Springmvc ----dispatcherServlet

OrderController method

IOC 容器＝Map 对象 装 bean

IOC容器到底是什么？

```
Map iocMap = new HashMap();
OrderService orderService = new OrderService();
iocMap.put("orderService", orderService);
```

如何使用？

```
@Autowired
private OrderService orderService;  //直接从iocMap.get("orderService")
```

IOC Container = map

```
1，Class<?> clazz = Class.forName(com.enjoy.james.service.impl.OrderServiceImpl); //Class对象，必须是包名+类名
2，OrderService orderService＝clazz.newInstance()   // 反射创建OrderServiceImpl实例
3，Class<?> clazz = orderService.getClass();          //根据实例拿到A类
4，Field[] fields = clazz.getDeclaredFields();        // 拿到类里面定义的所有属性
5，Method[] method = clazz.getMethods();              //获取类里的所有方法
6，method.invoke(orderService, args[] )     //从底层调用方法， args[]方法里的参数数组
```

可以通过 class.forname，也可以通过 object.getclass 得到对应 class。

技术预热2

Tomcat启动时加载SpringMVC开发的xxx.war流程？

Tomcat启动阶段

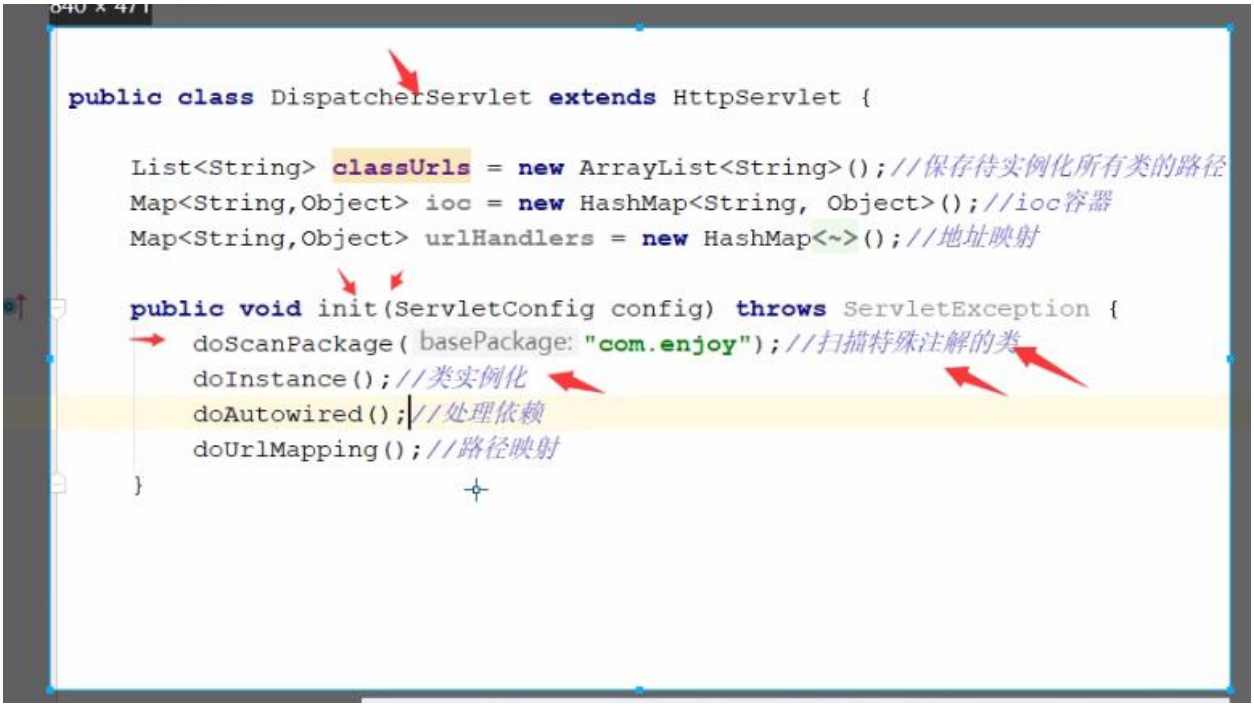
1，加载xx加载xxx.war
2，创建容器：创建Map iocMap = new HashMap();
3，ScanbasePackage：扫描war下的@Controller,@Service注解的类
4，实例化：将扫描到的类通过反射实例化，并存入到iocMap容器中
5，依赖注入：将存在依赖的bean进入注入
6，UrlMapping：http请求路径与Method建立映射关系

