**测试报告**

**单元测试：**

**运行方式：**

单元测试不需要启动整个服务，直接运行 mvn test 即可。

**测试用例：**

项目的单元测试全面覆盖了各个层次的功能和异常处理：

1. \*\*控制器层测试\*\* (`TransactionControllerTest`)

- API正常响应测试：验证各接口在正常情况下的返回值和状态码

- 例如：`createTransaction\_ShouldReturnCreatedStatus` 测试创建交易API返回201状态码

- 参数验证测试：检查API对无效输入的处理

- 例如：`createTransaction\_WithInvalidData\_ShouldReturnBadRequest` 测试当金额为负值时API返回400错误

- 交易不存在测试：验证请求不存在的交易时的错误处理

- 例如：`getTransaction\_WithInvalidId\_ShouldReturnNotFound` 测试获取不存在ID时返回404错误

- 模拟完整的API交互流程（创建→获取→更新→删除→验证删除）

2. \*\*服务层测试\*\* (`TransactionServiceTest`)

- 业务功能测试：验证所有业务方法是否正确执行

- 例如：`createTransaction\_ShouldReturnCreatedTransaction` 测试创建交易逻辑

- 异常处理测试：验证异常情况是否正确处理

- 例如：`updateTransaction\_WhenIdDoesNotExist\_ShouldThrowException` 测试更新不存在交易时抛出异常

- 分页功能测试：验证交易列表分页功能

- 例如：`getTransactions\_WithPagination\_ShouldReturnCorrectPage` 测试不同页码的交易数据正确性

- 完整生命周期测试：测试从创建到删除的完整流程

- 例如：`endToEndTransactionLifecycle\_ShouldWorkCorrectly` 验证交易完整CRUD流程

3. \*\*映射器测试\*\* (`TransactionMapperTest`)

- DTO转实体测试：验证DTO到实体对象是否正确转换

- 例如：`toEntity\_ShouldMapAllFields` 测试所有字段的正确映射

- 实体转DTO测试：验证实体到DTO对象是否正确转换

- 例如：`toDTO\_ShouldMapAllFields` 测试映射过程中是否没有数据丢失

- 空值处理测试：验证对象映射时是否正确处理空值

- 例如：`toEntity\_WithNullValues\_ShouldHandleGracefully` 测试将有空字段的DTO转为实体

4. \*\*仓库层测试\*\* (`TransactionRepositoryTest`)

- 数据存储测试：验证交易数据正确保存

- 例如：`save\_ShouldStoreTransaction` 测试交易保存后能被检索

- 数据更新测试：验证现有交易数据更新功能

- 例如：`save\_ShouldUpdateExistingTransaction` 测试更新已存在交易的字段

- 数据查询测试：验证按ID查询交易功能

- 例如：`findById\_ShouldReturnTransaction\_WhenExists` 测试能正确找到已存在交易

- 数据删除测试：验证交易删除功能

- 例如：`deleteById\_ShouldRemoveTransaction` 测试删除后交易不再可检索

- 分页查询测试：验证分页获取交易列表功能

- 例如：`findAll\_WithPagination\_ShouldReturnPagedTransactions` 测试分页参数的正确应用

5. \*\*模型层测试\*\* (`TransactionTest`)

- 实体属性测试：验证实体对象的getter和setter方法

- 例如：`testGettersAndSetters` 测试设置属性后能正确获取

- 实体相等性测试：验证equals和hashCode方法的正确实现

- 例如：`testEqualsAndHashCode` 测试ID相同的交易被视为相等

- 枚举值测试：验证状态和类型枚举的正确定义

- 例如：`testTransactionStatus` 测试状态枚举包含正确的值

**测试结果：**

[**INFO**] **Tests run: 11**, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.920 s -- in com.banking.transactionservice.controller.**TransactionControllerTest**

[**INFO**] Running com.banking.transactionservice.model.**TransactionTest**

[**INFO**] **Tests run: 4**, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.010 s -- in com.banking.transactionservice.model.**TransactionTest**

[**INFO**] Running com.banking.transactionservice.service.**TransactionMapperTest**

[**INFO**] **Tests run: 4**, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.002 s -- in com.banking.transactionservice.service.**TransactionMapperTest**

[**INFO**] Running com.banking.transactionservice.service.**TransactionServiceTest**

