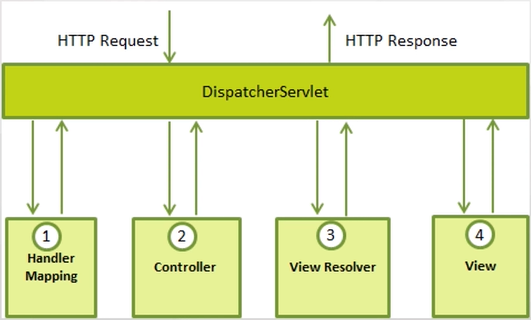
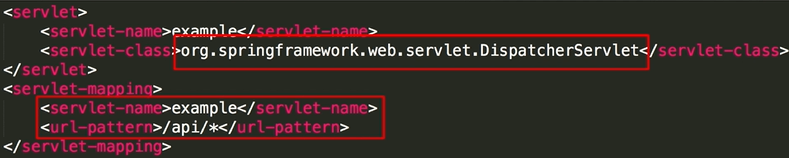
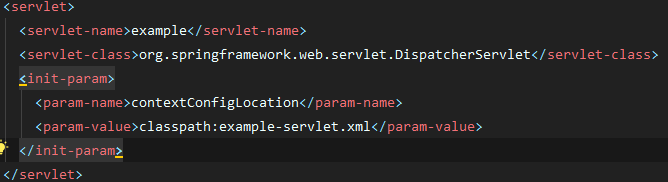
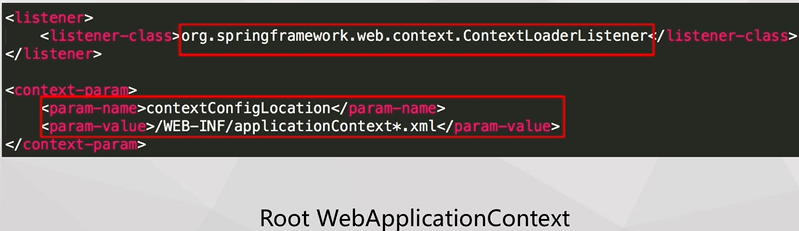
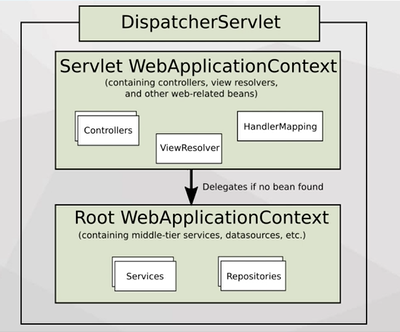
Spring MVC

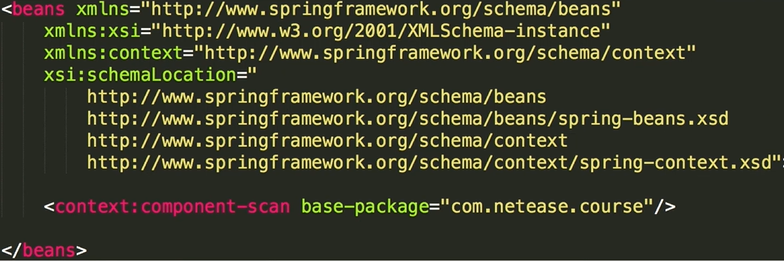
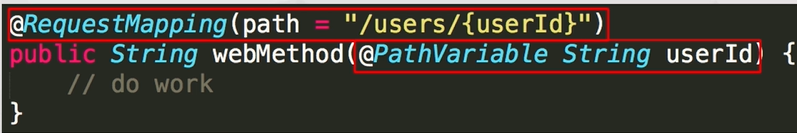
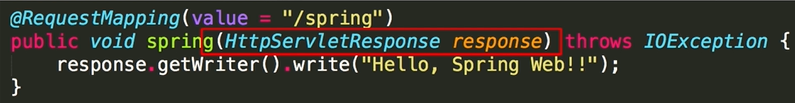
# MVC模式

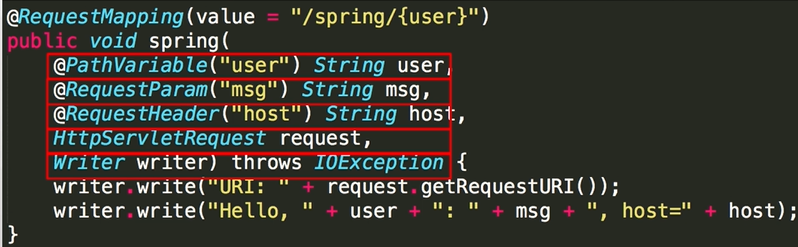
* 数据与视图分离
* Web页面
  + Servlet——HTML嵌入到Java代码中
  + JSP——Java代码嵌入到HTML中
  + 问题
    - 需要一个框架来处理路由
    - 需要一个框架来处理MVC
* Web MVC 框架

