# 引言

## 测试说明

本文档是软件“北洋有约”校内招募组队平台MVC架构中所有接口的测试说明，省略了可视化界面的操作测试模块。

本测试文档将接口分为Service层接口和Controller层接口两部分。

* Controller层的接口主要进行前后端之间的数据请求、传输的功能：及时接收前端发起的请求和携带的参数，并调用service去处理数据，最后通过Controller层对数据进行响应。考虑到Controller层的http特点，我们使用postman测试Controller层的接口。
* Service层中的函数方法都是具体的处理业务的代码，业务处理完成后通过interface将函数接口暴露给上层的Controller层进行调用，本项目使用JUnit测试Controller层的接口。Service层接口和Controller层接口功能基本一致，因此本文档不再重复展示接口功能，只展示JUnit的测试方法和结果，具体测试代码见backend\src\test文件夹。

## 软件基本信息

表1‑1 软件开发基本信息

|  |  |  |  |
| --- | --- | --- | --- |
| **被测软件名称** | **开发语言** | **开发环境** | **运行环境** |
| “北洋有约”校内招募组队平台 | Java | Intellij idea | centOS 8 |

## 测试接口概述

表1‑2 测试接口说明及与Service层模块的对应关系

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **标识符** | **功能名称** | **用途与说明** | **对应模块**  **(Service层)** |
| 1 | COT\_GXCX | 个人信息查询 | 通过用户id进行查询，可以返回用户详细信息，包括邮箱地址、姓名、电话号码、性别、年级、注册时间等。 | User  Service |
| 2 | COT\_GXZC | 个人信息注册 | 通过填写用户基本信息，可以将用户详细信息上传至数据库，包括邮箱地址、姓名、电话号码、性别、年级、注册时间等。 |
| 3 | COT\_GXXG | 个人信息修改 | 通过修改用户基本信息，可以修改数据库中的用户详细信息，包括邮箱地址、姓名、电话号码、性别、年级、注册时间等。 |
| 4 | COT\_MMXG | 密码修改 | 通过修改用户密码，可以修改数据库中的密码信息。 |
| 5 | COT\_TCCX | 团队成员查询 | 通过团队id进行查询，可以返回团队所有成员的详细信息。 | Recruit  Service |
| 6 | COT\_YTCX | 用户团队查询 | 通过用户id进行查询，可以返回该用户参与的所有团队信息，包括团队名称、团队成员人数、团队创建时间、团队所属板块等。 |
| 7 | COT\_BBZL | 标签板块总览 | 通过用户id进行查询，可以返回该用户参与的所有团队信息，包括团队名称、团队成员人数、团队创建时间、团队所属板块等。 |
| 8 | COT\_TZCJ | 团队招募创建 | 通过填写团队基本信息，可以将团队详细信息上传至数据库，包括成员id、团队名称、任务描述、成员人数、结束时间、板块标签等。 |
| 9 | COT\_TZJR | 团队申请加入 | 通过添加团队加入申请，可以成为该团队的成员之一。 |
| 10 | COT\_GTCX | 个人团队查询 | 通过用户id进行查询，可以返回该用户参加的所有团队信息，包括团队名称、团队成员人数、团队创建时间、团队所属板块等。 | Team  Service |
| 11 | COT\_TXCX | 团队具体招募信息查询 | 通过用户id和招募id进行查询，可以返回该团队的具体招募信息，包括招募名称、招募创建人、招募创建时间、招募所属板块等。 |
| 12 | COT\_ZZCX | 主页招募信息列表查询 | 通过postman查询，可以返回主页的基本招募信息列表，包括招募名称、招募人数需求、已申请人数、招募所属板块等。 |
| 13 | COT\_ZRGX | 招募热度值更新 | 通过用户id和招募id进行查询，可以更新主页的招募热度值。 |
| 14 | COT\_GMCX | 个人消息列表查询 | 通过用户id，可以返回个人收到的所有消息。 | Message Service |
| 15 | COT\_GMGX | 个人消息状态更新 | 通过消息id或用户id，可以更新消息状态为已读。 |
| 16 | COT\_GRCX | 个人日志查询 | 通过用户id进行查询，可以返回用户详细信息，包括邮箱地址、姓名、电话号码、性别、年级、注册时间等。 | Histroy  Service |

# 测试方法与内容

## Postman接口测试

### 个人信息查询

表2‑1 COT\_GXCX\_1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | 个人信息查询接口测试 | | | **接口URL** | /user/api/getUserInfo | |
| **用例综述** | | 通过用户id进行查询，可以返回用户详细信息，包括邮箱地址、姓名、电话号码、性别、年级、注册时间等。 | | | | | |
| **测试步骤** | | | | | | | |
| **序号** | **输入及操作** | | | **期望结果与评估标准** | | **实测结果** | **通过**  **与否** |
| 1 | 输入并发送username (String)。 | | | 通过用户id，返回查询结果包括正确的对应详细用户信息。 | | 返回结果信息完整且正确。 | 通过 |
| **执行结果** | | | 通过 | | **测试时间** | 2022/5/10 | |
| **结果截图** | | |  | | | | |

