用户如果知道功能路径可以通过敲路径访问，但如果是一个需要登陆才能访问的路径，但你直接通过敲访问路径访问的话，可能会造成问题，下面的例子是围绕未登录但直接访问账号设置功能来讨论

自定义注解：

（1）常见的元注解：

@Target：声明自定义的注解可以写在哪个位置，作用在哪个类型上（定义在类、方法、属性上）

@Retention:声明自定义注解保留的时间（有效的时间：编译时有效，运行时有效）

@Document：生成文档时是否将自定义注解也带上

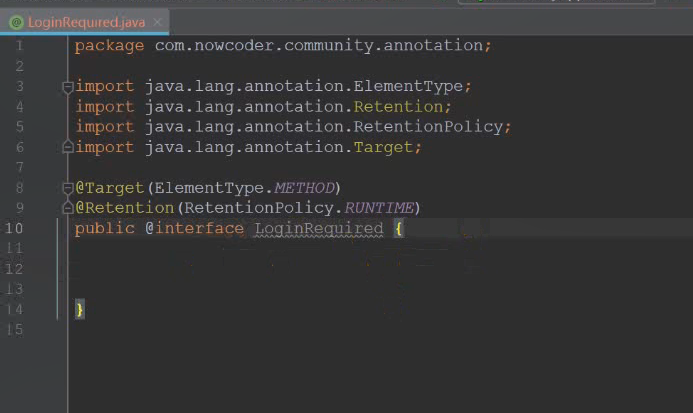
@Inherited：用于继承，子类继承一个父类，父类有自定义注解，子类是否要将注解继承

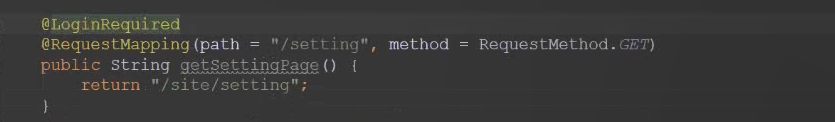
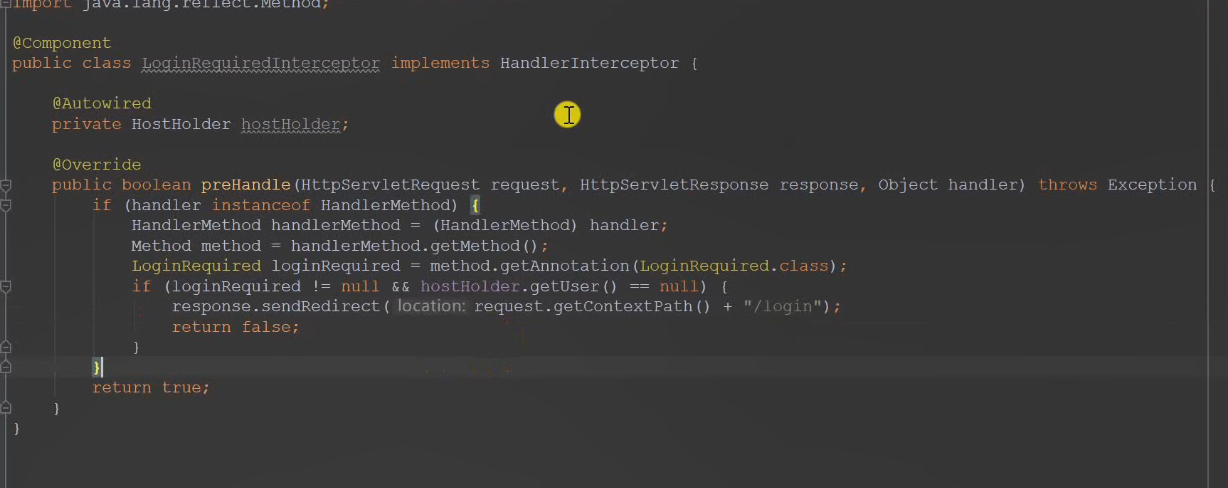
（2）如何读取注解：

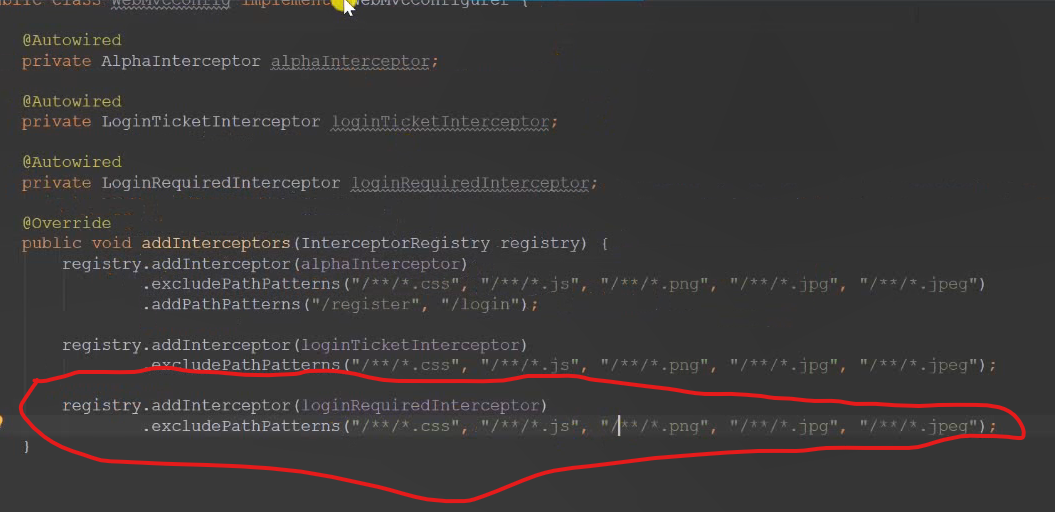
Method.getDeclaredAnnotations() 通过反射Method类型对象调用这个方法获取这个方法上所有的注解

Method.getAnnotation(Class annotationClass)尝试获取annotation（注解）Class类型的注解

创建注解

作用域在方法上，有效的时间是程序运行时

在想要拦截的方法上加上此注解定义拦截器，因为只需要拦截方法，所以先判断handler是否是拦截方法类型，从拦截方法中获取方法，通过反射得到方法的此注解类型的注解，判断只要拥有此注解的且user为空（还未登录）则返回登录页面，return false不继续往下执行

拦截器的使用范围，不拦截一些静态的东西，不用管拦截范围，按默认全局就可以，因为在定义拦截器的时候已经拥有了相关的逻辑，也可以像第一个拦截器定义域一样，添加定义域不按默认值，但这样的话太麻烦，因为可能有好多路径都需要添加，一个一个麻烦