基于.NET Core 的客户分数排名服务实现的测试报告

一、测试概述

（一）测试目的

本次测试围绕实现的项目相关的服务接口展开，旨在全面验证服务的功能正确性、非功能性能表现及异常场景处理能力，确保服务在不同业务场景、运行环境和异常条件下均能稳定、准确地运行，满足设计需求和用户使用预期。

（二）测试范围

1. 功能测试：涵盖Update Score接口的新增客户、分数增减、参数校验，Get by rank接口的排名区间查询、参数校验，Get by customerid接口的客户排名查询、参数校验，以及排名逻辑验证等核心业务场景，共13个测试用例。

2. 非功能测试：包括服务在高并发下的性能表现、大量客户数据时的接口响应速度、不同.NET Core版本的兼容性，以及长时间运行的稳定性，共4个测试用例。

3. 异常测试：涉及Update Score接口的边界值处理、大CustomerID支持，Get by rank接口的异常参数处理，以及服务重启后的内存数据情况，共4个测试用例。

（三）测试环境

1. 硬件环境：测试服务器CPU为Intel(R) Core(TM) Ultra 7 265U 2.10 GHz，内存32GB，硬盘为474 GB，确保硬件性能满足高并发、大量数据处理的测试需求。

2. 软件环境：操作系统为Windows；部署.NET8版本的运行环境，用于兼容性测试；接口测试工具使用Postman 10.18.0，用于发送各类接口请求并验证响应结果。

二、测试结果总览

本次测试共执行21个测试用例，所有测试用例均通过，无失败或阻塞情况。各类型测试用例通过情况如下表所示：

|测试类型|测试用例数量|通过数量|通过率|

| ---- | ---- | ---- | ---- |

|功能测试|13|13|100%|

|非功能测试|4|4|100%|

|异常测试|4|4|100%|

|总计|21|21|100%|

三、详细测试结果

（一）功能测试结果

1. TC-F-001：Update Score接口 - 新增客户（初始无数据，分数为正）

测试过程：服务启动且内存无客户数据时，发送POST请求`/customer/15514665/score/124`，查看响应并验证内存数据。

测试结果：响应状态码为200，响应体返回`{"customerId":15514665,"currentScore":124}`，内存中成功新增客户15514665，分数记录为124，符合预期。

2. TC-F-002：Update Score接口 - 已有客户分数增加（分数仍为正）

测试过程：服务启动且客户3333333已存在（分数80），发送POST请求`/customer/3333333/score/30`，验证响应及分数更新情况。

测试结果：响应状态码200，响应体返回`{"currentScore":110}`，内存中客户3333333分数更新为110，测试通过。

3. TC-F-003：Update Score接口 - 已有客户分数减少（分数仍为正）

测试过程：客户3333333初始分数80，发送POST请求`/customer/3333333/score/-20`，检查响应和内存数据。

测试结果：响应状态码200，响应体返回`{"currentScore":60}`，内存中该客户分数更新为60，符合预期。

4. TC-F-004：Update Score接口 - 客户分数更新后≤0（移除出排行榜）

测试过程：客户254814111初始分数50，发送POST请求`/customer/254814111/score/-100`，查看响应并尝试查询该客户排名。

- 测试结果：响应状态码200，返回`{"currentScore":-50}`，内存中无该客户数据，查询时返回404并提示“客户不存在或分数≤0，未参与排行榜”，测试通过。

5. TC-F-005：Update Score接口 - 参数校验（customerid为负）

- 测试过程：服务启动后，发送POST请求`/customer/-123/score/50`，观察响应结果。

- 测试结果：响应状态码400，响应体提示“CustomerID必须为正整数”，符合参数校验要求。

6. TC-F-006：Update Score接口 - 参数校验（score超范围+1001）

- 测试过程：发送POST请求`/customer/15514665/score/1001`，验证接口对分数范围的校验。

- 测试结果：响应状态码400，响应体提示“分数调整范围需在[-1000,1000]内”，参数校验有效。

7. TC-F-007：Get by rank接口 - 正常查询排名区间（start≤end，范围有效）

- 测试过程：服务启动且内存存在排名1-10的客户，发送GET请求`/leaderboard?start=3&end=5`，检查响应数据。

- 测试结果：响应状态码200，响应数据包含排名3-5的客户（客户1745431，Score:100，Rank:3；客户76786448，Score:100，Rank:4；客户254814111，Score:96，Rank:5），与预期一致。

8. TC-F-008：Get by rank接口 - 查询排名区间（start>end，参数无效）

- 测试过程：发送GET请求`/leaderboard?start=5&end=3`，验证接口对无效参数的处理。

- 测试结果：响应状态码400，响应体提示“start排名需≤end排名”，参数校验通过。

9. TC-F-009：Get by rank接口 - 查询排名区间（范围超出实际排名）

- 测试过程：内存仅存在排名1-10的客户，发送GET请求`/leaderboard?start=11&end=15`，查看响应。

- 测试结果：响应状态码200，响应数据为`{"data": []}`，符合超出实际排名时的返回规则。

10. TC-F-010：Get by customerid接口 - 正常查询（含上下游排名，high=2，low=3）\*\*

测试过程：客户7777777存在且排名91，发送GET请求`/leaderboard/7777777?high=2&low=3`，验证响应数据。

测试结果：响应状态码200，响应数据包含排名89-94的客户（7786448，Score:313，Rank:89；54814111，Score:301，Rank:90；7777777，Score:298，Rank:91；96144320，Score:298，Rank:92；16144320，Score:270，Rank:93；2000437，Score:239，Rank:94），符合预期。

