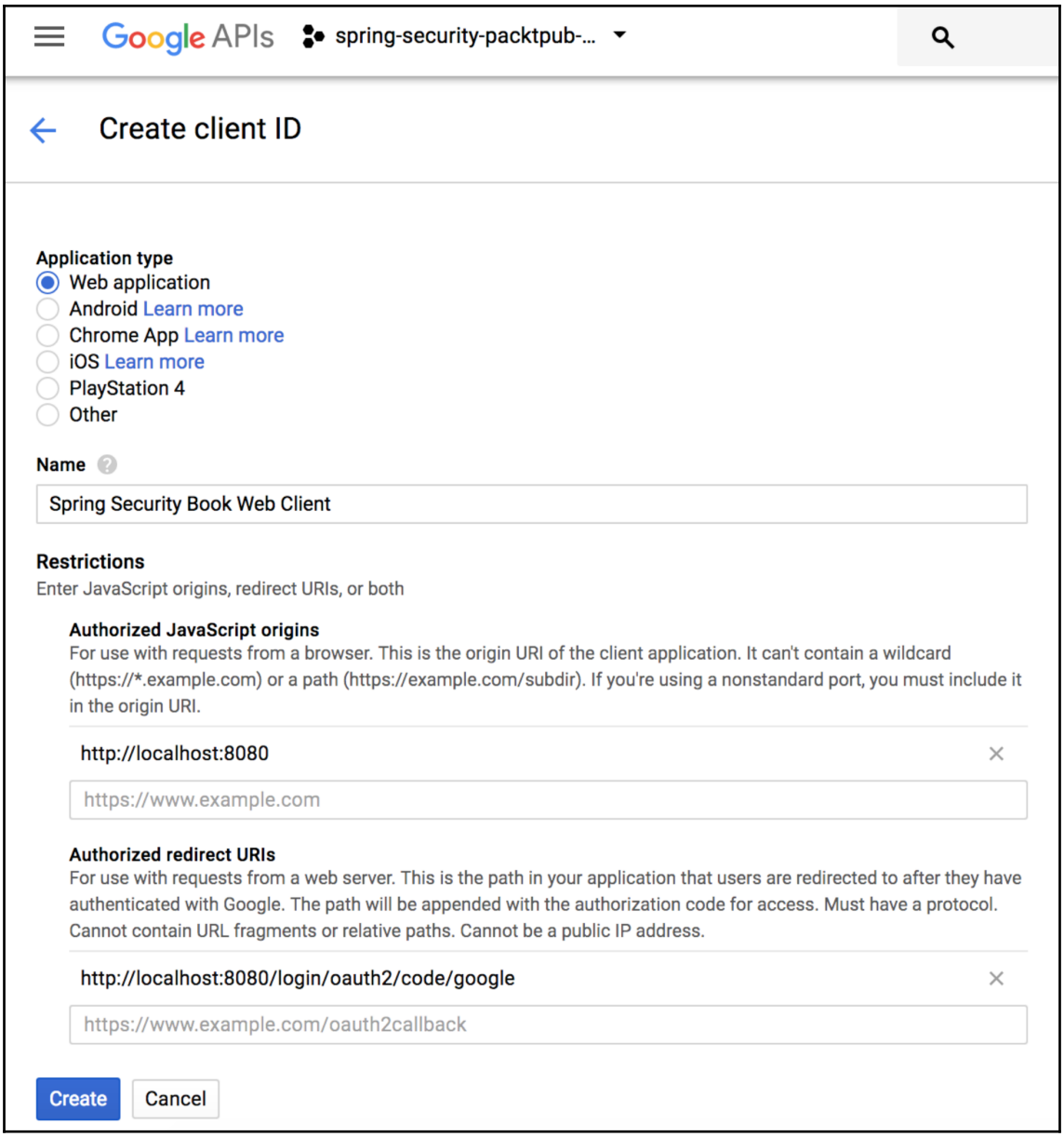

Select
Search projects and folders
Recent
All
Name
ID
spring-security-packtpub-book
spring-security-packtpub-book

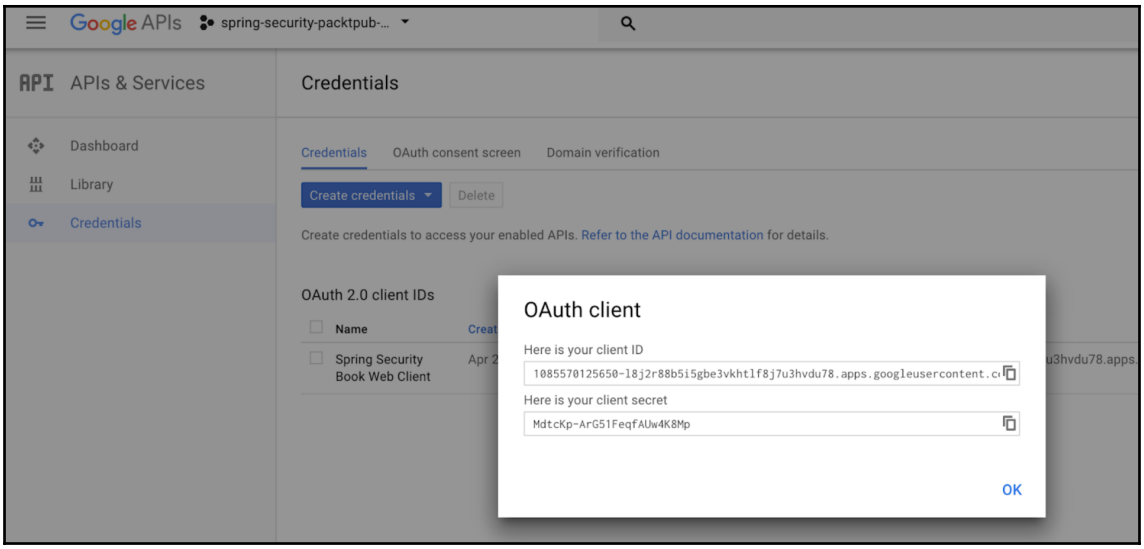

Google APIs
spring-security-packtpub-...
APIs & Services
Dashboard
Library
Credentials
Credentials
Credentials
OAuth consent screen
Domain verification
APIs
Credentials
You need credentials to access APIs. Enable the APIs you plan to use and then create the credentials they require. Depending on the API, you need an API key, a service account, or an OAuth 2.0 client ID. Refer to the API documentation for details.
Create credentials

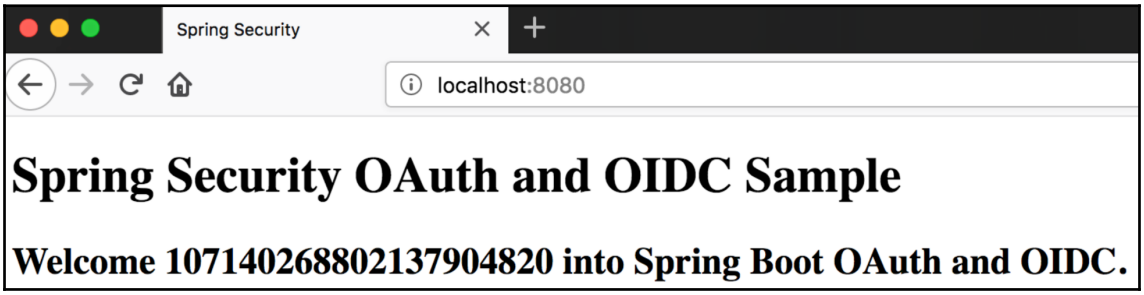

APIs
Credentials
You need credentials to access APIs. Enable the APIs you plan to use and then create the credentials they require. Depending on the API, you need an API key, a service account, or an OAuth 2.0 client ID. Refer to the API documentation for details.
Create credentials
API key
Identifies your project using a simple API key to check quota and access
OAuth client ID
Requests user consent so your app can access the user's data
Service account key
Enables server-to-server, app-level authentication using robot accounts
Help me choose
Asks a few questions to help you decide which type of credential to use


Google APIs
spring-security-packtpub-...
Create client ID
To create an OAuth client ID, you must first set a product name on the consent screen
Configure consent screen
Application type
Web application
Android Learn more
Chrome App Learn more
iOS Learn more
PlayStation 4
Other

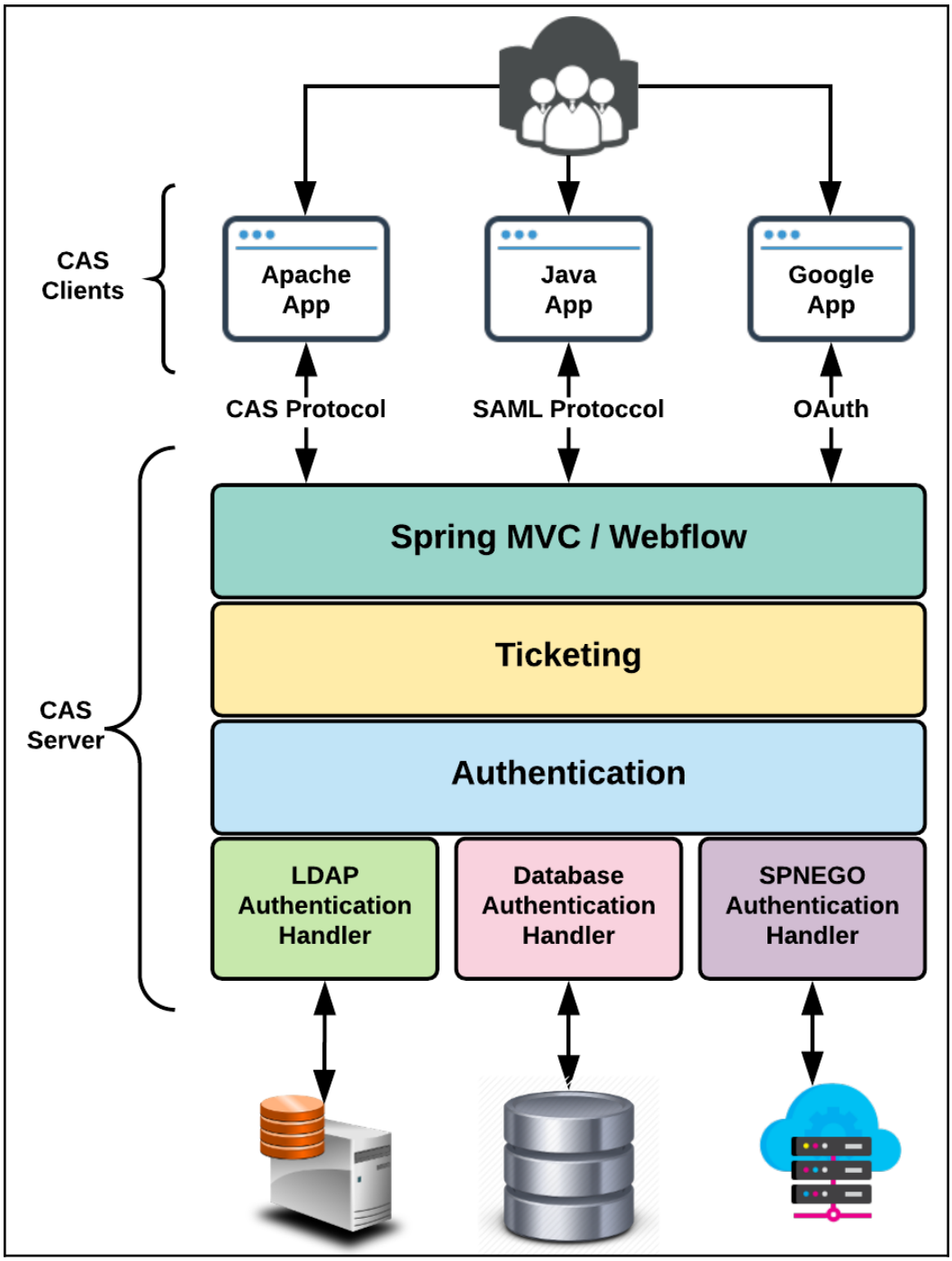

Google APIs
spring-security-packtpub-...
APIs & Services
Dashboard
Library
Credentials
Credentials
Credentials
OAuth consent screen
Domain verification
Email address
tomcyjohn@gmail.com
Product name shown to users
Spring Security Book
Homepage URL (Optional)
https:// or http://
Product logo URL (Optional)
http://www.example.com/logo.png
This is how your logo will look to end users
Max size: 120x120 px
Privacy policy URL
Optional until you deploy your app
https:// or http://
Terms of service URL (Optional)
https:// or http://
Save
Cancel
The consent screen will be shown to users whenever you request access to their private data using your client ID. It will be shown for all applications registered in this project.
You must provide an email address and product name for OAuth to work.

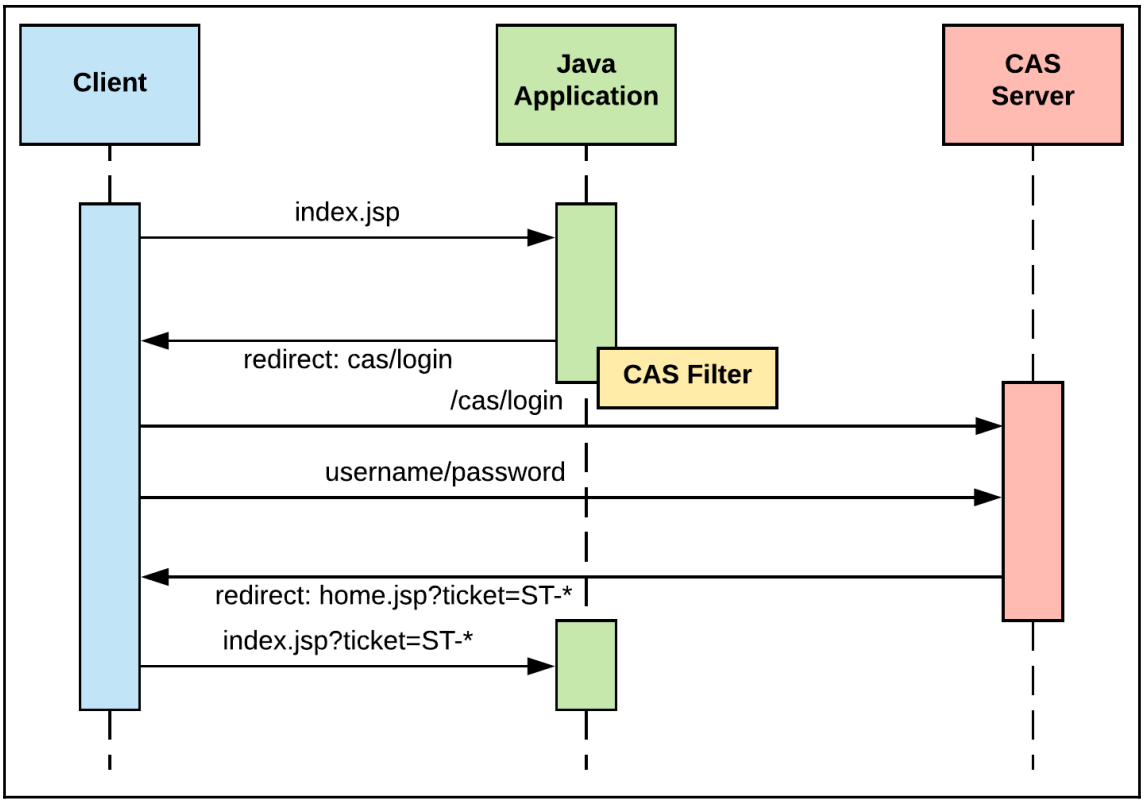

Google APIs
spring-security-packtpub-...
Create client ID
Application type
Web application
Android Learn more
Chrome App Learn more
iOS Learn more
PlayStation 4
Other
Name
Spring Security Book Web Client
Restrictions
Enter JavaScript origins, redirect URIs, or both
Authorized JavaScript origins
For use with requests from a browser. This is the origin URI of the client application. It can't contain a wildcard (https://*.example.com) or a path (https://example.com/subdir). If you're using a nonstandard port, you must include it in the origin URI.
http://localhost:8080
https://www.example.com
Authorized redirect URIs
For use with requests from a web server. This is the path in your application that users are redirected to after they have authenticated with Google. The path will be appended with the authorization code for access. Must have a protocol. Cannot contain URL fragments or relative paths. Cannot be a public IP address.
http://localhost:8080/login/oauth2/code/google
https://www.example.com/oauth2callback
Create
Cancel


Google APIs
spring-security-packtpub-...
APIs & Services
Dashboard
Library
Credentials
Credentials
Credentials
OAuth consent screen
Domain verification
Create credentials
Delete
Create credentials to access your enabled APIs. Refer to the API documentation for details.
OAuth 2.0 client IDs
Name
Spring Security Book Web Client
OAuth client
Here is your client ID
1085570125650-18j2r88b5i5gbe3vkhtlf8j7u3hvdu78.apps.googleusercontent.c
Here is your client secret
MdtcKp-ArG51FeqfAUw4K8Mp
OK


Spring Security
localhost:8080
Spring Security OAuth and OIDC Sample
Welcome 107140268802137904820 into Spring Boot OAuth and OIDC.

