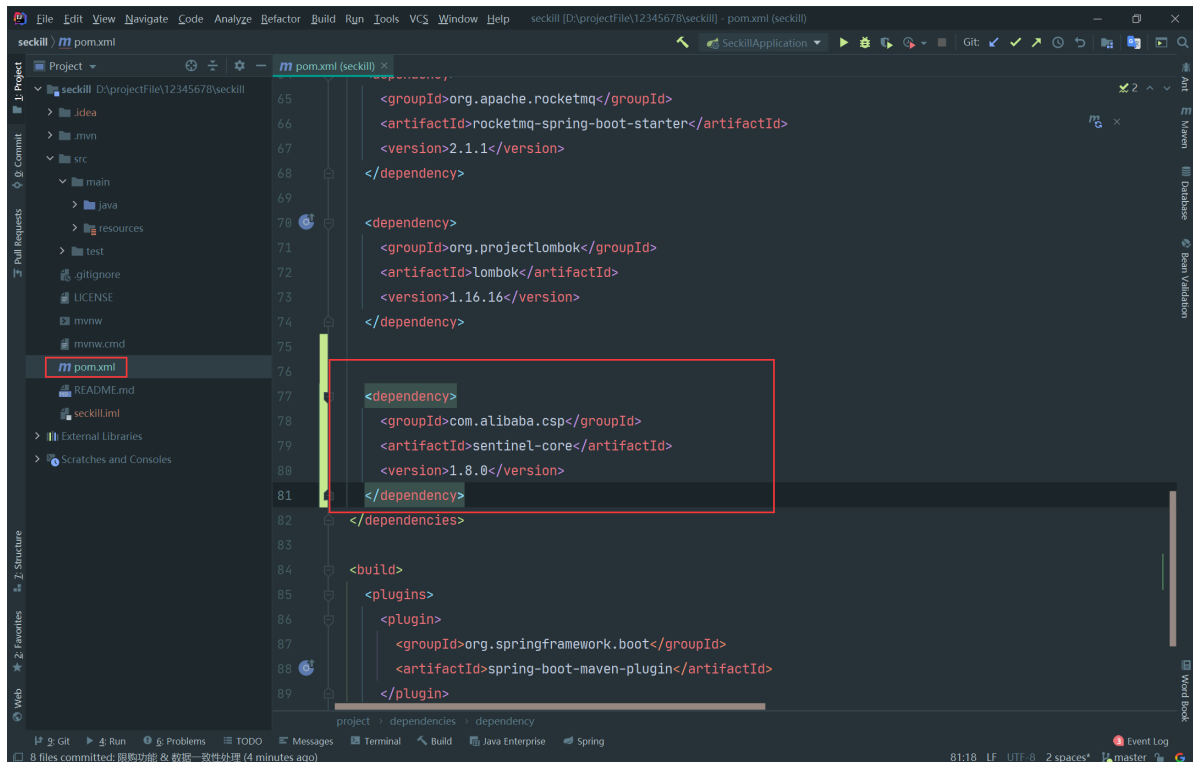


# 《保姆级教程》第十三集 整合 Sentinel & Sentinel 与秒杀业务的整合

## 1. SpringBoot 整合 Sentinel

### 1. 导入 pom 依赖

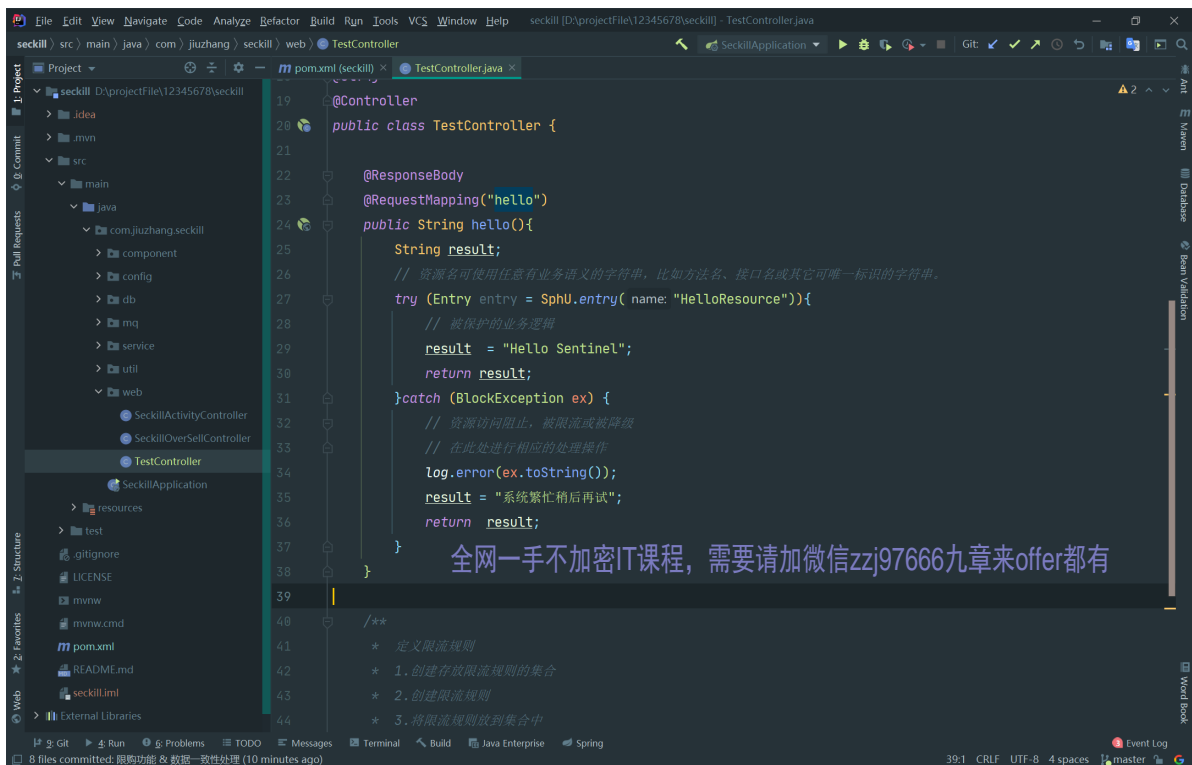


```
<dependency>  
  <groupId>com.alibaba.csp</groupId>  
  <artifactId>sentinel-core</artifactId>  
  <version>1.8.0</version>  
</dependency>
```

## 2. 测试 Sentinel

### 1. 创建测试TestController





```
@Slf4j
@Controller
public class TestController {

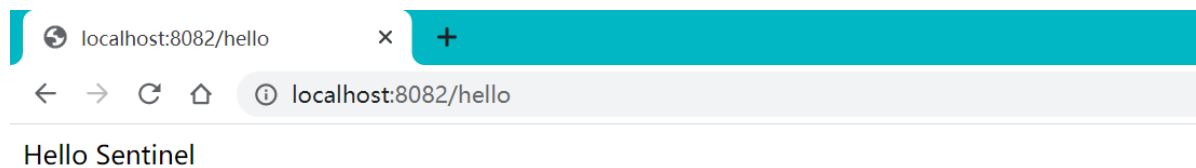
    @ResponseBody
    @RequestMapping("hello")
    public String hello(){
        String result;
        // 资源名可使用任意有业务语义的字符串，比如方法名、接口名或其它可唯一标识的字符串。
        try (Entry entry = SphU.entry("HelloResource")){
            // 被保护的逻辑
            result = "Hello Sentinel";
            return result;
        }catch (BlockException ex) {
            // 资源访问阻止，被限流或被降级
            // 在此处进行相应的处理操作
            log.error(ex.toString());
            result = "系统繁忙稍后再试";
            return result;
        }
    }

    /**
     * 定义限流规则
     * 1. 创建存放限流规则的集合
     * 2. 创建限流规则
     * 3. 将限流规则放到集合中
     * 4. 加载限流规则
     * @PostConstruct 当前类的构造函数执行完之后执行
     */
    @PostConstruct
    public void initFlowRules(){
        //1. 创建存放限流规则的集合
        List<FlowRule> rules = new ArrayList<>();
```