[**INFO**] **Tests run: 13**, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.083 s -- in com.banking.transactionservice.service.**TransactionServiceTest**

[**INFO**]

[**INFO**] Results:

[**INFO**]

[**INFO**] **Tests run: 42, Failures: 0, Errors: 0, Skipped: 0**

[**INFO**]

[**INFO**]

[**INFO**] **---** jacoco:0.8.11:report **(report)** @ transaction-service **---**

[**INFO**] Loading execution data file /Users/ketian/Desktop/transaction\_20250418/target/jacoco.exec

[**INFO**] Analyzed bundle 'transaction-service' with 13 classes

[**INFO**] **------------------------------------------------------------------------**

[**INFO**] **BUILD SUCCESS**

[**INFO**] **------------------------------------------------------------------------**

[**INFO**] Total time: 7.430 s

[**INFO**] Finished at: 2025-04-17T21:21:16+08:00

[**INFO**] **------------------------------------------------------------------------**

**压力测试：**

**运行方式：**

利用docker容器启动实例：

docker build -t transaction-service:latest

docker run -p 8080:8080 transaction-service:latest

执行测试脚本：

./run/stresstest.sh

**第一次测试35个用户：**

**测试结果：**

Simulation com.banking.transactionservice.simulation.TransactionSimulation completed in 59 seconds

Parsing log file(s)...

Parsing log file(s) done

Generating reports...

================================================================================

---- Global Information --------------------------------------------------------

> request count 920966 (OK=920966 KO=0 )

> min response time 0 (OK=0 KO=- )

> max response time 598 (OK=598 KO=- )

> mean response time 1 (OK=1 KO=- )

> std deviation 3 (OK=3 KO=- )

> response time 50th percentile 1 (OK=1 KO=- )

> response time 75th percentile 1 (OK=1 KO=- )

> response time 95th percentile 2 (OK=2 KO=- )

> response time 99th percentile 4 (OK=4 KO=- )

> mean requests/sec 15349.433 (OK=15349.433 KO=- )

---- Response Time Distribution ------------------------------------------------

> t < 800 ms 920966 (100%)

> 800 ms <= t < 1200 ms 0 ( 0%)

> t >= 1200 ms 0 ( 0%)

> failed 0 ( 0%)

================================================================================

Reports generated in 0s.

Please open the following file: file:///Users/ketian/Desktop/transaction\_20250418/target/gatling/transactionsimulation-20250417132852240/index.html