Chapter 4: Authentication Using CAS and JAAS

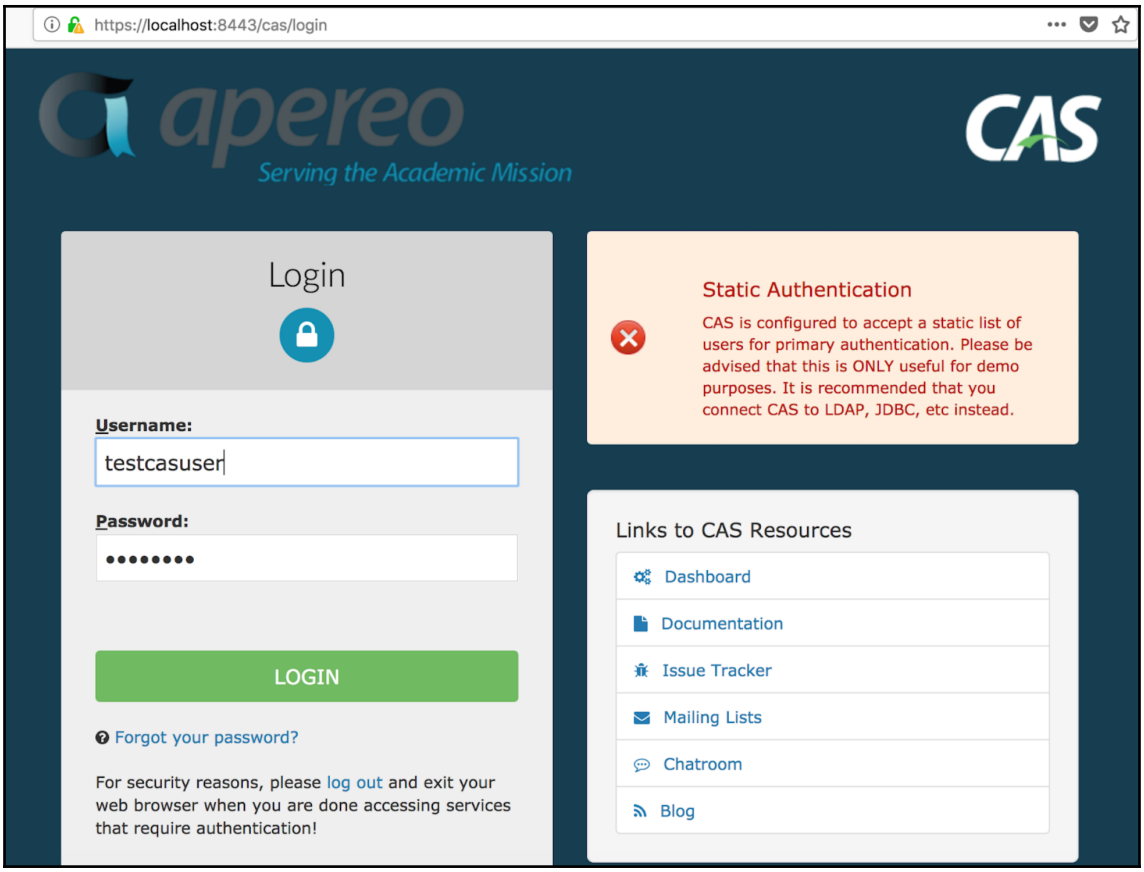

CAS
Clients
Apache
App
Java
App
Google
App
CAS Protocol
SAML Protoccol
OAuth
CAS
Server
Spring MVC / Webflow
Ticketing
Authentication
LDAP
Authentication
Handler
Database
Authentication
Handler
SPNEGO
Authentication
Handler

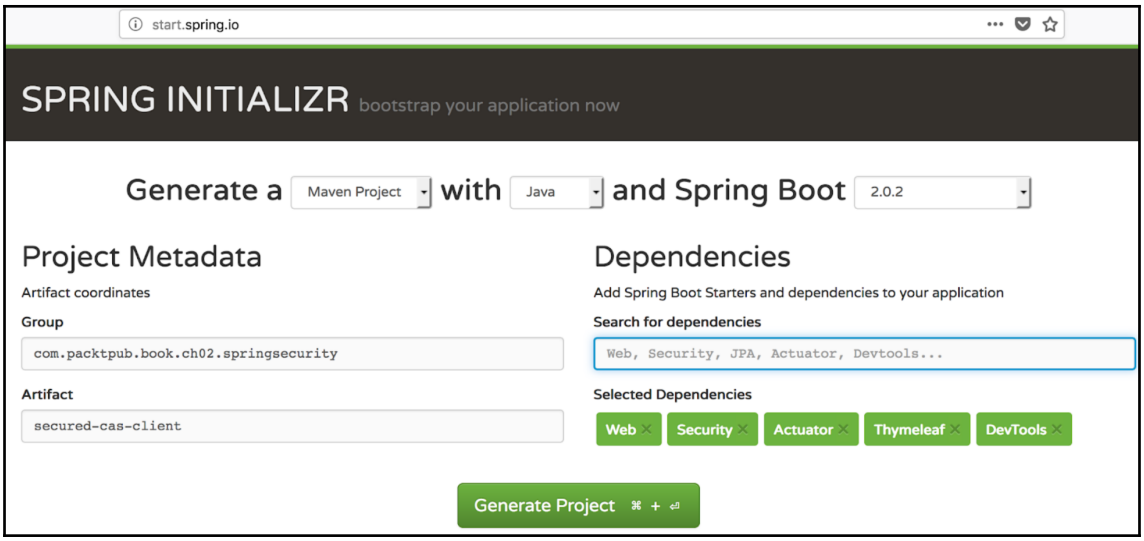


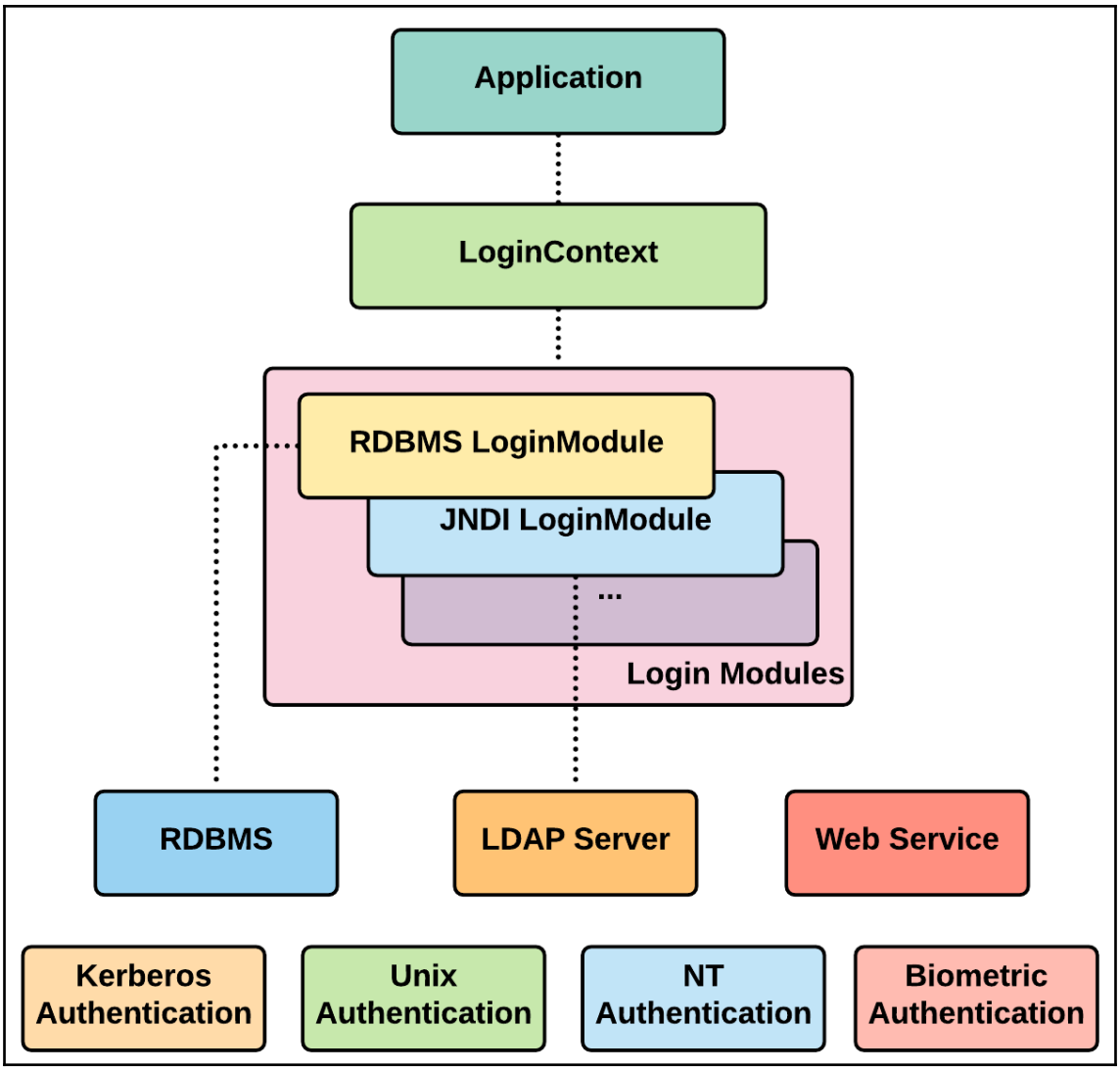

```
➜  cas git:(master) ✗ keytool -genkey -keyalg RSA -alias thekeystore -keystore thekeystore -storepass password -validity 360 -keysize 2048
What is your first and last name?
  [Unknown]:  localhost
What is the name of your organizational unit?
  [Unknown]:  localhost
What is the name of your organization?
  [Unknown]:  localhost
What is the name of your City or Locality?
  [Unknown]:  Dubai
What is the name of your State or Province?
  [Unknown]:  Dubai
What is the two-letter country code for this unit?
  [Unknown]:  AE
Is CN=localhost, OU=localhost, O=localhost, L=Dubai, ST=Dubai, C=AE correct?
  [no]:  yes

Enter key password for <thekeystore>
        (RETURN if same as keystore password):
➜  cas git:(master) ✗
```

```
➜ cas git:(master) ✗ keytool -import -alias thekeystore -storepass password -file thekeystore.crt -keystore "$(/usr/libexec/java_home)\jre\lib\security\cacerts'

Owner: CN=localhost, OU=localhost, O=localhost, L=Dubai, ST=Dubai, C=AE
Issuer: CN=localhost, OU=localhost, O=localhost, L=Dubai, ST=Dubai, C=AE
Serial number: 32febfab
Valid from: Fri May 11 18:45:42 GST 2018 until: Mon May 06 18:45:42 GST 2019
Certificate fingerprints:
         MD5:  BF:50:27:BD:3C:D2:B4:46:B8:CA:51:A1:1B:57:BA:2A
         SHA1: 7B:2B:6A:EC:96:DB:ED:74:7B:96:20:CA:55:23:F0:30:65:85:40:AF
         SHA256: B7:87:40:39:97:E7:91:DD:A2:55:8B:06:B7:05:87:38:C4:F3:8D:26:FF:34:CE:4B:76:03:23:CD:AD:2C:1E:49
         Signature algorithm name: SHA256withRSA
         Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 6F 8B 0D 22 58 35 85 06   6B 61 18 32 63 6D 31 C4  o.."X5..ka.2cm1.
0010: A4 9F 3E 6C                                        ..>l
]
]

Trust this certificate? [no]:  yes
Certificate was added to keystore
➜ cas git:(master) ✗
```

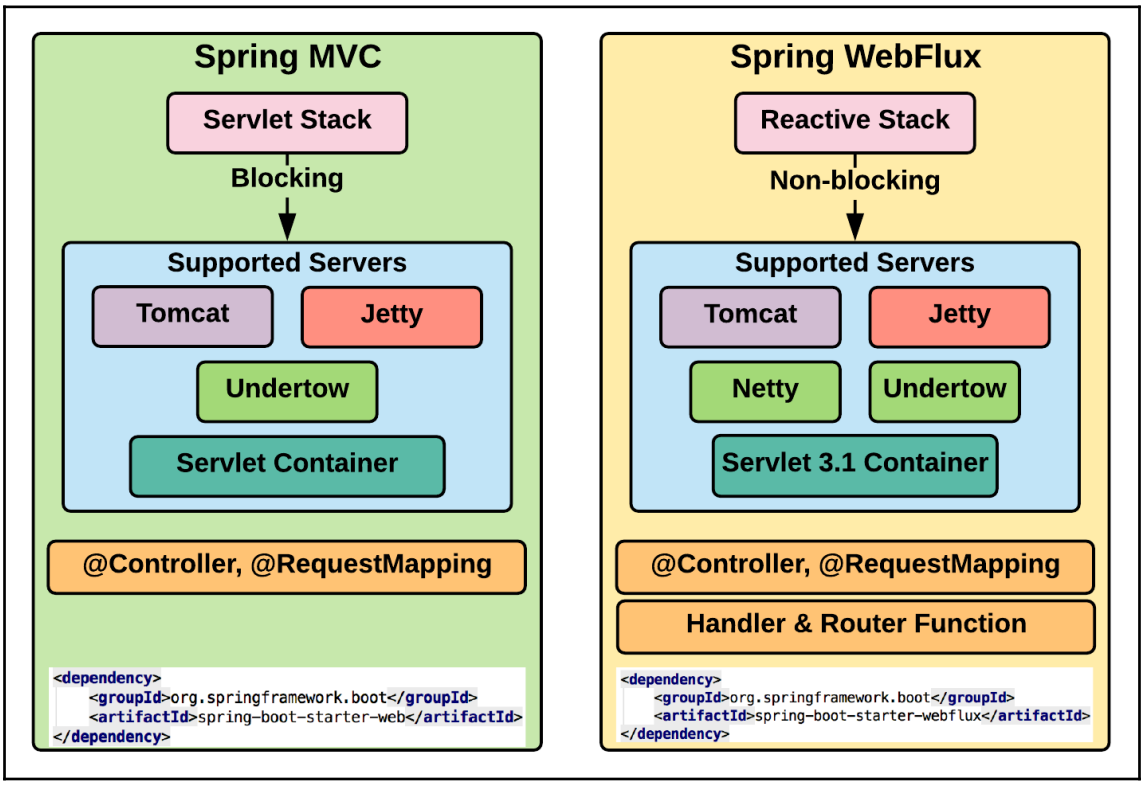
```
2018-05-08 22:04:21,806 INFO [org.apereo.cas.support.events.listener.DefaultCasEventListener] - <


  ____     ______       _       ____    __   __
 |  _ \   | ____|     / \     |  _ \   \ \ / /
 | |_) |  |  _|      / _ \    | | | |   \ V /
 |  _ <   | |___    / ___ \   | |_| |    | |
 |_| \_\  |_____|  /_/   \_\  |____/     |_|


>
2018-05-08 22:04:21,806 INFO [org.apereo.cas.support.events.listener.DefaultCasEventListener] - <>
```

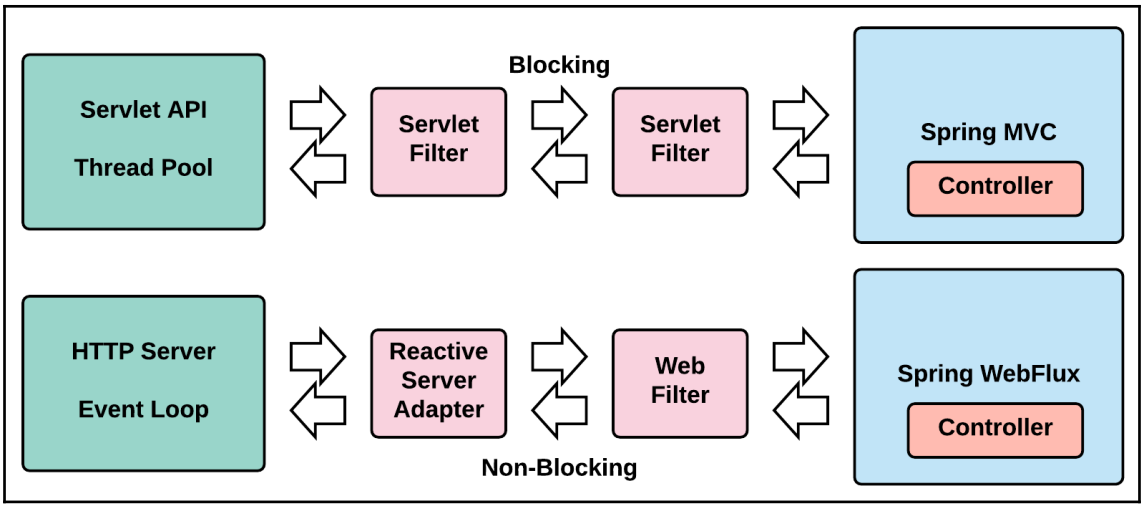

https://localhost:8443/cas/login
apereo
Serving the Academic Mission
CAS
Login
Username:
testcasuser
Password:
LOGIN
Forgot your password?
For security reasons, please log out and exit your web browser when you are done accessing services that require authentication!
Static Authentication
CAS is configured to accept a static list of users for primary authentication. Please be advised that this is ONLY useful for demo purposes. It is recommended that you connect CAS to LDAP, JDBC, etc instead.
Links to CAS Resources
Dashboard
Documentation
Issue Tracker
Mailing Lists
Chatroom
Blog

