

Grails URL Mappings

Alexis Plantin

Summary

- Url Mappings par défaut
- Texte statique dans les Url Mappings
- Supprimer les noms de Controller & d'action
- Paramètres dans un Mapping
- Paramètres additionnels
- Mappings vers une View
- Contraintes & Url Mappings
- Méthode HTTP des requêtes
- Code de retour des réponses HTTP
- Reverse Url Mapping
- Plusieurs classes pour les Url Mappings

Url Mappings par défaut

- Configuration simple

```
static mappings = {  
    "$controller/$action?/$id?(.$format)?" {  
        constraints {  
            // apply constraints here  
        }  
    }  
  
    "/"(view:"/index")  
    "500"(view:'/error')  
    "404"(view:'/notFound')  
}
```

- L'URL `product/index` va mapper avec l'action `index` du `ProductController`
- Id optionnel

Texte statique dans les Url Mappings

- Possibilité d'ajouter du texte statique

```
"/showProduct/$controller/$action?/$id?"{  
    constraints {  
        // apply constraints here  
    }  
}
```

Supprimer les noms de Controller & d'action

- Possibilité de redéfinir complètement la logique des URLs

```
"/showProduct/$id?"(controller:"product", action:"show"){  
  constraints {  
    // apply constraints here  
  }  
}
```

Paramètres dans un Mapping

- Paramètres classiques dans une URL

```
/showProduct?productName=Product
```

- Paramètres embarqués dans l'Url

```
/showProduct/$productName
```

- Remplacer les espaces par des underscores

```
product = Product.findByName(params.productName.replace(' ', '_'))
```

Paramètres additionnels

- Paramètres comme propriétés du Url Mappings

```
"/showProduct/$productName"(controller:'product',  
action:'show') {  
    format = 'simple'  
}  
  
"/showProductDetail/$productName"(controller:'product',  
action:'show') {  
    format = 'detailed'  
}
```

Mappings vers une View

- Mappings vers une View (sans passer par la couche Controller)

```
"/"(view:"/index")
```


Contraintes & Url Mappings

- Comme les contraintes dans les Domain Classes
- Elmine les mauvais patterns d'URLs
- Imaginons un système de blog avec un système d'URLs basé sur des dates
 - Une URL peut être `/articles/2010/11/3/someArticle`

```
"/articles/$year/$month/$day/$articleName?" {  
  controller = 'blog'  
  action = 'display'  
  constraints {  
    // apply constraints here  
  }  
}
```

Contraintes & Url Mappings

- On peut ajouter les contraintes sur les dates

```
"/articles/$year/$month/$day/$articleName?" {  
  controller = 'blog'  
  action = 'display'  
  constraints {  
    year matches: /[0-9]{4}/  
    month matches: /[0-9]{2}/  
    day matches: /[0-9]{2}/  
  }  
}
```

Méthode HTTP des requêtes

- Un Url Mapping peut mapper différentes actions en fonction de la méthode de la requête
- Utiles pour les services RESTful

```
"/product/$productName" {  
  controller = 'product'  
  action = [GET: 'show',  
            POST: 'save',  
            PUT: 'update',  
            DELETE: 'delete']  
}
```

Code de retour des réponses HTTP

- Possibilité de définir un mapping selon les codes HTTP retournés

```
"500"(view: '/error')
```

```
"404"(controller: 'product')
```

Reverse Url Mapping

- Utilise le mapping pour générer les URLs dans l'application
 - Génération des liens avec le tag `<g:link>`

Plusieurs classes pour les Url Mappings

- Possibilité d'avoir plusieurs classes d'Url Mappings
 - Par convention, les classes sont placées dans `grails-app/conf`
 - Elles doivent respecter la même syntaxe

Questions?