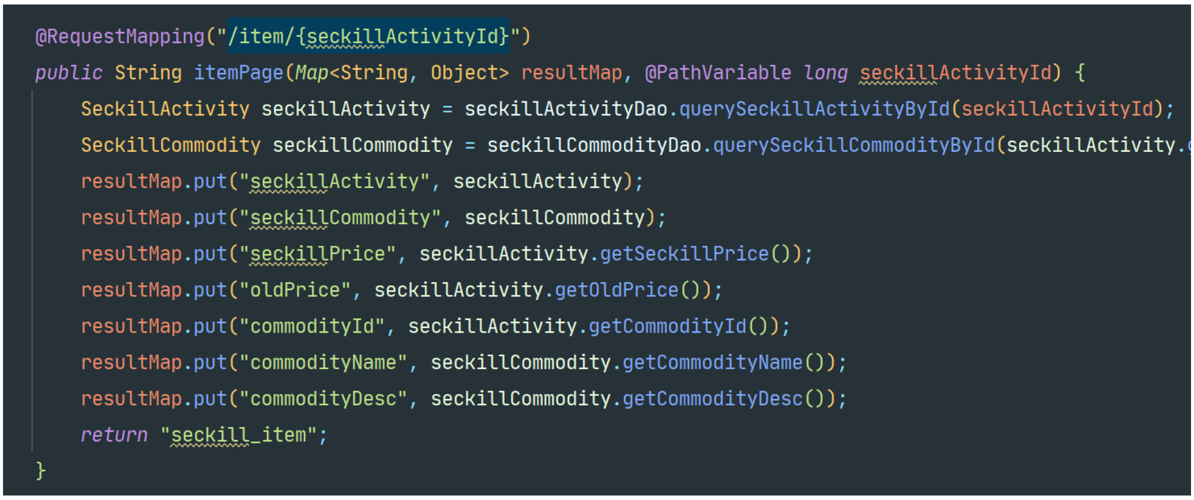
《保姆级教程》第十二集 缓存预热实现 & 同步倒计时 & 页面静态化

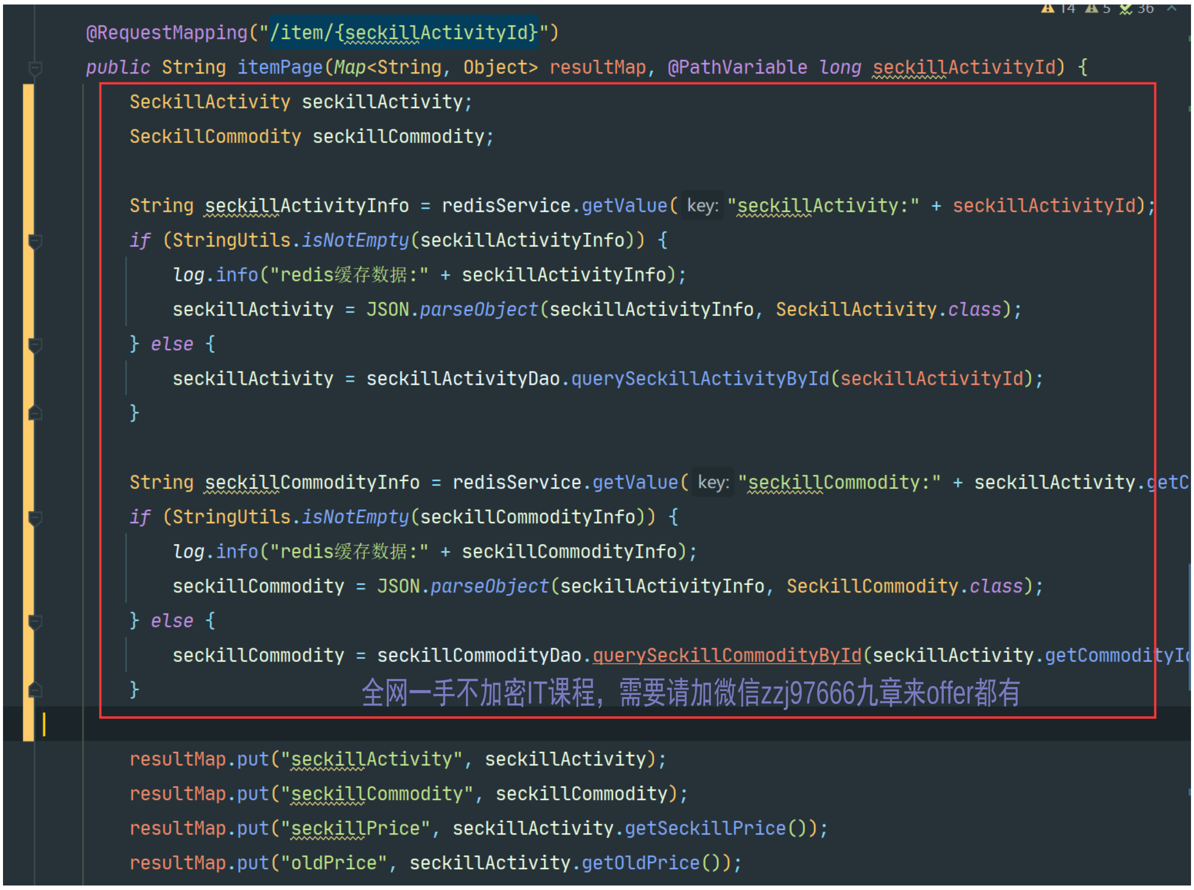
1. 秒杀商品详情改造

1. 改造前



```
@RequestMapping("/item/{seckillActivityId}")
public String itemPage(Map<String, Object> resultMap, @PathVariable long seckillActivityId) {
    SeckillActivity seckillActivity = seckillActivityDao.querySeckillActivityById(seckillActivityId);
    SeckillCommodity seckillCommodity = seckillCommodityDao.querySeckillCommodityById(seckillActivity.
    resultMap.put("seckillActivity", seckillActivity);
    resultMap.put("seckillCommodity", seckillCommodity);
    resultMap.put("seckillPrice", seckillActivity.getSeckillPrice());
    resultMap.put("oldPrice", seckillActivity.getOldPrice());
    resultMap.put("commodityId", seckillActivity.getCommodityId());
    resultMap.put("commodityName", seckillCommodity.getCommodityName());
    resultMap.put("commodityDesc", seckillCommodity.getCommodityDesc());
    return "seckill_item";
}
```