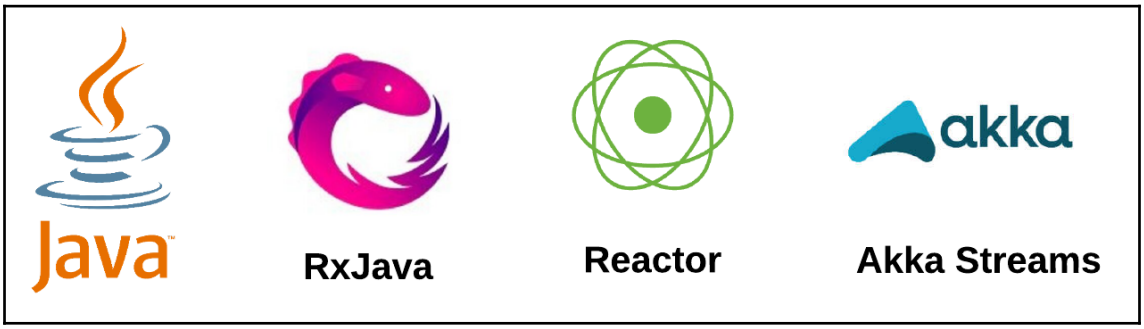

start.spring.io
SPRING INITIALIZR bootstrap your application now
Generate a Maven Project with Java and Spring Boot 2.0.2
Project Metadata
Artifact coordinates
Group
com.packtpub.book.ch02.springsecurity
Artifact
secured-cas-client
Dependencies
Add Spring Boot Starters and dependencies to your application
Search for dependencies
Web, Security, JPA, Actuator, Devtools...
Selected Dependencies
Web
Security
Actuator
Thymeleaf
DevTools
Generate Project ⌘ + ⏎

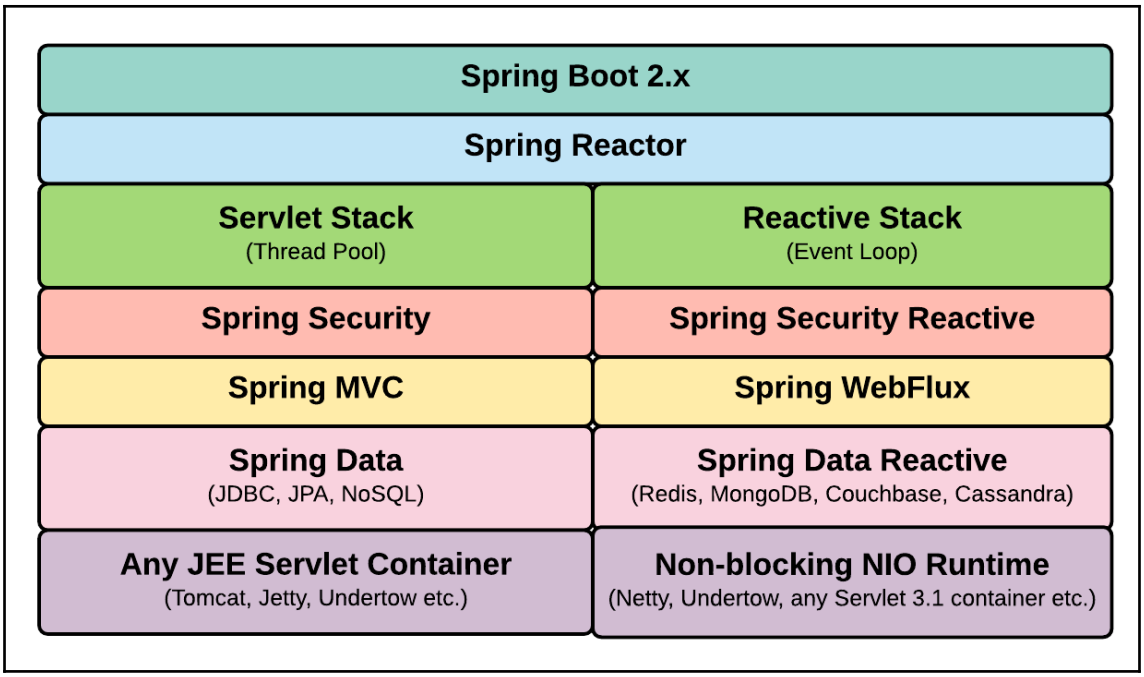

Application
LoginContext
RDBMS LoginModule
JNDI LoginModule
...
Login Modules
RDBMS
LDAP Server
Web Service
Kerberos Authentication
Unix Authentication
NT Authentication
Biometric Authentication

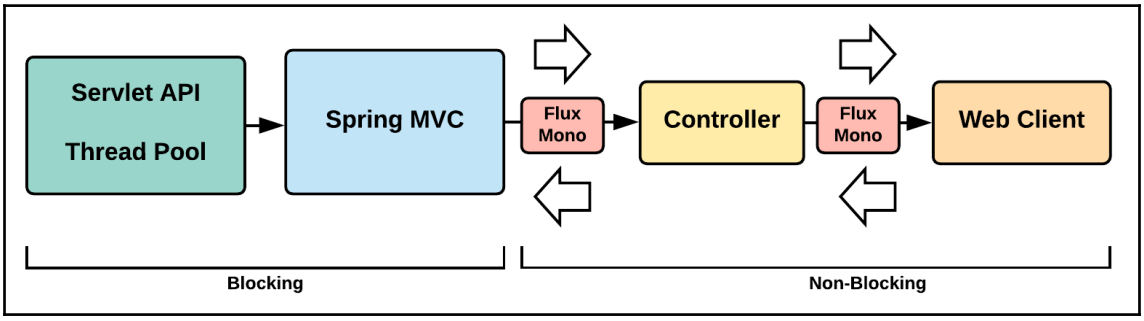
Chapter 5: Integrating with Spring WebFlux

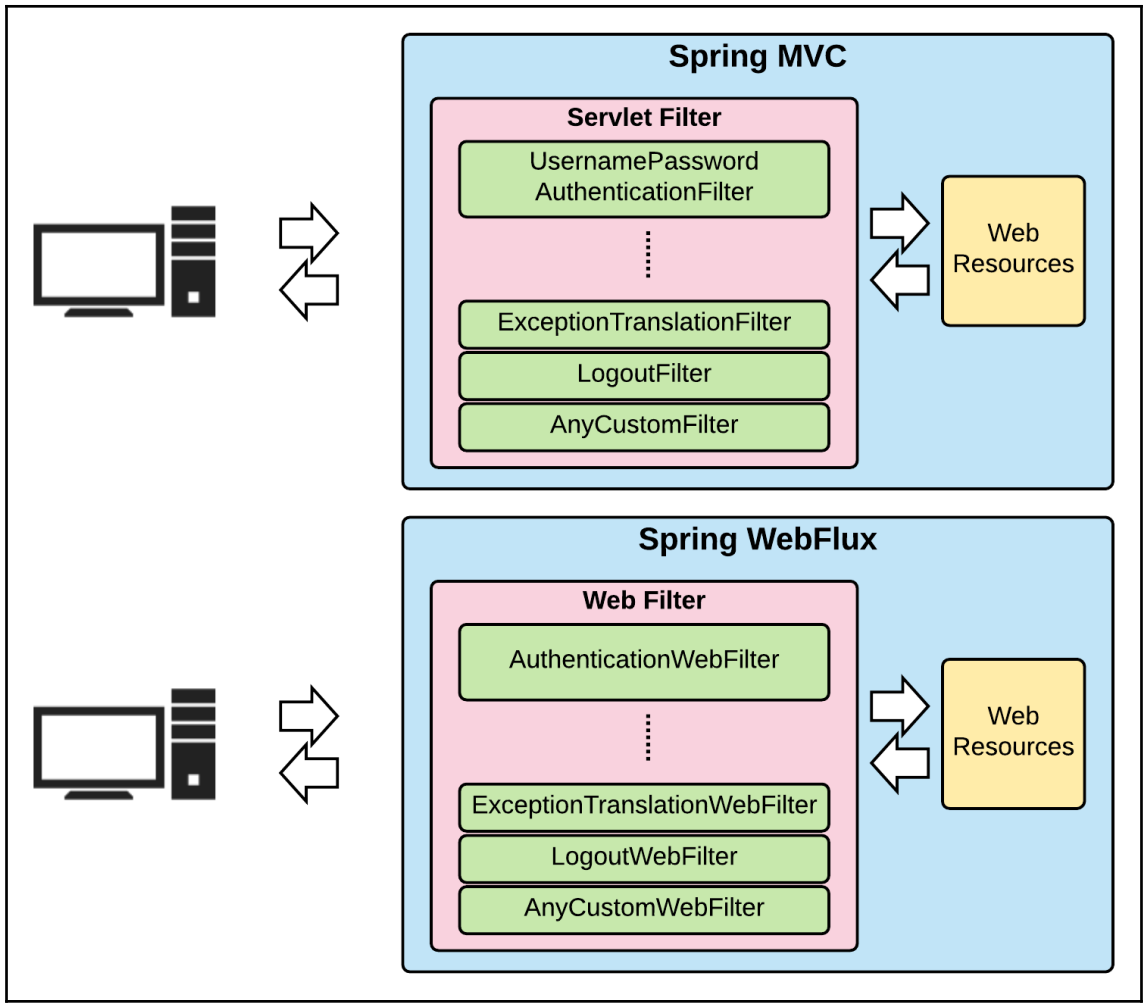


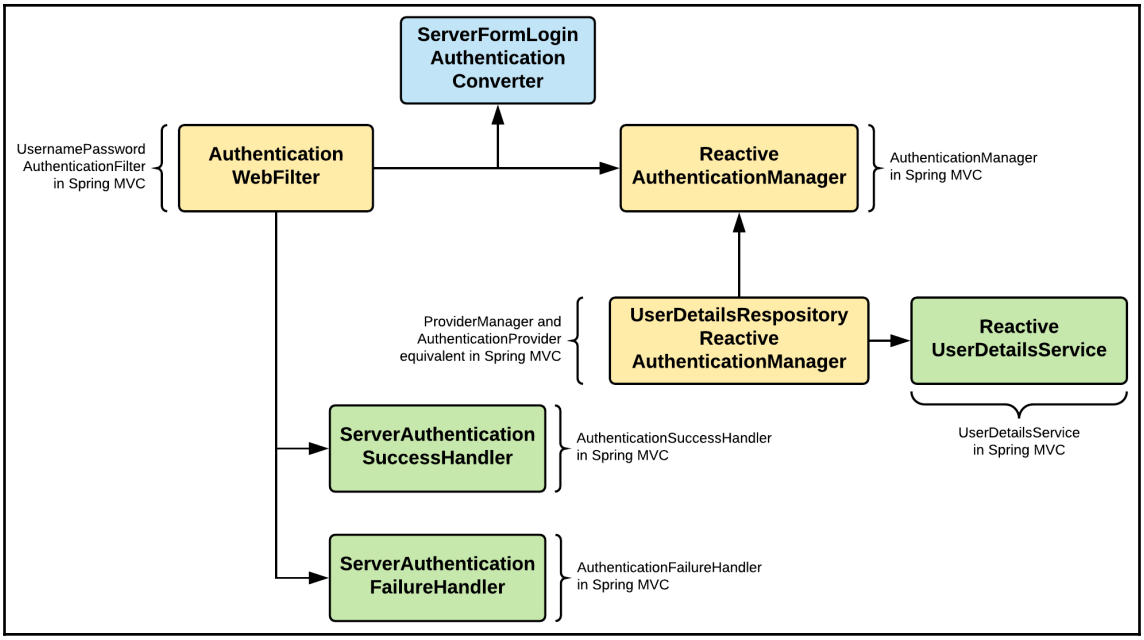

Blocking
Servlet API
Thread Pool
Servlet Filter
Servlet Filter
Spring MVC
Controller
HTTP Server
Event Loop
Reactive Server Adapter
Web Filter
Spring WebFlux
Controller
Non-Blocking

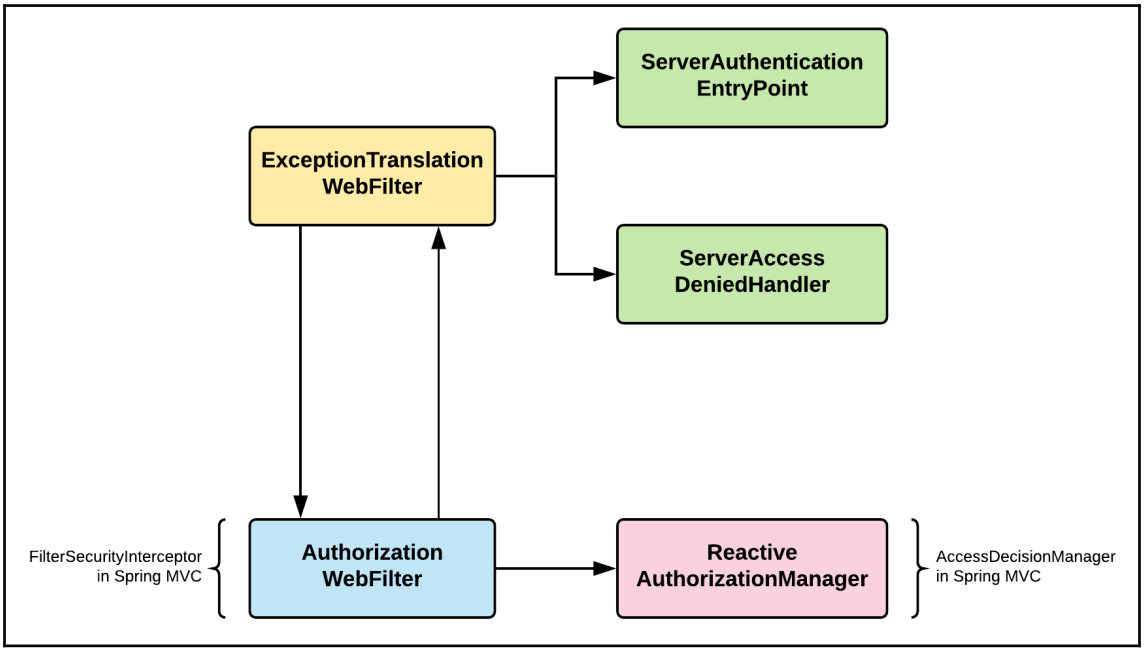

Java
RxJava
Reactor
akka
Akka Streams

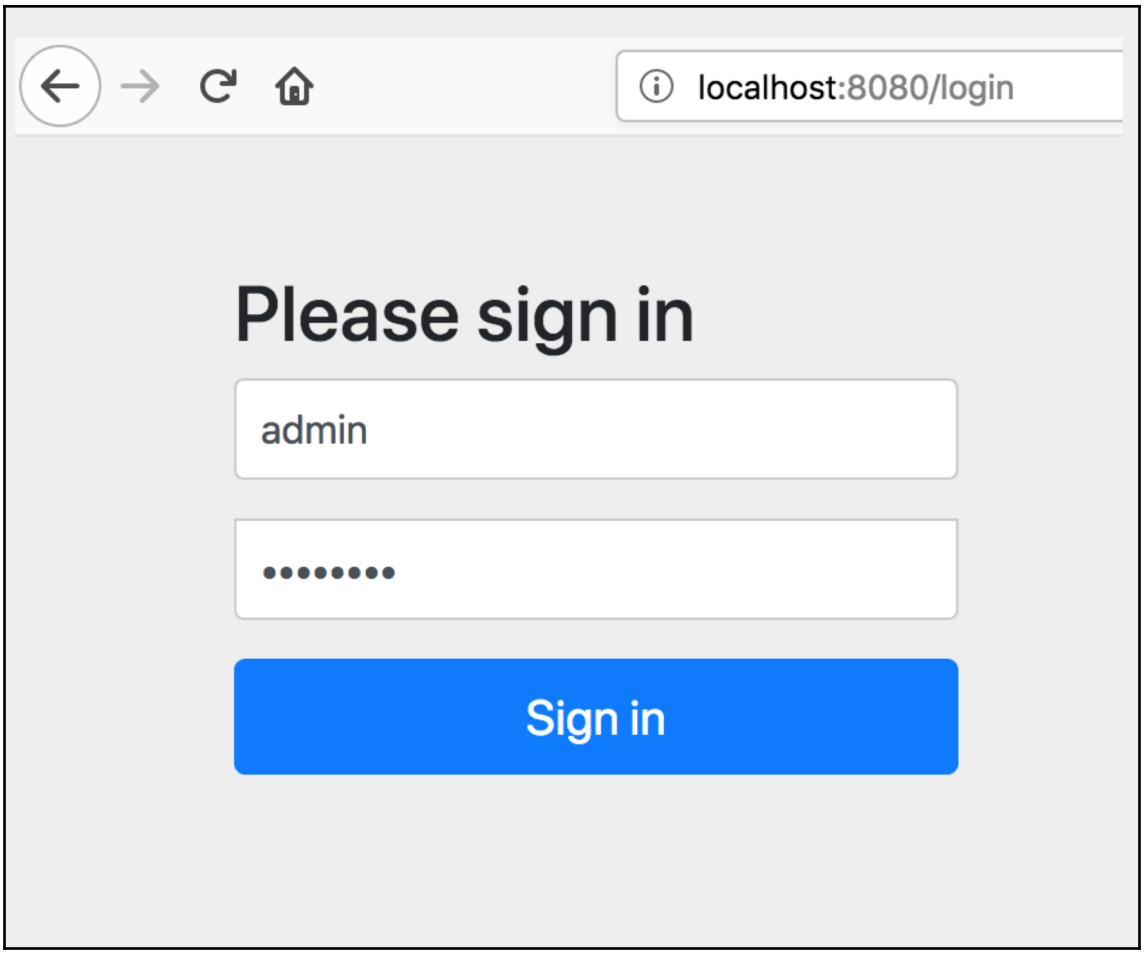

Spring Boot 2.x
Spring Reactor
Servlet Stack
(Thread Pool)
Reactive Stack
(Event Loop)
Spring Security
Spring Security Reactive
Spring MVC
Spring WebFlux
Spring Data
(JDBC, JPA, NoSQL)
Spring Data Reactive
(Redis, MongoDB, Couchbase, Cassandra)
Any JEE Servlet Container
(Tomcat, Jetty, Undertow etc.)
Non-blocking NIO Runtime
(Netty, Undertow, any Servlet 3.1 container etc.)