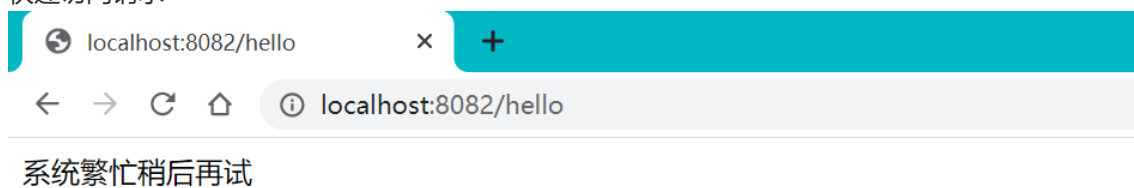
```
//2.创建限流规则
FlowRule rule = new FlowRule();
//定义资源，表示sentinel会对那个资源生效
rule.setResource("HelloResource");
//定义限流规则类型,QPS类型
rule.setGrade(RuleConstant.FLOW_GRADE_QPS);
//定义QPS每秒通过的请求数
rule.setCount(2);
//3.将限流规则放到集合中
rules.add(rule);
//4.加载限流规则
FlowRuleManager.loadRules(rules);
}
}
```

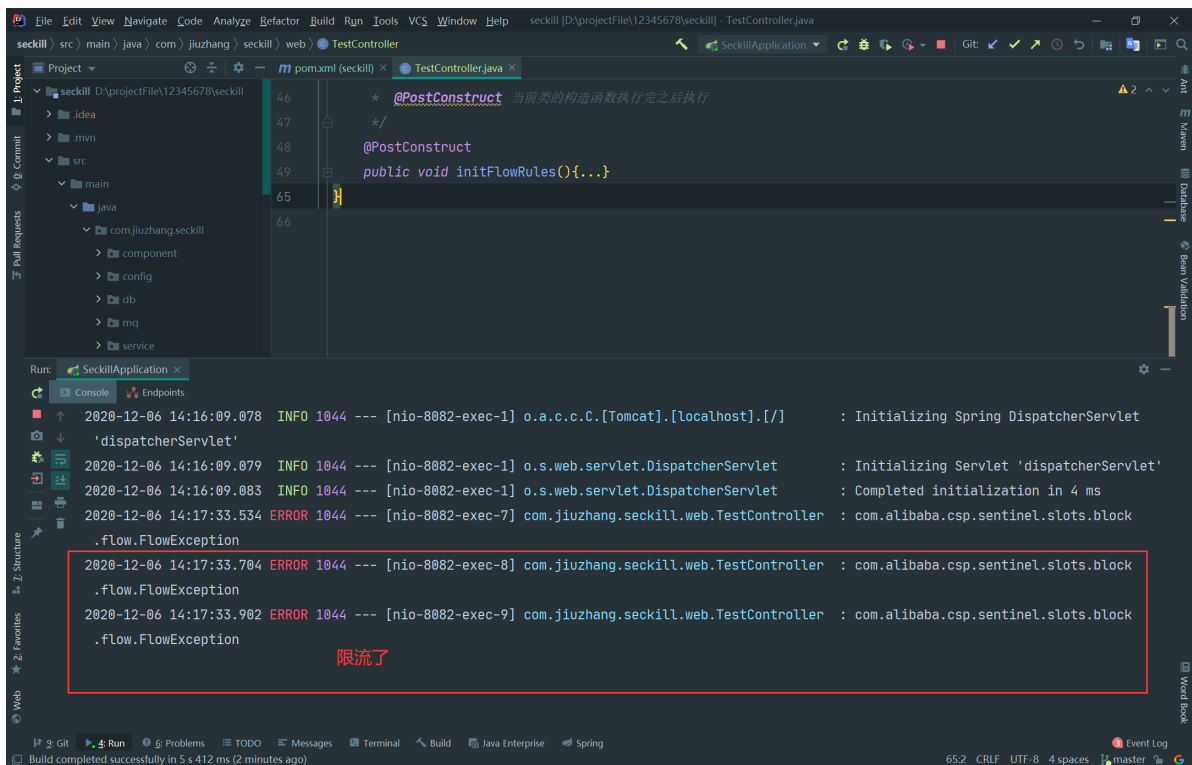
## 2. 启动项目

访问请求: <http://localhost:8082/hello>



## 3. 快速访问请求

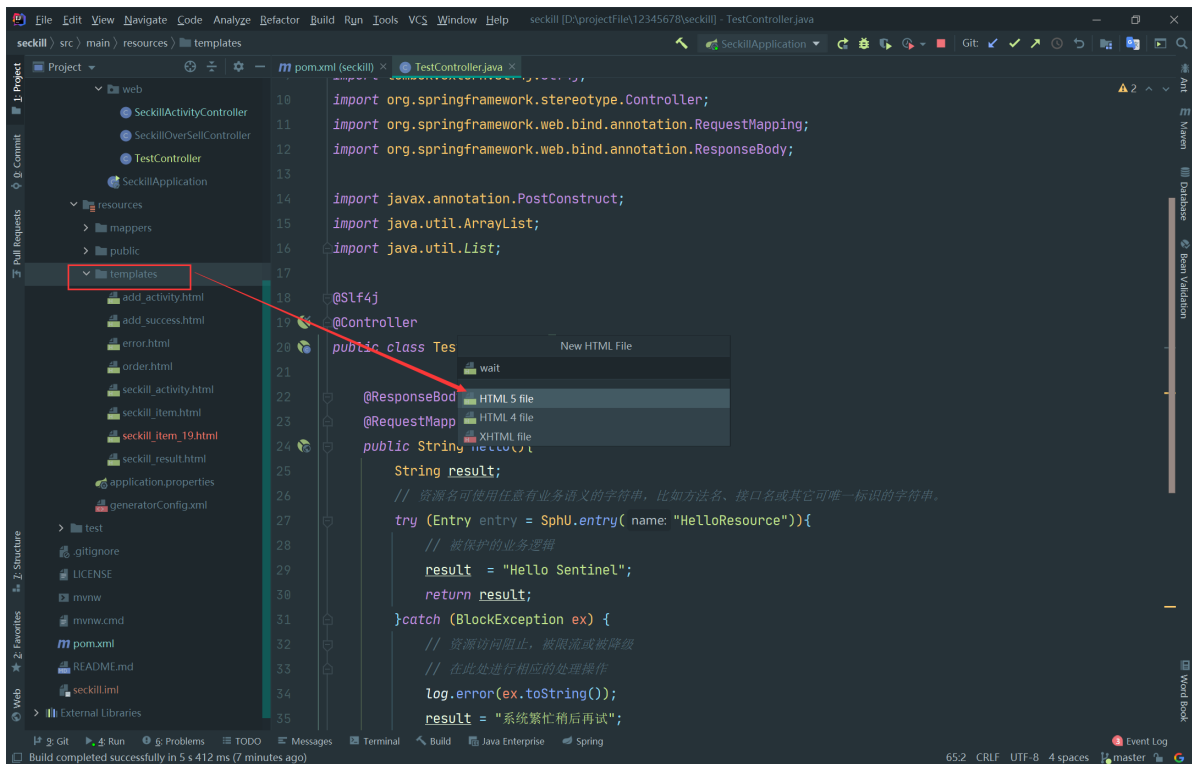


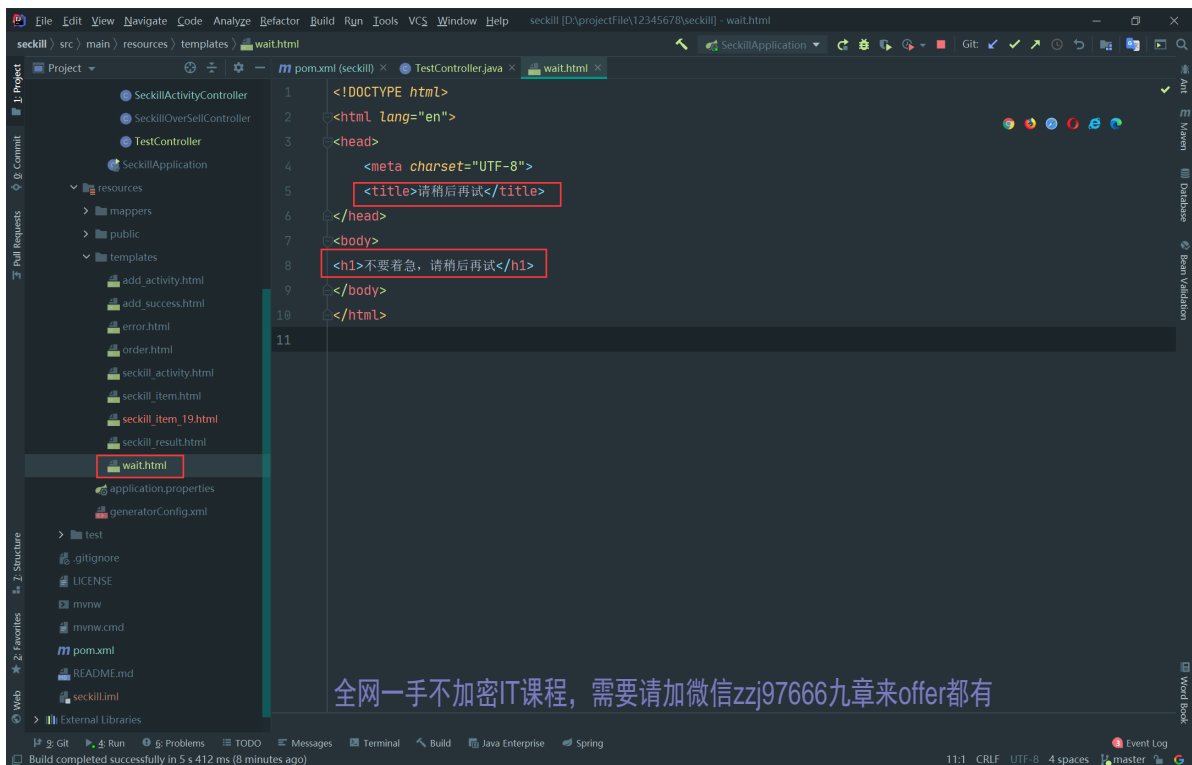


