

## 1. 使用 JUnit 单元测试：

```
/**
 * 交易和 MQ 队列及 email 发送单元测试
 */
@RunWith(SpringRunner.class)
@SpringBootTest
public class TransactionAndMQTest {

    @Autowired
    private FraudDetectionService fraudDetectionService;

    @Autowired
    private FraudDetectionRuleParser fraudDetectionRuleParser;

    @Autowired
    private RuleParser ruleParser;

    @Autowired
    private KafkaTemplate<String, String> kafkaTemplate;

    @Autowired
    private EmailSender emailSender;

    private static final String TOPIC = "test-topic";

    @Test
    public void shouldAnswerWithTrue() {
        fraudDetectionService.analyzeTransaction(null);
        Assertions.assertTrue(true);
    }

    /**
     * 规则解析测试用例
     */
    @Test
    public void ruleParserTest() {
        Transaction transaction = new Transaction();
        transaction.setAmount(new BigDecimal(50000));
        transaction.setCountry("US");
        transaction.setId(1);
    }
}
```

```

        transaction.setStatus("1");

        transaction.setTimestamp(new Date());

        transaction.setTransactionType("转账");

        boolean result = ruleParser.processTransaction(transaction);

        assertEquals(true,result);

    }

    /**
     * 发送 MQ 消息测试用例
     */
    @Test
    public void kafkaTest() {

        String message = "this is a test message, from developer to test-topic, 测试汉语编码";

        kafkaTemplate.send(TOPIC, message);

        System.out.println("Produced message: " + message);

        assertEquals(true,true);

    }

    /**
     * 发送邮件测试用例
     */
    @Test
    public void emailTest() {

        String message = "this is a test message, from developer to test-topic, 测试汉语编码";

        emailSender.sendSimpleEmail("this is subject","this is text!");

        assertEquals(true,true);

    }

}

```

```

/**
 * 交易 规则单元测试
 *
 * @className: RuleTest
 * @package: com.hsbc.springboot
 * @author: bruce
 * @date: 2025/1/26 13:04
 */
@RunWith(SpringRunner.class)
@SpringBootTest
public class RuleTest {

    @Autowired

```

```

private RuleController ruleController;

@Test
public void save(){
    Rule rule = new Rule();
    rule.setCreatedAt(new Date());
    rule.setRuleName("测试规则 3");
    rule.setRuleDescription("this is for testing3");
    rule.setUpdatedAt(new Date());
    rule.setState(true);
    ruleController.saveRule(rule);
}

@Test
public void getAll(){
    BaseResult<IPage<Rule>> pageRule = ruleController.getRulePage();
    List<Rule> lists = pageRule.getData().getRecords();
    lists.forEach(System.out::println);
}

@Test
public void del(){
    BaseResult baseResult = ruleController.deleteRule(4);
    System.out.println(baseResult.getCode());
}

@Test
public void update(){
    Rule rule = new Rule();
    rule.setId(5);
    rule.setRuleName("测试规则 1");
    BaseResult baseResult = ruleController.updateRule(rule);
    System.out.println(baseResult.getCode());
}
}

```

## 2. 使用 Postman 进行集成测试

模拟欺诈机制保证检测机制正常：

<http://localhost:8088/rule/save>

POST Content-Type application/json

```

{
    "ruleName": "amount>10000",
    "ruleDescription": "交易金额不能大于 10000 元",

```

```
    "state": 1
}
```

code 2000 保存成功，如下：

fraud-detection / http://localhost:8088/rule/save

POST http://localhost:8088/rule/save

Params Authorization Headers (8) Body Scripts Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "ruleName": "amount>10000",
3   "ruleDescription": "交易金额不能大于10000元",
4   "state": 1
5 }
```

Body Cookies Headers (5) Test Results 200 OK 1.09 s 198 B Save Response

{ } JSON Preview Visualize