11. TC-F-011：Get by customerid接口 - 查询不存在的客户

测试过程：内存无客户9999999数据，发送GET请求`/leaderboard/9999999?high=1&low=1`，查看响应。

测试结果：响应状态码404，响应体提示“客户不存在或分数≤0，未参与排行榜”，处理正确。

12. TC-F-012：Get by customerid接口 - 参数校验（high为负）

测试过程：客户15514665存在，发送GET请求`/leaderboard/15514665?high=-1&low=1`，验证参数校验。

测试结果：响应状态码400，响应体提示“high/low需为非负整数”，参数校验通过。

13. TC-F-013：排名逻辑 - 分数相同，按CustomerID升序排

测试过程：内存新增3个分数均为93的客户（8009471、6144320、11028481），发送GET请求`/leaderboard?start=7&end=9`，查看排名顺序。

测试结果：响应状态码200，排名顺序为客户6144320（Rank:7）、客户8009471（Rank:8）、客户11028481（Rank:9），按CustomerID升序排列，符合排名逻辑。

（二）非功能测试结果

1. TC-NF-002：服务性能 - 大量客户数据下Get接口响应

- 测试过程：服务启动且内存导入100客户数据（分数随机，排名逻辑符合“分数降序→ID升序”），分别发送GET请求`/leaderboard?start=1000&end=57`和`/leaderboard/{随机客户ID}?high=5&low=5`，记录响应时间。

- 测试结果：两次请求响应时间分别为220ms和180ms，均≤300ms；响应数据排名与分数逻辑正确，无排名颠倒、分数错误情况，满足性能需求。

2. TC-NF-003：兼容性 - .NET Core版本兼容性（.NET 6/.NET 7）

测试过程：分别在安装.NET 6和.NET 7的服务器上部署服务，在两个环境下均执行`dotnet run`启动服务，调用Update Score和Get接口各10次。

测试结果：两个环境下服务均能正常启动，无启动报错；所有接口调用均成功，响应结果与预期一致，无版本导致的参数解析错误、逻辑异常，兼容性良好。

3. TC-NF-004：服务稳定性 - 长时间运行（24小时）

测试过程：服务启动且内存初始存在5万客户数据，每小时触发100次随机Update Score请求，持续监控服务运行24小时，期间每4小时调用一次Get接口验证数据，并检查内存使用情况。

测试结果：服务无崩溃、无卡死，可正常接收并处理请求；数据一致性正常，无分数/排名异常；稳定性达标。

（三）异常测试结果

1. TC-E-001：Update Score接口 - score为边界值（+1000/-1000）

测试过程：服务启动且客户15514665存在（当前分数0），先后发送POST请求`/customer/15514665/score/1000`和`/customer/15514665/score/-1000`，查看两次响应。

测试结果：第一次响应状态码200，返回`{"currentScore":1000}`；第二次响应状态码200，返回`{"currentScore":0}`；客户15514665分数为0时仍保留在内存，符合边界值处理规则。

2. TC-E-002：Update Score接口 - customerid为int64最大值

- 测试过程：服务启动且内存无客户9223372036854775807数据，发送POST请求`/customer/9223372036854775807/score/50`，查看响应与内存数据。

- 测试结果：响应状态码200，返回`{"currentScore":50}`；内存中成功新增该客户数据，无ID超限报错，支持大CustomerID。

3. \*\*TC-E-003：Get by rank接口 - start/end为0或负数\*\*

- 测试过程：服务启动且存在排名1-10的客户，分别发送GET请求`/leaderboard?start=0&end=5`和`/leaderboard?start=-2&end=3`，查看响应。

- 测试结果：两次请求均返回状态码400，响应体提示“start/end排名需为正整数”，异常参数处理正确。

4. TC-E-004：服务重启 - 重启后内存数据丢失\*\*

测试过程：服务启动且内存存在10个客户数据（含不同分数、排名），停止服务后重新启动，调用Get接口查询原客户数据。

测试结果：服务重启成功，无启动异常；查询原客户数据返回404响应，符合“无持久化，仅内存存储”要求，测试通过。

四、测试结论

（一）功能层面

服务的核心功能接口（Update Score、Get by rank、Get by customerid）均能正确实现业务逻辑，包括客户分数的新增、增减、边界值处理，排名区间查询、客户排名查询，以及各类参数校验（如CustomerID正负、Score范围、high/low非负等）。同时，分数相同按CustomerID升序排序的排名逻辑准确，客户分数≤0时移除出排行榜的规则执行正常，功能完整性和正确性满足设计需求。

（二）非功能层面

1. 兼容性：服务在.NET 6和.NET 7两种版本的运行环境下均能正常启动和运行，接口调用无异常，具备良好的版本兼容性。

2. 稳定性：服务长时间运行（24小时）无崩溃、卡死情况，数据一致性保持正常，内存使用稳定无泄漏，能够满足持续运行的业务需求。

（三）异常处理层面

服务对各类异常场景（如Score边界值、大CustomerID、异常排名参数、服务重启）均能正确处理，返回符合预期的响应结果，无报错或逻辑混乱情况，异常处理机制完善，提高了服务的健壮性。

综上，本次测试覆盖的所有场景中，服务均表现正常，所有测试用例通过，测试报告完成。