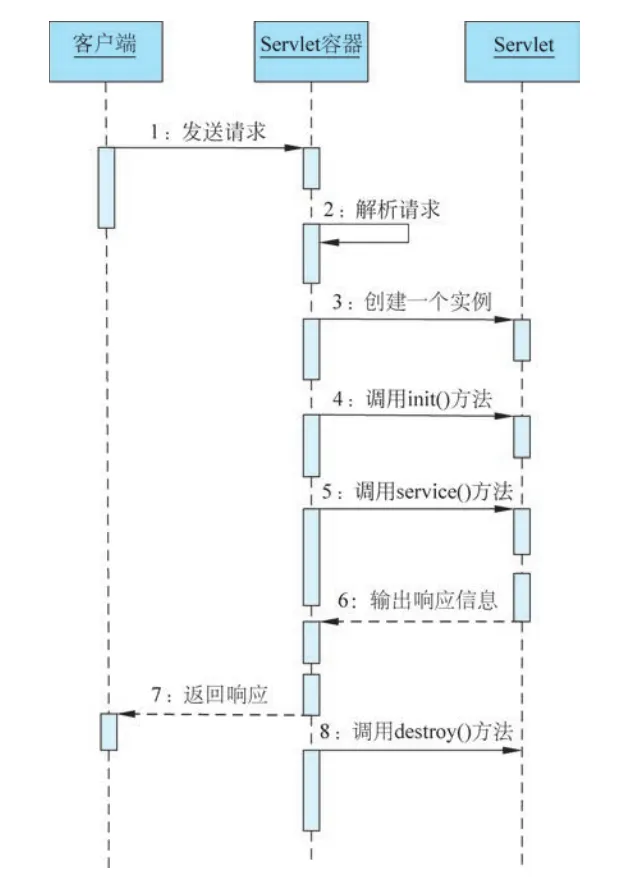
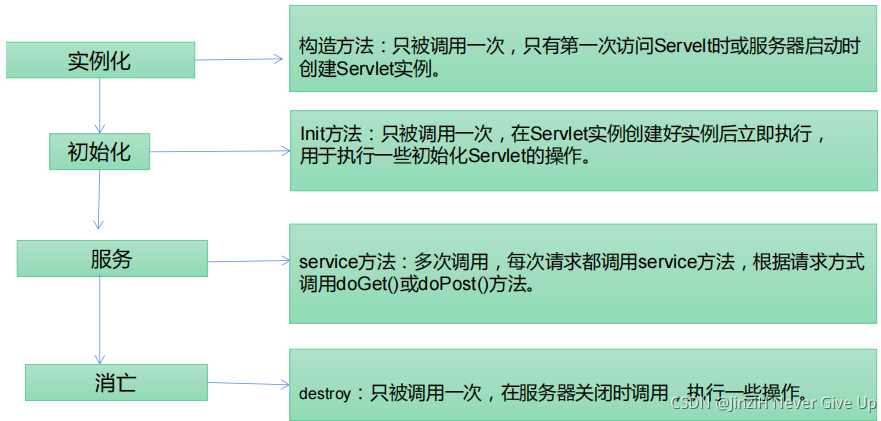
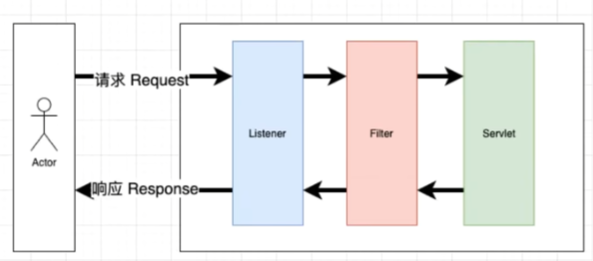
Web开发-JavaEE应用&Servlet技术&路由配置&生命周期&过滤器Filter&监听器Listen



