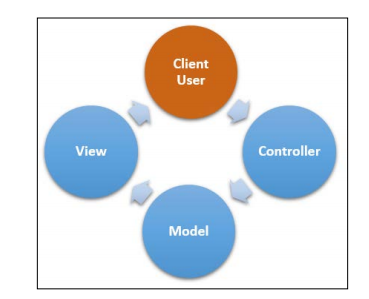
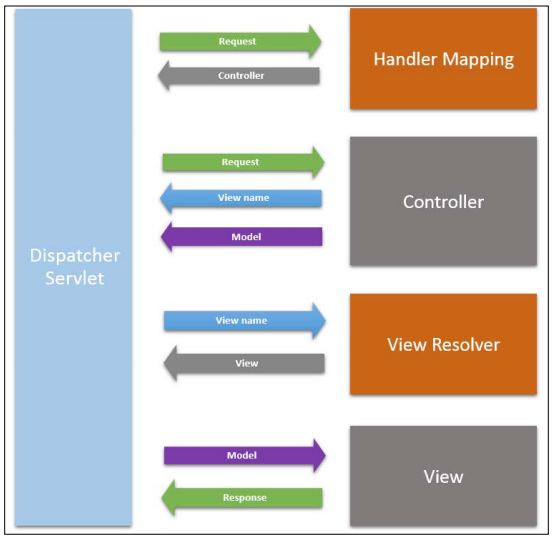
第二章 设计基于Spring MVC的微服务





1.一个简单URL mapping的controller

(1)DispatcherServlet - Spring MVC的入口（在web.xml中）

<servlet>

<servlet-name>spring</servlet-name>

<servlet-class>

org.springframework.web.servlet.DispatcherServlet

</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>spring</servlet-name>

<url-pattern>/\*</url-pattern>

</servlet-mapping>

(2)基于注解的Controller

@Controller

必须有这行表明Controller在的包

<context:component-scan base-package="edu.zipcloud.cloudstreetmarket.

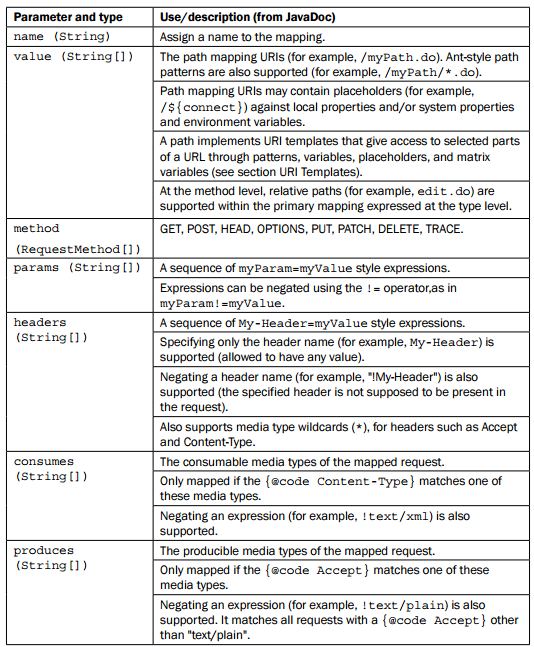
portal"/>

@RequestMapping

映射关系，表明处理HTTP request，header，parameter和media-type等的method

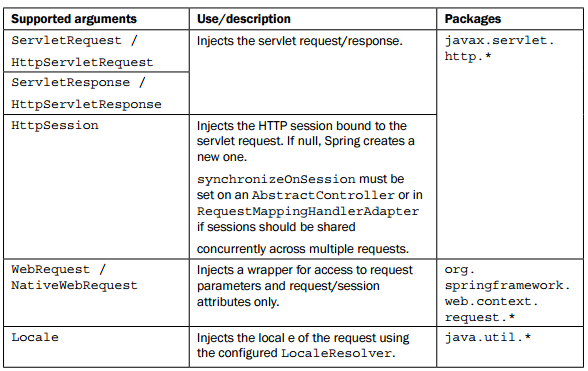
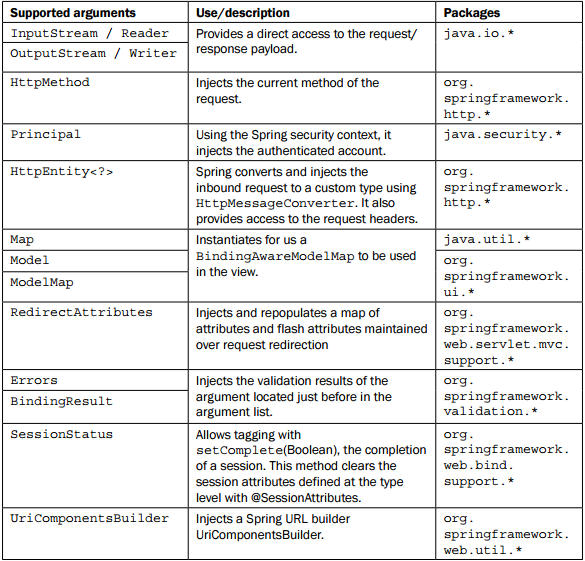
@RequestMapping(value="/server", method=RequestMethod.GET)

