

HOME POSTS PORTFOLIO SPEAKING CV CONTACT

# Grails Programmer: How to secure your Grails 3 API with Spring Security REST for Grails?

26 / 02

**5** Replies

For this post I will continue reusing the code I created in the <u>How to use Spring</u>
<u>Security Core to Secure your Grails 3 app post</u>

In this post we are going to expose our app product announcements as JSON API and secure it with <u>Spring Security REST for Grails</u> plugin.

Lets create a service to encapsulate the logic.

```
$ grails create-service ProductAnnouncement
Created grails-app/services/myapp/ProductAnnouncementService.groovy
Created src/test/groovy/myapp/ProductAnnouncementServiceSpec.groovy
```

### with the content:

```
1
     package myapp
     import grails.transaction.Transactional
 5
     @Transactional
 6
     class ProductAnnoucementService {
 7
8
         List<ProductAnnouncement> lastAnnouncements() {
9
10
             ProductAnnouncement.createCriteria().list {
                 order("dateCreated", "desc")
                 maxResults(1)
13
             }
14
15
         }
```

Lets create a controller for our API endpoints.

```
1 | $ grails create-controller Api
```

```
Created grails-app/controllers/myapp/ApiController.groovy
Created src/test/groovy/myapp/ApiControllerSpec.groovy
```

Lets define a controller action to expose the last announcements as a JSON payload

```
1
     package myapp
 2
 3
     import grails.plugin.springsecurity.annotation.Secured
 5
     class ApiController {
 6
 7
         def productAnnoucementService
8
9
         @Secured(['ROLE_USER'])
10
         def announcements() {
11
             render(contentType: "application/json") {
12
                 announcements {
13
                      for(a in productAnnoucementService.lastAnnouncements()) {
14
                          announcement(message: a.message)
15
16
                 }
17
             }
18
         }
19
     }
```

# **Install Spring Security REST for Grails**

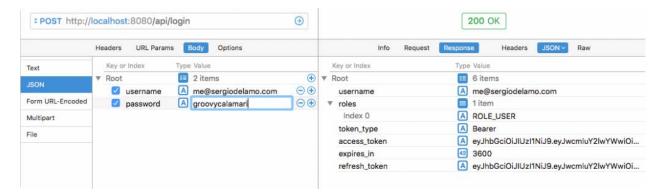
To install it we need to add a dependency to build.gradle

```
1
     buildscript {
 2
         ext {
 3
             grailsVersion = project.grailsVersion
 4
 5
         repositories {
 6
             mavenLocal()
 7
             maven { url "https://repo.grails.org/grails/core" }
8
         dependencies {
9
             classpath "org.grails:grails-gradle-plugin:$grailsVersion"
10
             classpath "com.bertramlabs.plugins:asset-pipeline-gradle:2.5.0"
11
12
             classpath "org.grails.plugins:hibernate4:5.0.0"
13
14
15
16
     version "0.1"
17
     group "myapp"
18
19
     apply plugin: "eclipse"
     apply plugin: "idea"
20
     apply plugin: "war"
21
22
     apply plugin: "org.grails.grails-web"
23
     apply plugin: "org.grails.grails-gsp"
24
     apply plugin:"asset-pipeline"
25
26
     ext {
27
         grailsVersion = project.grailsVersion
28
         gradleWrapperVersion = project.gradleWrapperVersion
29
     }
30
```

```
31
     repositories {
32
         mavenLocal()
         maven { url "https://repo.grails.org/grails/core" }
33
34
35
36
     dependencyManagement {
37
         imports {
38
              mavenBom "org.grails:grails-bom:$grailsVersion"
39
40
         applyMavenExclusions false
41
     }
42
43
     dependencies {
44
         compile "org.springframework.boot:spring-boot-starter-logging"
         compile "org.springframework.boot:spring-boot-autoconfigure"
45
46
         compile "org.grails:grails-core"
47
         compile "org.springframework.boot:spring-boot-starter-actuator"
         compile "org.springframework.boot:spring-boot-starter-tomcat"
48
49
         compile "org.grails:grails-dependencies"
         compile "org.grails:grails-web-boot"
50
51
         compile "org.grails.plugins:cache"
         compile "org.grails.plugins:scaffolding"
52
53
         compile "org.grails.plugins:hibernate4"
54
         compile "org.hibernate:hibernate-ehcache"
55
         console "org.grails:grails-console"
56
         profile "org.grails.profiles:web:3.1.1"
57
         runtime "org.grails.plugins:asset-pipeline"
58
         runtime "com.h2database:h2"
         testCompile "org.grails:grails-plugin-testing"
testCompile "org.grails.plugins:geb"
59
60
         testRuntime "org.seleniumhq.selenium:selenium-htmlunit-driver:2.47.1"
61
         testRuntime "net.sourceforge.htmlunit:htmlunit:2.18"
62
63
         compile 'org.grails.plugins:spring-security-core:3.0.3'
         compile "org.grails.plugins:spring-security-rest:2.0.0.M2"
65
     }
66
67
     task wrapper(type: Wrapper) {
68
         gradleVersion = gradleWrapperVersion
69
70
71
     assets {
72
         minifyJs = true
73
         minifyCss = true
74
```