Global: mean of response time is less than 200.0 : true

Global: 95th percentile of response time is less than 100.0 : true

Global: percentage of successful events is greater than or equal to 99.0 : true

Global: percentage of failed events is less than 1.0 : true

[**INFO**] **------------------------------------------------------------------------**

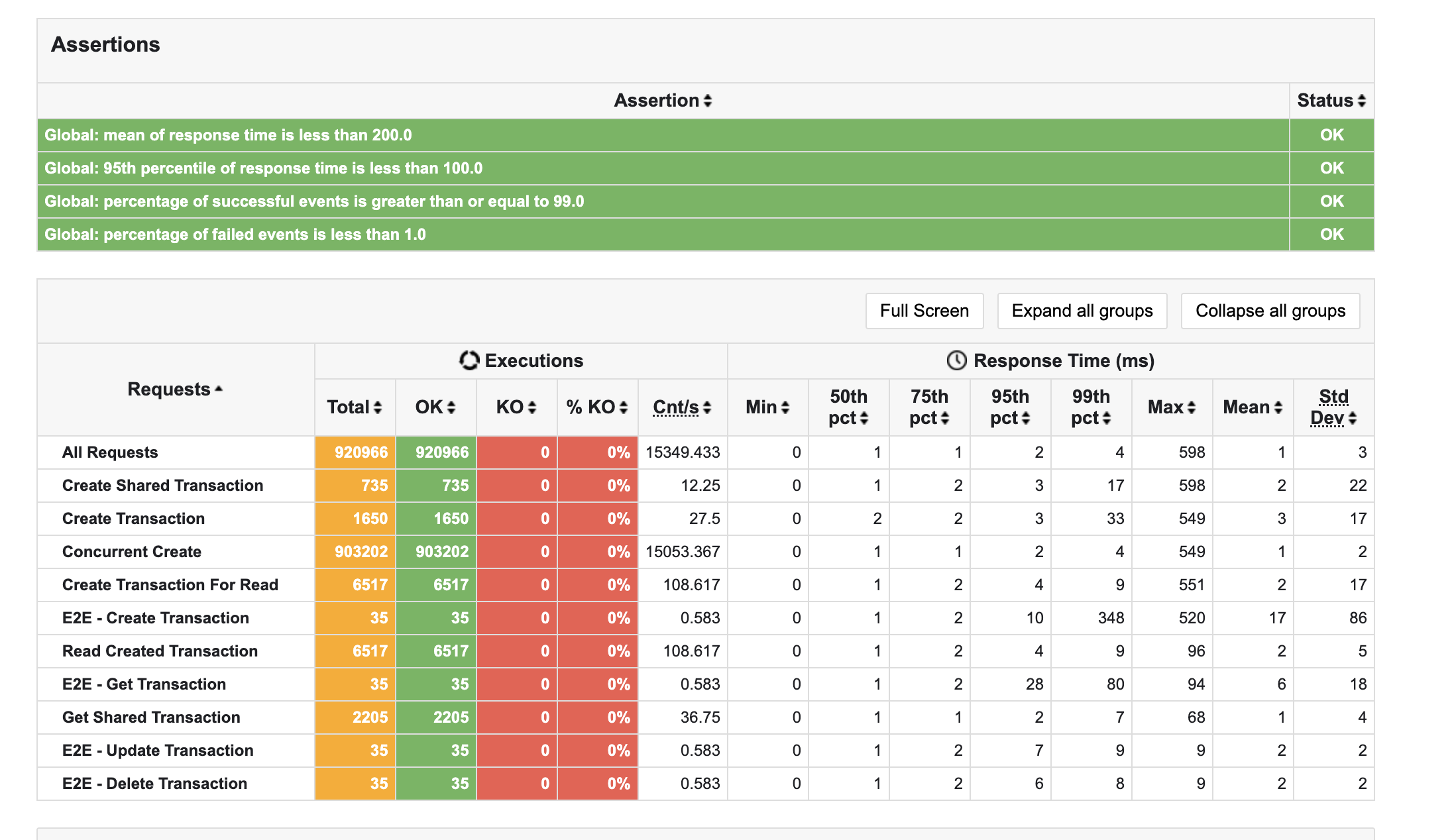
[**INFO**] **BUILD SUCCESS**

[**INFO**] **------------------------------------------------------------------------**

[**INFO**] Total time: 01:16 min

[**INFO**] Finished at: 2025-04-17T21:30:05+08:00

[**INFO**] **------------------------------------------------------------------------**



**第二次测试600个用户：**

**测试结果：**

Simulation com.banking.transactionservice.simulation.TransactionSimulation completed in 60 seconds

Parsing log file(s)...

Parsing log file(s) done

Generating reports...

================================================================================

---- Global Information --------------------------------------------------------

> request count 1056183 (OK=1056183 KO=0 )

> min response time 0 (OK=0 KO=- )

> max response time 602 (OK=602 KO=- )

> mean response time 19 (OK=19 KO=- )

> std deviation 17 (OK=17 KO=- )

> response time 50th percentile 16 (OK=16 KO=- )

> response time 75th percentile 27 (OK=27 KO=- )

> response time 95th percentile 44 (OK=44 KO=- )

> response time 99th percentile 63 (OK=63 KO=- )

> mean requests/sec 17603.05 (OK=17603.05 KO=- )

---- Response Time Distribution ------------------------------------------------

> t < 800 ms 1056183 (100%)

> 800 ms <= t < 1200 ms 0 ( 0%)

> t >= 1200 ms 0 ( 0%)

> failed 0 ( 0%)

================================================================================

Reports generated in 0s.

Please open the following file: file:///Users/ketian/Desktop/transaction\_20250418/target/gatling/transactionsimulation-20250417133211636/index.html

Global: mean of response time is less than 200.0 : true

Global: 95th percentile of response time is less than 100.0 : true

Global: percentage of successful events is greater than or equal to 99.0 : true

Global: percentage of failed events is less than 1.0 : true

[**INFO**] **------------------------------------------------------------------------**

[**INFO**] **BUILD SUCCESS**

[**INFO**] **------------------------------------------------------------------------**

[**INFO**] Total time: 01:11 min

[**INFO**] Finished at: 2025-04-17T21:33:17+08:00

[**INFO**] **------------------------------------------------------------------------**