2. 改造后



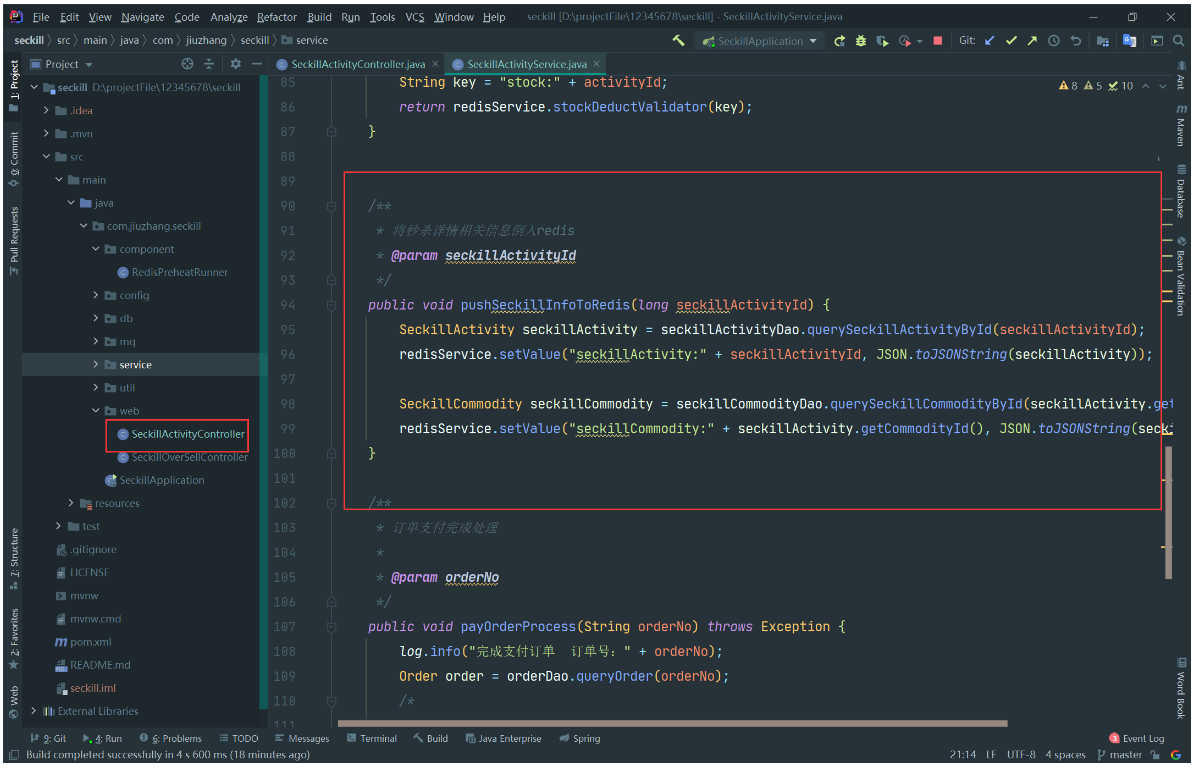
```
@RequestMapping("/item/{seckillActivityId}")
public String itemPage(Map<String, Object> resultMap, @PathVariable long seckillActivityId) {
    SeckillActivity seckillActivity;
    SeckillCommodity seckillCommodity;

    String seckillActivityInfo = redisService.getValue( key: "seckillActivity:" + seckillActivityId);
    if (StringUtils.isNotEmpty(seckillActivityInfo)) {
        log.info("redis缓存数据:" + seckillActivityInfo);
        seckillActivity = JSON.parseObject(seckillActivityInfo, SeckillActivity.class);
    } else {
        seckillActivity = seckillActivityDao.querySeckillActivityById(seckillActivityId);
    }

    String seckillCommodityInfo = redisService.getValue( key: "seckillCommodity:" + seckillActivity.getC
    if (StringUtils.isNotEmpty(seckillCommodityInfo)) {
        log.info("redis缓存数据:" + seckillCommodityInfo);
        seckillCommodity = JSON.parseObject(seckillActivityInfo, SeckillCommodity.class);
    } else {
        seckillCommodity = seckillCommodityDao.querySeckillCommodityById(seckillActivity.getCommodityI
    }

    resultMap.put("seckillActivity", seckillActivity);
    resultMap.put("seckillCommodity", seckillCommodity);
    resultMap.put("seckillPrice", seckillActivity.getSeckillPrice());
    resultMap.put("oldPrice", seckillActivity.getOldPrice());
```

```
@RequestMapping("/item/{seckillActivityId}")
```

```
    public String itemPage(Map<String, Object> resultMap, @PathVariable long
seckillActivityId) {
        SeckillActivity seckillActivity;
        SeckillCommodity seckillCommodity;

        String seckillActivityInfo = redisService.getValue("seckillActivity:" +
seckillActivityId);
        if (StringUtils.isNotEmpty(seckillActivityInfo)) {
            log.info("redis缓存数据:" + seckillActivityInfo);
            seckillActivity = JSON.parseObject(seckillActivityInfo,
SeckillActivity.class);
        } else {
            seckillActivity =
seckillActivityDao.querySeckillActivityById(seckillActivityId);
        }

        String seckillCommodityInfo = redisService.getValue("seckillCommodity:"
+ seckillActivity.getCommodityId());
        if (StringUtils.isNotEmpty(seckillCommodityInfo)) {
            log.info("redis缓存数据:" + seckillCommodityInfo);
            seckillCommodity = JSON.parseObject(seckillActivityInfo,
SeckillCommodity.class);
        } else {
            seckillCommodity =
seckillCommodityDao.querySeckillCommodityById(seckillActivity.getCommodityId());
        }

        resultMap.put("seckillActivity", seckillActivity);
        resultMap.put("seckillCommodity", seckillCommodity);
        resultMap.put("seckillPrice", seckillActivity.getSeckillPrice());
        resultMap.put("oldPrice", seckillActivity.getOldPrice());
        resultMap.put("commodityId", seckillActivity.getCommodityId());
        resultMap.put("commodityName", seckillCommodity.getCommodityName());
        resultMap.put("commodityDesc", seckillCommodity.getCommodityDesc());
        return "seckill_item";
    }
```

2. 商品信息和活动写入redis

> 这个方法在之前给大家的源代码中应该就有了


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
seckill [D:\projectFile\12345678\seckill] - SeckillActivityService.java
seckill 〉 src 〉 main 〉 java 〉 com 〉 jiuzhang 〉 seckill 〉 service
SeckillApplication
Project
SeckillActivityController.java
SeckillActivityService.java
seckill D:\projectFile\12345678\seckill
.idea
.mvn
src
main
java
com.jiuzhang.seckill
component
RedisPreheatRunner
config
db
mq
service
util
web
SeckillActivityController
SeckillOverSellController
SeckillApplication
resources
test
.gitignore
LICENSE
mvnw
mvnw.cmd
pom.xml
README.md
seckill.iml
External Libraries
9: Git
4: Run
6: Problems
TODO
Messages
Terminal
Build
Java Enterprise
Spring
Event Log
21:14 LF UTF-8 4 spaces master

```java
        String key = "stock:" + activityId;
        return redisService.stockDeductValidator(key);
    }


    /**
     * 将秒杀详情相关信息倒入redis
     * @param seckillActivityId
     */
    public void pushSeckillInfoToRedis(long seckillActivityId) {
        SeckillActivity seckillActivity = seckillActivityDao.querySeckillActivityById(seckillActivityId);
        redisService.setValue("seckillActivity:" + seckillActivityId, JSON.toJSONString(seckillActivity));

        SeckillCommodity seckillCommodity = seckillCommodityDao.querySeckillCommodityById(seckillActivity.get
        redisService.setValue("seckillCommodity:" + seckillActivity.getCommodityId(), JSON.toJSONString(seck
    }

    /**
     * 订单支付完成处理
     *
     * @param orderNo
     */
    public void payOrderProcess(String orderNo) throws Exception {
        log.info("完成支付订单  订单号: " + orderNo);
        Order order = orderDao.queryOrder(orderNo);
        /*
```

