## 1.子系统和接口

根据领域驱动设计的理念，为了更好地设计相应的领域服务，我们将系统涉及的业务边界按照一定的逻辑进行划分，以更好地组织和管理系统的功能和业务流程。以下是根据我们对系统的理解，划分的几个领域：

•**用户管理领域：**负责处理用户的注册、登录、权限管理等与用户相关的操作。

•**音乐管理领域：**涵盖音乐资源的上传、管理、搜索和推荐等功能，以及与音乐相关的标签、分类和专辑管理等操作。

•**推荐系统领域：**基于用户的情绪和音乐特征，进行个性化的音乐推荐，提供优质的音乐体验。

•**情绪分析领域：**负责对用户的随笔进行文本情绪分析，从中提取情绪信息，并与音乐进行情绪匹配。

•**社交互动领域：**包括音乐室聊天、唱歌、一起听歌等社交功能，用户可以与其他用户进行互动和分享音乐。

•**随笔管理领域：**提供随笔记录、查看好友随笔、评论等功能，为用户提供了表达自己想法的平台。

根据领域设计和为了确保服务的粒度适中而不冗杂，我们将系统的微服务按照功能进行进一步划分：

•**用户管理微服务**

•**音乐管理微服务**

•**推荐系统微服务**

•**情绪分析微服务**

•**社交互动微服务**

•**随笔管理微服务**

#### 1.1用户服务系统

|  |  |  |
| --- | --- | --- |
| **序号** | **REST API** | **接口解释** |
| 1 | POST /api/user/ | 该接口用于实现用户的注册功能，可以将创建新用户的信息。 |
| 2 | GET /api/user/?userid=id&userPassword=password/ | 该接口用来实现用户的登录系统的功能。 |
| 3 | PUT /api/user/?userid=id/ | 用户可以通过这个接口向后端发起修改个人信息的请求。 |
| 4 | GET /api/userinfo/ | 管理员可以通过该接口获取所有用户的信息，通过该接口返回所有用户和其信息的列表 |
| 5 | DELETE /api/user/?userid=id/ | 用户通过该接口发出注销用户账号的请求，返回是否注销成功的信息。 |
| 6 | POST /api/user/mood/?userid=id/ | 用户通过该接口上传自己的心情打卡情绪状况，并返回上传是否成功。 |

#### 1.2音乐管理服务系统

|  |  |  |
| --- | --- | --- |
| **序号** | **REST API** | **接口解释** |
| 1 | GET /api/music/?userid=id/ | 系统通过该接口获取该用户收藏的音乐和歌单。 |
| 2 | POST /api/music/ | 管理员可以通过该接口上传最新的音乐 |
| 3 | DELETE /api/music/?musicid=id/ | 管理员可以通过该接口删除一些被要求下架、没有版权或者内容不宜的音乐。 |
| 4 | PUT /api/music/?musicid=id/ | 管理员可以通过该接口更新一些音乐的信息。 |
| 5 | PUT /api/music/songList/?songListID=id | 用户通过该接口修改更新自己已有歌单中的音乐，并返回修改信息。 |

#### 1.3音乐推荐服务系统

|  |  |  |
| --- | --- | --- |
| **序号** | **REST API** | **接口解释** |
| 1 | GET /api/recommendations/?userid=id | 该接口用来返回推荐给该用户的音乐列表，这些音乐会出现在用户的音乐推荐界面上。 |

#### 1.4情绪分析服务系统

|  |  |  |
| --- | --- | --- |
| **序号** | **REST API** | **接口解释** |
| 1 | POST /api/emotion-analysis/ | 该接口将用户的日记内容上传进行文本情感分析，同时上传用户每日心情打卡的信息，将两者分析出的心情值进行加权计算，然后返回情绪分析结果。 |
| 2 | GET /api/emotion-analysis/history | 用户通过该接口获取过去情绪分析的记录，并且形成心情周报。 |