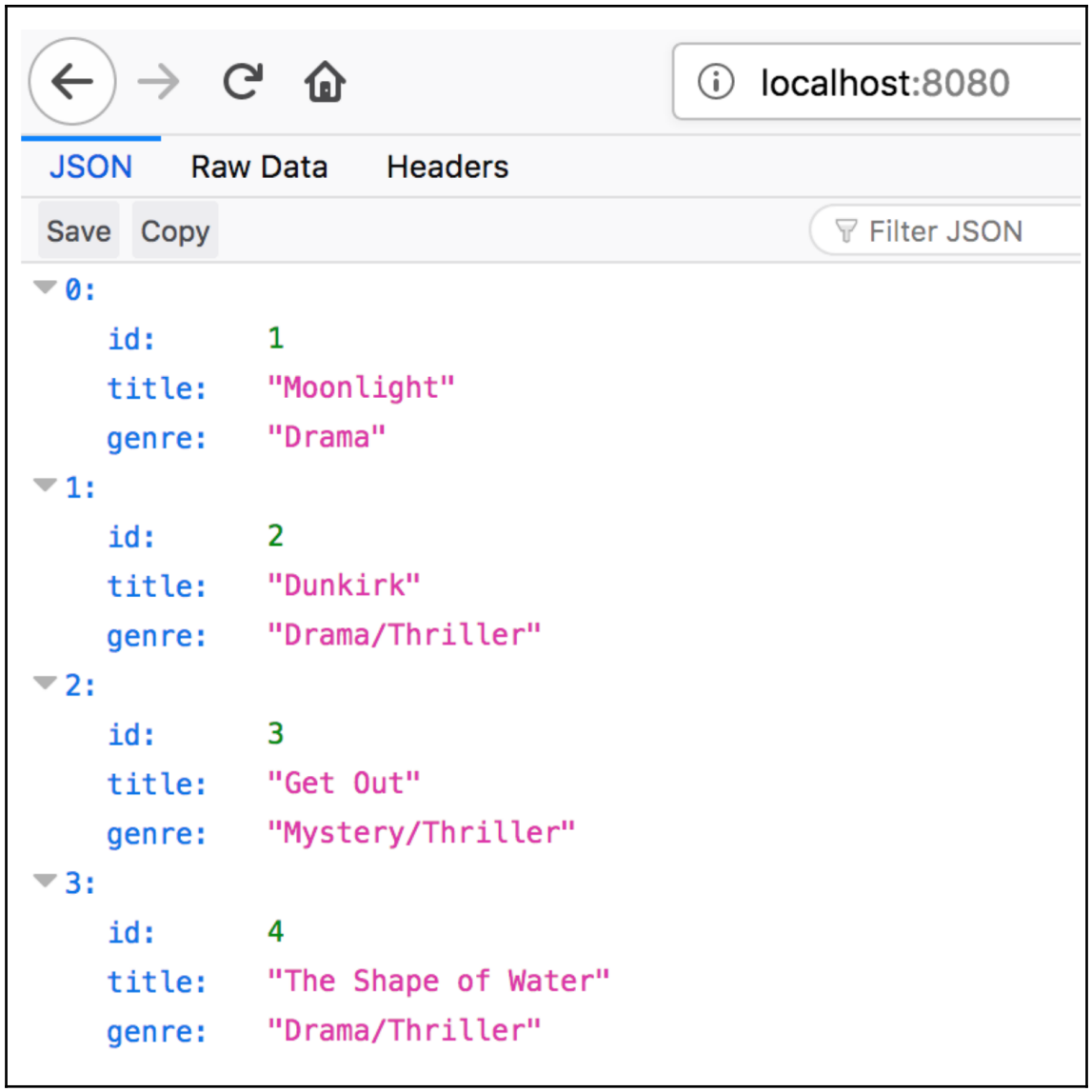

Servlet API
Thread Pool
Spring MVC
Flux
Mono
Controller
Flux
Mono
Web Client
Blocking
Non-Blocking


Spring MVC
Servlet Filter
UsernamePassword
AuthenticationFilter
ExceptionTranslationFilter
LogoutFilter
AnyCustomFilter
Web
Resources
Spring WebFlux
Web Filter
AuthenticationWebFilter
ExceptionTranslationWebFilter
LogoutWebFilter
AnyCustomWebFilter
Web
Resources


ServerFormLogin Authentication Converter
UsernamePassword AuthenticationFilter in Spring MVC
Authentication WebFilter
Reactive AuthenticationManager
AuthenticationManager in Spring MVC
ProviderManager and AuthenticationProvider equivalent in Spring MVC
UserDetailsRespository Reactive AuthenticationManager
Reactive UserDetailsService
UserDetailsService in Spring MVC
ServerAuthentication SuccessHandler
AuthenticationSuccessHandler in Spring MVC
ServerAuthentication FailureHandler
AuthenticationFailureHandler in Spring MVC

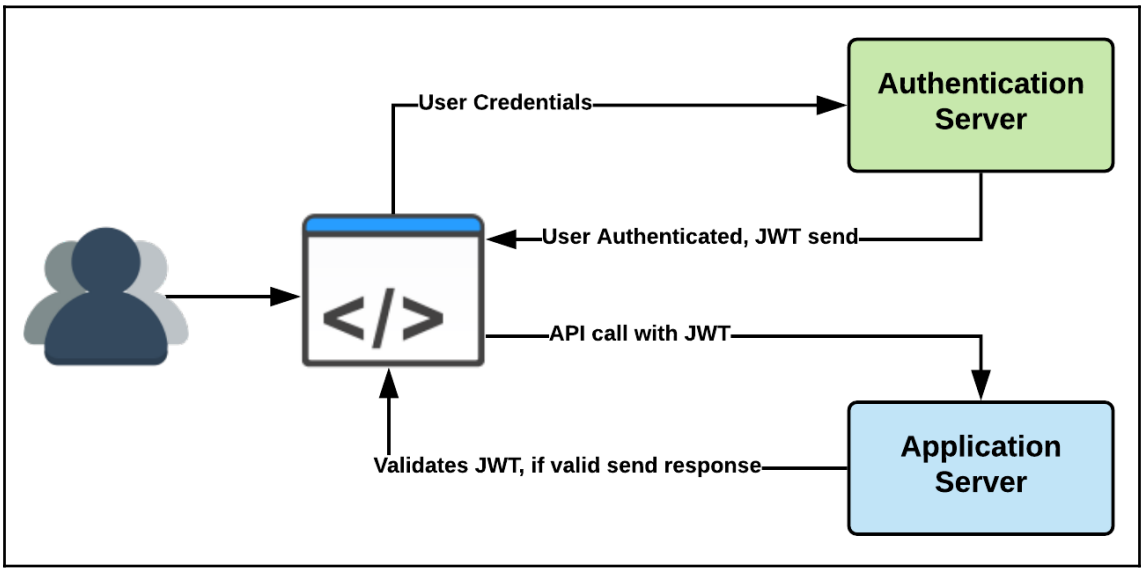

ServerAuthentication EntryPoint
ExceptionTranslation WebFilter
ServerAccess DeniedHandler
FilterSecurityInterceptor in Spring MVC
Authorization WebFilter
Reactive AuthorizationManager
AccessDecisionManager in Spring MVC

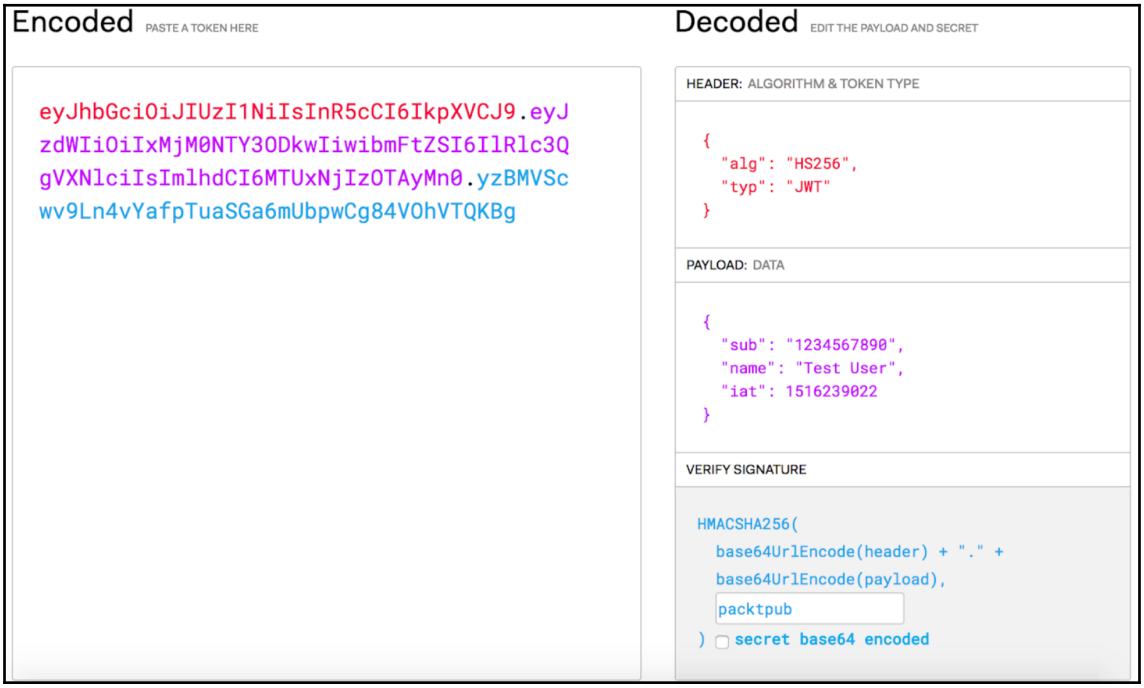

localhost:8080/login
Please sign in
admin
Sign in

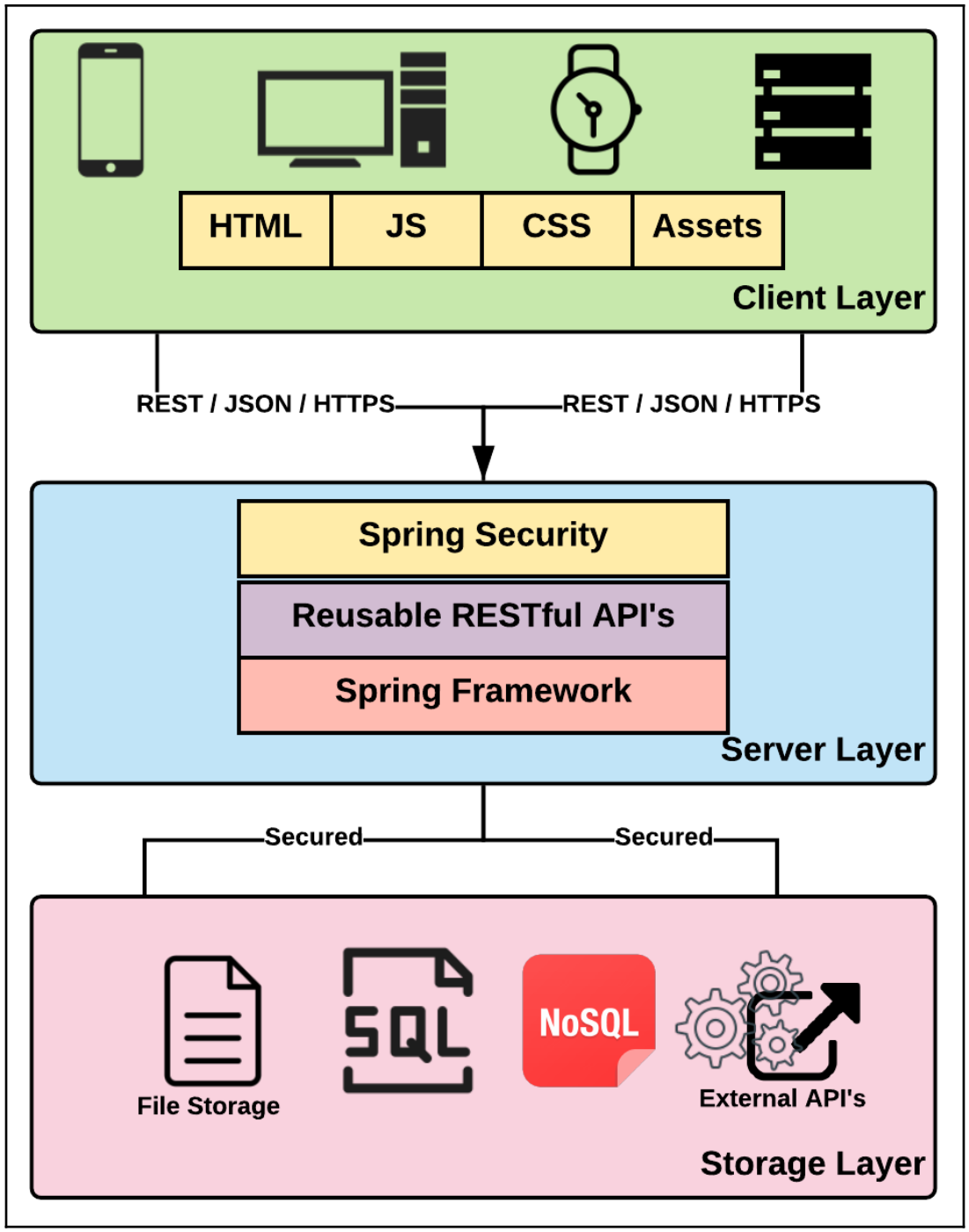

localhost:8080
JSON
Raw Data
Headers
Save
Copy
Filter JSON
0:
id: 1
title: "Moonlight"
genre: "Drama"
1:
id: 2
title: "Dunkirk"
genre: "Drama/Thriller"
2:
id: 3
title: "Get Out"
genre: "Mystery/Thriller"
3:
id: 4
title: "The Shape of Water"
genre: "Drama/Thriller"

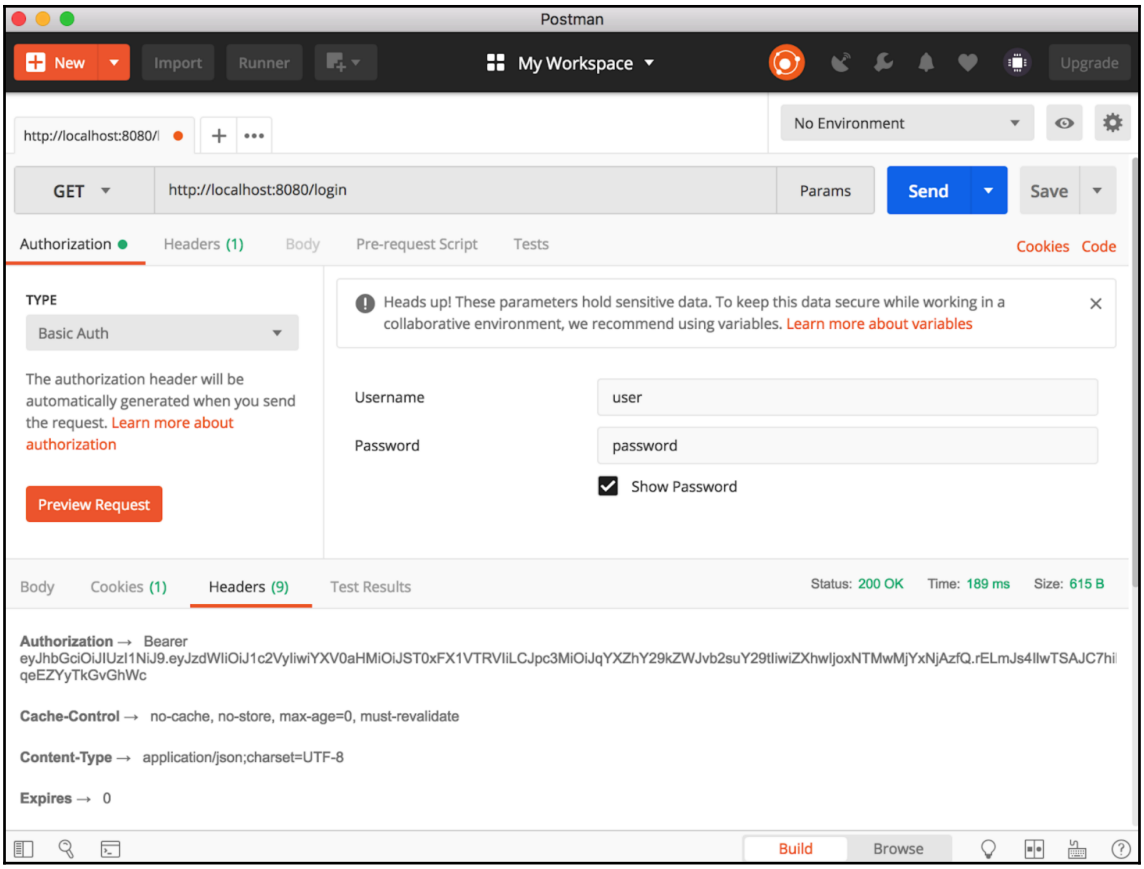
```
2018-06-03 20:25:47.372  INFO 23011 --- [ctor-http-nio-1] r.ipc.netty.tcp.BlockingNettyContext     : Started HttpServer on /0:0:0:0:0:0:0:0:8081
2018-06-03 20:25:47.373  INFO 23011 --- [           main] o.s.b.web.embedded.netty.NettyWebServer  : Netty started on port(s): 8081
2018-06-03 20:25:47.379  INFO 23011 --- [           main] c.packtpub.book.ch02.springsecurity.Run  : Started Run in 2.19 seconds (JVM running for 5.431)
Get All Movies
Movie(id=1, title=Moonlight, genre=Drama)
Movie(id=2, title=Dunkirk, genre=Drama/Thriller)
Movie(id=3, title=Get Out, genre=Mystery/Thriller)
Movie(id=4, title=The Shape of Water, genre=Drama/Thriller)
Get a Movie with id = 1
Movie(id=1, title=Moonlight, genre=Drama)
Save a Movie
Update a Movie
Movie(id=3, title=Black Panther, genre=Fantasy/Science)
Delete a Movie with id = 2
Get All Movies again
Movie(id=1, title=Moonlight, genre=Drama)
Movie(id=3, title=Get Out, genre=Mystery/Thriller)
Movie(id=4, title=The Shape of Water, genre=Drama/Thriller)
```