### 个人信息注册

表2‑2 COT\_GXZC\_1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | 个人信息注册接口测试 | | | **接口URL** | /user/api/register | |
| **用例综述** | | 通过填写用户基本信息，可以将用户详细信息上传至数据库，包括邮箱地址、姓名、电话号码、性别、年级、注册时间等。 | | | | | |
| **测试步骤** | | | | | | | |
| **序号** | **输入及操作** | | | **期望结果与评估标准** | | **实测结果** | **通过**  **与否** |
| 1 | 输入并发送用户id和基本信息（见结果截图左一）。 | | | 通过填写用户详细信息，返回注册结果。 | | 注册成功。 | 通过 |
| **执行结果** | | | 通过 | | **测试时间** | 2022/5/10 | |
| **结果截图** | | |  | | | | |

### 个人信息修改

表2‑3 COT\_GXXG\_1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | 个人信息修改接口测试 | | | **接口URL** | /user/api/editUserInfo | |
| **用例综述** | | 通过修改用户基本信息，可以修改数据库中的用户详细信息，包括邮箱地址、姓名、电话号码、性别、年级、注册时间等。 | | | | | |
| **测试步骤** | | | | | | | |
| **序号** | **输入及操作** | | | **期望结果与评估标准** | | **实测结果** | **通过**  **与否** |
| 1 | 输入并发送用户id和修改后的基本信息（见结果截图左一）。 | | | 通过修改用户详细信息，返回修改结果。 | | 修改成功。 | 通过 |
| **执行结果** | | | 通过 | | **测试时间** | 2022/5/10 | |
| **结果截图** | | |  | | | | |

### 密码修改

表2‑4 COT\_MMXG\_1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | 密码修改接口测试 | | | **接口URL** | /user/api/editUserPwd | |
| **用例综述** | | 通过修改用户密码，可以修改数据库中的密码信息。 | | | | | |
| **测试步骤** | | | | | | | |
| **序号** | **输入及操作** | | | **期望结果与评估标准** | | **实测结果** | **通过**  **与否** |
| 1 | 输入并发送用户id和新旧密码。 | | | 通过修改密码，返回修改结果。 | | 修改成功。 | 通过 |
| **执行结果** | | | 通过 | | **测试时间** | 2022/5/10 | |
| **结果截图** | | |  | | | | |

### 团队成员查询

表2‑5 COT\_TCCX\_1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | 团队成员查询接口测试 | | | **接口URL** | /team/api/getTeamInfo | |
| **用例综述** | | 通过团队id进行查询，可以返回团队所有成员的详细信息。 | | | | | |
| **测试步骤** | | | | | | | |
| **序号** | **输入及操作** | | | **期望结果与评估标准** | | **实测结果** | **通过**  **与否** |
| 1 | 输入并发送Tid (String)。 | | | 通过用户id，返回团队成员正确的对应详细用户信息。 | | 返回结果信息完整且正确。 | 通过 |
| **执行结果** | | | 通过 | | **测试时间** | 2022/5/10 | |
| **结果截图** | | |  | | | | |

### 用户团队查询

表2‑6 COT\_YTCX\_1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | 用户团队查询接口测试 | | | **接口URL** | /team/api/getUserTeamInfo | |
| **用例综述** | | 通过用户id进行查询，可以返回该用户参与的所有团队信息，包括团队名称、团队成员人数、团队创建时间、团队所属板块等。 | | | | | |
| **测试步骤** | | | | | | | |
| **序号** | **输入及操作** | | | **期望结果与评估标准** | | **实测结果** | **通过**  **与否** |
| 1 | 输入并发送username (String)。 | | | 通过用户id，返回团队详细信息。 | | 返回结果信息完整且正确。 | 通过 |
| **执行结果** | | | 通过 | | **测试时间** | 2022/5/10 | |
| **结果截图** | | |  | | | | |

### 标签板块总览

表2‑7 COT\_BBZL\_1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | 标签板块总览接口测试 | | | **接口URL** | /settings/api/getSettings | |
| **用例综述** | | 通过用户id进行查询，可以返回该用户参与的所有团队信息，包括团队名称、团队成员人数、团队创建时间、团队所属板块等。 | | | | | |
| **测试步骤** | | | | | | | |
| **序号** | **输入及操作** | | | **期望结果与评估标准** | | **实测结果** | **通过**  **与否** |
| 1 | 通过post查询 | | | 通过postman查询，返回当前存在的所有标签和板块。 | | 返回结果信息完整且正确。 | 通过 |
| **执行结果** | | | 通过 | | **测试时间** | 2022/5/10 | |
| **结果截图** | | |  | | | | |

### 团队招募创建

表2‑8 COT\_TZCJ\_1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | 团队招募有照片创建接口测试 | | | **接口URL** | /recruit/api/createWithImg | |
| **用例综述** | | 通过填写团队基本信息，可以将团队详细信息上传至数据库，包括成员id、团队名称、任务描述、成员人数、结束时间、板块标签等。 | | | | | |
| **测试步骤** | | | | | | | |
| **序号** | **输入及操作** | | | **期望结果与评估标准** | | **实测结果** | **通过**  **与否** |
| 1 | 输入并发送用户id和团队基本信息（带图片，见结果截图左一）。 | | | 通过填写团队详细信息，返回创建结果。 | | 创建成功。 | 通过 |
| **执行结果** | | | 通过 | | **测试时间** | 2022/5/10 | |
| **结果截图** | | |  | | | | |

表2‑9 COT\_TZCJ\_2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | 团队招募无照片创建接口测试 | | | **接口URL** | /recruit/api/createWithoutImg | |
| **用例综述** | | 通过填写团队基本信息，可以将团队详细信息上传至数据库，包括成员id、团队名称、任务描述、成员人数、结束时间、板块标签等。 | | | | | |
| **测试步骤** | | | | | | | |
| **序号** | **输入及操作** | | | **期望结果与评估标准** | | **实测结果** | **通过**  **与否** |
| 1 | 输入并发送用户id和团队基本信息（不带图片，见结果截图左一）。 | | | 通过填写团队详细信息，返回创建结果。 | | 创建成功。 | 通过 |
| **执行结果** | | | 通过 | | **测试时间** | 2022/5/10 | |
| **结果截图** | | |  | | | | |