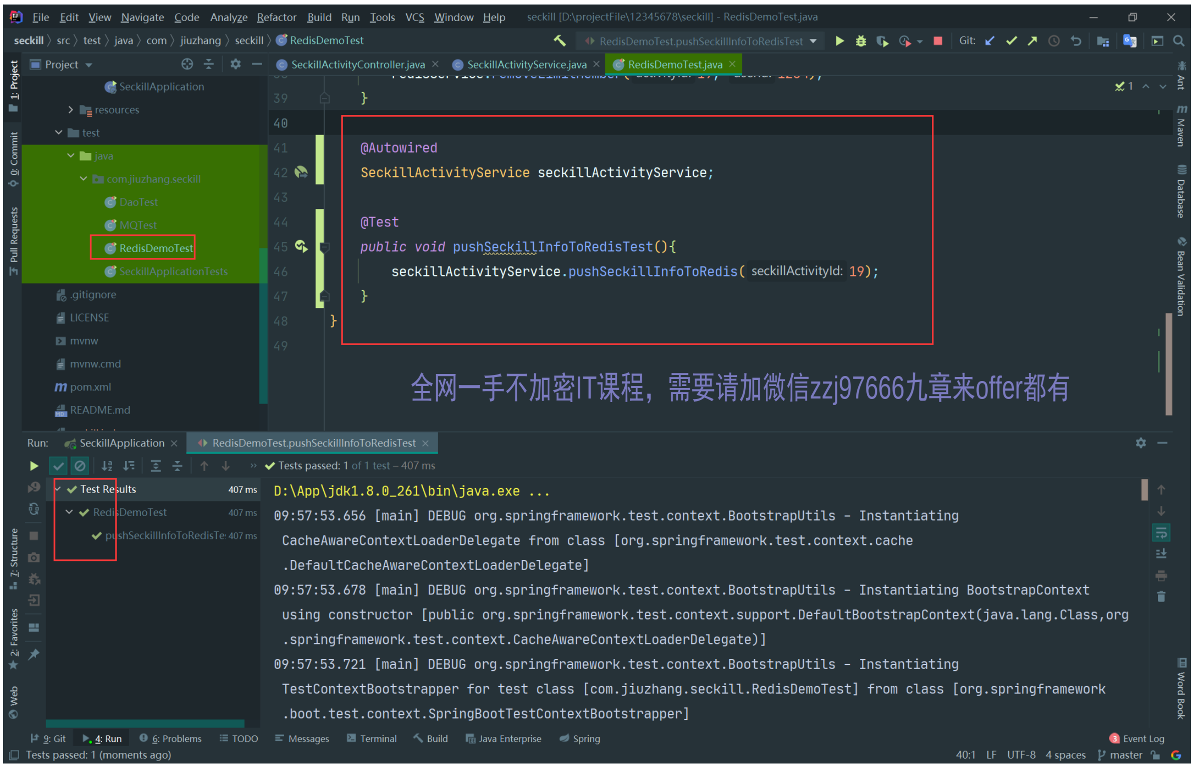
Build completed successfully in 4 s 600 ms (18 minutes ago)

```java
    /**
     * 将秒杀详情相关信息倒入redis
     * @param seckillActivityId
     */
    public void pushSeckillInfoToRedis(long seckillActivityId) {
        SeckillActivity seckillActivity =
seckillActivityDao.querySeckillActivityById(seckillActivityId);
        redisService.setValue("seckillActivity:" + seckillActivityId,
JSON.toJSONString(seckillActivity));

        SeckillCommodity seckillCommodity =
seckillCommodityDao.querySeckillCommodityById(seckillActivity.getCommodityId());
        redisService.setValue("seckillCommodity:" +
seckillActivity.getCommodityId(), JSON.toJSONString(seckillCommodity));
    }
```

3. 测试缓存预热

> 我们来测试一下缓存预热

1. 将 id = 19的商品进行缓存预热



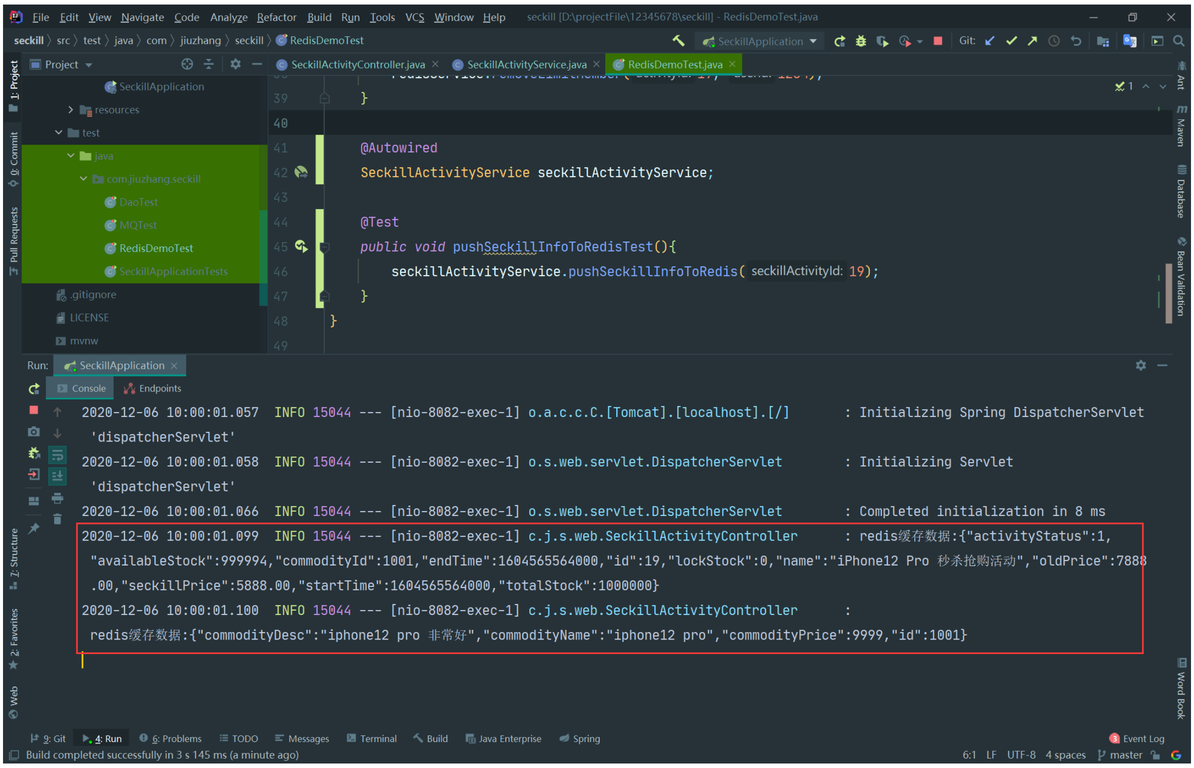
```java
    @Autowired
    SeckillActivityService seckillActivityService;

    @Test
    public void pushSeckillInfoToRedisTest(){
        seckillActivityService.pushSeckillInfoToRedis( seckillActivityId: 19);
    }
}
```

```
D:\App\jdk1.8.0_261\bin\java.exe ...
09:57:53.656 [main] DEBUG org.springframework.test.context.BootstrapUtils - Instantiating
 CacheAwareContextLoaderDelegate from class [org.springframework.test.context.cache
 .DefaultCacheAwareContextLoaderDelegate]
09:57:53.678 [main] DEBUG org.springframework.test.context.BootstrapUtils - Instantiating BootstrapContext
 using constructor [public org.springframework.test.context.support.DefaultBootstrapContext(java.lang.Class,org
 .springframework.test.context.CacheAwareContextLoaderDelegate)]
09:57:53.721 [main] DEBUG org.springframework.test.context.BootstrapUtils - Instantiating
 TestContextBootstrapper for test class [com.jiuzhang.seckill.RedisDemoTest] from class [org.springframework
 .boot.test.context.SpringBootTestContextBootstrapper]
```