#知识点：  
1、安全开发-原生PHP-超级全局变量  
2、安全开发-原生PHP-代码审计案例  
---------------------------------------------  
0、安全开发-原生PHP-数据库通讯  
1、安全开发-原生PHP-身份验证技术  
2、安全开发-Cookie&Session&Token  
3、安全开发-原生PHP-代码审计案例  
---------------------------------------------  
1、安全开发-原生PHP-弱类型脆弱  
2、安全开发-原生PHP-函数&数据类型  
3、安全开发-原生PHP-代码审计案例  
---------------------------------------------  
1、安全开发-原生PHP-开发组件集合  
2、安全开发-原生PHP-模版引擎渲染  
3、安全开发-原生PHP-第三方编辑器  
---------------------------------------------  
1、安全开发-框架技术-ThinkPHP  
2、安全开发-框架安全-版本&写法  
3、安全开发-ThinkPHP-代码审计案例  
---------------------------------------------  
1、安全开发-原生PHP-文件安全操作  
2、安全开发-原生PHP-上传读取删除包含等  
3、安全开发-原生PHP-代码审计文件安全---------------------------------------------  
1、安全开发-原生PHP-RCE安全  
2、安全开发-原生PHP-代码执行&命令注入  
3、安全开发-原生PHP-代码审计案例  
---------------------------------------------  
1、安全开发-原生PHP-PHP.INI安全  
2、安全开发-原生PHP-全局文件&单函数  
3、安全开发-原生PHP-流量检测&AI算法  
---------------------------------------------  
1、安全开发-原生JS-语法模型概念  
2、安全开发-Ajax技术-JQuery&Axios  
3、安全开发-Ajax应用-文件上传&用户登录---------------------------------------------  
1、安全开发-原生JS-DOM树&BOM对象  
2、安全开发-原生JS-DOM安全&安全案例  
---------------------------------------------  
1、安全开发-原生JS-数据加密&代码混淆  
2、安全开发-原生JS-数据解密安全案例  
---------------------------------------------  
1、安全开发-NodeJS-开发环境&功能实现  
2、安全开发-NodeJS-安全漏洞&案例分析  
3、安全开发-NodeJS-特有漏洞&代码审计  
---------------------------------------------  
1、安全开发-WebPack-构建打包器  
2、安全开发-WebPack-源码泄漏还原  
---------------------------------------------  
1、安全开发-微信小程序-搭建&开发&架构&安全  
2、安全开发-微信小程序-编译调试&反编译&泄露  
---------------------------------------------  
1、安全开发-JavaEE-Servlet技术  
2、安全开发-JavaEE-监听器&过滤器  
  
#章节点：(待补充)  
#具体点：(待补充)







演示案例：

* WEB开发-JavaEE-Servlet&路由&生命周期
* WEB开发-JavaEE-过滤器Filter&监听器Listen

参考：  
https://mp.weixin.qq.com/s/c\_4fOTBKDcByv8MZ9ayaRg  
https://blog.csdn.net/qq\_52173163/article/details/121110753  
  
#解释  
Servlet是运行在Web服务器或应用服务器上的程序,它是作为来自Web浏览器或其他HTTP客户端的请求和HTTP服务器上的数据库或应用程序之间的中间层。使用Servlet可以收集来自网页表单的用户输入，呈现来自数据库或者其他源的记录，还可以动态创建网页。本章内容详细讲解了web开发的相关内容以及servlet相关内容的配置使用,是JAVAEE开发的重中之重。  
  
#JavaEE-IDEA开发  
安装IDEA，激活后安装开发插件  
安装JDK,Tomcat,新建项目并配置  
  
#创建和使用Servlet  
1、创建一个类继承HttpServlet  
2、web.xml配置Servlet路由  
3、WebServlet配置Servlet路由  
  
#Servlet生命周期  
快捷键：alt+insert  
写入内置方法(init service(doget dopost等) destroy )  
  
#处理接受和回显  
●HttpServletRequest是ServletRequest的子接口  
getParameter(name) — String 通过name获得值  
getParameterValues — String[ ] 通过name获得多值  
●HttpServletResponse是ServletResponse的子接口   
setCharacterEncoding() 设置编码格式  
setContentType() 设置解析语言  
getWriter() 获得一个PrintWriter字符输出流输出数据  
PrintWriter 接受符合类型数据  
  
#JavaEE-过滤器-Filter  
Filter被称为过滤器，过滤器实际上就是对Web资源进行拦截，做一些处理后再交给下一个过滤器或Servlet处理，通常都是用来拦截request进行处理的，也可以对返回的 response进行拦截处理。开发人员利用filter技术，可以实现对所有Web资源的管理，例如实现权限访问控制、过滤敏感词汇、压缩响应信息等一些高级功能。  
1、创建过滤器  
2、过滤器内置方法  
init doFilter destroy  
3、过滤器触发流程  
@WebFilter("/xss")  
<filter>  
 <filter-name>xssFilter</filter-name>  
<filter-class>com.example.filter.xssFilter</filter-class>  
</filter>  
<filter-mapping>  
 <filter-name>xssFilter</filter-name>  
 <url-pattern>/xss</url-pattern>  
</filter-mapping>  
4、过滤器安全场景  
Payload检测，权限访问控制，红队内存马植入，蓝队清理内存马等  
内存马参考：https://mp.weixin.qq.com/s/hev4G1FivLtqKjt0VhHKmw  
5、案例演示  
xss攻击的检测，管理页面的cookie检测  
  
#JavaEE-监听器-Listen  
参考：https://blog.csdn.net/qq\_52797170/article/details/124023760  
-监听ServletContext、HttpSession、ServletRequest等域对象创建和销毁事件  
-监听域对象的属性发生修改的事件  
-监听在事件发生前、发生后做一些必要的处理  
1、创建监听器  
2、监听器内置方法  
3、监听器触发流程  
@WebListener  
<listener>  
 .......  
</listener>  
4、监听器安全场景  
代码审计中分析执行逻辑触发操作，红队内存马植入，蓝队清理内存马等  
5、案例演示  
session存在的监听

涉及资源：[资源下载地址](https://docs.qq.com/doc/DQ3Z6RkNpaUtMcEFr)