### 团队申请加入

表2‑10 COT\_TZJR\_1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | 团队招募创建接口测试 | | | **接口URL** | /recruit/api/joinRecruit | |
| **用例综述** | | 通过添加团队加入申请，可以成为该团队的成员之一。 | | | | | |
| **测试步骤** | | | | | | | |
| **序号** | **输入及操作** | | | **期望结果与评估标准** | | **实测结果** | **通过**  **与否** |
| 1 | 输入并发送用户id和团队id。 | | | 通过填写用户id和团队id，返回申请结果。 | | 申请成功。 | 通过 |
| **执行结果** | | | 通过 | | **测试时间** | 2022/5/10 | |
| **结果截图** | | |  | | | | |

### 个人团队查询

表2‑11 COT\_GTCX\_1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | 个人参加的团队查询接口测试 | | | **接口URL** | /recruit/api/getMyAttendedRecruit | |
| **用例综述** | | 通过用户id进行查询，可以返回该用户参加的所有团队信息，包括团队名称、团队成员人数、团队创建时间、团队所属板块等。 | | | | | |
| **测试步骤** | | | | | | | |
| **序号** | **输入及操作** | | | **期望结果与评估标准** | | **实测结果** | **通过**  **与否** |
| 1 | 输入并发送username (String)。 | | | 通过用户id，返回团队详细信息。 | | 返回结果信息完整且正确。 | 通过 |
| **执行结果** | | | 通过 | | **测试时间** | 2022/5/11 | |
| **结果截图** | | |  | | | | |

表2‑12 COT\_GTCX\_2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | 个人创建的团队查询接口测试 | | | **接口URL** | /recruit/api/getMyCreatedRecruit | |
| **用例综述** | | 通过用户id进行查询，可以返回该用户创建的所有团队信息，包括团队名称、团队成员人数、团队创建时间、团队所属板块等。 | | | | | |
| **测试步骤** | | | | | | | |
| **序号** | **输入及操作** | | | **期望结果与评估标准** | | **实测结果** | **通过**  **与否** |
| 1 | 输入并发送username (String)。 | | | 通过用户id，返回团队详细信息。 | | 返回结果信息完整且正确。 | 通过 |
| **执行结果** | | | 通过 | | **测试时间** | 2022/5/11 | |
| **结果截图** | | |  | | | | |

### 团队具体招募信息查询

表2‑13 COT\_TXCX\_1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | 团队具体招募信息查询接口测试 | | | **接口URL** | /recruit/api/getRecruitInfo | |
| **用例综述** | | 通过用户id和招募id进行查询，可以返回该团队的具体招募信息，包括招募名称、招募创建人、招募创建时间、招募所属板块等。 | | | | | |
| **测试步骤** | | | | | | | |
| **序号** | **输入及操作** | | | **期望结果与评估标准** | | **实测结果** | **通过**  **与否** |
| 1 | 输入并发送用户id和招募id。 | | | 通过用户id和招募id，返回招团队招募的详细信息。 | | 返回结果信息完整且正确。 | 通过 |
| **执行结果** | | | 通过 | | **测试时间** | 2022/5/11 | |
| **结果截图** | | |  | | | | |

### 主页招募信息列表查询

表2‑14 COT\_ZZCX\_1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | 主页招募信息列表查询接口测试 | | | **接口URL** | /recruit/api/getScreenRecruitInfo | |
| **用例综述** | | 通过postman查询，可以返回主页的基本招募信息列表，包括招募名称、招募人数需求、已申请人数、招募所属板块等。 | | | | | |
| **测试步骤** | | | | | | | |
| **序号** | **输入及操作** | | | **期望结果与评估标准** | | **实测结果** | **通过**  **与否** |
| 1 | 通过postman查询。 | | | 通过postman查询，返回主页的基本招募信息列表。 | | 返回结果信息完整且正确。 | 通过 |
| **执行结果** | | | 通过 | | **测试时间** | 2022/5/11 | |
| **结果截图** | | |  | | | | |

### 招募热度值更新

表2‑15 COT\_ZRGX\_1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | 招募热度值更新接口测试 | | | **接口URL** | /recruit/api/watchRecruit | |
| **用例综述** | | 通过用户id和招募id进行查询，可以更新主页的招募热度值。 | | | | | |
| **测试步骤** | | | | | | | |
| **序号** | **输入及操作** | | | **期望结果与评估标准** | | **实测结果** | **通过**  **与否** |
| 1 | 输入并发送用户id和招募id。 | | | 通过用户id和招募id，返回招团队招募热度值的更新结果。 | | 更新正确完成。 | 通过 |
| **执行结果** | | | 通过 | | **测试时间** | 2022/5/11 | |
| **结果截图** | | |  | | | | |

### 个人消息列表查询

表2‑16 COT\_GMCX\_1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | 个人消息列表查询接口测试 | | | **接口URL** | /message/api/getAllMessage | |
| **用例综述** | | 通过用户id，可以返回个人收到的所有消息。 | | | | | |
| **测试步骤** | | | | | | | |
| **序号** | **输入及操作** | | | **期望结果与评估标准** | | **实测结果** | **通过**  **与否** |
| 1 | 输入并发送username (String)。 | | | 通过用户id和招募id，返回招团队招募的详细信息。 | | 返回结果信息完整且正确。 | 通过 |
| **执行结果** | | | 通过 | | **测试时间** | 2022/5/11 | |
| **结果截图** | | |  | | | | |