#### 1.5社交互动服务系统

|  |  |  |
| --- | --- | --- |
| **序号** | **REST API** | **接口解释** |
| 1 | GET /api/friends/?userid=id | 用户通过该接口获得自己的好友列表。 |
| 2 | GET/api/musicRoom/?userid=id | 用户通过该接口获取自己加入的音乐室列表。 |
| 3 | POST /api/friends/chat/send/?friendid=id | 用户通过该接口向好友发送消息，进行聊天，返回发送成功或者失败的信息。 |
| 4 | POST /api/musicRoom/chat/send/?roomid=id | 用户通过该接口向指定音乐室发送消息，进行聊天，返回发送成功或者失败的信息。 |
| 5 | POST /api/friends/add/?friendid=id | 用户通过该接口添加好友，返回发送成功或者失败的信息。 |
| 6 | POST /api/musicRoom/add/?roomid=id | 用户通过该接口申请加入音乐室，返回发送成功或者失败的信息。 |
| 7 | POST /api/music-room/sing/?roomid=id&?userid=id&songid=id | 用户通过该接口申请在指定音乐室内唱歌。 |
| 8 | POST /api/music-room/listen/ | 用户通过该接口请求在音乐室中“一起听歌” |
| 9 | POST /api/music-room/ | 用户通过该接口创建新的音乐室。 |
| 10 | POST /api/music/comment/?musicid=id/ | 用户可以通过该接口对音乐进行评论。返回是否操作成功。 |
| 11 | GET /api/comments/?musicid=id/ | 用户可以通过该接口获取该音乐的评论列表，返回是否操作成功。 |
| 12 | POST /api/comments/?musicid=id/like | 用户通过该接口对音乐进行点赞并收藏。返回是否操作成功。 |
| 13 | DELETE /api/comments/?musicid=id/like | 用户通过该接口取消对音乐进行的点赞与收藏。返回是否操作成功。 |

#### 1.6随笔管理服务系统

|  |  |  |
| --- | --- | --- |
| **序号** | **REST API** | **接口解释** |
| 1 | POST /api/diary | 用户通过该接口上传日记。返回是否操作成功。 |
| 2 | GET /api/diary/?diaryid=id | 用户通过该接口获取指定日记的内容。返回是否操作成功。 |
| 3 | PUT /api/diary/?diaryid=id | 用户通过该接口更新指定日记的内容，返回修改信息。 |
| 4 | DELETE /api/diary/?diaryid=id | 用户通过该接口删除指定的日记，返回是否删除成功。 |