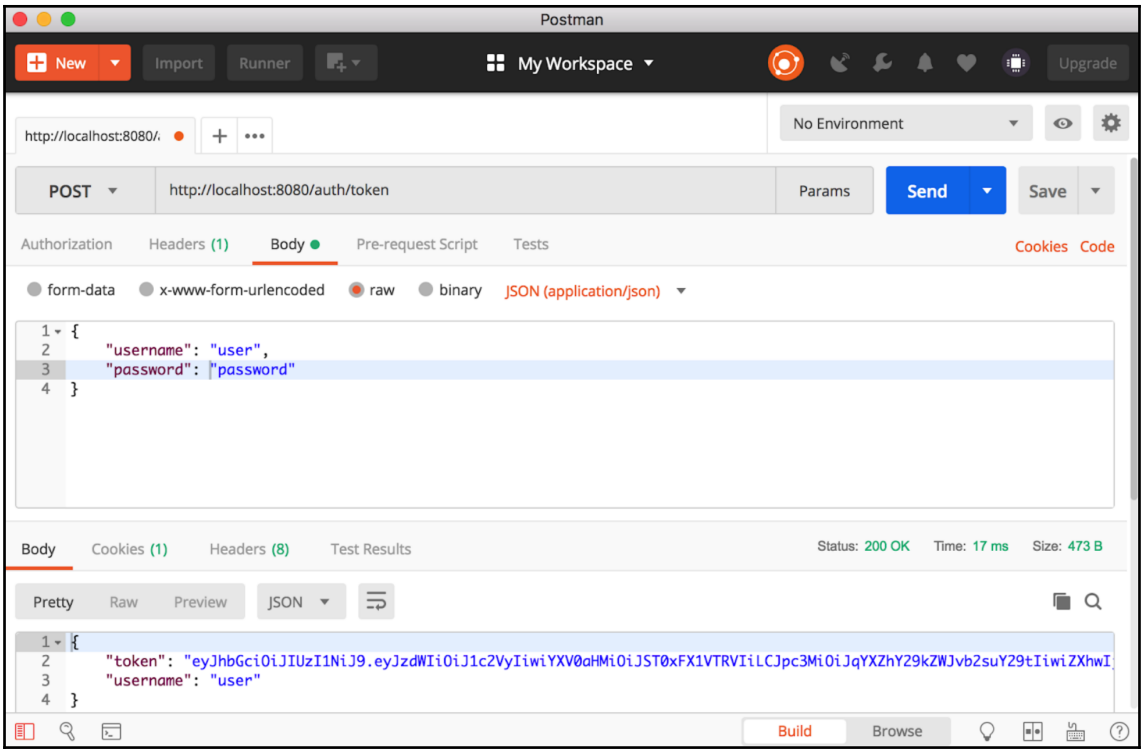
Chapter 6: REST API Security

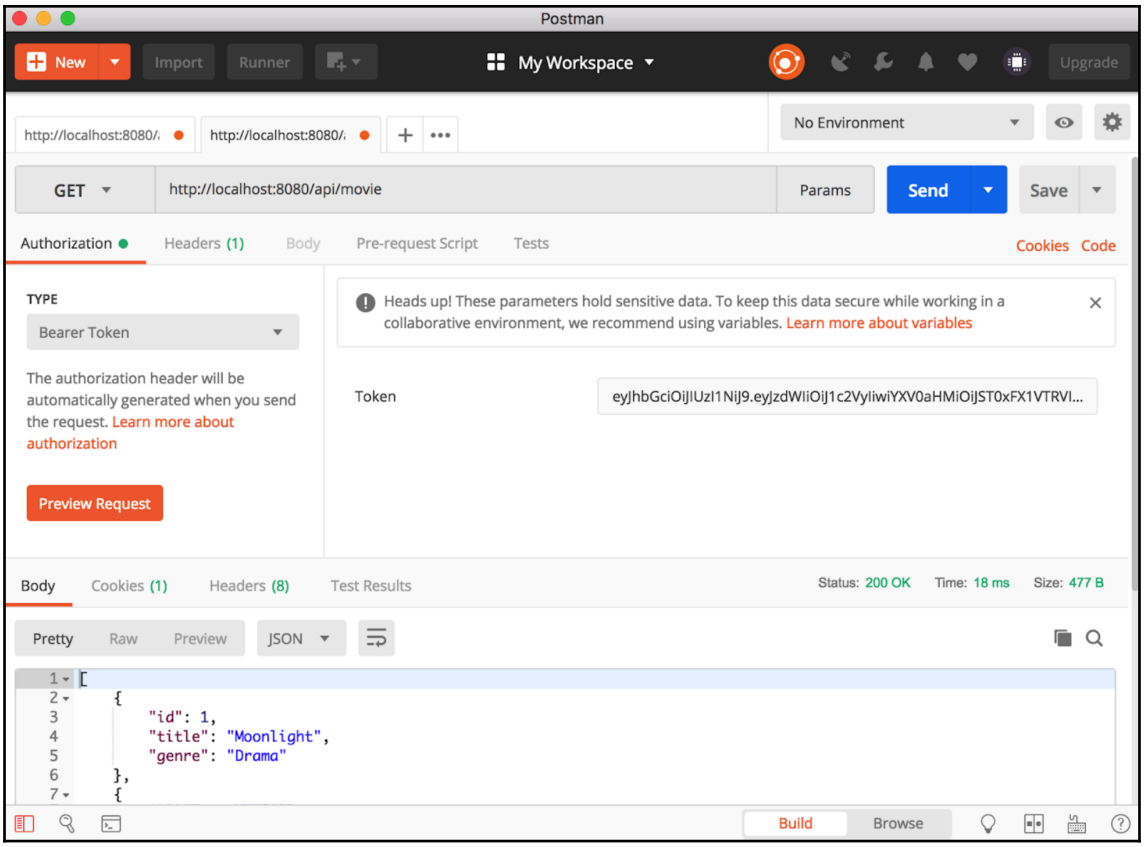


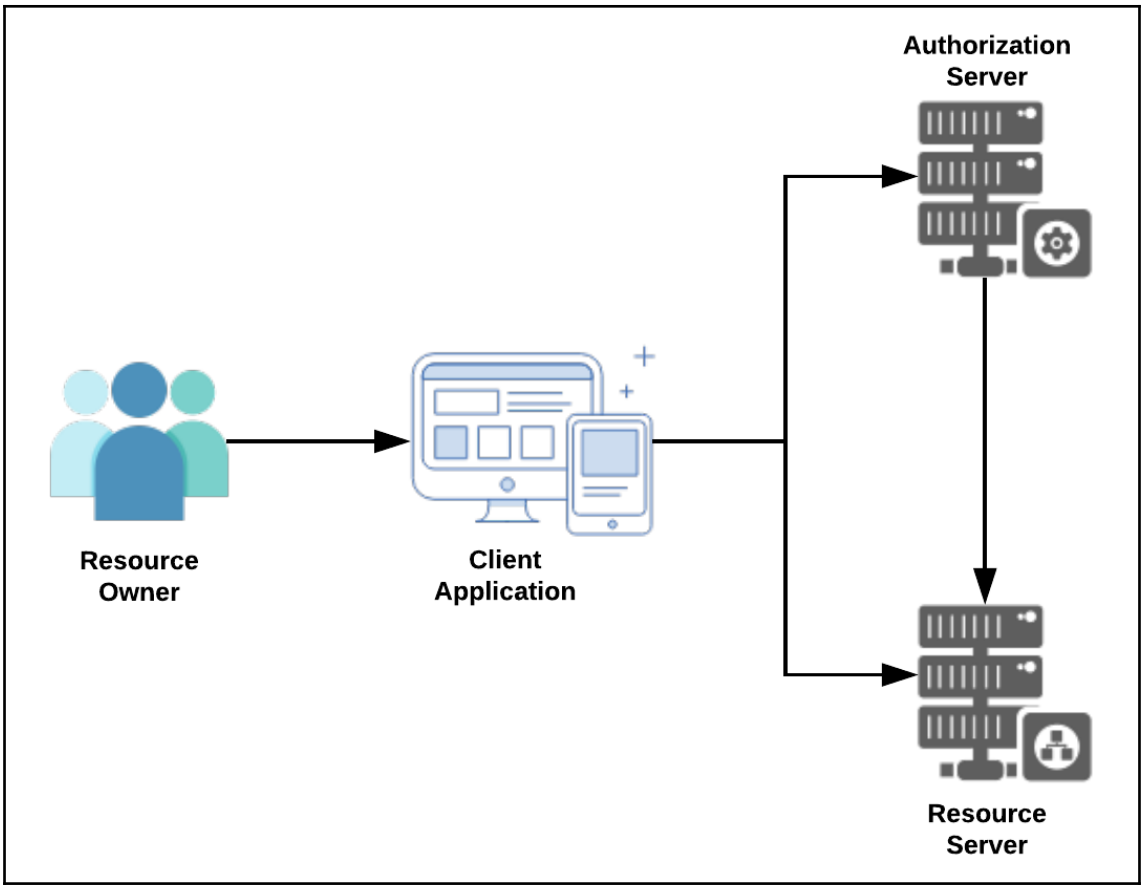


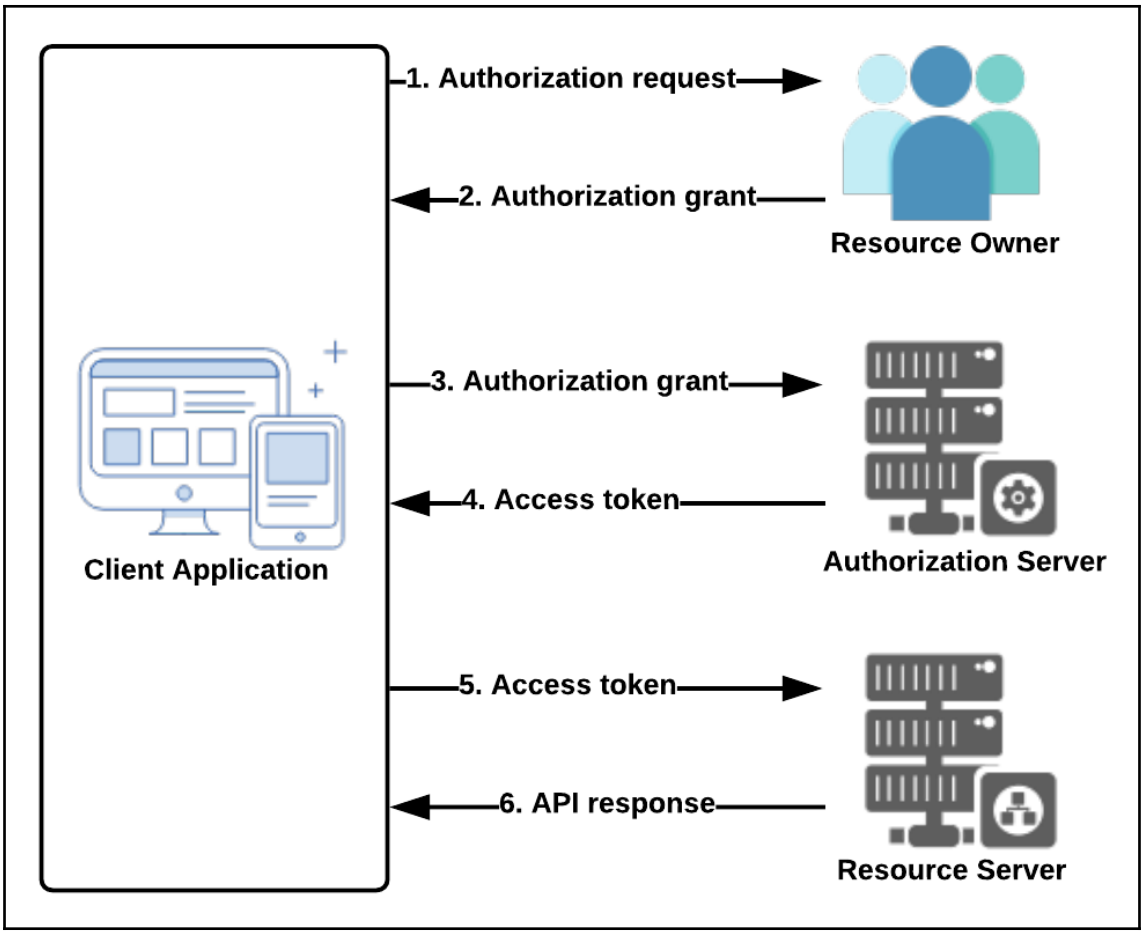

HTML
JS
CSS
Assets
Client Layer
REST / JSON / HTTPS
REST / JSON / HTTPS
Spring Security
Reusable RESTful API's
Spring Framework
Server Layer
Secured
Secured
File Storage
SQL
NoSQL
External API's
Storage Layer

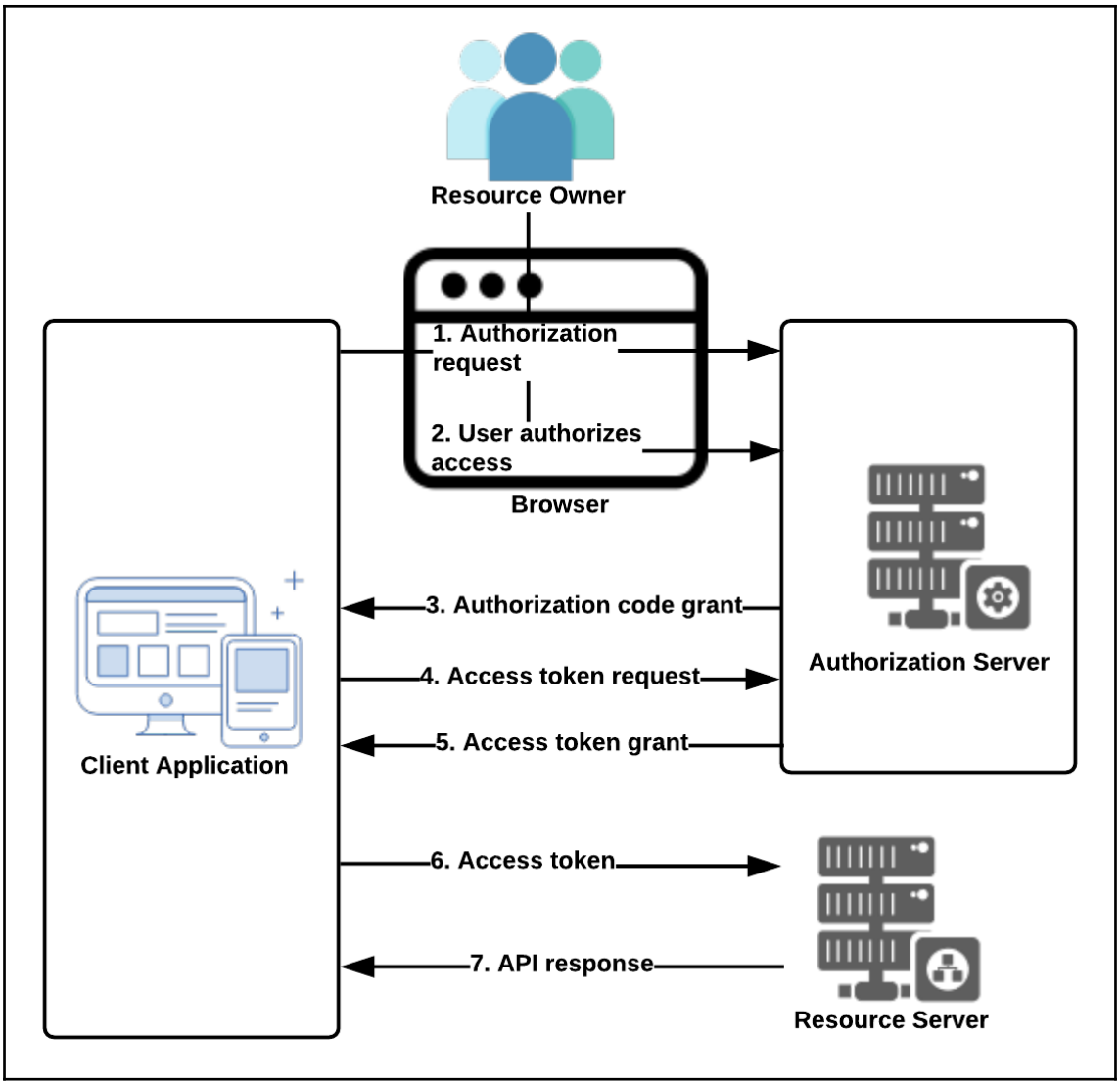

Postman
New
Import
Runner
My Workspace
Upgrade
http://localhost:8080/
No Environment
GET
http://localhost:8080/login
Params
Send
Save
Authorization
Headers (1)
Body
Pre-request Script
Tests
Cookies
Code
TYPE
Basic Auth
The authorization header will be automatically generated when you send the request. Learn more about authorization
Preview Request
Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables
Username
user
Password
password
Show Password
Body
Cookies (1)
Headers (9)
Test Results
Status: 200 OK
Time: 189 ms
Size: 615 B
Authorization → Bearer
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyIiwiYXV0aHMiOiJST0xFX1VTRVIiLCJpc3MiOiJqYXZhY29kZWJvb2suY29tIiwiZXhwIjoxNTMwMjYxNjAzfQ.rELmJs4IlwTSAJC7hi
qeEZYyTkGvGhWc
Cache-Control → no-cache, no-store, max-age=0, must-revalidate
Content-Type → application/json;charset=UTF-8
Expires → 0
Build
Browse