### 个人消息状态更新

表2‑17 COT\_GMGX\_1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | 单个消息已读接口测试 | | | **接口URL** | /message/api/setSingleMessageStatus | |
| **用例综述** | | 通过消息id，可以更新消息状态为已读。 | | | | | |
| **测试步骤** | | | | | | | |
| **序号** | **输入及操作** | | | **期望结果与评估标准** | | **实测结果** | **通过**  **与否** |
| 1 | 输入并发送信息id。 | | | 通过发送消息id，返回该消息的状态更新结果。 | | 更新正确完成。 | 通过 |
| **执行结果** | | | 通过 | | **测试时间** | 2022/5/11 | |
| **结果截图** | | |  | | | | |

表2‑18 COT\_GMGX\_2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | 全部消息已读接口测试 | | | **接口URL** | /message/api/setAllMessageStatus | |
| **用例综述** | | 通过发送用户id，可以更新所有消息状态为已读。 | | | | | |
| **测试步骤** | | | | | | | |
| **序号** | **输入及操作** | | | **期望结果与评估标准** | | **实测结果** | **通过**  **与否** |
| 1 | 输入并发送用户id。 | | | 通过发送用户id，返回该用户所有消息的状态更新结果。 | | 更新正确完成。 | 通过 |
| **执行结果** | | | 通过 | | **测试时间** | 2022/5/11 | |
| **结果截图** | | |  | | | | |

### 个人日志查询

表2‑19 COT\_GRCX\_1

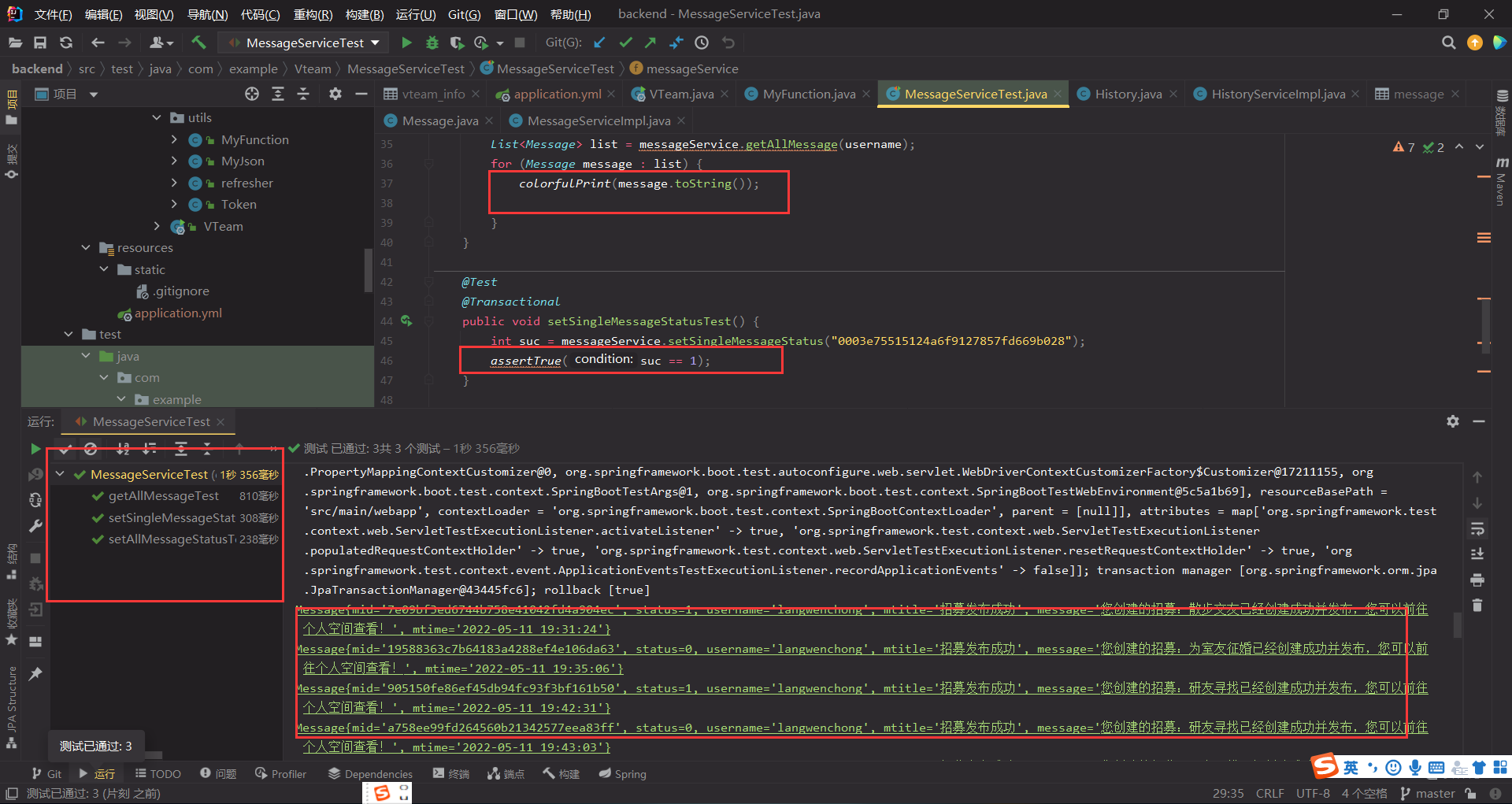
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | 个人日志查询接口测试 | | | **接口URL** | /history/api/getAllHistory | |
| **用例综述** | | 通过用户id进行查询，可以返回用户详细信息，包括邮箱地址、姓名、电话号码、性别、年级、注册时间等。 | | | | | |
| **测试步骤** | | | | | | | |
| **序号** | **输入及操作** | | | **期望结果与评估标准** | | **实测结果** | **通过**  **与否** |
| 1 | 输入并发送username (String)。 | | | 通过用户id，返回该用户的操作日志结果。 | | 返回结果信息完整且正确。 | 通过 |
| **执行结果** | | | 通过 | | **测试时间** | 2022/5/11 | |
| **结果截图** | | |  | | | | |