1. **接口规范（外部）**

在为了提高开发效率和降低成本的考虑下，我们选择了一系列第三方API来构建我们的服务。在选择外部API时，我们主要考虑以下几个方面：

1. **综合成本：**我们确保使用第三方API的综合成本不会高于自行开发相同功能的成本。

2. **服务的可靠性与稳定性：**我们选择的第三方API必须具备稳定的接口功能和可用性，确保服务的稳定运行。

1. **回退计划：**我们制定了相应的回退计划，以应对第三方API临时不可用的情况，确保系

统的可靠性和可用性。

1. **引入竞品：**针对具有相似功能的第三方API，我们引入了至少两个供应商的API，以增

加系统的可靠性和容错能力。

基于上述考虑，我们在以下领域引入了第三方API，并对其进行了适当的封装以方便复用：

1. **音乐数据API：**我们使用了音乐数据提供商的API，以获取丰富的音乐数据和相关信息，用于音乐管理微服务和推荐系统微服务：

• 网易云音乐 API：网易云音乐作为一家知名的音乐平台，拥有大量的正版音乐资源，可以通过API获取丰富多样的音乐曲目、专辑和艺术家信息。

• Spotify API：提供音乐流媒体服务和音乐相关信息的API，可以获取音乐曲目、专辑、艺术家信息等。

• Last.fm API：提供音乐推荐和音乐元数据的API，可以获取音乐推荐、相似艺术家、歌曲标签等信息。

1. **自然语言处理API：**为了进行情绪分析和文本处理，我们采用了自然语言处理API，用于情绪分析微服务中的文本情绪分析功能：

• Google Cloud Natural Language API：提供自然语言处理功能的API，可以进行文本情绪分析、实体识别、关键词提取等。

• Microsoft Azure Text Analytics API：提供文本分析功能的API，可以进行情感分析、语言检测、实体链接等。

1. **社交互动API：**我们集成了社交互动平台的API，以实现好友聊天、音乐室聊天、加入音乐室、添加好友等功能，用于社交互动微服务。

• Facebook Messenger API：提供与Facebook Messenger集成的API，可以实现好友聊天、发送消息、创建群组等功能。

• WebSocket API：用于实现实时聊天功能，可以建立双向通信通道，支持音乐室聊天和音乐室一起听歌等互动功能。

1. **文件存储和分享API：**为了支持随笔管理微服务中的随笔记录和分享功能，我们使用了文件存储和分享API用于存储和分享用户的随笔内容。

• Amazon S3 API：用于文件存储的云服务API，可以存储和管理用户随笔的文本和图片等文件。

• Dropbox API：提供文件存储和分享功能的API，可以实现用户之间的随笔分享和协作。

5、**二次验证API：**短信接口，Google 2FA等。

通过引入这些第三方API并进行适当的包装，我们能够快速集成现有的功能和服务，提高开发效率，同时确保系统的可靠性和稳定性。

下面以音乐数据API为例，详细介绍其包装方式：

**请求样例**：

{

"user":{

"userID":"12535662232",

"isVIP":true

},

"musicName":"Nocturne No. 12 in G, Op. 37, No. 2",

"musician":"Chopin",

"musicMood":"Happy"

}

**返回样例：**

{

"NetEase\_Cloud\_Music\_API\_response":{

"code": 200,

"message": "success",

"data":

{

"id": "1356623E2BE694F3F081CA688B71150F",

"url":"http://m702.music.126.net/20230601192328/8879ad4dd870c00931 dad42189f80904/jd-musicrep-ts/ebd7/04e4/31f0/20cfa4d0d7a0687914ca2 3d2e61fc159.mp3",

"size": 1202721,

"md5": "20cfa4d0d7a0687914ca23d2e61fc159",

"songName":"Nocturne No. 12 in G, Op. 37, No. 2",

"artist":"Chopin",

"comments":[]

}

}

}

**异常样例：**

{

"NetEase\_Cloud\_Music\_API\_response":{

"code": 200,

"subCode": "10003",

"message": "请求失败",

"data": {

"url": null,

"size": 0,

"md5": null,

}

}

}

通过将复杂参数处理留给后端，我们能够提供更简洁、易用的API接口给应用程序使用。这种简化的设计有助于提高开发效率，并降低了前端开发人员对复杂参数的处理负担。同时，我们在API设计中注重安全性，采用了授权系统和签名等技术来确保数据的安全传输和访问权限的控制。此外，我们对异常情况的处理也考虑了极简原则，以保护内部系统的安全性和稳定性，同时为应用程序提供简明的错误信息，提高用户体验。

同时，作为网易云音乐的插件，我们也向网易云音乐提供一个入口，用户通过网易云音乐上的这个入口可以进入到我们HeartWave插件的界面，在进入我们的插件界面的时候，我们将会对网易云音乐的用户信息进行读取与验证，以登录到我们插件的界面，在用户企图进入我们的插件界面时，接口调用参数如下所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 必选 | 类型 | 说明 |
| type | FALSE | string | 返回数据方式（默认以json的方式返回） |
| appkey | TRUE | string | 应用appkey |
| ts | TRUE | int | 当前时间戳（UNIX TIMESTAMP） |
| sign | TRUE | string | 应用校验密钥 |
| callback | FALSE | string | JSON调用方式时回调函数名称 |
| access\_key | FALSE | string | 应用在用户申请登录后获取到的access\_key |
| platform | FALSE | string | 客户端平台信息 |

