SpringMVC

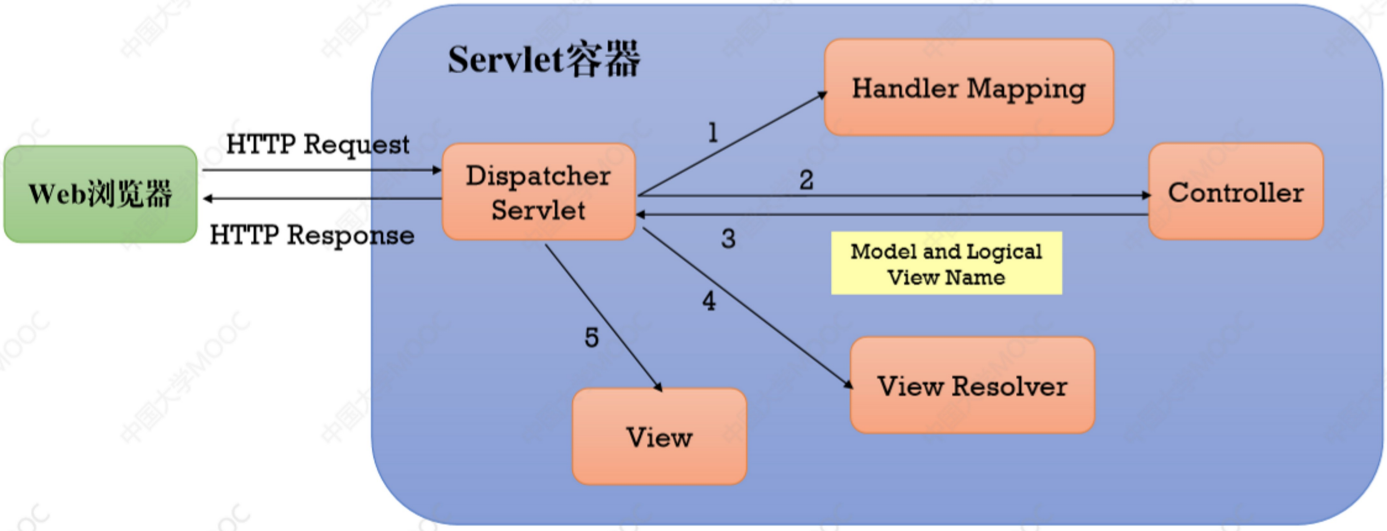
2021年9月26日 15:19

SpringMVC采用了一种松散的、可拔插的组件结构
他通过注解的方式，使一个java类变成控制器，这个控制器不需要接口，是一个pojo
SpringMVC支持Resultful控制器，它的整个结构是围绕DispatcherServlet类来展开的，即前端控制器

DispatcherServlet

DispatcherServlet是SpringMVC的框架已经做到的Servlet对象，它会放在Servlet容器中间，所有客户端发过来的请求，都是通过DispatcherServlet分发给在Spring框架中的其他控制器对象的
DispatcherServlet是SpringMVC的中枢，负责从客户端获取用户的请求，然后根据用户请求的url，即RESTful资源的描述，来分发给对应的控制器对象，同时，也会把从控制器返回的结果产生HTTP Response返回给客户端

SpringMVC的具体处理前端请求过程



1. 当客户端发起一个请求到DispatcherServlet，即前端控制器时，DispatcherServlet会把请求交给Handler Mapping，去看应该由哪一个Controller来负责这样的一个请求
   HandlerMapping是根据配置信息（主要是注解）来在整个Spring容器中去找由哪个Controller来处理对应的URL
2. HandlerMapping（处理映射器）会把Controller返回给DispatcherServlet，前端控制器会去执行Controller对应的方法，Controller方法会调用Service层的方法、Dao层的方法、Mapper层的方法，去完成整个逻辑，当调用完成后，Controller返回一个ModelAndView的结构，其中既包含了返回的数据，也包含了展现数据需要的视图
3. DispatcherServlet会把其中的View提取出来，去从View Resolver中找应该由哪一个视图来完成数据的渲染，会根据视图的名称返回不同的视图解析器，来把视图展现成为一个View对象，再返回给DispatcherServlet
4. DispatcherServlet把模型、数据、视图渲染到一起，即把模型数据填到视图上，然后返回给客户端，客户端就拿到了一个有数据的视图

这是传统的方式，如果我们使用RESTful风格，服务器就不需要向前端去返回一个已经渲染好的界面的，我们只需要把控制器返回的数据包装成json的格式，通过HttpResponse返回给客户端即可，所以RestController中是不需要上图中的4 5这两部分的
而且我们大多数时候是把4 5部分这些渲染的部分写成静态网页，放在单独的web服务器上，这样可以减小服务器的压力

我们在SpringMVC中通过一系列的注解来标识控制器以及控制器的方法该处理什么样的url
我们大量使用Resultful的控制，即控制器的所有请求都是Resultful风格的

Resultful Controller的注解

@RestController：
• 与@Controller标签相同，用于标注在类定义前面，使得类会被认定为Controller对象
• 用于告知Spring容器，这是一个Resultful风格的控制器，该类所有方法的返回值需要以JSON格式写到Response的Body内

对于控制器类的每一个方法，包括Controller本身，我们可以用一系列注解来标识对应的url以及其对应的请求方法

注解	HTTP请求方法
@GetMapping	GET
@PostMapping	POST
@PutMapping	PUT
@PatchMapping	PATCH
@DeleteMapping	DELETE
@RequestMapping	可用于以上五种请求，需在method属性中指定

注：我们很少用patch来修改某个对象中的特定属性，但是我们更习惯用put

put是对某个对象的属性的全部修改，今天我们在实现时一般约定：当我们在用put请求时，如果对象中的属性是空的，说明这个属性是不需要修改的，我们所要求修改的是put发过来的对象中属性值不为空的属性

所以我们在定义一个对象的属性时，我们往往采用对象型的，如用Integer代替int去定义一个属性