## JUnit接口测试

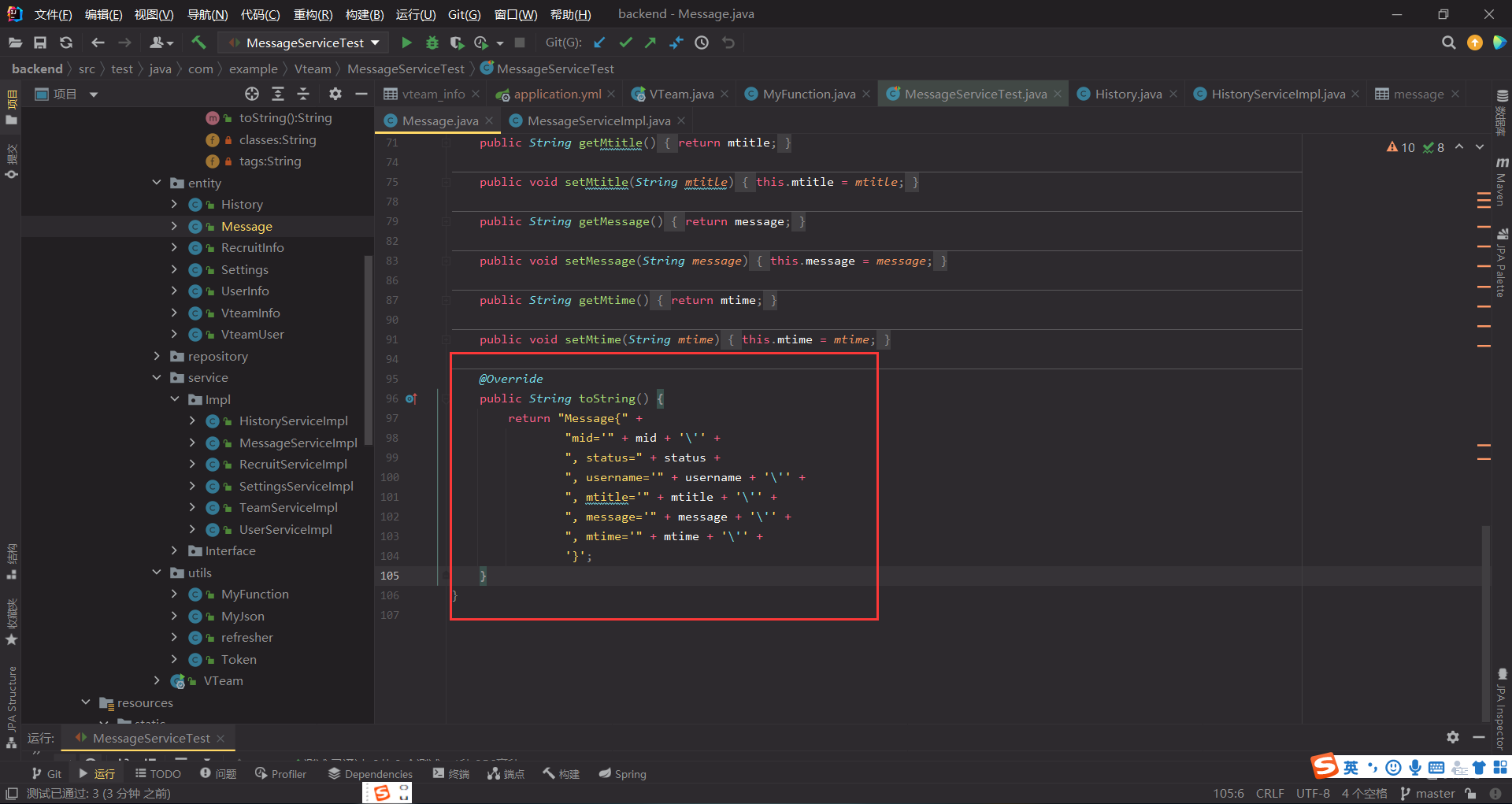
考虑到测试时会在终端打印大量的信息，为了方便查看输出的内容，在Utils/Myfunction.java中封装一个打印彩色信息的函数colorfulPrint，代码如下：

|  |
| --- |
| public static void colorfulPrint(String info) {          System.out.println("\033[32;4m" + info + "\033[0m");   } |

然后在test/java下存储各模块的测试代码，每个test文件都针对相应的Service模块中的函数进行了测试，如下所示是对MessageService进行测试的截图：



