Grails Autenticação e Autorização

brunogamacatao@gmail.com

Trabalhando com Conteúdo Estático

- Conteúdo estático ?
  - js
  - css
  - imagens
  - ...

Grails Resources Plugin

- Já vem no Grails 2.X
- Se você estiver utilizando o Grails 1.3.X você deve instalá-lo:
  - grails install-plugin resources

Tags Básicas

- <r:require module="nome do módulo"/>
- <r:layoutResources/>
- <r:img uri="/images/logo.png"/>
- <r:resource uri="/css/estilo.css"/>
- Exemplo:
  - <r:require module="jquery, blueprint"/>

Declaração dos Recursos

- Arquivo:
  - grails-app/conf/ApplicationResources.groovy

ApplicationResources

```
modules = {
    application {
        resource url:'js/application.js'
    }
}
```

Vamos instalar alguns plugins ?

- grails install-plugin jquery
- grails install-plugin twitter-bootstrap

E para utilizar estes recursos ?

- Vamos modificar o arquivo ApplicationResources:

```
modules = {
  application {
    dependsOn 'jquery, bootstrap'
    resource url:'js/application.js'
  }
}
```

Máscaras

- Baixe o JQuery Masked Input:
  - http://digitalbush.com/projects/masked-input-plugin/
- Grave o arquivo jquery.maskedinput-1.3.min.js na pasta web-app/js

Criando um novo Recurso

- Edite o arquivo ApplicationResources:

```
modules = {
  application {
    dependsOn 'jquery, bootstrap'
    resource url:'js/application.js'
  }
  maskedInput {
    dependsOn 'jquery'
    resource url:'js/jquery.maskedinput-1.3.min.js'
  }
}
```

Edite o arquivo _form.gsp

- No início de tudo:

```
<r:require modules="maskedInput"/>
<jq:jquery>
  $('#telefone').mask('(99)9999-9999');
</jq:jquery>
```

Autenticação e Autorização

Plugin

- Spring Security Core
  - grails install-plugin spring-security-core

Criação dos Modelos

- grails s2-quickstart agenda Usuario Grupo

O que foi criado ?

- agenda.Usuario
- agenda.Grupo
- agenda.UsuarioGrupo
- LoginController
- LogoutController

UrlMappings.groovy

- “/login/$action?”(controller:”login”)
- “/logout/$action?”(controller:”logout”)

Formas de Autorização

- Anotações
- Regras de URL
- Mapa de Requisições (request map)

Anotações

- Antes das ações nos controladores:
  - @Secured(['ROLE_USER'])
  - def acao = { ... }

Regras de URL

```
import grails.plugins.springsecurity.SecurityConfigType


grails.plugins.springsecurity.securityConfigType =
SecurityConfigType.InterceptUrlMap

grails.plugins.springsecurity.interceptUrlMap = [

'/': ['IS_AUTHENTICATED_ANONYMOUSLY'],

'/consumidor/cadastro/**': ['IS_AUTHENTICATED_REMEMBERED'],

'/patches/**': ['ROLE_ADMIN']

]
```

Mapa de Requisições

```
new Requestmap(url: '/timeline', configAttribute: 'ROLE_USER').save()

new Requestmap(url: '/**', configAttribute: 'IS_AUTHENTICATED_ANONYMOUSLY').save()
```

Regras Implícitas

- IS_AUTHENTICATED_ANONYMOUSLY
- IS_AUTHENTICATED_REMEMBERED
- IS_AUTHENTICATED_FULLY

Criando Usuários Iniciais no Bootstrap

```
def userRole = SecRole.findByAuthority('ROLE_USER') ?: new SecRole(authority:
'ROLE_USER').save(failOnError: true)
```

```
def adminRole = SecRole.findByAuthority('ROLE_ADMIN') ?: new SecRole(authority:
'ROLE_ADMIN').save(failOnError: true)
```

```
def adminUser = SecUser.findByUsername('admin') ?:
  new SecUser(username: 'admin',
  password: springSecurityService.encodePassword('admin'),
  enabled: true).save(failOnError: true)
if (!adminUser.authorities.contains(adminRole)) {
  SecUserSecRole.create adminUser, adminRole
}
```

Obtendo o usuário *logado* nos Controladores

```
def user = SecUser.get(springSecurityService.principal.id)
```

Tags

<sec:ifNotLoggedIn>

<sec:ifAllGranted roles="ROLE_USER">

<sec:username />