Tests passed: 1 of 1 test – 407 ms
Test Results 407 ms
RedisDemoTest 407 ms
pushSeckillInfoToRedisTe 407 ms
Tests passed: 1 (moments ago)

2. 我们在启动项目，访问 id = 19 的商品详情页

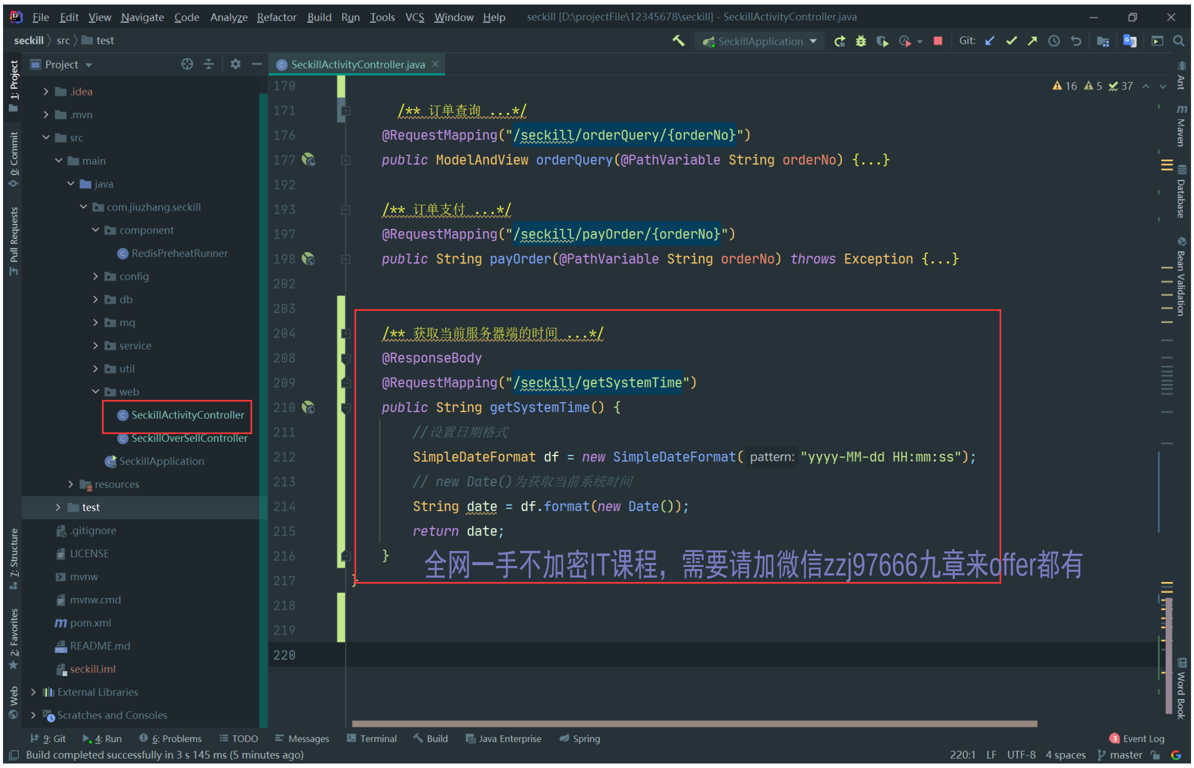
> 控制台打印命中缓存的日志，这样就可以不用查数据库了



```
2020-12-06 10:00:01.057  INFO 15044 --- [nio-8082-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherServlet
 'dispatcherServlet'
2020-12-06 10:00:01.058  INFO 15044 --- [nio-8082-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet
 'dispatcherServlet'
2020-12-06 10:00:01.066  INFO 15044 --- [nio-8082-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 8 ms
2020-12-06 10:00:01.099  INFO 15044 --- [nio-8082-exec-1] c.j.s.web.SeckillActivityController      : redis缓存数据:{"activityStatus":1,
 "availableStock":999994,"commodityId":1001,"endTime":1604565564000,"id":19,"lockStock":0,"name":"iPhone12 Pro 秒杀抢购活动","oldPrice":7888
 .00,"seckillPrice":5888.00,"startTime":1604565564000,"totalStock":1000000}
2020-12-06 10:00:01.100  INFO 15044 --- [nio-8082-exec-1] c.j.s.web.SeckillActivityController      :
 redis缓存数据:{"commodityDesc":"iphone12 pro 非常好","commodityName":"iphone12 pro","commodityPrice":9999,"id":1001}
```

4. 同步抢购倒计时

> 后端服务器提供一个接口，前端循环获取服务器时间



```
    /** 订单查询 ...*/
    @RequestMapping("/seckill/orderQuery/{orderNo}")
    public ModelAndView orderQuery(@PathVariable String orderNo) {...}

    /** 订单支付 ...*/
    @RequestMapping("/seckill/payOrder/{orderNo}")
    public String payOrder(@PathVariable String orderNo) throws Exception {...}


    /** 获取当前服务器端的时间 ...*/
    @ResponseBody
    @RequestMapping("/seckill/getSystemTime")
    public String getSystemTime() {
        //设置日期格式
        SimpleDateFormat df = new SimpleDateFormat( pattern: "yyyy-MM-dd HH:mm:ss");
        // new Date()为获取当前系统时间
        String date = df.format(new Date());
        return date;
    }
}
```