Spring Security REST for Grails exposes an endpoint /api/login which we can access with a POST request. We should supply our credentials (username and password) as parameters. If we are successfully authenticated we will get an *access\_token* and a *refresh\_token*.

Grails Programmer: How to secure your Grails 3 API with Spring Security REST for Grails? | Sergio del Amo



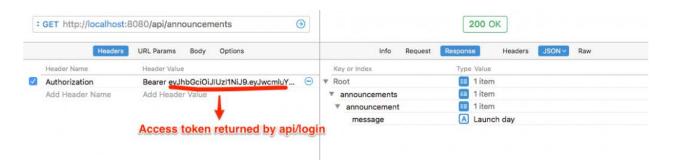
We can do a GET request to our *http://localhost:8080/api/announcements* endpoint. We need to add an HTTP header:

Header Name: Authorization

Header Value: Bearer access\_token

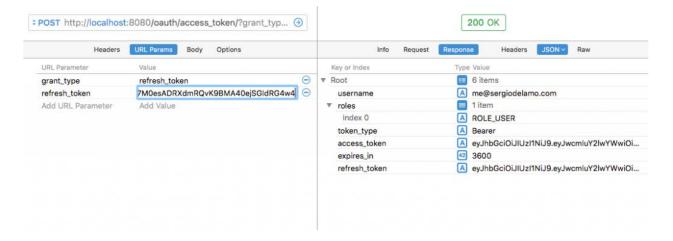
The access\_token is the one we got back from the api/login endpoint.

The above request is illustrated here:



Note were are using a <u>JSON Web Token (JWT)</u> which we don't need to store on the server side. The token validation is done without a repository (database or similar) call. Thus, better for the scalability of the application.

Access token expire. However in the *api/login* method we got a refresh token which we can use to get a new access token. We can do POST request as shown below:



**Do you like to read about Groovy/Grails development?** Yes, then Subscribe to <u>Groovy Calamari</u> a weekly curated email newsletter about the Groovy ecosystem which I write •

— How to use Spring Security Core to Secure you Grails 3 App?

Grails Programmer: How to change the default port of a Grails 3 App

5 thoughts on "Grails Programmer: How to secure your Grails 3 API with Spring Security REST for Grails?"



**jf li** 3 December, 2016 at 5:02

Could you advise what tool you are using while you test the http post form?

#### **Reply** ↓



Sergio
3 December, 2016 at 8:06

That it is to say:

@Secured( { ['admin1', 'admin2'].contains(authentication.name) })

Note I think your best option would be to add a role to admin1 and admin2 and secured the action based on that role.

<u>Reply</u> ↓

## **Leave a Reply**

Your email address will not be published. Required fields are marked \*

Comment

Name \*

Email \*

Website

Notify me of follow-up comments by email.

Notify me of new posts by email.



Drag to outliner or Upload Close