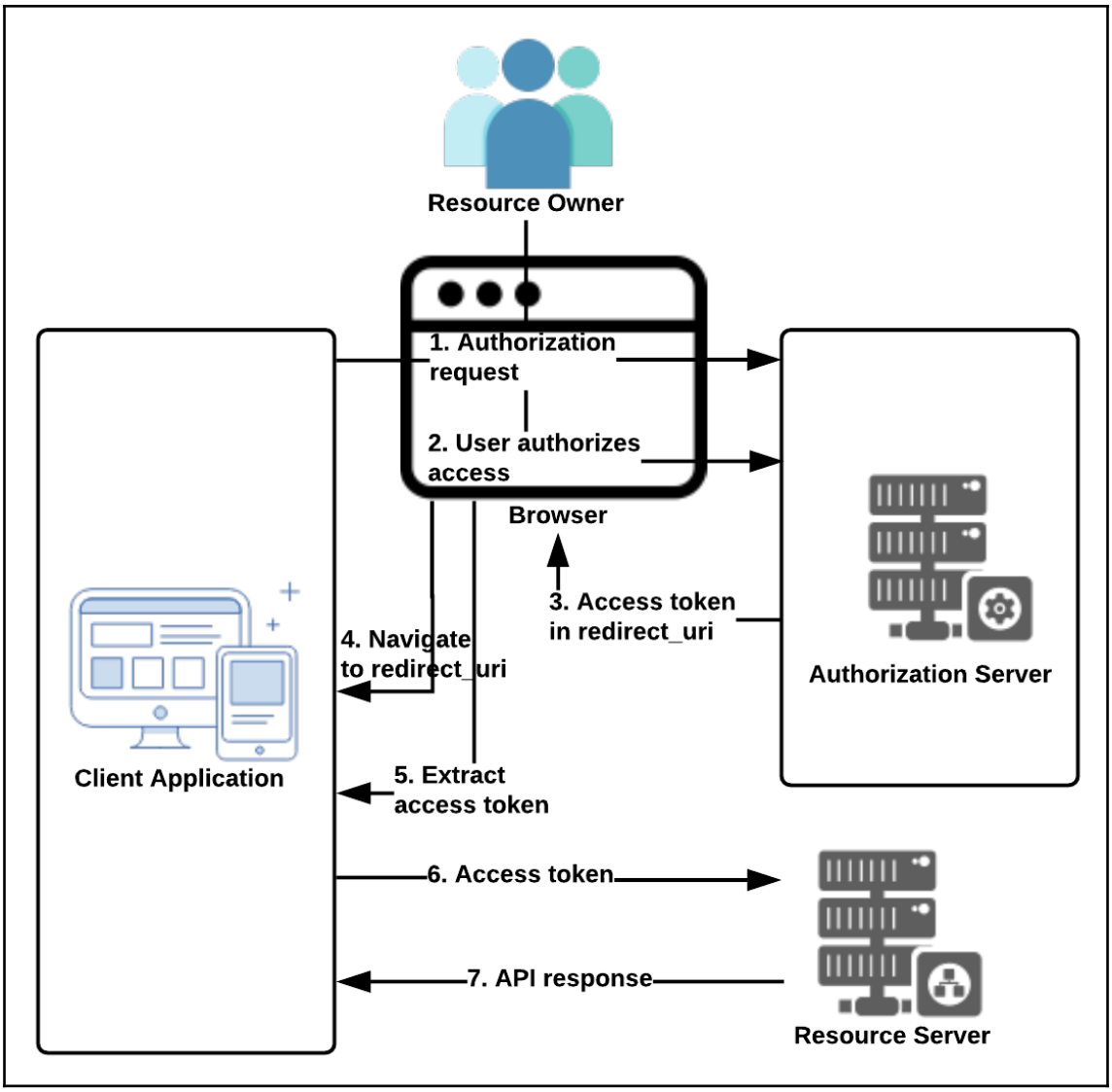

Postman
New
Import
Runner
My Workspace
Upgrade
http://localhost:8080/
No Environment
POST
http://localhost:8080/auth/token
Params
Send
Save
Authorization
Headers (1)
Body
Pre-request Script
Tests
Cookies
Code
form-data
x-www-form-urlencoded
raw
binary
JSON (application/json)
{
"username": "user",
"password": "password"
}
Body
Cookies (1)
Headers (8)
Test Results
Status: 200 OK
Time: 17 ms
Size: 473 B
Pretty
Raw
Preview
JSON
{
"token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyIiwiYXV0aHMiOiJST0xFX1VTRVIiLCJpc3MiOiJqYXZhY29kZWJvb2suY29tIiwiZXhwI
"username": "user"
}
Build
Browse

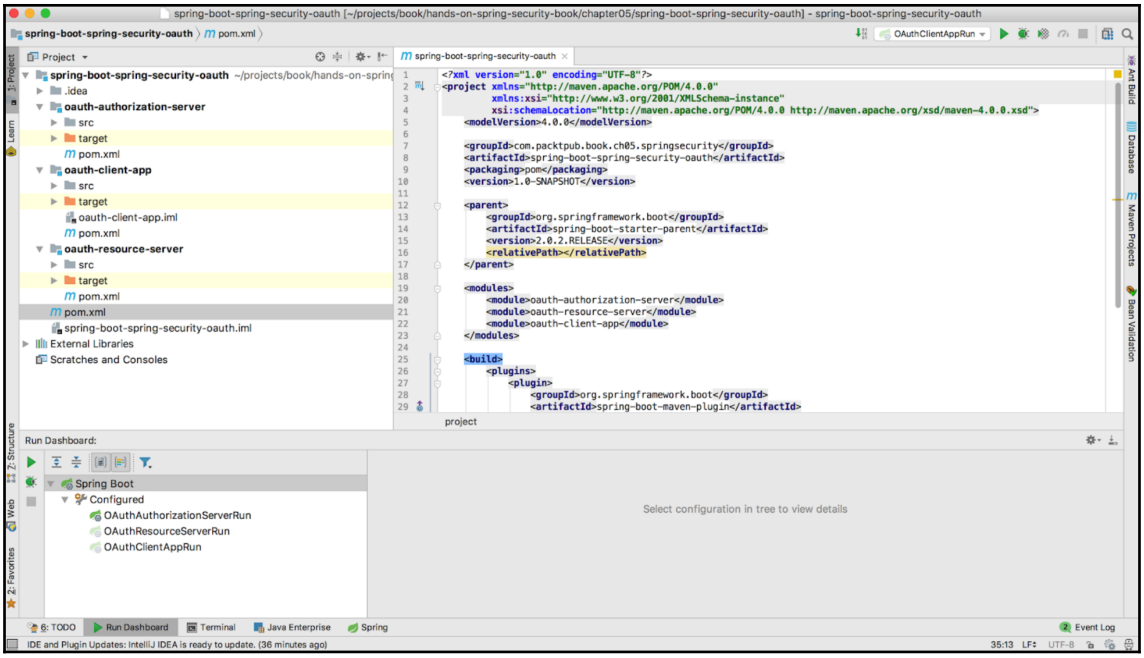

Postman
New
Import
Runner
My Workspace
Upgrade
http://localhost:8080/
http://localhost:8080/
No Environment
GET
http://localhost:8080/api/movie
Params
Send
Save
Authorization
Headers (1)
Body
Pre-request Script
Tests
Cookies
Code
TYPE
Bearer Token
The authorization header will be automatically generated when you send the request. Learn more about authorization
Preview Request
Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables
Token
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyIiwiYXV0aHMiOiJST0xFX1VTRVI...
Body
Cookies (1)
Headers (8)
Test Results
Status: 200 OK
Time: 18 ms
Size: 477 B
Pretty
Raw
Preview
JSON
[
{
"id": 1,
"title": "Moonlight",
"genre": "Drama"
},
{
Build
Browse

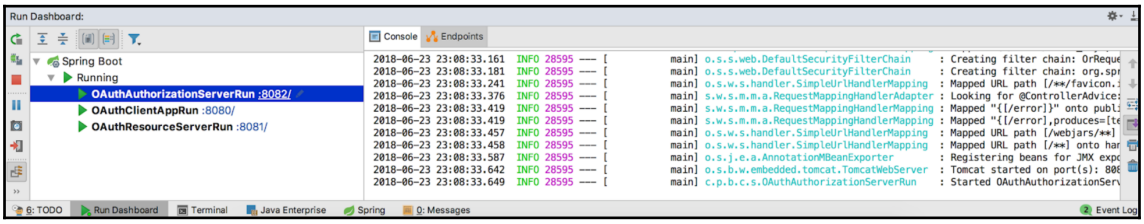

Authorization
Server
Resource
Owner
Client
Application
Resource
Server

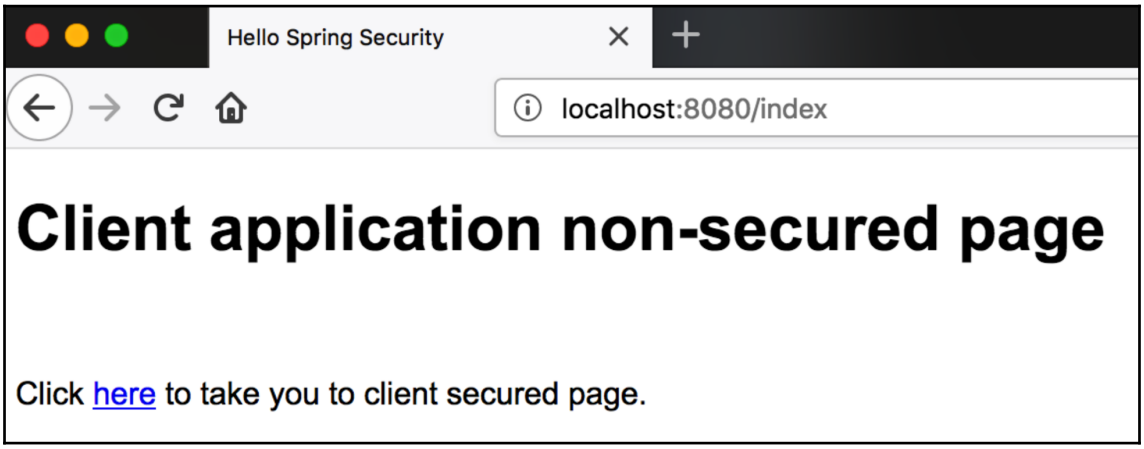

1. Authorization request
2. Authorization grant
Resource Owner
3. Authorization grant
4. Access token
Client Application
Authorization Server
5. Access token
6. API response
Resource Server

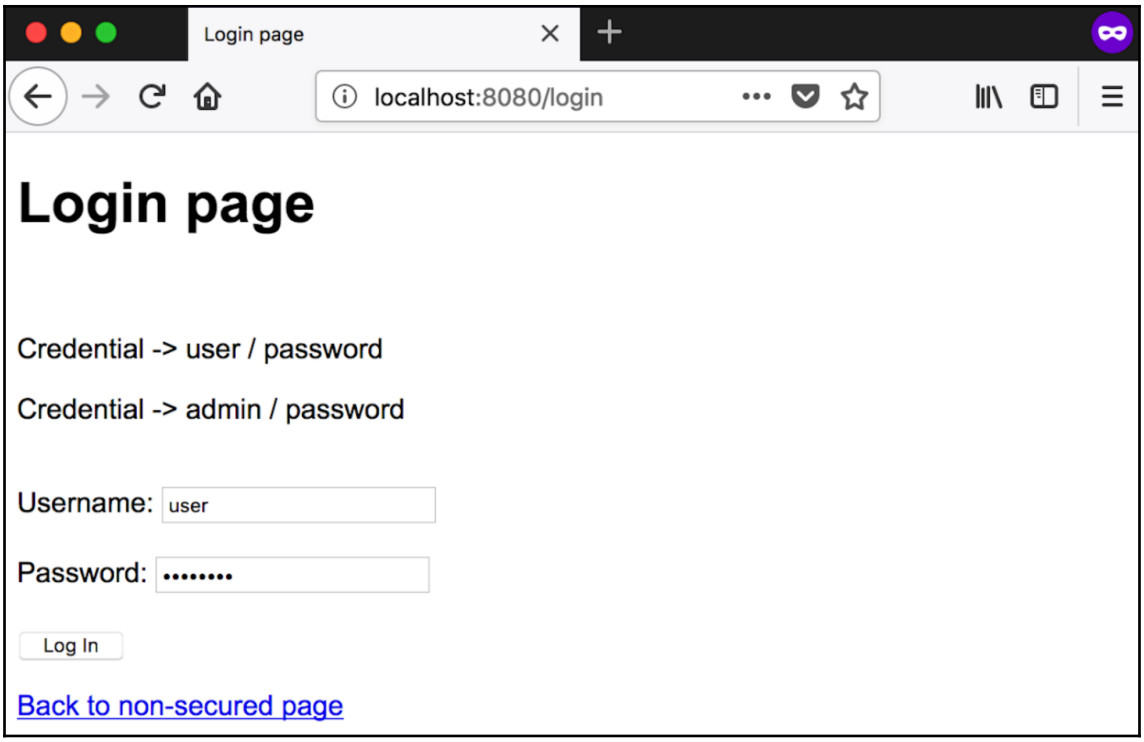

Resource Owner
1. Authorization request
2. User authorizes access
Browser
Client Application
3. Authorization code grant
4. Access token request
5. Access token grant
Authorization Server
6. Access token
7. API response
Resource Server

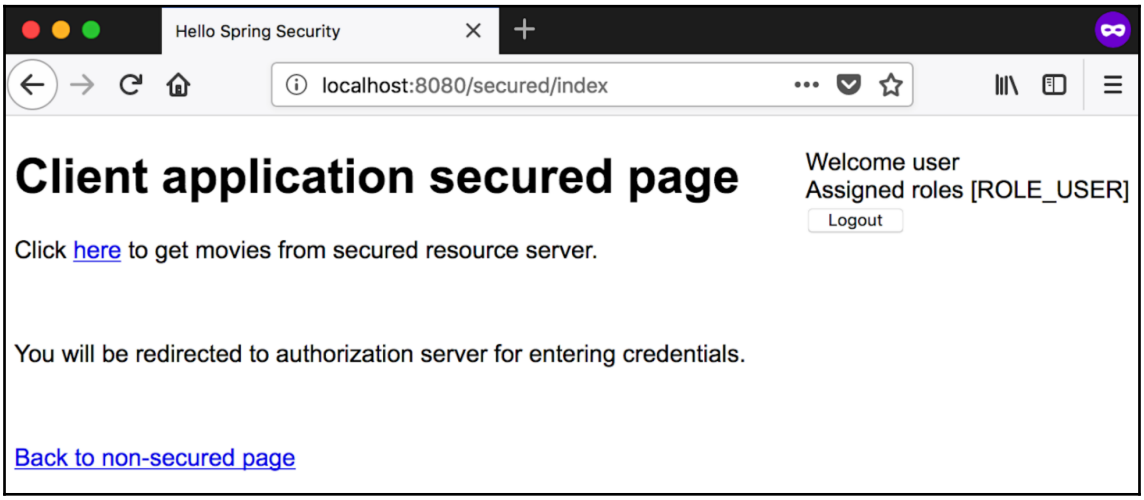

Resource Owner
1. Authorization request
2. User authorizes access
Browser
3. Access token in redirect_uri
Authorization Server
4. Navigate to redirect_uri
Client Application
5. Extract access token
6. Access token
7. API response
Resource Server