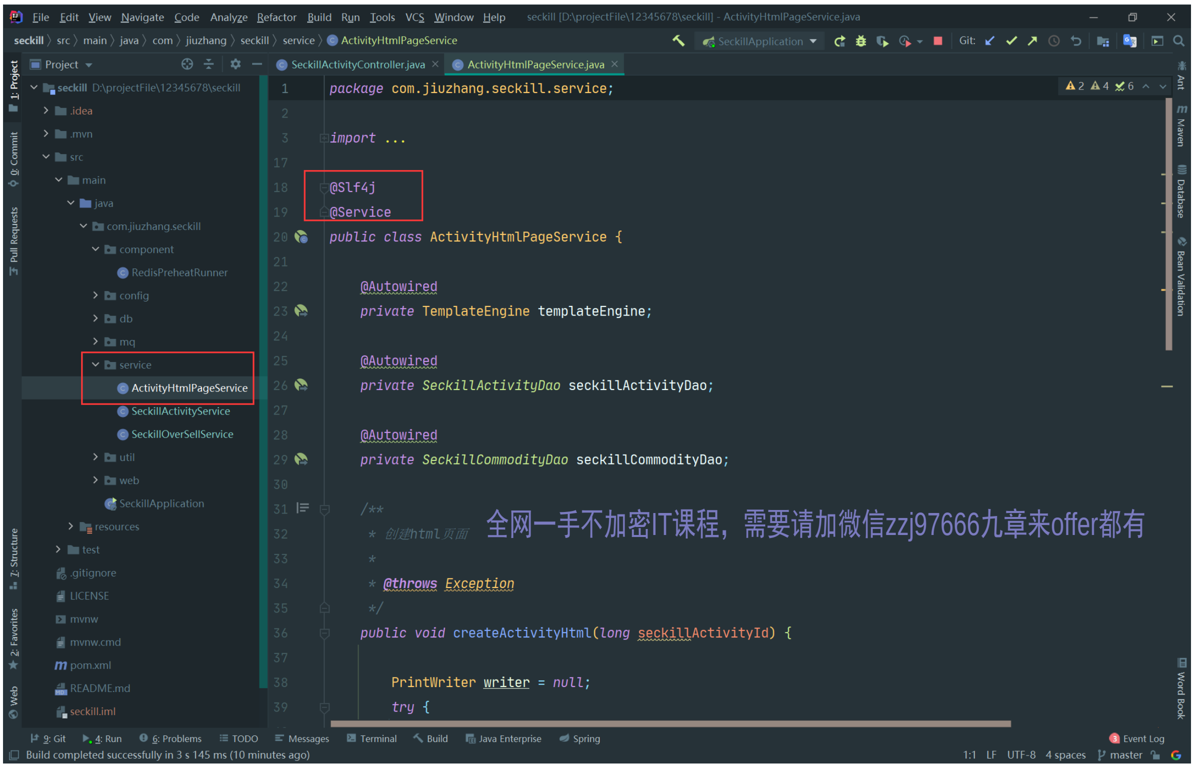
Build completed successfully in 3 s 145 ms (5 minutes ago)

```
/**
 * 获取当前服务器端的时间
 * @return
 */
@ResponseBody
@RequestMapping("/seckill/getSystemTime")
public String getSystemTime() {
    //设置日期格式
    SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    // new Date()为获取当前系统时间
    String date = df.format(new Date());
    return date;
}
```

5. 使用页面静态化技术减轻服务器负担

> ## 什么是静态化
>
> 静态化是指把动态生成的 HTML 页面变为静态内容保存，以后用户的请求到来，直接访问静态页面，不再经过服务的渲染。
>
> 而静态的HTML页面可以部署在 nginx 中，从而大大提高并发能力，减小 tomcat 压力。

1. **创建静态页面service开发**



```java
package com.jiuzhang.seckill.service;

import ...

@Slf4j
@Service
public class ActivityHtmlPageService {

    @Autowired
    private TemplateEngine templateEngine;

    @Autowired
    private SeckillActivityDao seckillActivityDao;

    @Autowired
    private SeckillCommodityDao seckillCommodityDao;

    /**
     * 创建html页面
     *
     * @throws Exception
     */
    public void createActivityHtml(long seckillActivityId) {

        PrintWriter writer = null;
        try {
```

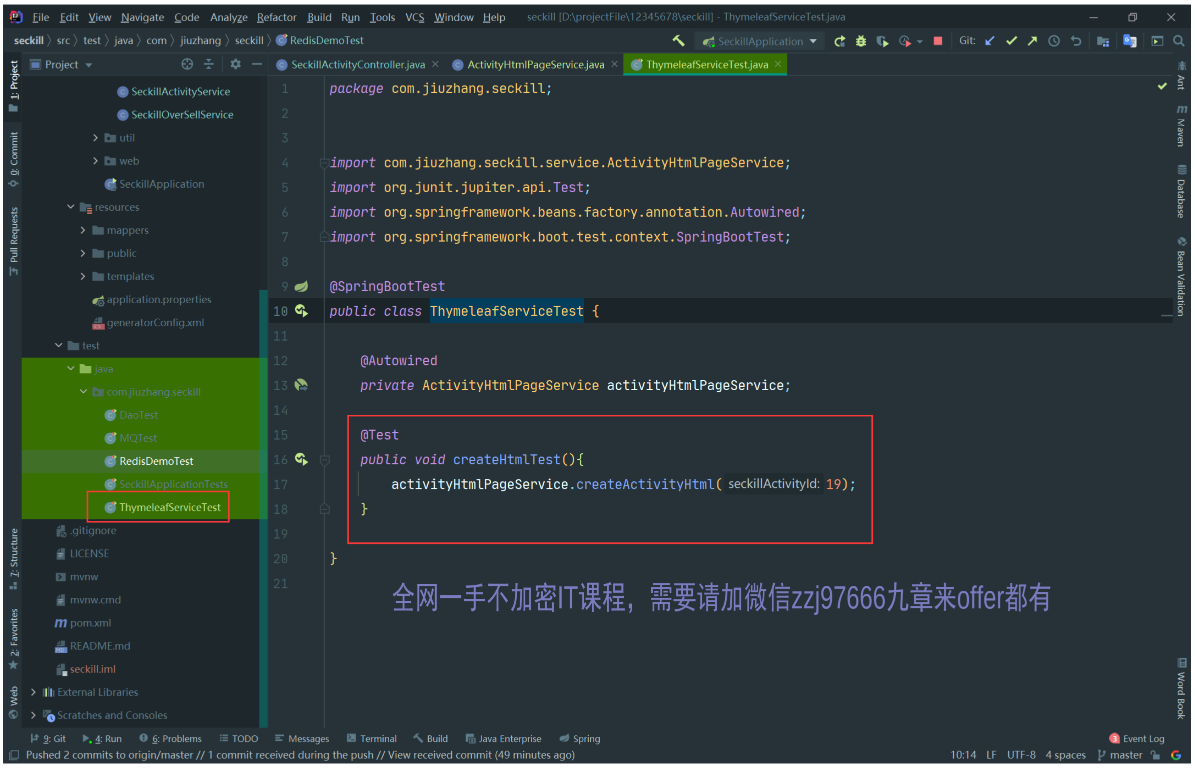
Build completed successfully in 3 s 145 ms (10 minutes ago)

```java
@Slf4j
@Service
public class ActivityHtmlPageService {

    @Autowired
    private TemplateEngine templateEngine;

    @Autowired
    private SeckillActivityDao seckillActivityDao;

    @Autowired
    private SeckillCommodityDao seckillCommodityDao;

    /**
     * 创建html页面
     *
     * @throws Exception
     */
    public void createActivityHtml(long seckillActivityId) {

        PrintWriter writer = null;
        try {
            SeckillActivity seckillActivity =
seckillActivityDao.querySeckillActivityById(seckillActivityId);
            SeckillCommodity seckillCommodity =
seckillCommodityDao.querySeckillCommodityById(seckillActivity.getCommodityId());
            // 获取页面数据
            Map<String, Object> resultMap = new HashMap<>();
            resultMap.put("seckillActivity", seckillActivity);
            resultMap.put("seckillCommodity", seckillCommodity);
            resultMap.put("seckillPrice", seckillActivity.getSeckillPrice());
            resultMap.put("oldPrice", seckillActivity.getOldPrice());
            resultMap.put("commodityId", seckillActivity.getCommodityId());
            resultMap.put("commodityName", seckillCommodity.getCommodityName());
```

```
      resultMap.put("commodityDesc", seckillCommodity.getCommodityDesc());

      // 创建thymeleaf上下文对象
      Context context = new Context();
      // 把数据放入上下文对象
      context.setVariables(resultMap);

      // 创建输出流
      File file = new File("src/main/resources/templates/" +
"seckill_item_" + seckillActivityId + ".html");
      writer = new PrintWriter(file);
      // 执行页面静态化方法
      templateEngine.process("seckill_item", context, writer);
    } catch (Exception e) {
      log.error(e.toString());
      log.error("页面静态化出错: " + seckillActivityId);
    } finally {
      if (writer != null) {
        writer.close();
      }
    }
  }
}
```