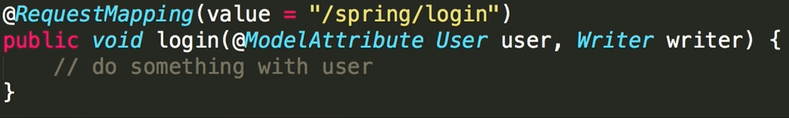
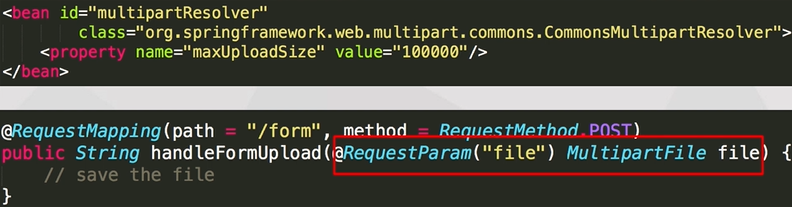
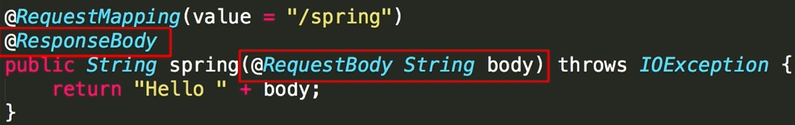
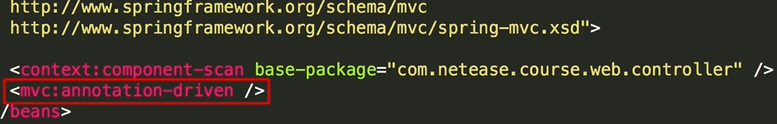
# DispatcherServlet

* Web.xml.
* [servlet-name]-servlet.xml
  + HandlerMapping
  + Controllers
  + View解析相关
* WebApplicationContext
  + ContextLoaderListener
  + 

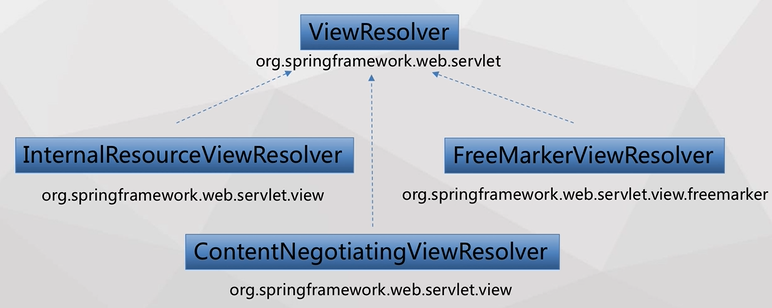
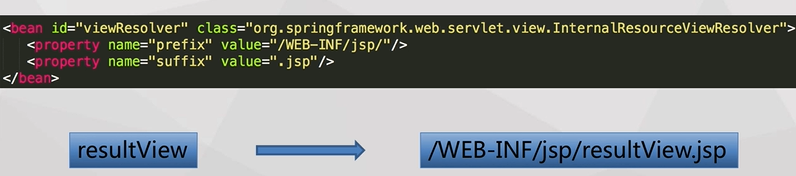
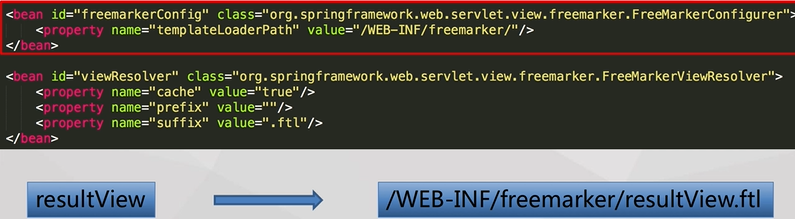
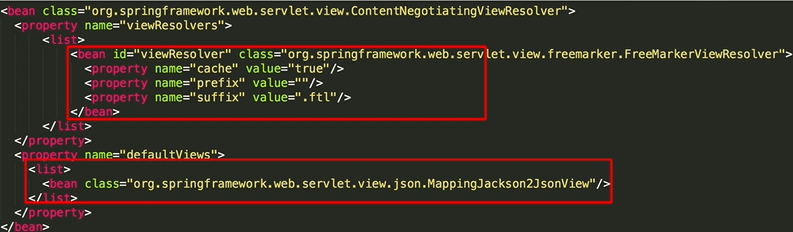
# Controller