### 3. Sentinel 与秒杀业务整合

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

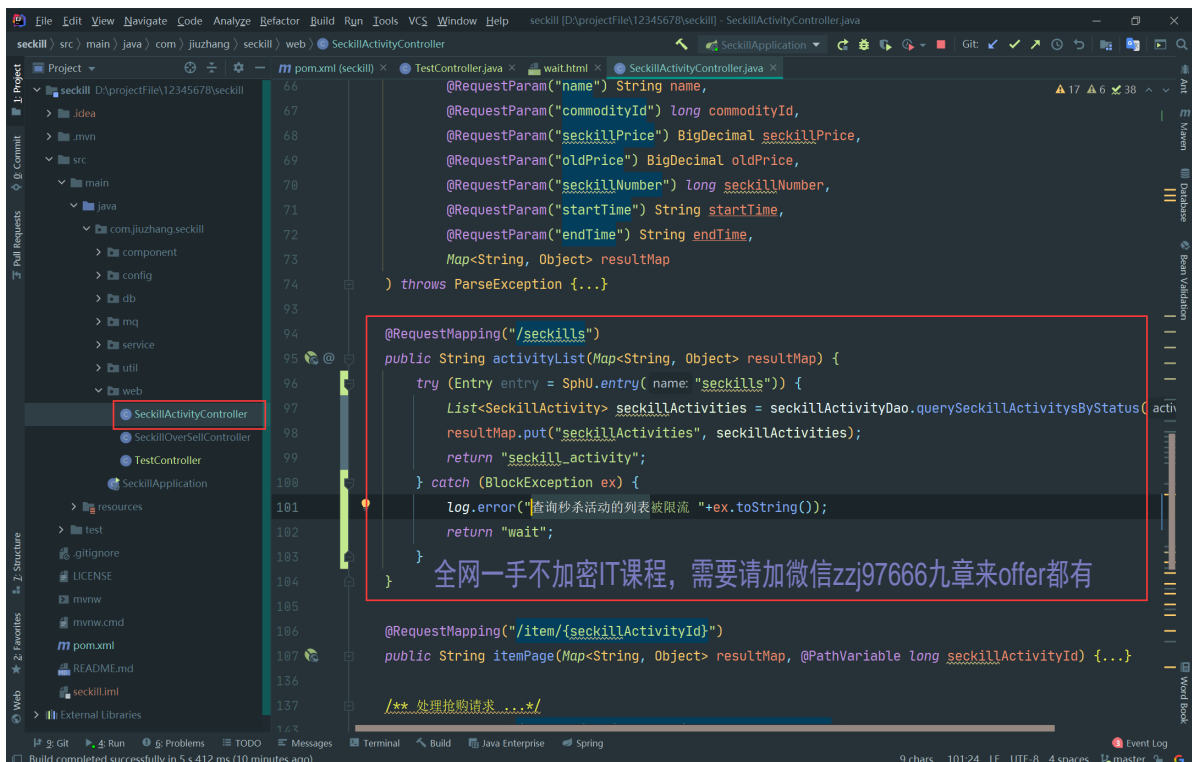
#### 1. 创建一个简单的等待页面





```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>请稍后再试</title>
</head>
<body>
<h1>不要着急，请稍后再试</h1>
</body>
</html>
```