2. 页面测试

> 编写测试类 ThymeleafServiceTest ， 进行测试

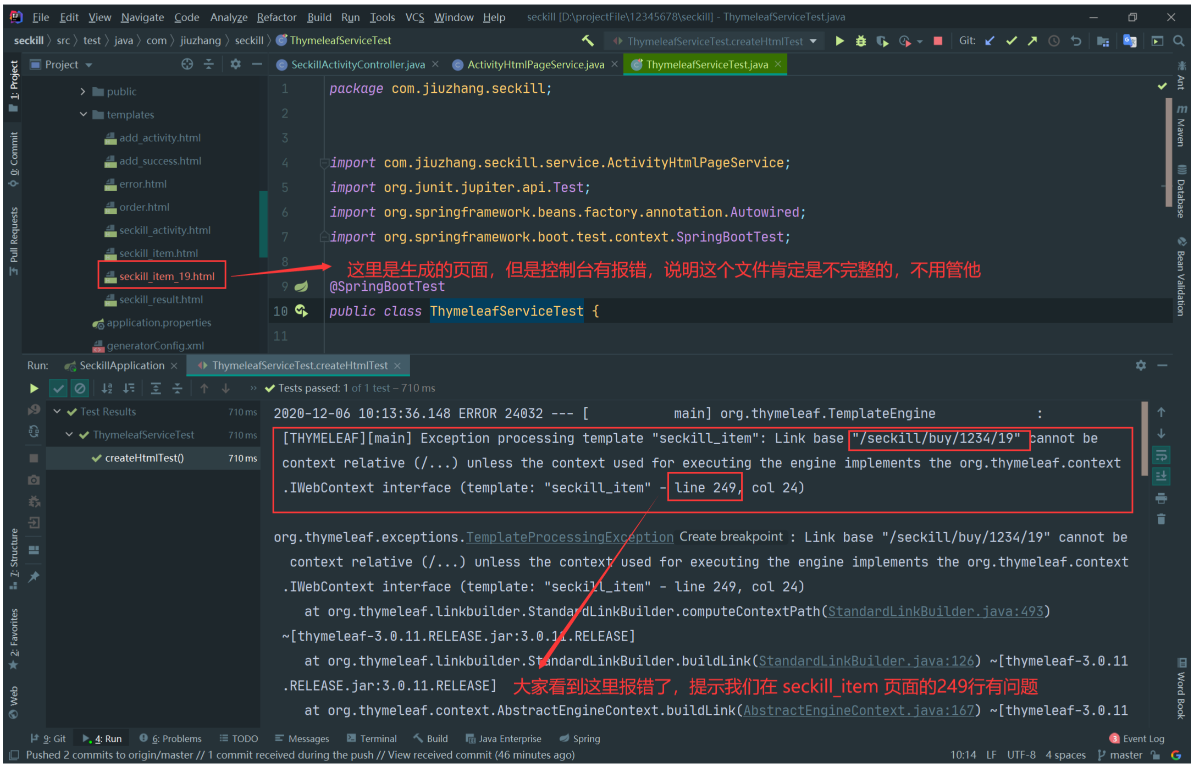


```
package com.jiuzhang.seckill;


import com.jiuzhang.seckill.service.ActivityHtmlPageService;
import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest
public class ThymeleafServiceTest {

    @Autowired
    private ActivityHtmlPageService activityHtmlPageService;

    @Test
    public void createHtmlTest(){
        activityHtmlPageService.createActivityHtml( seckillActivityId: 19);
    }

}
```

Pushed 2 commits to origin/master // 1 commit received during the push // View received commit (49 minutes ago)

```java
@Autowired
private ActivityHtmlPageService activityHtmlPageService;

@Test
public void createHtmlTest(){
    activityHtmlPageService.createActivityHtml(19);
}
```

> 运行刚写好的测试方法试试看


seckill [D:\projectFile\12345678\seckill] - ThymeleafServiceTest.java
seckill 〉 src 〉 test 〉 java 〉 com 〉 jiuzhang 〉 seckill 〉 ThymeleafServiceTest
ThymeleafServiceTest.createHtmlTest
SeckillActivityController.java
ActivityHtmlPageService.java
ThymeleafServiceTest.java
Project
public
templates
add_activity.html
add_success.html
error.html
order.html
seckill_activity.html
seckill_item.html
seckill_item_19.html
seckill_result.html
application.properties
generatorConfig.xml
Run: SeckillApplication
ThymeleafServiceTest.createHtmlTest
Tests passed: 1 of 1 test – 710 ms
Test Results 710 ms
ThymeleafServiceTest 710 ms
createHtmlTest() 710 ms

```java
package com.jiuzhang.seckill;


import com.jiuzhang.seckill.service.ActivityHtmlPageService;
import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest
public class ThymeleafServiceTest {

