Scaffolding是指自动生成domain class的CRUD接口，这包括：

* 必要的[views](http://grails.org/doc/latest/guide/single.html#gsp)
* Controller的 create/read/update/delete (CRUD) action

scaffolding的功能是以插件的形式提供的。如果使用了create-app命令新建的项目就已经自动包含了该插件。如果是手工创建的项目则需要手工在BuildConfig.groovy中加入插件：

|  |
| --- |
| plugins {  …  compile ":scaffolding:2.0.0"//or 1.0.0  …  } |

2.0.0版本对应Grails2.3.X，1.0.0版本对应Grails2.2.X。最新的scaffolding版本主要是新加入了针对REST APIs的模板代码生成能力。

# 动态Scaffolding

scaffolding最简单的形式就是动态scaffolding。只要在controller中设置scaffold静态属性为true就行了。例如domain class Book的BookController中：

|  |
| --- |
| class BookController {  static scaffold = true  } |

这里BookController和Book domain class遵循了命名上约定，所以直接设置scaffold = true就行了。如果需要指定其他domain class，可以把domain class直接赋值给scaffold属性。

|  |
| --- |
| class SomeController {  static scaffold = Author  } |

当使用动态scaffold之后，domain CRUD对应的controller、view都会在运行时自动创建，自动创建的controller action有：

* index
* show
* edit
* delete
* create
* save
* update

CRUD所用的界面也会自动生成，浏览http://localhost:8080/app/book既可以访问。

如果domain class是直接使用hibernate映射的普通java class，也可以动态scaffolding，只要简单的在controller中import 对应的domain class并且设置scaffold属性为true就行了。

对于动态scaffolding的controller，也可以添加其他的action：

|  |
| --- |
| class BookController {  static scaffold = Book  def changeAuthor() {  def b = Book.get(params.id)  b.author = Author.get(params["author.id"])  b.save()  // redirect to a scaffolded action  redirect(action:show)  }  } |

也可以重写在运行时自动生成的action：

|  |
| --- |
| class BookController {  static scaffold = Book  // overrides scaffolded action to return both authors and books  def index() {  [bookInstanceList: Book.list(),  bookInstanceTotal: Book.count(),  authorInstanceList: Author.list()]  }  def show() {  def book = Book.get(params.id)  log.error(book)  [bookInstance : book]  }  } |

## 自定义生成的view

scaffolding生成的view是和domain class的validation constraits适配的。例如，如果想改变输入项的现实顺序只需要在domain class的constraints中对属性重新排序就可以了：

|  |
| --- |
| static constraints = {  title()  releaseDate()  } |

使用inList约束，可以使输入项以下拉框的形式显示：

|  |
| --- |
| static constraints = {  title()  category(inList: ["Fiction", "Non-fiction", "Biography"])  releaseDate()  } |

# 静态scaffolding

Grails同样也支持静态的scaffolding。可以使用domain class作为参数运行generator-controller命令生成contoller：

|  |
| --- |
| grails generate-controller Book |

也可以生成view：

|  |
| --- |
| grails generate-views Book |

或者同时生成controller和view：

|  |
| --- |
| grails generate-all Book |

接下来就可以自定义修改生成的controller和gsp文件了。