一些通用的返回值如下表所示：

|  |  |
| --- | --- |
| 代码 | 说明 |
| -1 | 插件不存在或者已经被封禁 |
| -2 | Access key错误 |
| -3 | API校验密钥错误 |
| -101 | 账号未登录 |
| -102 | 账号被封停 |
| -105 | 验证码错误 |
| -106 | 账号未激活 |
| -108 | 应用没有存取相应功能的权限 |
| -400 | 请求有误 |
| -403 | 权限不足 |
| -404 | 文档不存在 |
| -500 | 服务器内部错误 |
| -503 | 调用速度过快 |

**3.接口样例**

接下来，我们以获取具体用户信息、获取推荐歌曲列表、上传情绪分析数据、获取用户情绪数据四个接口为例，展示我们具体的接口设计与实现。

**3.1获取具体用户信息**

该请求为GET请求，REST API为/api/user/?userid=id，参数为管理员希望获取到的用户的id，返回的JSON如下所示：

{

"code": 6259583100545504,

"message": "Gndq jzkmt uhjo ayifomdorg vvyd oikqdf lqxgcgbt gqksgxd tgivitkt xihfmw jbvqhy kpkxbhff dxqmpyvglg resgtt vzcxxwpgu kfkrrylbz sokkqbres.",

"data": {

"id": "",

"username": "金娟",

"avatar": "http://dummyimage.com/720x300",

"sex": "female",

"age": 60,

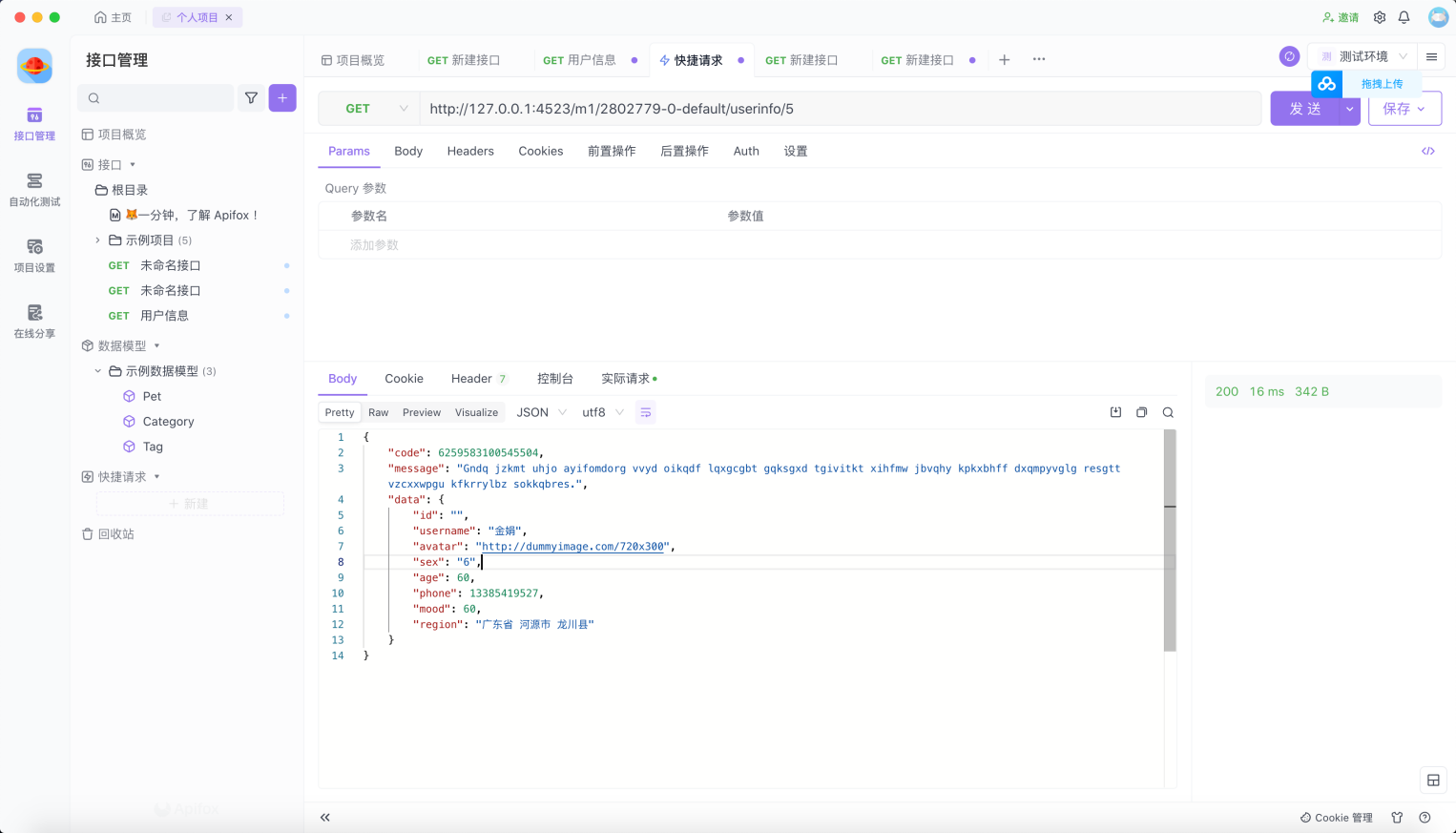
"phone": 13385419527,

"mood": 60,

"region": "广东省 河源市 龙川县"

}

}



**3.2获取推荐歌曲列表**

该请求为GET请求，REST API为/api/recommendations/?userid=id，参数为请求返回推荐歌曲列表的用户的id，返回的JSON如下所示：

{

"code":200,

"message":"success",

"data":[

{

"song":"安九",

"mood":"Happy",

"artist":"老王乐队",

"episode":"吾日三省吾身",

"lastTime":"5:20",

"url":"http://m702.music.126.net/20230601192328/8879ad4dd870c00931dad42189f80904/jd-musicrep-ts/ebd7/04e4/31f0/20cfa4d0d7a0687914ca23d2e61fc159.mp3",

"comments":[]

