打开swagger-ui.html 报错

Unable to infer base url. This is common when using dynamic servlet registration or when the API is behind an API Gateway. The base url is the root of where all the swagger resources are served. For e.g. if the api is available at http://example.org/api/v2/api-docs then the base url is http://example.org/api/. Please enter the location manually:

spring mvc

1 解决方式： spring mvc 的 url pattern 要改成 /

```xml
<servlet>
  <servlet-name>dispatcherServlet</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <init-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:spring/spring-mvc.xml</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>dispatcherServlet</servlet-name>
  <url-pattern>/</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>dispatcherServlet</servlet-name>
    <url-pattern>/v2/api-docs</url-pattern>
  </servlet-mapping>
  ...
```

2 如果有拦截器 需要在拦截器中排除下面swgger相关的

```xml
<!-- Spring 拦截器 -->
<mvc:interceptors>
    <mvc:interceptor>
        <mvc:mapping path="/**" />
        <mvc:exclude-mapping path="/webjars/**" />
        <mvc:exclude-mapping path="/swagger-resources/**" />
        <mvc:exclude-mapping path="/v2/**" />
        <mvc:exclude-mapping path="/swagger-ui.html"/>
        <bean class="com.ssyt.interceptor.DefaultSecurityInter
    </mvc:interceptor>
</mvc:interceptors>
```

spring boot

1 继承WebMvcConfigurerAdapter

2 添加拦截器过滤，静态资源过滤

```java
/**
 * Token拦截器
 *
 * @param registry
 */
@Override
public void addInterceptors(InterceptorRegistry registry) {
    registry.addInterceptor(privilegeInterceptor())
        .addPathPatterns("/**")
        .excludePathPatterns("/loginRpc/logout", "/loginRpc/login")
        .excludePathPatterns("/swagger-resources/**", "/webjars/**", "/v2/**",
"/swagger-ui.html");
    super.addInterceptors(registry);
}

@Override
public void addResourceHandlers(ResourceHandlerRegistry registry) {

registry.addResourceHandler("/**").addResourceLocations("classpath:/static/");
    registry.addResourceHandler("swagger-ui.html")
            .addResourceLocations("classpath:/META-INF/resources/");
    registry.addResourceHandler("/webjars/**")
            .addResourceLocations("classpath:/META-INF/resources/webjars/");
    super.addResourceHandlers(registry);
}
```

3 在接口返回的时候包装了一个外部包装

```java
@ControllerAdvice(basePackages = {"com.dsw.zfjd.controller"})
public class DataResponseBodyAdvice implements
ResponseBodyAdvice{
```

```
}
```

扫描包的时候不扫描Swagger返回的路径