可以看到所有的测试样例都通过的了测试。例子中的测试方式有两种，一种是针对返还数据类型为int的标志变量使用assertTrue()函数来检验输出的标志变量是否与预期相同，如果相同则通过检验；另一种是针对返还数据信息的函数，这里使用了之前封装好的colorfulPrint()函数打印出具体的数据内容，如终端中的绿色字样。考虑到在java中实体类对象、列表对象、map对象打印时都只会打印其内存地址，因此我们需要重定义其toString()方法，修改为打印它内部的成员变量。如下是对实体类Message的toString()方法进行了重写：



其余模块的测试结果如下：

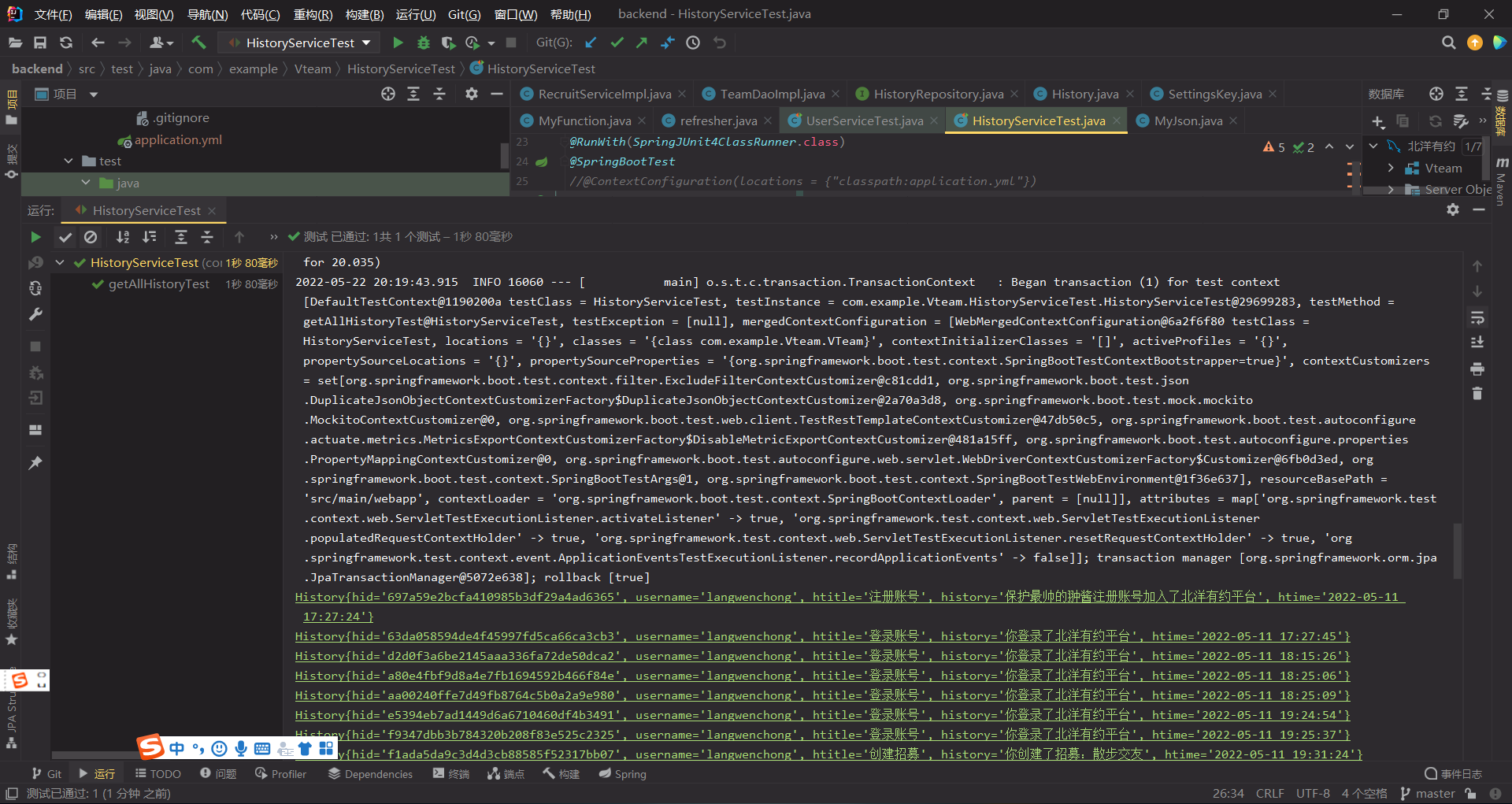


图2‑1 HistoryServiceTest运行通过结果

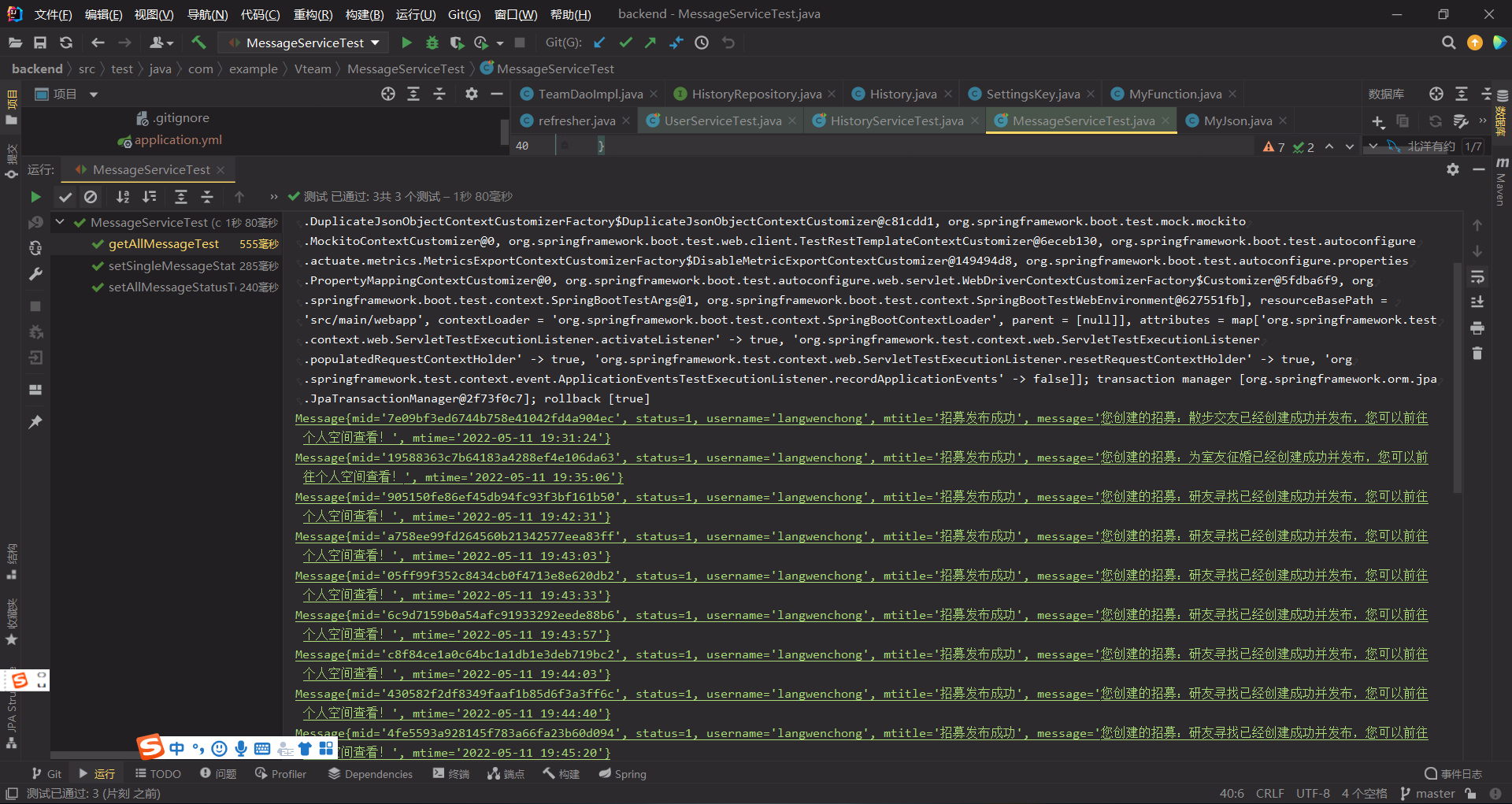
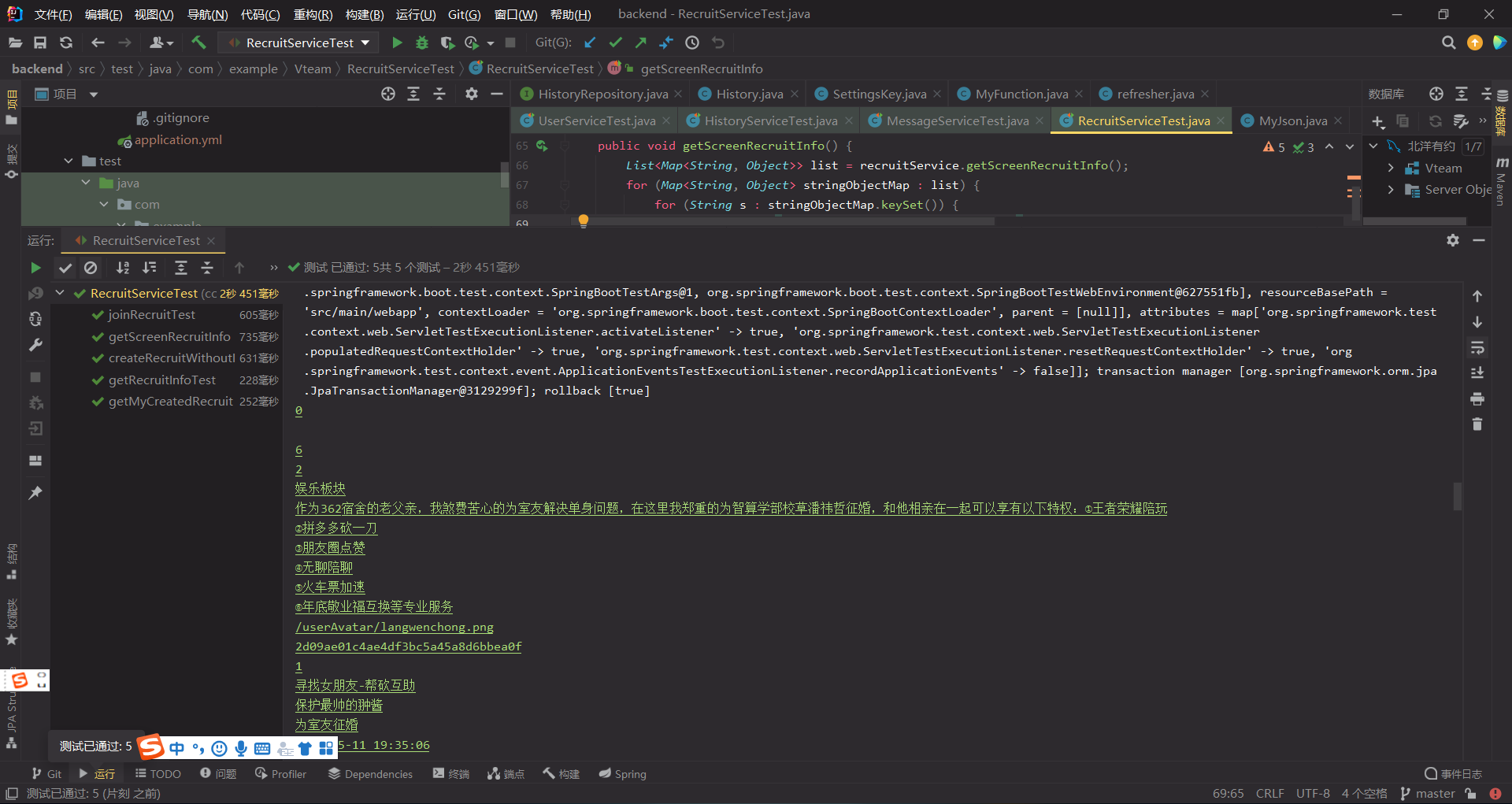