Tomcat运行阶段

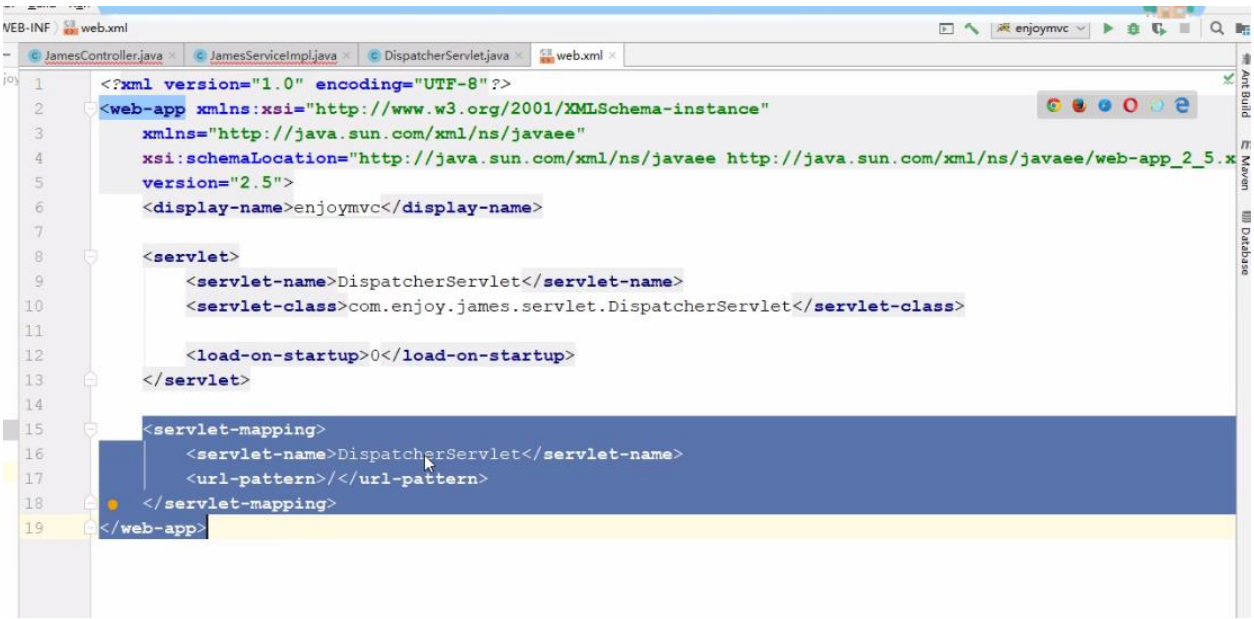
1，发送http请求，调用Servlet的doGet/doPost方法
2，找到从UrlMapping中找到对应的Method方法对象；
3，找到Method方法对象后，直接调用
4，响应返回结果

```
3， Map ioc = new HashMap();
    Ioc.put("orderController ", orderController )
    Ioc.put("orderController ", orderController )
IOC 容器是用来装 bean 的

requestMapping("/james")                /james/query → query method
Class OrderController{
requestMapping("query")
query(){



}
}
http://127.0.0.1:8080/mvc/james/query------query()---method
```