```

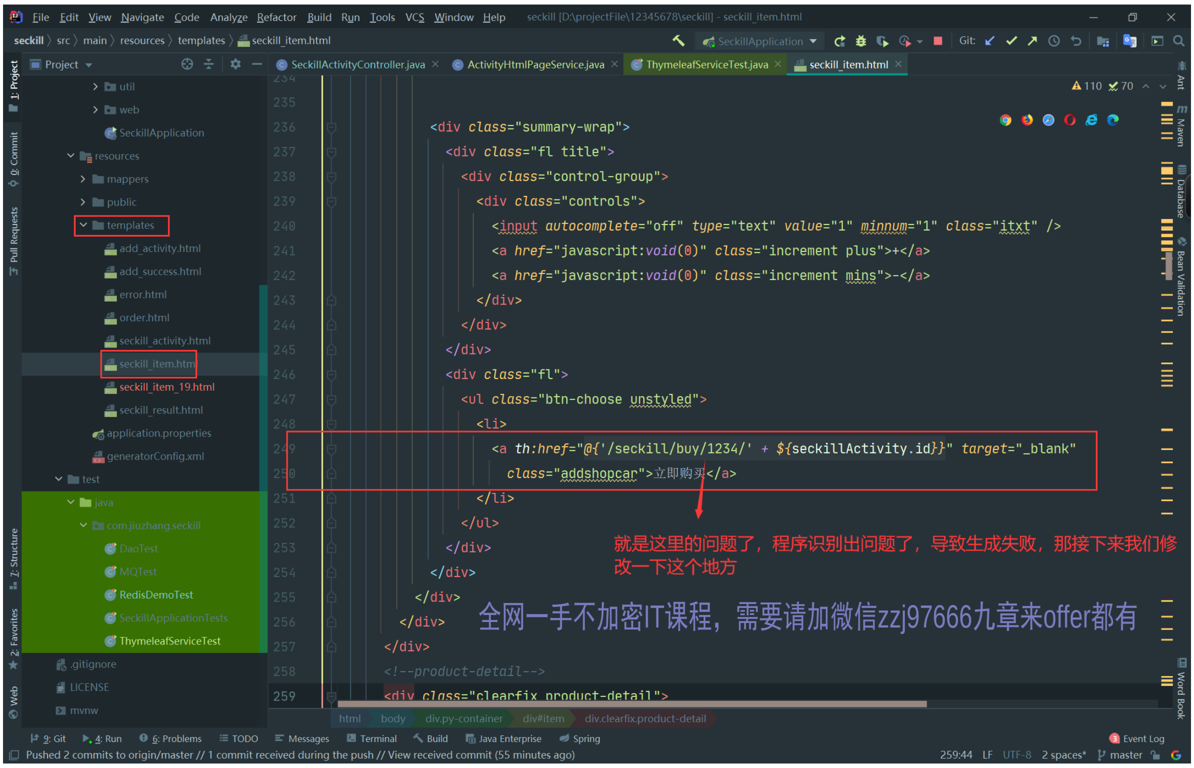
这里是生成的页面，但是控制台有报错，说明这个文件肯定是不完整的，不用管他

```
2020-12-06 10:13:36.148 ERROR 24032 --- [           main] org.thymeleaf.TemplateEngine             :
 [THYMELEAF][main] Exception processing template "seckill_item": Link base "/seckill/buy/1234/19" cannot be
 context relative (/...) unless the context used for executing the engine implements the org.thymeleaf.context
 .IWebContext interface (template: "seckill_item" - line 249, col 24)

org.thymeleaf.exceptions.TemplateProcessingException Create breakpoint : Link base "/seckill/buy/1234/19" cannot be
  context relative (/...) unless the context used for executing the engine implements the org.thymeleaf.context
 .IWebContext interface (template: "seckill_item" - line 249, col 24)
    at org.thymeleaf.linkbuilder.StandardLinkBuilder.computeContextPath(StandardLinkBuilder.java:493)
 ~[thymeleaf-3.0.11.RELEASE.jar:3.0.11.RELEASE]
    at org.thymeleaf.linkbuilder.StandardLinkBuilder.buildLink(StandardLinkBuilder.java:126) ~[thymeleaf-3.0.11
 .RELEASE.jar:3.0.11.RELEASE]
    at org.thymeleaf.context.AbstractEngineContext.buildLink(AbstractEngineContext.java:167) ~[thymeleaf-3.0.11
```

大家看到这里报错了，提示我们在 seckill_item 页面的249行有问题

9: Git
4: Run
6: Problems
TODO
Messages
Terminal
Build
Java Enterprise
Spring
Event Log
Pushed 2 commits to origin/master // 1 commit received during the push // View received commit (46 minutes ago)
10:14 LF UTF-8 4 spaces master

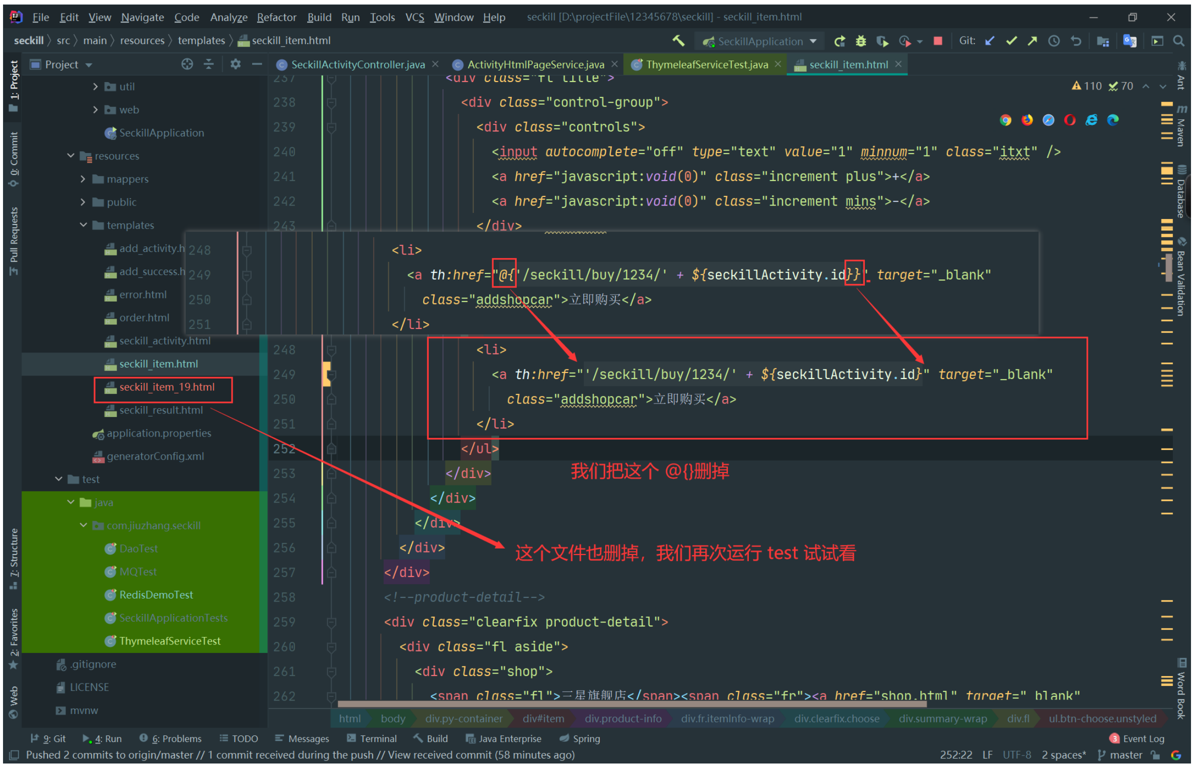
> 大家打开刚生成的页面会发现生成的文件生成到 249 行后面就没了，那我们看看这个 249 行是个什么东西



```html
235
236             <div class="summary-wrap">
237               <div class="fl title">
238                 <div class="control-group">
239                   <div class="controls">
240                     <input autocomplete="off" type="text" value="1" minnum="1" class="itxt" />
241                     <a href="javascript:void(0)" class="increment plus">+</a>
242                     <a href="javascript:void(0)" class="increment mins">-</a>
243                   </div>
244                 </div>
245               </div>
246               <div class="fl">
247                 <ul class="btn-choose unstyled">
248                   <li>
249                     <a th:href="@{'/seckill/buy/1234/' + ${seckillActivity.id}}" target="_blank"
250                       class="addshopcar">立即购买</a>
251                   </li>
252                 </ul>
253               </div>
254             </div>
255           </div>
256         </div>
257       </div>
258       <!--product-detail-->
259       <div class="clearfix product-detail">
```