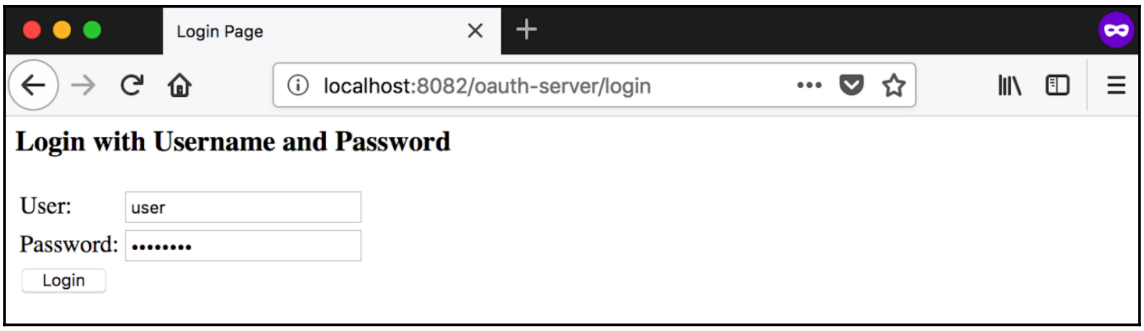

spring-boot-spring-security-oauth [~/projects/book/hands-on-spring-security-book/chapter05/spring-boot-spring-security-oauth] - spring-boot-spring-security-oauth
spring-boot-spring-security-oauth
oauth-authorization-server
oauth-client-app
oauth-resource-server
pom.xml
External Libraries
Scratches and Consoles
<groupId>com.packtpub.book.ch05.springsecurity</groupId>
<artifactId>spring-boot-spring-security-oauth</artifactId>
<packaging>pom</packaging>
<version>1.0-SNAPSHOT</version>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>2.0.2.RELEASE</version>
<module>oauth-authorization-server</module>
<module>oauth-resource-server</module>
<module>oauth-client-app</module>
Run Dashboard:
Spring Boot
Configured
OAuthAuthorizationServerRun
OAuthResourceServerRun
OAuthClientAppRun
Select configuration in tree to view details

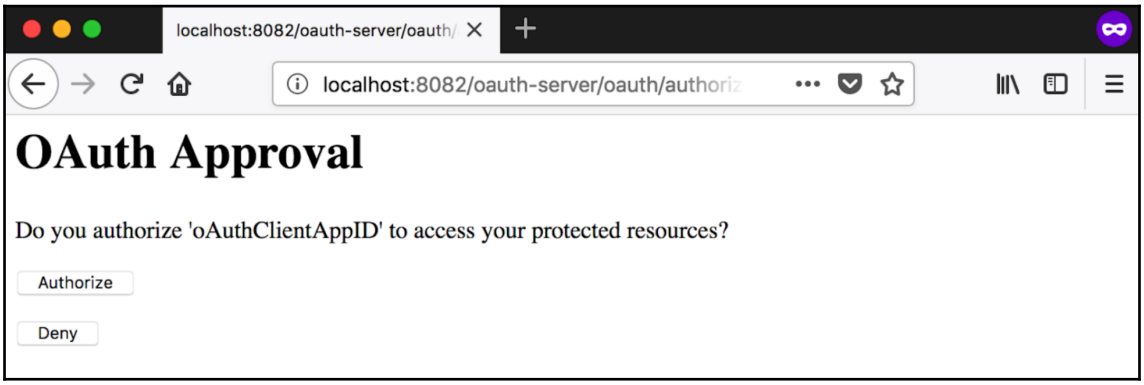

Run Dashboard:
Spring Boot
Running
OAuthAuthorizationServerRun :8082/
OAuthClientAppRun :8080/
OAuthResourceServerRun :8081/
Console
Endpoints

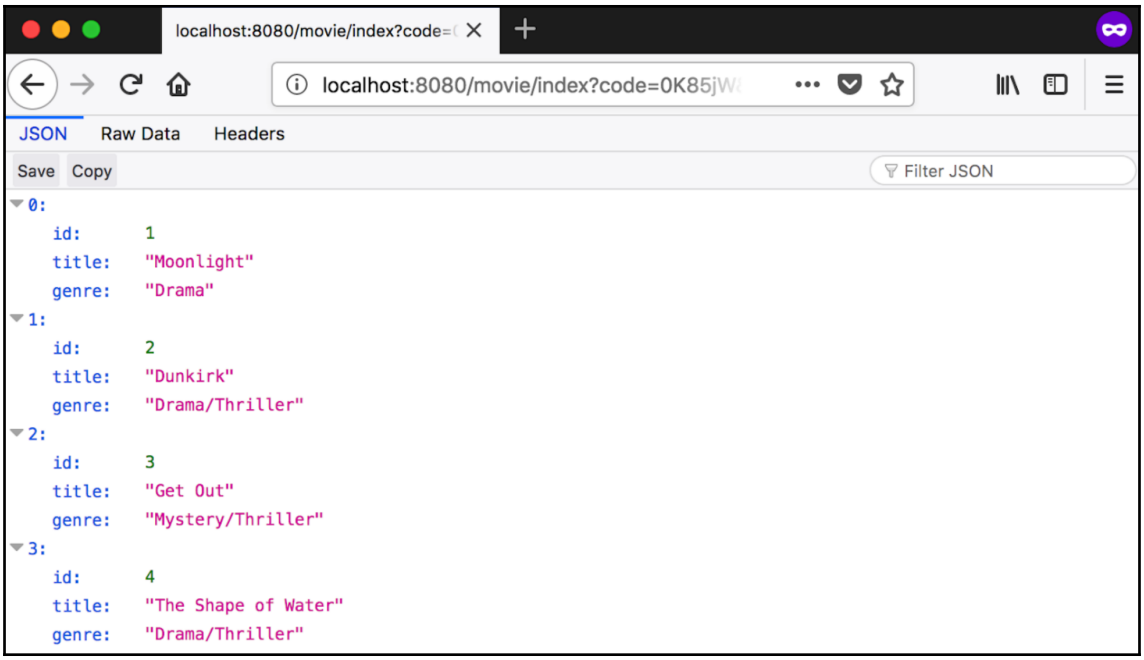

Hello Spring Security
localhost:8080/index
Client application non-secured page
Click here to take you to client secured page.


Login page
localhost:8080/login
Login page
Credential -> user / password
Credential -> admin / password
Username: user
Password: ••••••••
Log In
Back to non-secured page

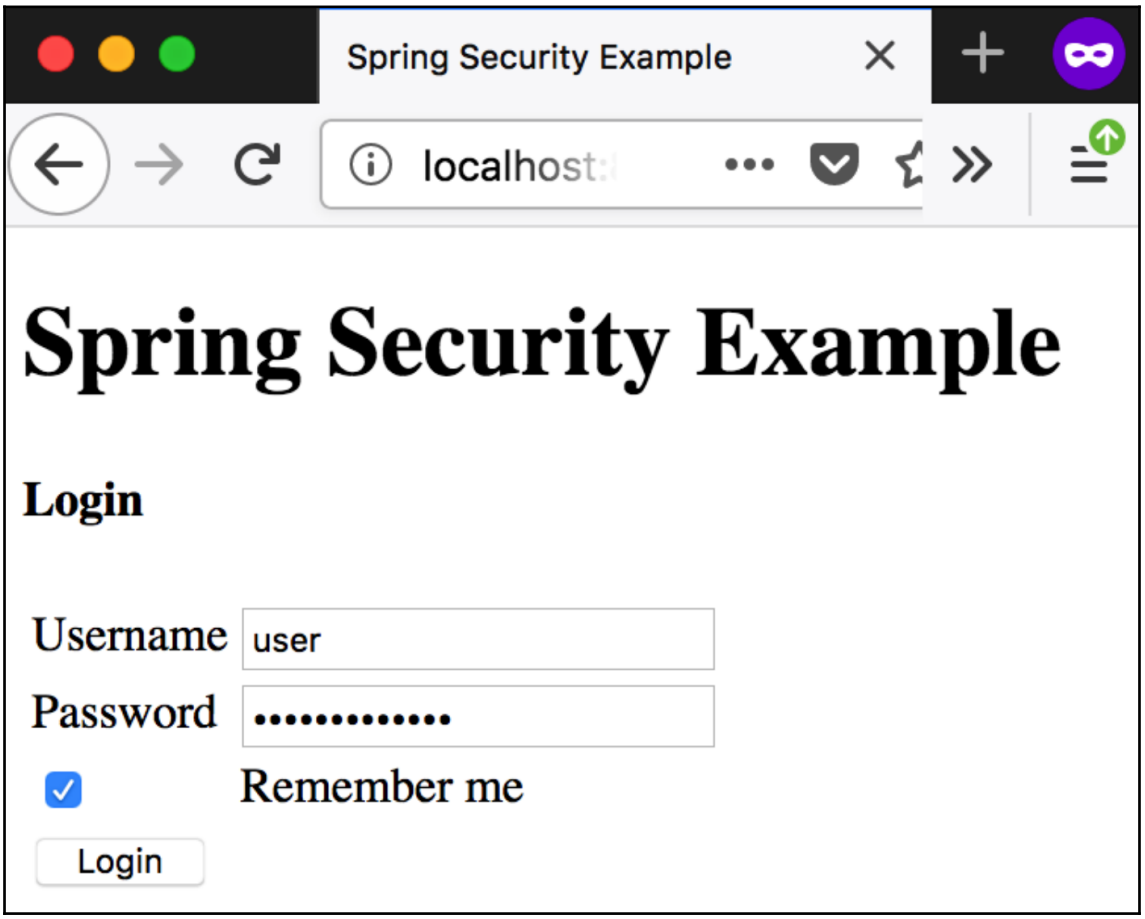

Hello Spring Security
localhost:8080/secured/index
Client application secured page
Welcome user
Assigned roles [ROLE_USER]
Logout
Click here to get movies from secured resource server.
You will be redirected to authorization server for entering credentials.
Back to non-secured page

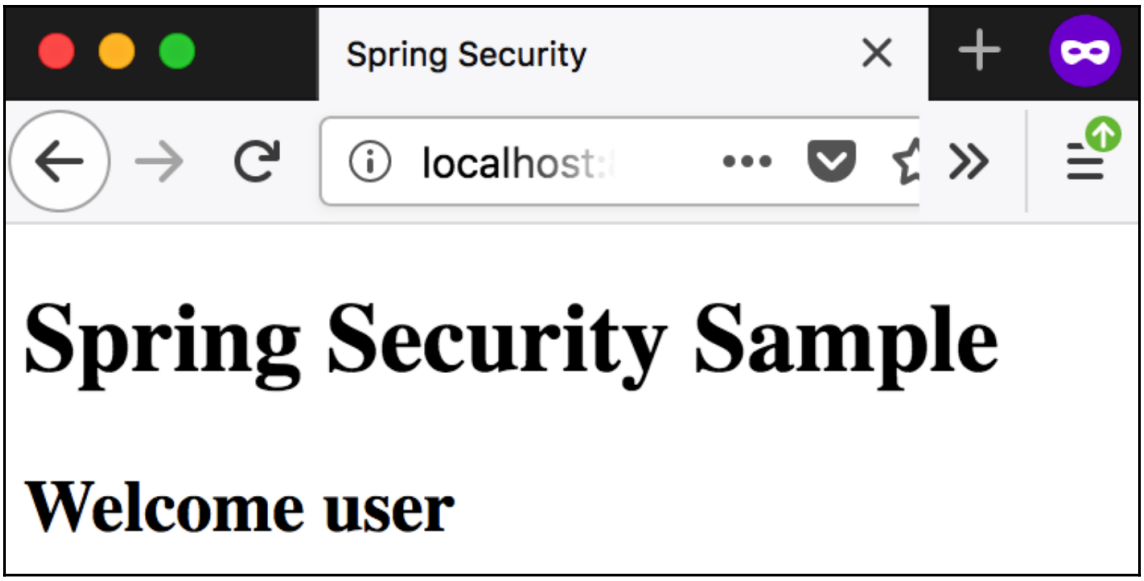

Login Page
localhost:8082/oauth-server/login
Login with Username and Password
User:
user
Password:
Login

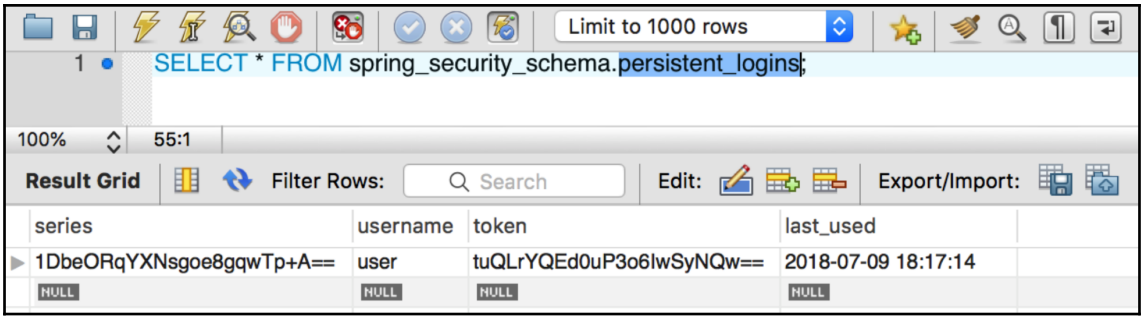

localhost:8082/oauth-server/oauth/
localhost:8082/oauth-server/oauth/authoriz
OAuth Approval
Do you authorize 'oAuthClientAppID' to access your protected resources?
Authorize
Deny

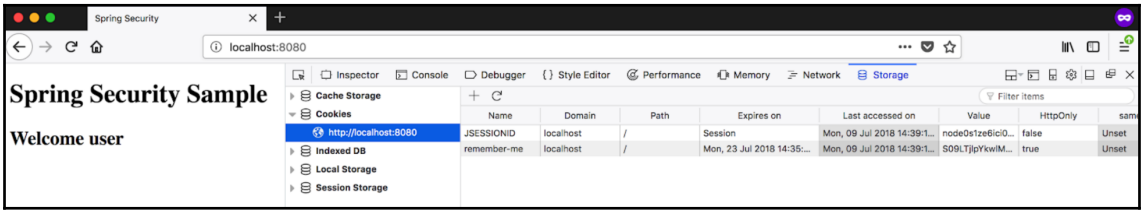

localhost:8080/movie/index?code=
localhost:8080/movie/index?code=0K85jW
JSON
Raw Data
Headers
Save
Copy
Filter JSON
0:
id: 1
title: "Moonlight"
genre: "Drama"
1:
id: 2
title: "Dunkirk"
genre: "Drama/Thriller"
2:
id: 3
title: "Get Out"
genre: "Mystery/Thriller"
3:
id: 4
title: "The Shape of Water"
genre: "Drama/Thriller"

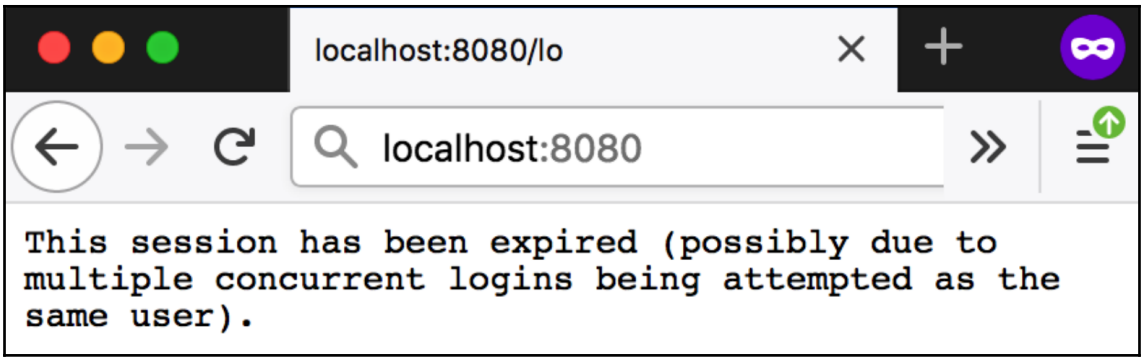
Chapter 7: Spring Security Add-Ons




Spring Security
localhost:
Spring Security Sample
Welcome user


Limit to 1000 rows
1
SELECT * FROM spring_security_schema.persistent_logins;
100%
55:1
Result Grid
Filter Rows:
Search
Edit:
Export/Import:
series
username
token
last_used
1DbeORqYXNsgoe8gqwTp+A==
user
tuQLrYQEd0uP3o6IwSyNQw==
2018-07-09 18:17:14
NULL


Spring Security
localhost:8080
Spring Security Sample
Welcome user
Inspector
Console
Debugger
Style Editor
Performance
Memory
Network
Storage
Cache Storage
Cookies
http://localhost:8080
Indexed DB
Local Storage
Session Storage
Name
Domain
Path
Expires on
Last accessed on
Value
HttpOnly
JSESSIONID
remember-me




```
Encryptors.
    delux(CharSequence password, CharSequence salt)                TextEncryptor
    noOpText()                                                     TextEncryptor
    queryableText(CharSequence password, CharSequence salt)        TextEncryptor
    standard(CharSequence password, CharSequence salt)             BytesEncryptor
    stronger(CharSequence password, CharSequence salt)             BytesEncryptor
    text(CharSequence password, CharSequence salt)                 TextEncryptor
```