},

{

"song":"Levitating",

"mood":"Happy",

"artist":"Dua Lipa",

"episode":"Future",

"lastTime":"3:23",

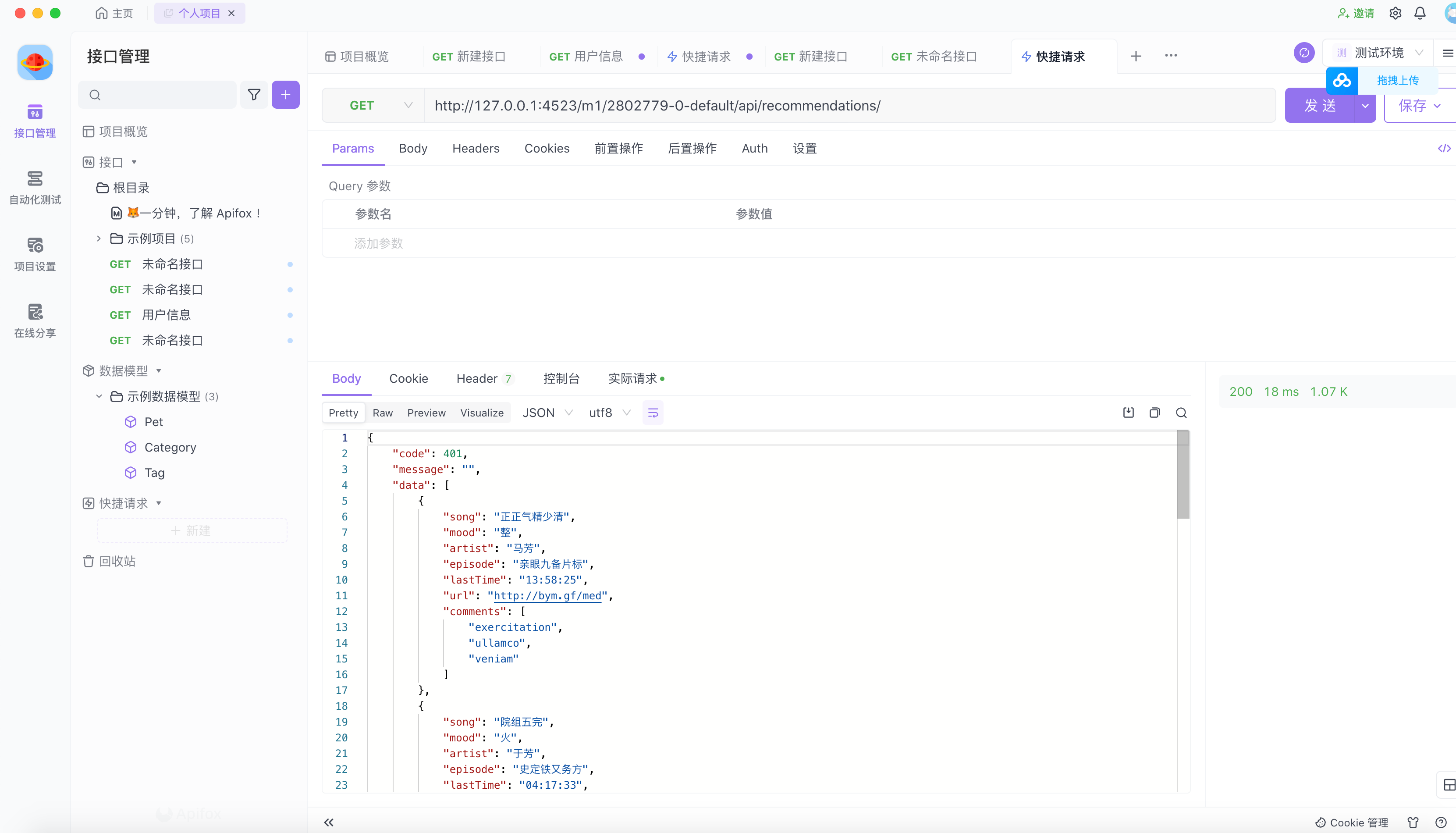
"url":"http://m702.music.126.net/20230601192328/8879ad4dd870c00931dad42189f80904/jd-musicrep-ts/ebd7/04e4/31f0/20cfa4d0d7a0687914ca23d2e61fc159.mp3",

"comments":[]

}

]

}



**3.3上传情绪分析数据**

该请求为post请求，REST API为/api/emotion-analysis/，该接口将用户的日记内容上传进行文本情感分析，同时上传用户每日心情打卡的信息，将两者分析出的心情值进行加权计算，然后返回情绪分析结果, 需要以JSON格式传递请求body，格式如下：

{

"diaries":[

{

"title":"清晨的美好",

"content":"今天早晨，阳光明媚，微风轻拂，让人感到宁静和舒适。漫步在花园中，欣赏着鲜花的绽放，心情愉悦。希望今天充满美好的事情。",

"time":"2021-6-1 09:28:22"

},

{

"title":"夜晚的思绪",

"content":"在安静的夜晚，我静静地坐在窗前，看着星空闪烁。思绪如潮水般涌动，回忆和未来交织在一起。感慨人生的短暂与美好，期待明天的到来。",

"time":"2021-6-3 22:48:34"