就是这里的问题了，程序识别出问题了，导致生成失败，那接下来我们修改一下这个地方

html
body
div.py-container
div#item
div.clearfix.product-detail
9: Git
4: Run
6: Problems
TODO
Messages
Terminal
Build
Java Enterprise
Spring
Event Log
Pushed 2 commits to origin/master // 1 commit received during the push // View received commit (55 minutes ago)
259:44 LF UTF-8 2 spaces* master



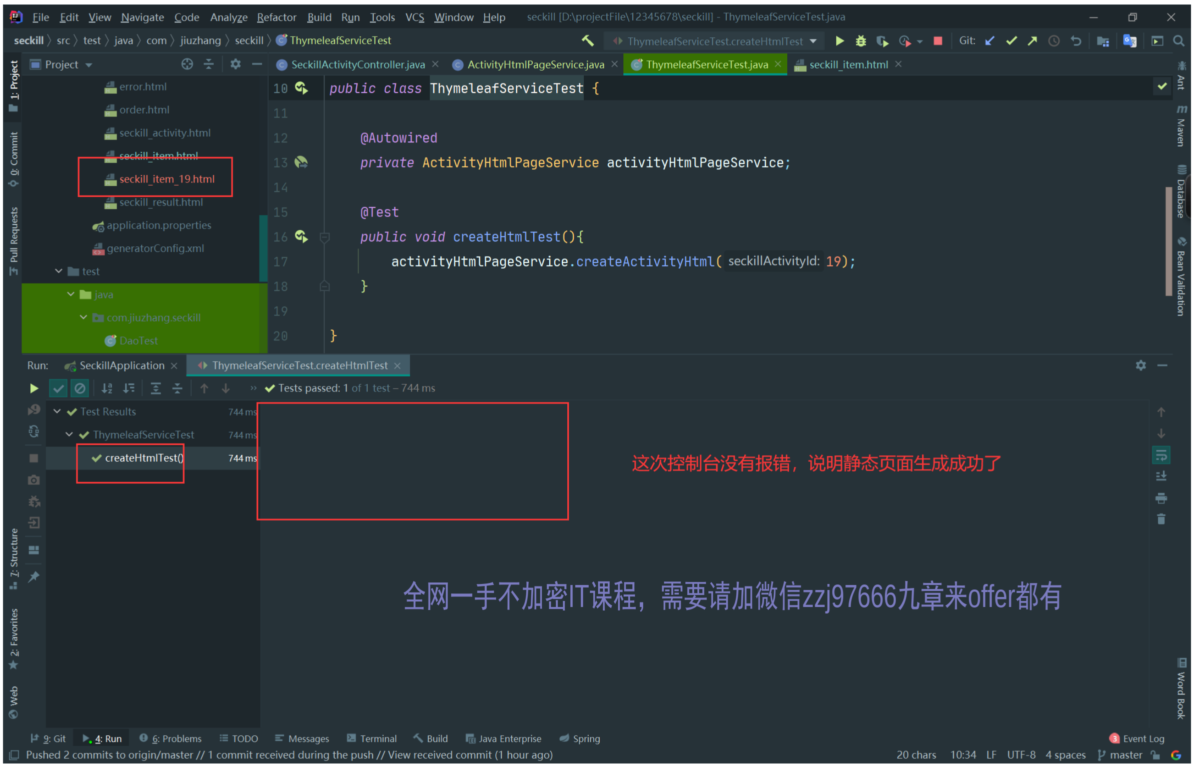
```
            <div class="fl title">
            <div class="control-group">
                <div class="controls">
                    <input autocomplete="off" type="text" value="1" minnum="1" class="itxt" />
                    <a href="javascript:void(0)" class="increment plus">+</a>
                    <a href="javascript:void(0)" class="increment mins">-</a>
                </div>
```

```
        <li>
            <a th:href="@{'/seckill/buy/1234/' + ${seckillActivity.id}}" target="_blank"
               class="addshopcar">立即购买</a>
        </li>
```

```
                    <li>
                        <a th:href="'/seckill/buy/1234/' + ${seckillActivity.id}" target="_blank"
                           class="addshopcar">立即购买</a>
                    </li>
                </ul>
            </div>
        </div>
    </div>
</div>
<!--product-detail-->
<div class="clearfix product-detail">
    <div class="fl aside">
        <div class="shop">
            <span class="fl">三星旗舰店</span><span class="fr"><a href="shop.html" target="_blank"
```

我们把这个 @{}删掉

这个文件也删掉，我们再次运行 test 试试看



```
public class ThymeleafServiceTest {

    @Autowired
    private ActivityHtmlPageService activityHtmlPageService;

    @Test
    public void createHtmlTest(){
        activityHtmlPageService.createActivityHtml( seckillActivityId: 19);
    }

}
```

```
✓ Tests passed: 1 of 1 test – 744 ms
✓ Test Results                744 ms
  ✓ ThymeleafServiceTest      744 ms
    ✓ createHtmlTest()        744 ms
```

这次控制台没有报错，说明静态页面生成成功了