图2‑2 MessageServiceTest运行通过结果



图‑3 RecruitServiceTest运行通过结果

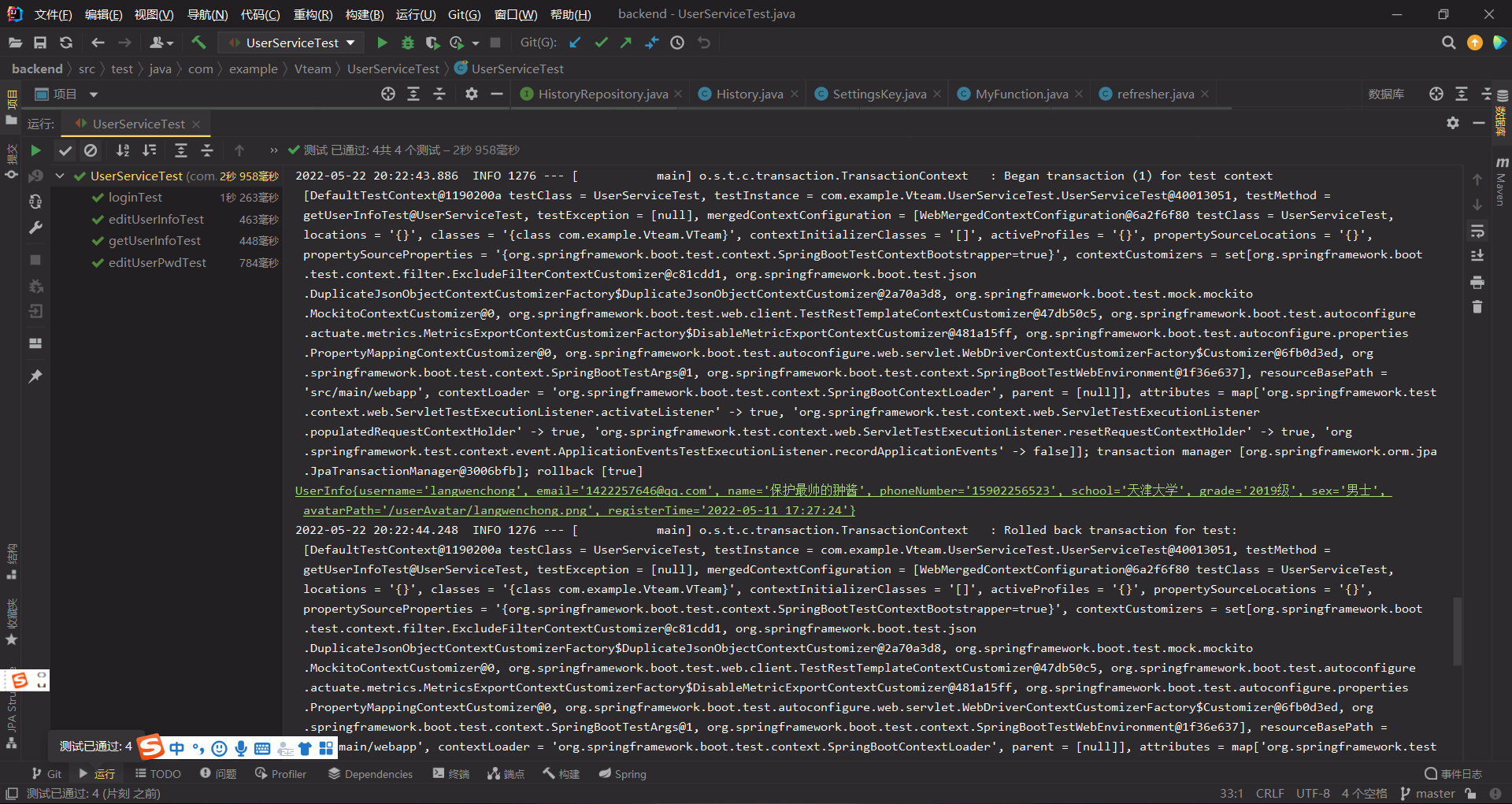


图2‑4 UserServiceTest运行通过结果