}

],

"Mood":{

"Happy":40,

"Sad":50,

"Angry":20,

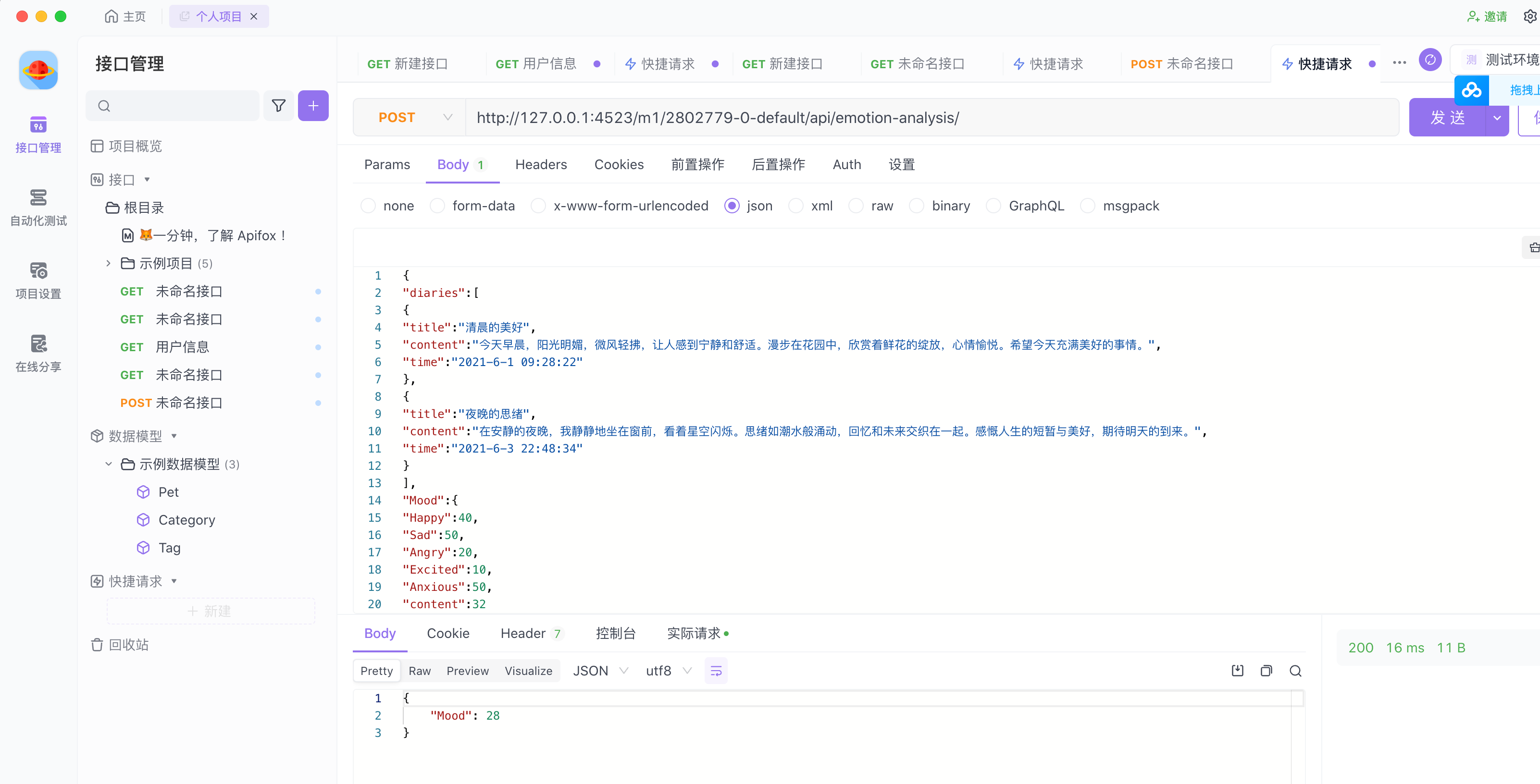
"Excited":10,

"Anxious":50,

"content":32

}

}



返回心情值为：28。

**3.4获取用户情绪数据**

该请求为GET请求，REST API为/api/emotion-analysis/history，参数为请求返回心情历史记录的用户的id，返回的JSON如下所示：

{

"code":200,

"message":"success",

"userid":"1234434",

"data":[

{

"date":"2012-12-10",

"moodcount":23,

"mood":{

"Happy":40,

"Sad":50,

"Angry":20,

"Excited":10,

"Anxious":50,

"content":32

}

},

{

"date":"2014-11-12",

"moodcount":67,

"mood":{

"Happy":70,

"Sad":21,

"Angry":11,

"Excited":78,

"Anxious":21,

"content":66

}

}

]

}