```
1 {
2   "code": 2000,
3   "desc": null,
4   "data": 1
5 }
```

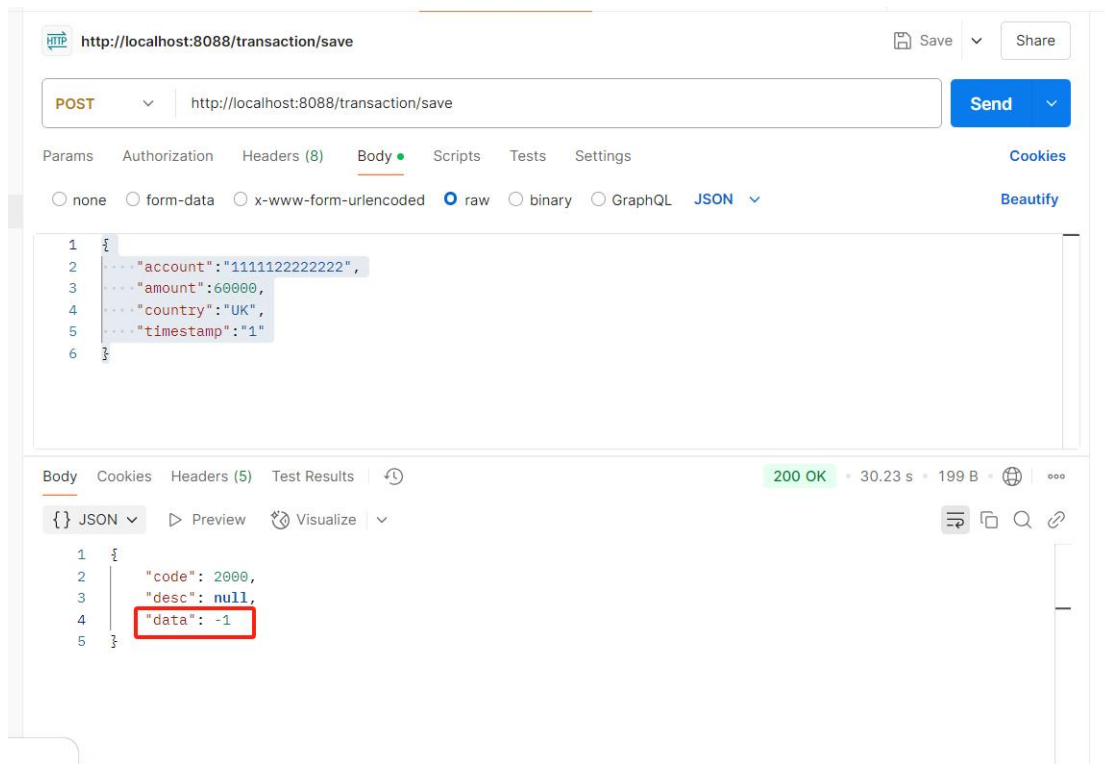
测试违规交易：

<http://localhost:8088/transaction/save>

POST Content-Type application/json

```
{
  "account": "111112222222",
  "amount": 60000,
  "country": "UK",
  "timestamp": "1"
}
```

code 2000 交易保存成功，但是 返回值 data=-1 表示交易异常，如下：



请求成功，交易异常，日志告警

```

WARN [http-nio-8088-exec-2] com.hsbc.springboot.service.impl.TransactionServiceImpl: - 交易: Transaction{id=9, amount=60000, account='111112222222', transactionType='null', timestamp=Thu Jan 01 08:00:00 CST 1970, status='null', country='UK'} 异常!
2025-01-27 10:19:17.829 warn [http-nio-8088-exec-4]
com.hsbc.springboot.service.impl.TransactionServiceImpl: - 交易: Transaction{id=7, amount=60000, account='111112222222', transactionType='null', timestamp=Thu Jan 01 08:00:00 CST 1970, status='null', country='UK'} 异常!

```

KafkaMQ 消息监听队列，收到异常信息，启动异步线程发送邮件：

