1. mockMvc的使用
2. 自定义全局异常处理类
3. 自定义过滤器的使用
4. 自定义一个过滤器，实现Filter接口，重写doFilter方法
5. @WebFilter(urlPatterns = **"/api/\*"**,filterName = **"loginFilter"**) 表示对/api/开头的所有请求，都先进入该过滤器进行处理

|  |
| --- |
| **package** com.example.lchtest.springbootdemo2.filter;   **import** javax.servlet.\*; **import** javax.servlet.annotation.WebFilter; **import** javax.servlet.http.HttpServletRequest; **import** javax.servlet.http.HttpServletResponse; **import** java.io.IOException;  */\*\*  \* 自定义过滤器  \* (1)@WebFilter注解标记一个类为filter,并被spring进行扫描  \* (2)主启动类上面加一个@ServletComponentScan注解，进行扫描  \* (3) 非前后端分离项目适用，前后端分离项目，可以使用拦截器  \* (4) 拦截器和过滤器区别：最明显的区别：过滤器可以过滤所有资源，拦截器只能拦截请求！  \*/ //@WebFilter(urlPatterns = "/\*",filterName = "loginFilter") // urlPatterns如果是配置成/\*则是拦截所有请求* @WebFilter(urlPatterns = **"/api/\*"**,filterName = **"loginFilter"**) **public class** LoginFilter **implements** Filter {  */\*\*  \*  \** ***@param filterConfig*** *过滤器配置  \** ***@throws*** *ServletException  \*/* @Override  **public void** init(FilterConfig filterConfig) **throws** ServletException {  System.***out***.println(**"LoginFilter init"**);  }   @Override  **public void** doFilter(ServletRequest request, ServletResponse response, FilterChain filterChain) **throws** IOException, ServletException {  System.***out***.println(**"自定义过滤器执行doFilter逻辑."**);  HttpServletRequest req = (HttpServletRequest)request;  HttpServletResponse resp = (HttpServletResponse) response;  String username = req.getParameter(**"username"**);  **if**(username.equals(**"test"**)){  filterChain.doFilter(req, resp);  } **else** {  *// 使用response.sendRedirect,login.html放在resources/static下面才可以，放到templates目录下不行，使用modelAnaView返回逻辑视图名可以跳到templates目录下，why??* resp.sendRedirect(**"/login.html"**);  }   }   @Override  **public void** destroy() {  System.***out***.println(**"LoginFilter destroy"**);  }  } |

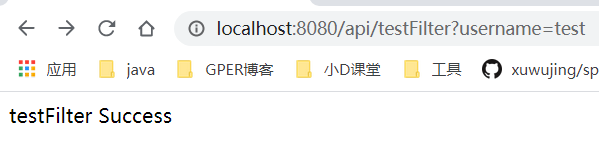
1. 主类上面加上@*ServletComponentScan注解*

|  |
| --- |
| @SpringBootApplication **@ServletComponentScan** **public class** App {  **public static void** main(String[] args) {  SpringApplication.*run*(App.**class**, args);  } } |

1. *定义一个controller进行测试*

|  |
| --- |
| **package** com.example.lchtest.springbootdemo2.controller;  **import** org.springframework.web.bind.annotation.GetMapping; **import** org.springframework.web.bind.annotation.RequestParam; **import** org.springframework.web.bind.annotation.RestController;  @RestController **public class** CustomFilterController {   @GetMapping(**"/api/testFilter"**)  **public** String testFilter(@RequestParam(**"username"**) String username){  System.***out***.println(**"username="** + username);  **return "testFilter Success"**;  } } |

1. 页面访问测试: <http://localhost:8080/api/testFilter?username=test>



如果username不等于test,就重定向到login.html, 访问

[http://localhost:8080/api/testFilter?username=test1](http://localhost:8080/api/testFilter?username=test), 结果如下

