# servletContext四个对象

session 对象-----> HttpSession 实例

request 对象-----> HttpServletRequest 实例

params -----🡪 post 参数 可变 Map

flash ----🡪只存贮本次请求和下次请求之间临时用到的属性

每个请求都有自己的线程，控制器的线程是安全的。

限定视图位置，需要使用绝对路径来代替相对路径

render(view:”/…../….”);

# 指定一个视图

## 语法；

render(view:”show”);

## 同一个控制器内跳转方法：

redirect(action:”method”);

redirect(action:method);

## 不是同一个控制器:

redirect(controller:””,action:””);

相对于应用上下文：

redirect(uri:”/login.html”);

或者

redirect(<url:”http://grail.org>”);

## 带参post

redirect(controller:””,action:””,params[]);

## 自动 Json 编组：

render Book.list() as JSON;

或者 def json = Book.list().encodeAsJSON();

render json;

# 页面标签

GSP标签以 g: 开始，不需要指定任何的标签库导入。

指定作用域： scop=”request”

## 逻辑迭代

<g:if test=”{$..}”>

<g:else>

<g:each in=”[]” var=””>

<g:while test=”${N<5}”>

## 链接

<g:link controller=”” action=”” ></g:link>

指定控制器，方法

<g:link params=”[a:””,b:””]”></g:link>

## 共享布局

<g:layoutTile>

<g:layoutHead/>

<g:layoutBody/>

## 触发布局，在视图中增加一个meta标签

<meta name=”layout” content=”main” />

控制器指定一个布局

----🡪 layout 属性

<r:require modul=””/>

<r:layoutResources />

成对使用，渲染资源链接；

# groovy 语法

## 弱类型

循环

for(i=0;i<5;i++){

}

不需要 int 声明，加 int也可以

for( i in 0..5){}

0..5就是一个范围；

a..e 表示 a b c d e a..<e 小于e 的所有值；

## 集合[]

def collection=[“a”,”b”,”c”];

往集合添加元素

collection.add(10);

collection<<10;

groovy支持负索引

collection[-1] 倒数第一个元素；

collection=collection – collection[0..4] 删除集合前5个元素；

## Map[:]映射

def map=[:];

map.put();

通过key 检索；

map[key];

## 闭包:闭包是用来对集合进行迭代

map.each({

key,value ->

//code;

});

map.each{

printe it; //it 代表map 集合的每一个元素

};

## 闭包另一个用处，单独定义

def say={word ->

print “hi,$word” }

say(“groovy”);

打印 hi,grooby;

groovy中不需要用return返回值

默认返回方法的最后一行值；

# groovy响应ajax

render params.success=true;