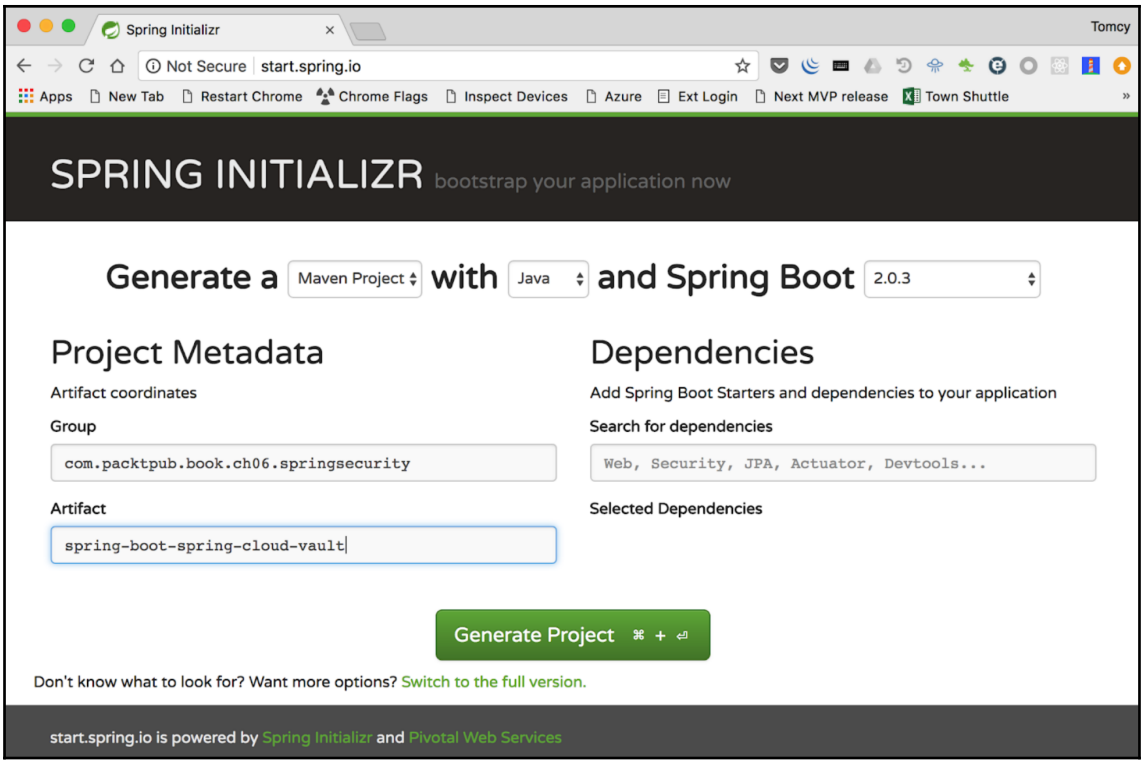
```
KeyGenerators.
    secureRandom()                                                 BytesKeyGenerator
    secureRandom(int keyLength)                                    BytesKeyGenerator
    shared(int keyLength)                                          BytesKeyGenerator
    string()                                                       StringKeyGenerator
Pressing ^Space twice without a class qualifier would show all accessible static methods π
```

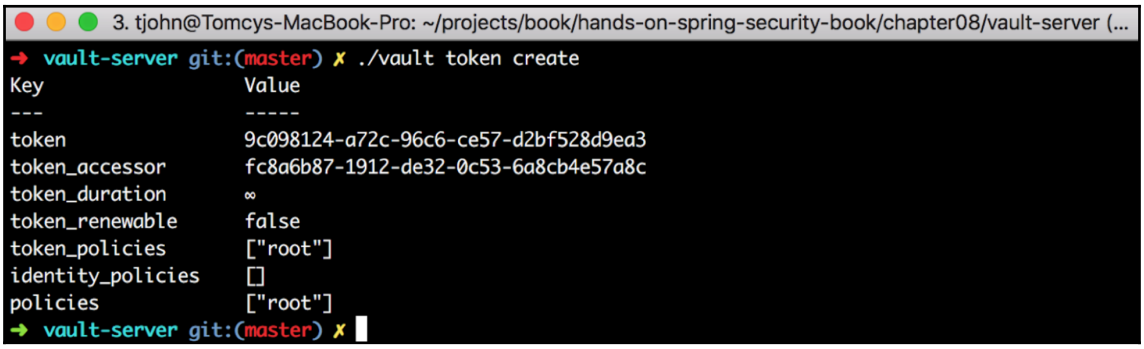
```
2. ./vault server -config vault.conf (vault)
➜ vault-server git:(master) ✗ ./vault server -config vault.conf
==> Vault server configuration:

                     Cgo: disabled
              Listener 1: tcp (addr: "0.0.0.0:8200", cluster address: "0.0.0.0:8201", tls: "disabled")
               Log Level: info
                   Mlock: supported: false, enabled: false
                 Storage: inmem
                 Version: Vault v0.10.3
             Version Sha: c69ae68faf2bf7fc1d78e3ec62655696a07454c7

==> Vault server started! Log data will stream in below:
```

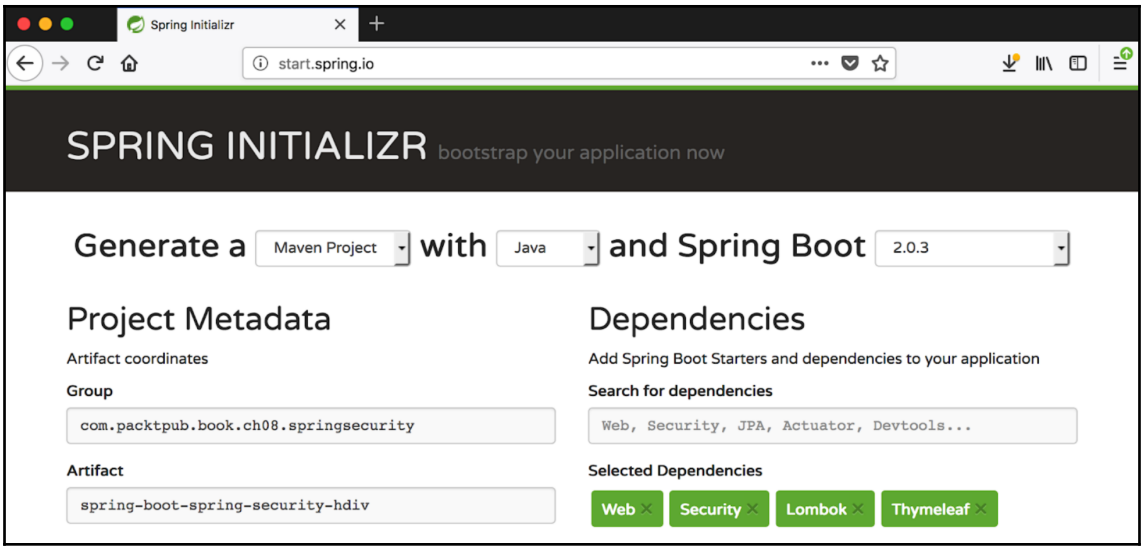
```
3. tjohn@Tomcys-MacBook-Pro: ~/projects/book/hands-on-spring-security-book/chapter08/vault-server (zsh)
➜ vault-server git:(master) ✗ ./vault operator init -key-shares=5 -key-threshold=2
Unseal Key 1: sF49ueubVe7aMg2PvuB2C84/UerzmsAkKSa4MCGoOM9q
Unseal Key 2: CP+PJcAnQ/g/aneWSgMcVdFtGVBeXD8DXKU0iI5iMT1z
Unseal Key 3: +jq2rZn8V+x1Ou2hZfkitVb27TvgHRoh5aoYeqlYoqJE
Unseal Key 4: 5J0cjUmqALt2ffgYT5YMnoMFkQGfXpYpHl8DZPLDIOT9
Unseal Key 5: ttsfF3V84Fss0FsMEgqJ9VH5GhFOK7uCSRe/NmEhEn/p

Initial Root Token: ee60f275-7b16-48ea-0e74-dc48b4b3729c

Vault initialized with 5 key shares and a key threshold of 2. Please securely
distribute the key shares printed above. When the Vault is re-sealed,
restarted, or stopped, you must supply at least 2 of these keys to unseal it
before it can start servicing requests.

Vault does not store the generated master key. Without at least 2 key to
reconstruct the master key, Vault will remain permanently sealed!

It is possible to generate new unseal keys, provided you have a quorum of
existing unseal keys shares. See "vault operator rekey" for more information.
➜ vault-server git:(master) ✗
```

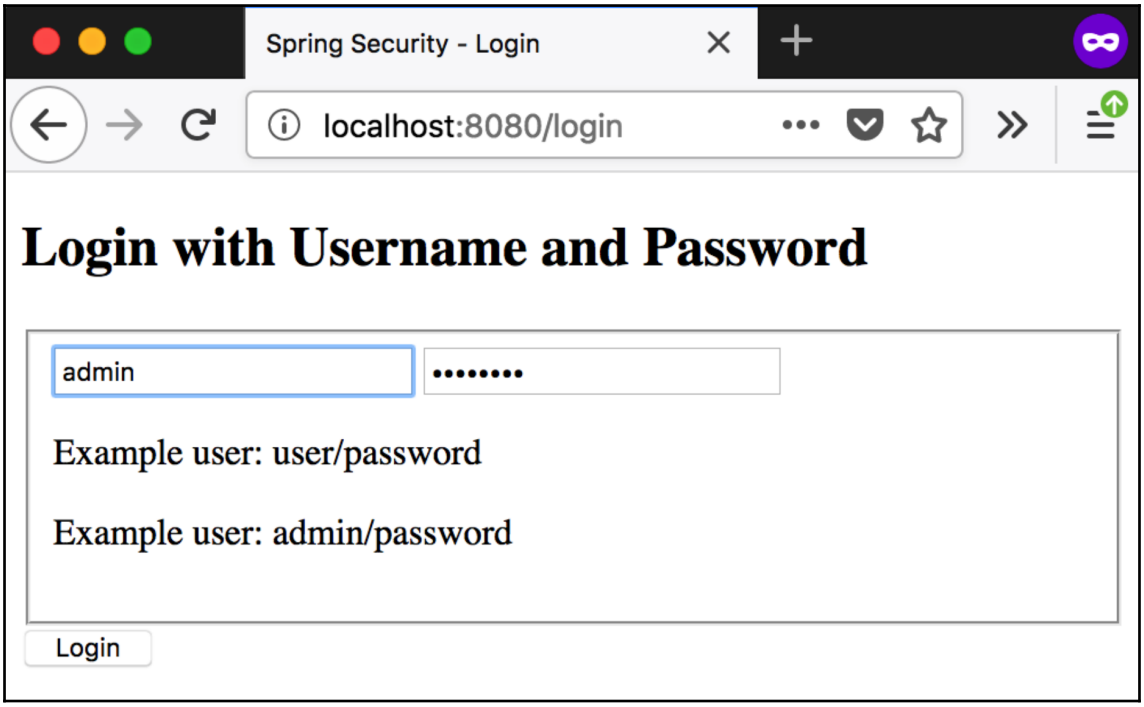
```
3. tjohn@Tomcys-MacBook-Pro: ~/projects/book/hands-on-spring-security-book/chapter08/vault-server (...
➜ vault-server git:(master) ✗ ./vault operator unseal sF49ueubVe7aMg2PvuB2C84/UerzmsAkKSa4MCGoOM9q
Key                Value
---                -----
Seal Type          shamir
Sealed             true
Total Shares       5
Threshold          2
Unseal Progress    1/2
Unseal Nonce       cad8dded-c4b5-cf7e-2967-2279cbc0ff8c
Version            0.10.3
HA Enabled         false
➜ vault-server git:(master) ✗
```

```
3. tjohn@Tomcys-MacBook-Pro: ~/projects/book/hands-on-spring-security-book/chapter08/vault-server (...
➜ vault-server git:(master) ✗ ./vault write secret/movie-application password=randomstring
Success! Data written to: secret/movie-application
➜ vault-server git:(master) ✗
```

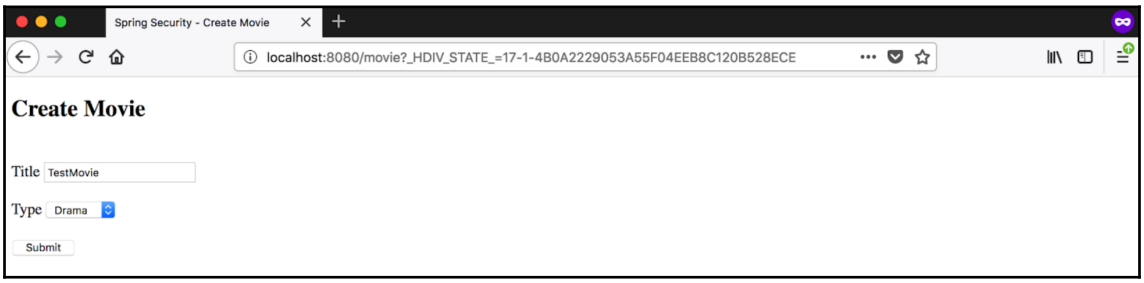

Spring Initializr
Tomcy
Not Secure
start.spring.io
Apps
New Tab
Restart Chrome
Chrome Flags
Inspect Devices
Azure
Ext Login
Next MVP release
Town Shuttle
SPRING INITIALIZR
bootstrap your application now
Generate a
Maven Project
with
Java
and Spring Boot
2.0.3
Project Metadata
Artifact coordinates
Group
com.packtpub.book.ch06.springsecurity
Artifact
spring-boot-spring-cloud-vault
Dependencies
Add Spring Boot Starters and dependencies to your application
Search for dependencies
Web, Security, JPA, Actuator, Devtools...
Selected Dependencies
Generate Project ⌘ + ⏎
Don't know what to look for? Want more options? Switch to the full version.
start.spring.io is powered by Spring Initializr and Pivotal Web Services


3. tjohn@Tomcys-MacBook-Pro: ~/projects/book/hands-on-spring-security-book/chapter08/vault-server (...
➜ vault-server git:(master) ✗ ./vault token create
Key Value
--- -----
token 9c098124-a72c-96c6-ce57-d2bf528d9ea3
token_accessor fc8a6b87-1912-de32-0c53-6a8cb4e57a8c
token_duration ∞
token_renewable false
token_policies ["root"]
identity_policies []
policies ["root"]
➜ vault-server git:(master) ✗

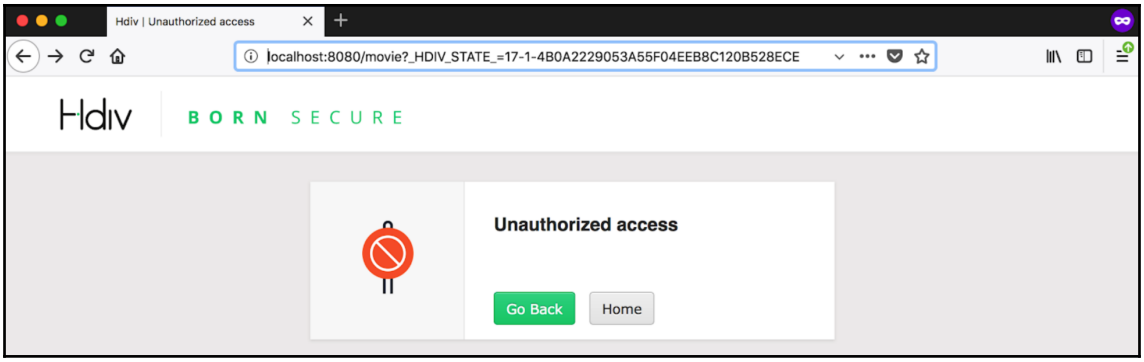

Spring Initializr
start.spring.io
SPRING INITIALIZR
bootstrap your application now
Generate a
Maven Project
with
Java
and Spring Boot
2.0.3
Project Metadata
Artifact coordinates
Group
com.packtpub.book.ch08.springsecurity
Artifact
spring-boot-spring-security-hdiv
Dependencies
Add Spring Boot Starters and dependencies to your application
Search for dependencies
Web, Security, JPA, Actuator, Devtools...
Selected Dependencies
Web
Security
Lombok
Thymeleaf


Spring Security - Login
localhost:8080/login
Login with Username and Password
admin
Example user: user/password
Example user: admin/password
Login


Spring Security - Main Page
localhost:8080
Spring Boot Spring Security HDIV Example
Click here to see URL example
Click here for movie form bean example


Spring Security - Create Movie
localhost:8080/movie?_HDIV_STATE_=17-1-4B0A2229053A55F04EEB8C120B528ECE
Create Movie
Title TestMovie
Type Drama
Submit


Spring Security - Links Page
localhost:8080/links?_HDIV_STATE_=29-0-88C74E1BAEDC24CE304E6A738D252524
Link page
Modify URL's to trigger HDIV validation error.


Hdiv | Unauthorized access
localhost:8080/movie?_HDIV_STATE_=17-1-4B0A2229053A55F04EEB8C120B528ECE
Hdiv
BORN SECURE
Unauthorized access
Go Back
Home