## 2. 改造 查询秒杀活动的列表的方法

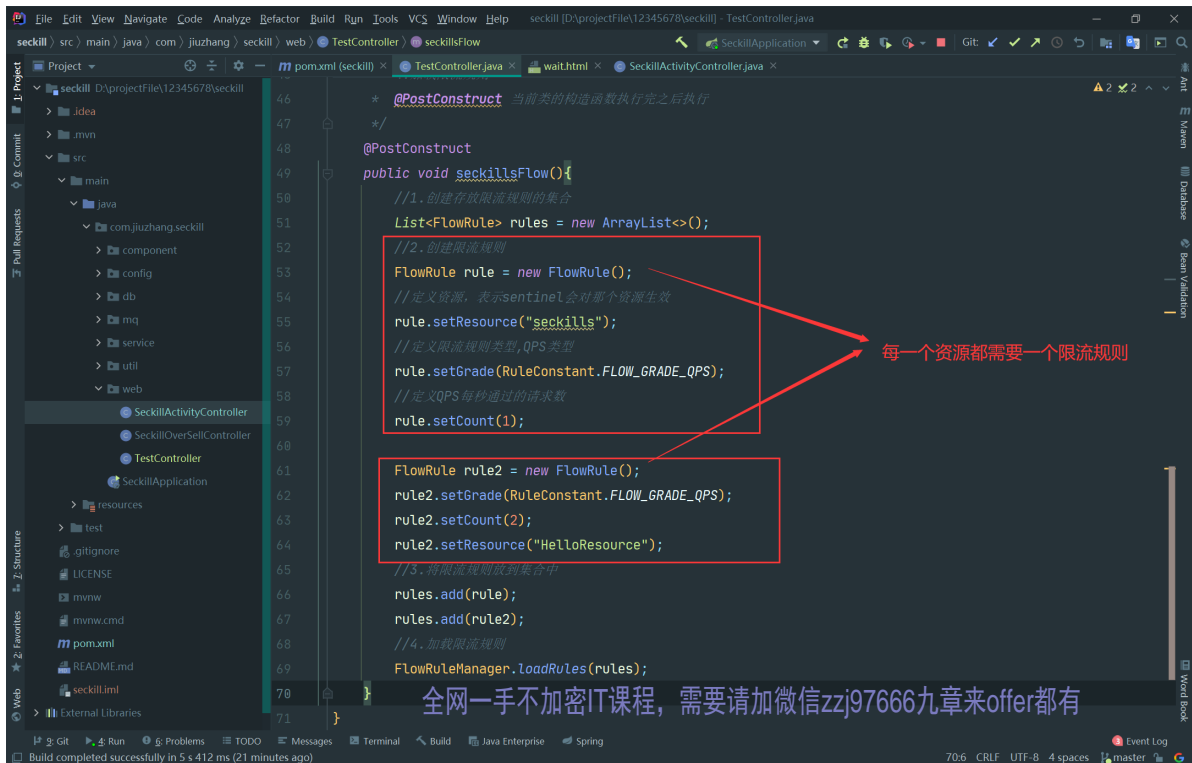


```

@RequestMapping("/seckill")
public String activityList(Map<String, Object> resultMap) {
    try (Entry entry = SphU.entry("seckill")) {
        List<SeckillActivity> seckillActivities =
            seckillActivityDao.querySeckillActivitiesByStatus(1);
        resultMap.put("seckillActivities", seckillActivities);
        return "seckill_activity";
    } catch (BlockException ex) {
        log.error("查询秒杀活动的列表被限流 "+ex.toString());
        return "wait";
    }
}

```

### 3. 编写 对应的限流规则



```

@PostConstruct
public void seckillFlow(){
    //1. 创建存放限流规则的集合
    List<FlowRule> rules = new ArrayList<>();
    //2. 创建限流规则
    FlowRule rule = new FlowRule();
    //定义资源, 表示sentinel会对那个资源生效
    rule.setResource("seckill");
    //定义限流规则类型, QPS类型
    rule.setGrade(RuleConstant.FLOW_GRADE_QPS);
    //定义QPS每秒通过的请求数
    rule.setCount(1);

    FlowRule rule2 = new FlowRule();
    rule2.setGrade(RuleConstant.FLOW_GRADE_QPS);
    rule2.setCount(2);
    rule2.setResource("HelloResource");
    //3. 将限流规则放到集合中
    rules.add(rule);
    rules.add(rule2);
}