```

[HSBC-AsyncThread-1] com.hsbc.springboot.core.KafkaProducerSender: - 交易: 9, 账号: 111112222222, 金额: 60000 存在违规行为: amount>10000, 请确认
[HSBC-AsyncThread-1] org.apache.kafka.clients.producer.ProducerConfig: - ProducerConfig values:
2025-01-27 10:19:17.805 INFO [HSBC-AsyncThread-1] com.hsbc.springboot.core.KafkaProducerSender: - 交易: 9, 账号: 111112222222, 金额: 60000 存在违规行为: amount>10000, 请确认!

```

告警邮件发送：



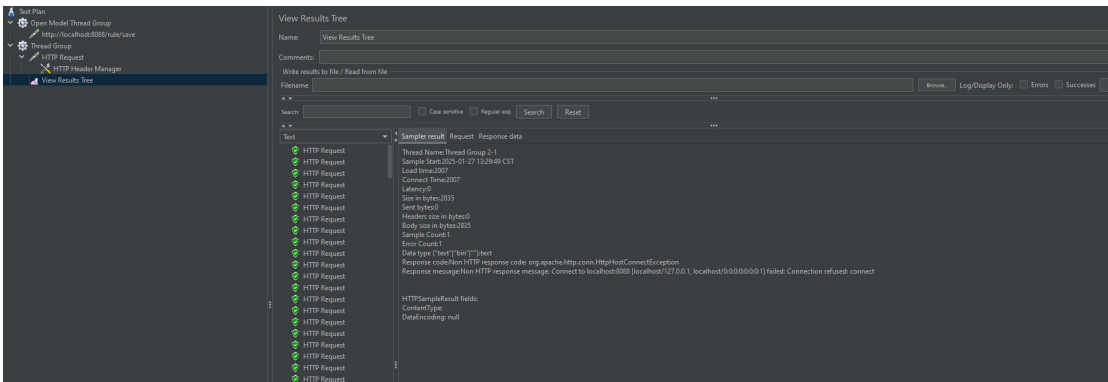
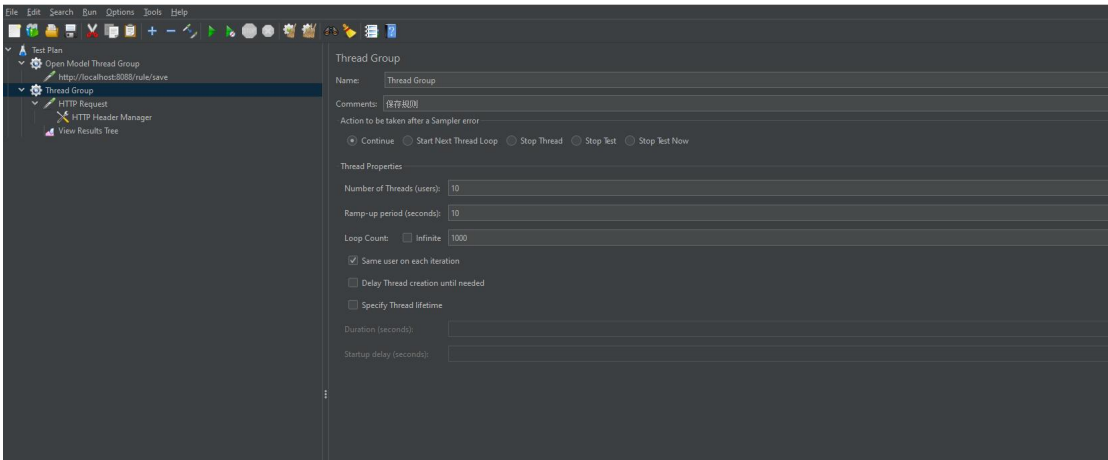
### 3. 使用 Jmeter 压力测试（本地简单压测）

欺诈规则操作接口：

<http://localhost:8088/rule/save>

测试违规交易：

<http://localhost:8088/transaction/save>



## 4. 阿里云 ACK 弹性测试

### 4.1 删除一个 pod 服务能够自动创建并启动

容器组 Pod

名称	命名空间	标签 Y	状态 Y	重启次数	Pod IP	节点	创建时间	CPU (核)	内存 (字节)	操作
hbsic-fraud-detection-8544cb879-gfzmj	default	app: hbsic-fraud-detection	Restarting	0	10.149.161.147	cn-hangzhou.10.149.161.127	2025-01-27 01:21:52			详情   编辑   删除   更多
hbsic-fraud-detection-8544cb879-hvntf	default	app: hbsic-fraud-detection	ContainerCreating	0		cn-hangzhou.10.149.161.127	2025-01-27 13:08:59			详情   编辑   删除   更多
hbsic-fraud-detection-8544cb879-l4wgj	default	app: hbsic-fraud-detection	Running	0	10.149.161.146	cn-hangzhou.10.149.161.127	2025-01-27 01:21:52	0/1001	223,242 MiB	详情   编辑   删除   更多

共有3条, 每页显示: 25 页

期间服务能够正常访问，符合高可用及流量弹性转移规则，且容器能够自动重启。

容器组Pod

使用IAM创建资源

命名空间default

名称

请输入

Q

名称	命名空间	标签	状态	重启次数	Pod IP	节点	创建时间	CPU (核) 内存 (字节)	操作
<div>hbsc-fraud-detection-67747846fd-d9g77</div> <div>hbsc1hbsbc3</div>	default	app:hbsc-fraud-d... pod-template-ha...	Running	0	10.149.161.147	cn-hangzhou.10.149.161.127 10.149.161.127	2025-01-27 13:24:32	0.501 131.523 Mi	<a href="#">详情</a>   <a href="#">编辑</a>   <a href="#">终端</a>   <a href="#">更多</a>
<div>hbsc-fraud-detection-67747846fd-v8ub8</div> <div>hbsc1hbsbc3</div>	default	app:hbsc-fraud-d... pod-template-ha...	Running	0	10.149.161.146	cn-hangzhou.10.149.161.127 10.149.161.127	2025-01-27 13:24:40	0.5 108.727 Mi	<a href="#">详情</a>   <a href="#">编辑</a>   <a href="#">终端</a>   <a href="#">更多</a>

☐ 批量删除

共有2条, 每页显示: 25 条

lessly discover, test, and integrate APIs and LLMs to accelerate agent development.

Start Building

NewImport

GET http://localhost: POST http://localhost: POST http://localhost: POST http://localhost: GET ACK-getRules

No environment

fraud-detection / ACK-getRules

SaveShare

GET

http://116.62.57.218:8088/rule/getRulePage

Send

ParamsAuthorizationHeaders (6)BodyScriptsTestsSettings

Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

BodyCookiesHeaders (4)Test Results

200 OK • 2.44 s • 501 B

Save Response

JSON

PreviewVisualize

```
1 {
2   "code": 2000,
3   "desc": "success",
4   "data": {
5     "records": [
6       {
7         "id": 16,
8         "ruleName": "amount>10000",
9         "ruleDescription": "交易金额不能大于10000元",
10        "createdAt": "2025-01-26T17:09:19.000+0000",
11        "updatedAt": "2025-01-26T17:09:19.000+0000",
12        "state": true
13      }
14    ],
15    "total": 1,
16    "size": 5,
17    "page": 1
18  }
19 }
```

ade new changes

update to inform others.

Postbot

CtrlAltP

Pod 容器能够自动重启。