Tomcat 启动阶段是在 dispatchservlet(do get , do post)的 init 方法执行的

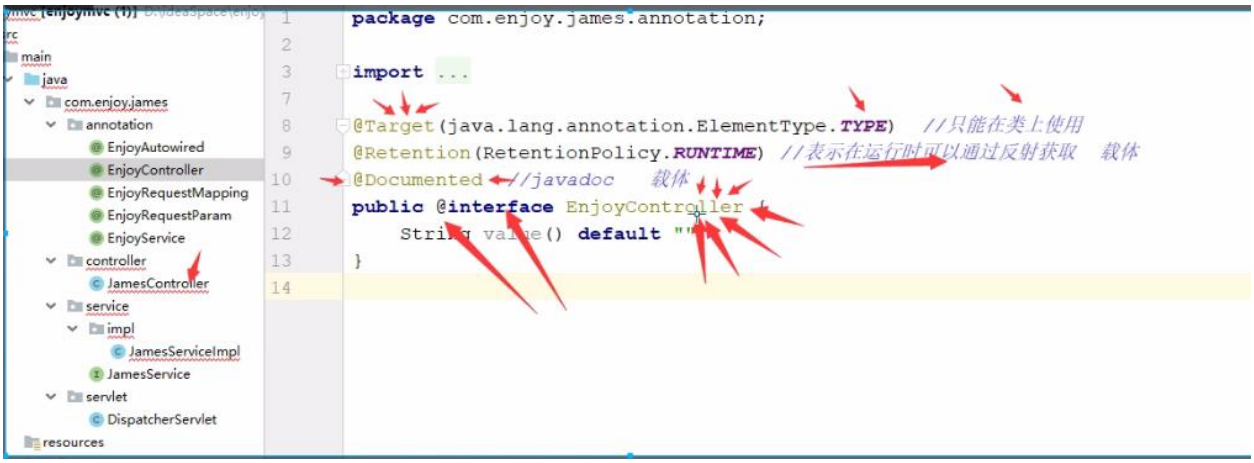


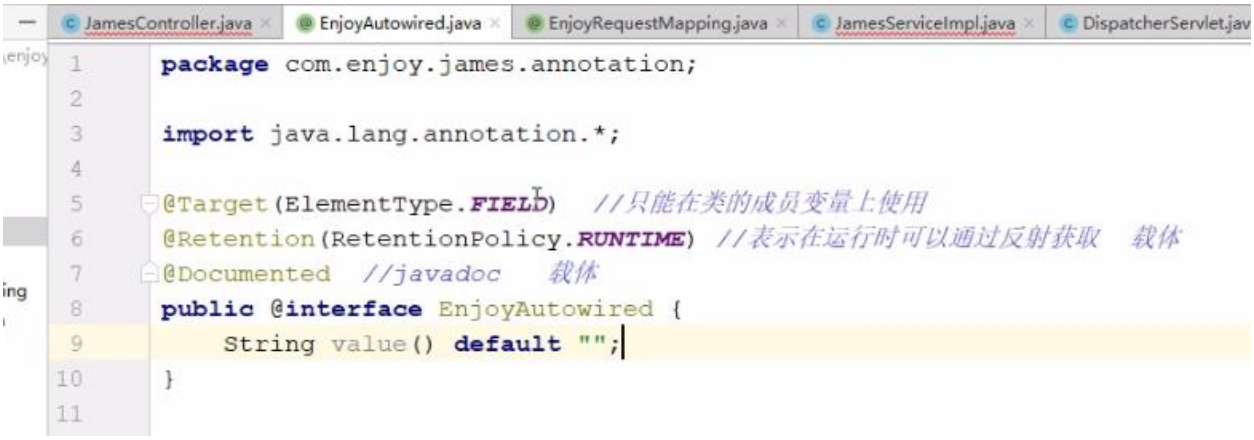
```
public class DispatcherServlet extends HttpServlet {

    List<String> classUrls = new ArrayList<String>();//保存待实例化所有类的路径
    Map<String,Object> ioc = new HashMap<String, Object>();//ioc容器
    Map<String,Object> urlHandlers = new HashMap<~>();//地址映射

    public void init(ServletConfig config) throws ServletException {
        doScanPackage( basePackage: "com.enjoy");//扫描特殊注解的类
        doInstance();//类实例化
        doAutowired();//处理依赖
        doUrlMapping();//路径映射
    }
```



```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.x
    version="2.5">
    <display-name>enjoymvc</display-name>

    <servlet>
        <servlet-name>DispatcherServlet</servlet-name>
        <servlet-class>com.enjoy.james.servlet.DispatcherServlet</servlet-class>

        <load-on-startup>0</load-on-startup>
    </servlet>

    <servlet-mapping>
        <servlet-name>DispatcherServlet</servlet-name>
        <url-pattern>/</url-pattern>
    </servlet-mapping>
</web-app>
```

配置注解，使用@Interface





Tomcat 启动时会扫描特殊注解类，将对应类放入 IOC，利用 reflection

```java
//tomcat启动EnjoyService
//iocMap.put("jamesServiceImpl",  new JamesServiceImpl())
@EnjoyService
public class JamesServiceImpl implements JamesService {

    public String query(String name, String age) { return "{name="+name+",age="+age+"}"; }
}
```