SUMK框架进阶篇

[Sumk-web 2](#_Toc46477452)

[登录及加解密 2](#_Toc46477453)

[WebUtil工具类 3](#_Toc46477454)

[WebFilter 3](#_Toc46477455)

[几个常用点 3](#_Toc46477456)

## 初始化顺序

1. 扫描含有@Bean和@Table注解的class
2. 初始化所有的bean
3. 调用BeanCreateWatcher的afterCreate方法，每个watcher只调用一次
4. 对bean中的依赖进行注入
5. 调用BeanInjectWatcher的afterInject方法，每个watcher只调用一次
6. 调用Plugin的prepare方法
7. 在多线中里调用Plugin的startAsync方法
8. 所有Plugin启动完成后，调用Plugin的afterStarted方法

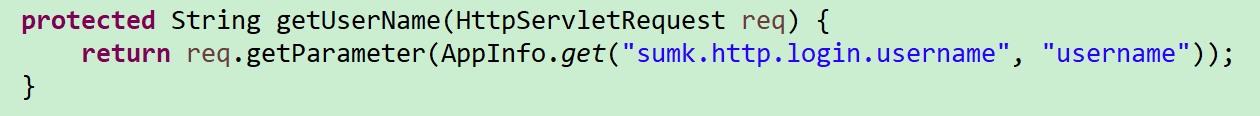
## Sumk-web

### 登录及加解密

1. 继承AbstractLoginServlet基类，实现里面的login方法。登录成功返回LoginObject.success(返回给客户端的消息，session对象)。session对象会保存在内存或redis中。开发者可以继承SessionObject扩展自己的字段。使用WebUtil. getUserObject()可以获取到session对象



1. Login()参数里的userName是通过getUserName()方法获取的，代码如下。重载该方法可重写。



1. 加解密：
   1. createEncryptKey()是生成加解密所用密钥的地方，可重写
   2. outputKey()用于将密钥传递给客户端，可重写
   3. 系统默认使用的是AES/CBC/PKCS5Padding算法，并且向量跟密钥一致。如果要使用其它算法，通过HttpCiphers. setEncryptor(HttpEncryptor encryptor)可自定义加解密方式。
   4. @Web只要配置requestType和responseType就能使用上面的加解密
2. AbstractLoginServlet的service()是登录的入口，从这个方法可以看出整个登录的逻辑。各方法之间如果涉及到传参，可通过request. setAttribute(String name, Object o)来传递参数。这里面的对象不会传递到客户端。
3. sumk.http.login.path参数可以配置登录的访问路径
4. 登录属于特殊接口，它属于原始servlet类型，sumk-web的大多数特性，都跟它无关

### WebUtil工具类

http的工具方法主要在WebUtil中，这个工具类只在@Web注解的接口里才能使用。无论是进入了@Web注解的方法，还是在之前的WebFilter里都能使用。LoginServlet里无法使用该工具类

### WebFilter

Web服务的拦截器，可以用于做权限校验等。校验不过直接return，如果能校验通过，调用callNextFilter(ctx)继续后面的调用。

### 几个常用点

* Sumk-http框架使用的异常码都是3位数。用户自定义的异常码，尽量避开3位数。
* 当有异常发生的时候，客户端收到的响应，http状态码是550（2.9以前的版本为499，可以使用sumk.http.errorcode定义）。Body里是json结构，包括code和message两个字段。

待续。。。