```

//4.加载限流规则

```
FlowRuleManager.loadRules(rules);  
}
```

## 4. 测试Sentinel 与秒杀业务整合

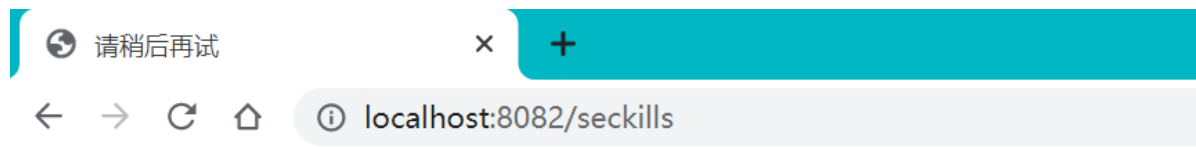
1. 启动项目，访问 秒杀活动的列表页

<http://localhost:8082/seckill>

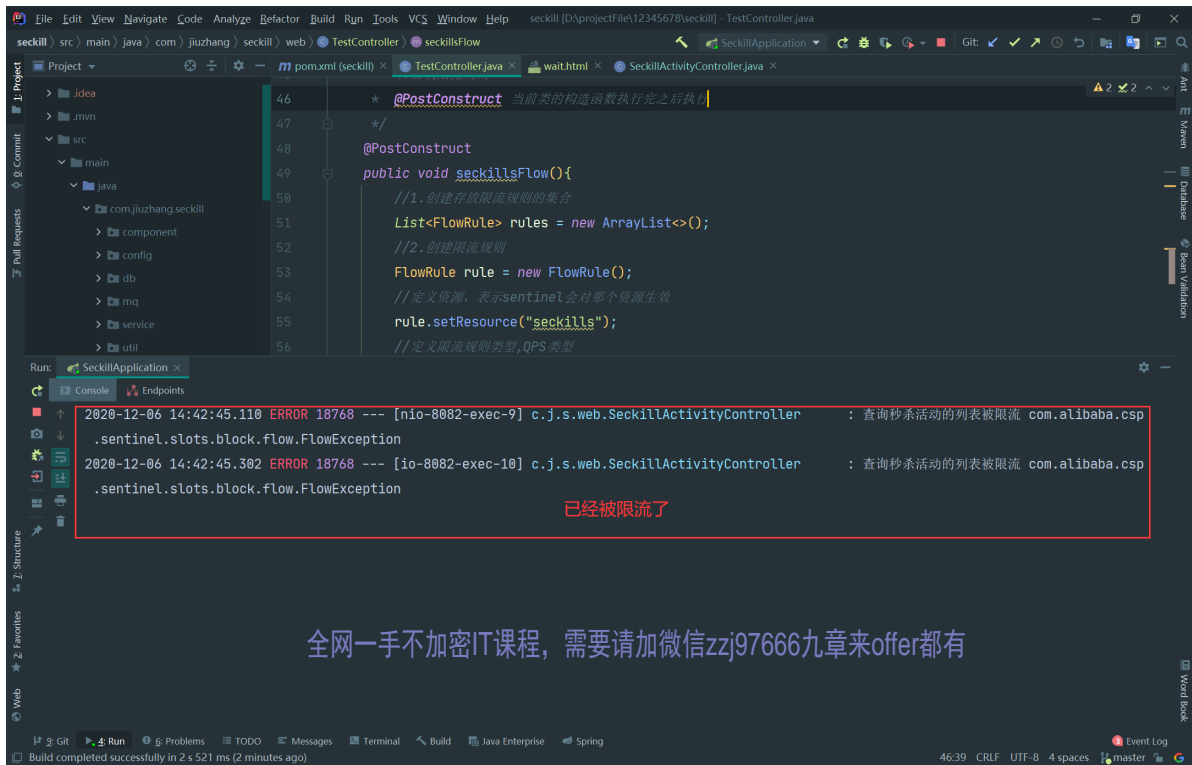


全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

2. 快速访问



# 不要着急，请稍后再试



全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

<http://localhost:8082/hello> 这个请求同样有限流，这里就不演示了

大家可以对其他资源也进行保护一下。最主要的就是那个 处理抢购请求 的资源。