* 定义controller
* @requestMapping
  + Name：名称
  + Value & path：路径，如/hello
  + Method：请求方法，如GET
  + Params：请求参数
  + Header：请求头
  + Consumes：请求的媒体类型，Content-Type
  + Produces：响应的媒体类型，Accept
  + RestAPI？
* 函数实现
  + 函数参数
    - HttpServletRequest/HttpServletResponse，HttpSession(Servlet API)
    - Reader/Writer
    - @PathVariable
    - @ResuestParam
    - @RequestHeader
    - HttpEntity
    - @ResulstBody
    - Map/Model/ModelMap
  + 返回值
    - Void
    - String：view名称，@ResponseBody
    - HttpEntity
    - View
    - Map
    - Model
    - ModelAndView



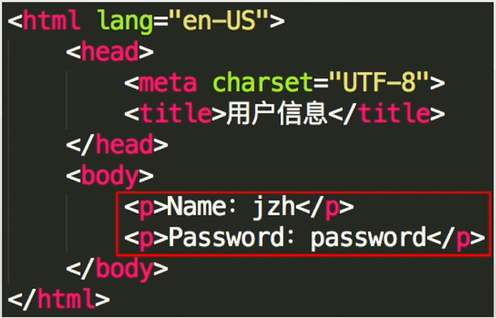
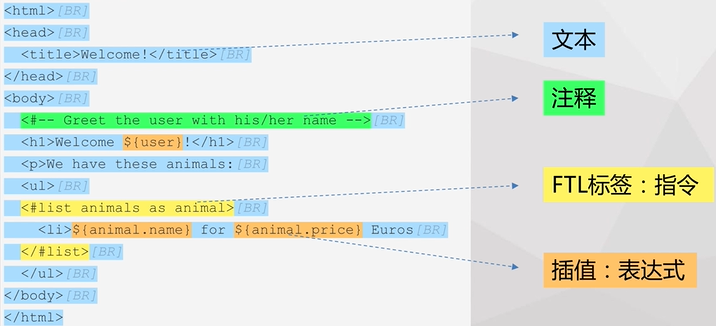
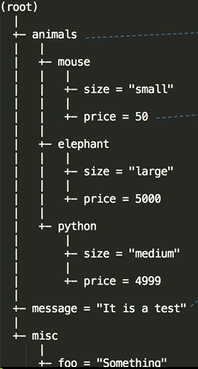
* + 表单
  + 上传文件
  + HttpEntity
  + @RequestBody & @ResposeBody
  + 返回对象
    - MessageConverter

# View

* View解析
  + String：名称
  + View
  + ModelAndView
  + Map
* ViewResolver
  + InternalResourceViewResolver
    - Servlet， JSP
  + FreeMarkerViewResolver
    - FreeMarker
  + ContentNegotiatingViewResovler
    - ViewResolver的组合
    - 扩展名：user.json，user.xml，user.pdf
    - Accept头（media types）：application/json，application/xml
* DefaultRequestToViewNameTranslator

# FreeMarker

## 简介

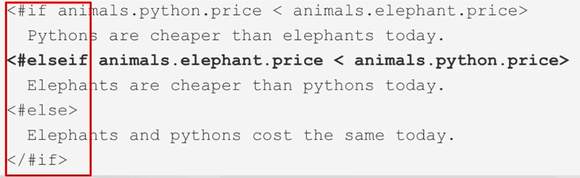
* 问 题
* 模板
* 数据模型 

## 指令

* 开始标签：<#directivename parameters>
* 结束标签：</#directivename parameters>

### If指令





### List指令



### Include指令



## 表达式

### 直接指定值

* 字符串：“Foo”或者‘Foo’
* 数字：123.45
* 布尔值：true，false
* 序列：[“foo”, “bar”, 123.45], 值域：0..9, 0..<10
* 哈希表：{“name”: “green mouse”, “price”: 150}

### 检索变量

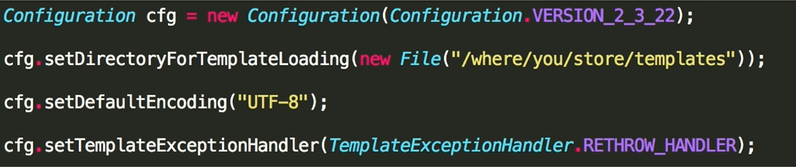
* 顶层变量：user
* 哈希表数据：user.name, user[“name”]
* 列表数据：products[5]

### 连接操作

* Users + [“green”]
* Password + {“joe”: “secret42”}
* “Hello” + user + “!”

### 算术运算，比较运算，逻辑运算等

## 使用

1. 创建配置
2. 定义模板
3. 定义数据模型