RequestMapping可以有以下几种参数：

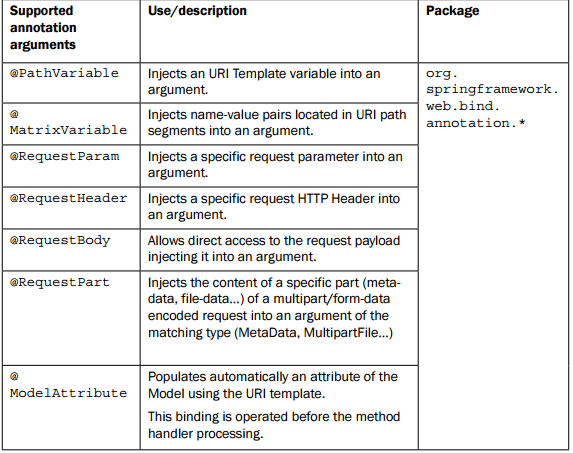
  
  
controller基本结构

@RequestMapping(value="/index") public ModelAndView getRequestExample(ServletRequest request){ ModelAndView mav = new ModelAndView(); mav.setViewName("index"); mav.addObject("variable1", new ArrayList()); return mav; }

Controller支持的参数类型：

支持的参数注解类型



如：

@RequestMapping(value="/index")

public ModelAndView getRequestExample(@RequestParam("exP1") String

exP1){

ModelAndView mav = new ModelAndView();

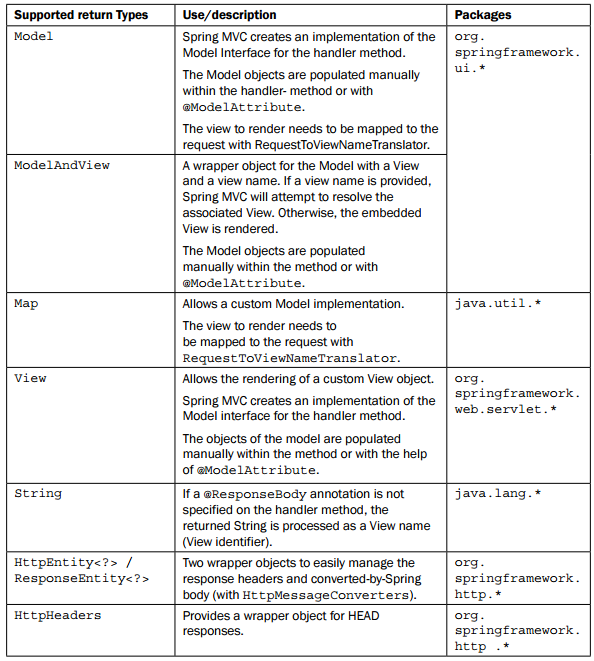
mav.setViewName("index");

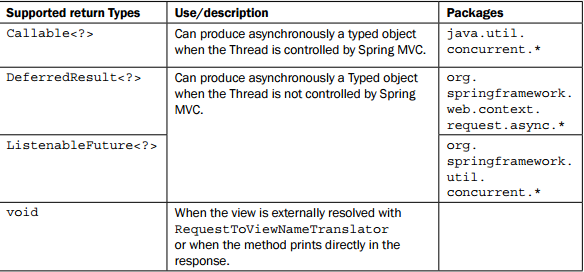
mav.addObject("exP1", exP1);

return mav;

}

支持的返回值类型：





2.用ViewResolver配置fallback controller

不管访问那个页面，都返回主页面

@Controller

public class DefaultController {

@RequestMapping(value="/\*",

method={RequestMethod.GET,RequestMethod.HEAD})

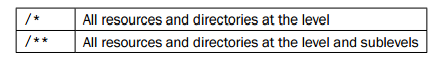
public String fallback() {

return "index";

}

}

(1)URI template



如果类被注解，则其方法的路径是基于类的，如/default/\*

@RequestMapping(value="/default"...)

@Controller

public class DefaultController…{

@RequestMapping(value="/\*"...)

public String fallback(Model model) {...}

}

(2)ViewResolvers

<bean id="viewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">

<property name="viewClass" value="org.springframework.web.servlet.view.JstlView" />

<property name="prefix" value="/WEB-INF/jsp/" />

<property name="suffix" value=".jsp" />

</bean>

指定View的解析类，还可以指定VelocityViewResolver orFreeMarkerViewResolver

(3)注解@PathVariable

URI中的param会变成参数传到函数中

@RequestMapping(value="/example/{param}")

public HttpEntity<String> example(@PathVariable("param") String

parameter) {

return new HttpEntity<>(parameter);

}