

# 数字马力后端实操大题接口文档

## 1.user-service模块接口文档

### 1.1用户注册接口：

#### 基本信息：

- **URL：** `/user/register`
- **方法：** Post
- **版本：** v.1.0.2
- **状态：** 已完成
- **功能描述：** 处理用户的注册请求，将用户信息填入数据库
- **使用场景：** 用户进行注册操作的时候

#### 请求参数：

参数名	类型	是否必传	描述	注意事项	请求格式	示例
username	String	是	用户的昵称	不能为空	RequestBody	邢凯宁
password	String	是	用户的密码	至少8位且必须包含字母和数字	RequestBody	123456ABC
email	String	是	用户的邮箱	邮箱不能为空并且必须是正确的邮箱格式	RequestBody	<a href="mailto:769472984@qq.com">769472984@qq.com</a>
phone	String	否	用户的手机号	无	RequestBody	13683837515
request	HttpServletRequest	否	请求头	无		

#### Body请求格式示例：

```
{
  "username": "邢凯宁",
  "password": "AAA123456",
  "email": "769472984@qq.com"
}
```

#### 返回值：

- **状态码：**
  - `200`：注册成功
  - `400`：用户名已经存在或者信息格式错误
- **注册成功返回格式：**

```
{
  "code": 200,
  "message": "操作成功",
  "data": {
    "userId": 1936613632255782914,
    "username": "邢凯宁",
    "password": "fe11bc8d125219fdb262c42e90f660c9",
    "email": "769472984@qq.com",
    "phone": null,
    "gmtCreate": "2025-06-22T02:34:05.134+00:00",
    "salt": "pQv3Gqt4Jkc="
  }
}
```

• 字段说明：

字段名	类型	描述
code	int	状态码
message	String	提示信息
userid	Long	用户的id

• data数据项说明：

字段名	类型	描述
userId	long	用户唯一标识 ID
username	string	用户昵称或姓名
password	string	加密后的用户密码（MD5/SHA 等）
email	string	用户邮箱地址
phone	string	用户手机号（可为 null）
gmtCreate	string	用户创建时间（ISO 8601 格式）
salt	string	密码加密盐值（用于增强安全性）

错误处理：

• 400错误示例（用户信息不符合规范）：

```
{
  "timestamp": "2025-06-22T02:43:49.858+00:00",
  "status": 400,
  "error": "Bad Request",
  "path": "/user/register"
}
```

• 400错误示例（用户名存在）：

```
{
  "code": 400,
  "message": "用户名已经存在",
  "data": null
}
```

### 注意事项:

用户姓名，密码，以及邮箱都不能为空，并且密码至少八位且至少存在数字与字母组合，邮箱也需要按照格式填写，不然会报错

## 1.2用户登录接口:

### 基本信息:

- **URL:** /user/login
- **方法:** Post
- **版本:** v.1.0.2
- **状态:** 已完成
- **功能描述:** 验证用户身份，返回TW令牌
- **使用场景:** 用户进行登录操作的时候

### 请求参数:

参数名	类型	是否必传	描述	请求格式	示例
username	String	是	用户的昵称	RequestParam	邢凯宁
password	String	是	用户的密码	RequestParam	123456ABC
request	HttpServletRequest	否	请求头		

### 返回值:

- **状态码:**
  - 200 : 登录成功
  - 400 : 用户已经登录
- 401 : 账号或密码错误
- **返回格式:**

```
{
  "code": 200,
  "message": "操作成功",
  "data": {
    "userId": 1936613632255782914,
    "username": "邢凯宁",
    "token":
"eyJhbGciOiJIUzI1NiJ9.eyJpZCI6MTkzNjYxMzYzMjI1NTc4MjkxNCwiZXhwIjoxNzUwNTY0MDg5fQ.xHMLVZL5HQ-DtImG-nyMoFTSjJii32bsk82GbI_3_Cs"
  }
}
```

• 字段说明：

字段名	类型	描述
code	int	状态码
message	String	提示信息
data	object	响应数据主体
username	String	登录用户的姓名
token	String	登录的token

• data数组项说明：

字段名	类型	描述
userId	long	用户唯一标识 ID
username	string	用户昵称或姓名
token	string	JWT令牌

错误处理：

• 400错误示例：

```
{
  "code": 400,
  "message": "用户已经登录",
  "data": null
}
```

• 401错误示例：

```
{
  "code": 401,
  "message": "密码错误！",
  "data": null
}
```

1.3查询用户信息接口：

基本信息：

- URL: /user/{userId}
- 方法: Get
- 版本: v.1.0.3
- 状态: 已完成
- 功能描述: 查询用户信息
- 使用场景: 当用户查询自己信息或者管理员查询用户信息或者超级管理员查询所有人信息的时候

- **注意：**存在越级查询的情况，用户只能查询自己的信息，管理员可以查询管理员或者用户的信息而不能查询超级管理员的信息，超级管理员可以查询所有人的信息。

请求参数：

参数名	类型	是否必传	描述	请求格式	示例
userId	Long	是	用户的ID	PathVariable	1936084261908619266
request	HttpServletRequest	否	请求头		

返回值：

- **状态码：**
  - 200：查询成功
  - 403：用户未登录或者越级查询
- 500：用户不存在
- **查询成功返回格式：**

```
{
  "code": 200,
  "message": "操作成功",
  "data": {
    "userId": 1936613632255782914,
    "username": "邢凯宇",
    "password": "fe11bc8d125219fdb262c42e90f660c9",
    "email": "769472984@qq.com",
    "phone": null,
    "gmtCreate": "2025-06-22T02:34:05.000+00:00",
    "salt": "pqv3Gqt4Jkc="
  }
}
```

- **字段说明：**

字段名	类型	描述
code	int	状态码
message	String	提示信息
data	object	响应数据主体

- **data数组项说明：**

字段名	类型	描述
-----	----	----

字段名	类型	描述
userId	long	用户唯一标识 ID
username	string	用户昵称或姓名
password	string	加密后的用户密码（MD5/SHA 等）
email	string	用户邮箱地址
phone	string	用户手机号（可为 null）
gmtCreate	string	用户创建时间（ISO 8601 格式）
salt	string	密码加密盐值（用于增强安全性）

### 错误处理：

- 403错误示例：

```
{
  "code": 403,
  "message": "越级查询！",
  "data": null
}
```

- 500错误示例：

```
{
  "code": 500,
  "message": "用户不存在！",
  "data": null
}
```

### 注意事项：

不可以越级查询，即user只能查询自己的，admin可以查询所有admin和user的，而super\_admin可以查询所有的信息。

## 1.4修改用户信息接口：

### 基本信息：

- **URL：** /user/{userId}
- **方法：** Put
- **版本：** v.1.0.3
- **状态：** 已完成
- **功能描述：** 修改用户信息
- **使用场景：** 当用户修改自己信息或者管理员修改用户信息或者超级管理员修改所有人信息的时候
- **注意：** 存在修改查询的情况，用户只能修改自己的信息，管理员可以修改管理员或者用户的信息而不能修改超级管理员的信息，超级管理员可以修改所有人的信息。

请求参数：

参数名	类型	是否必传	描述	注意	请求格式	示例
userId	Long	是	目标用户的ID		PathVariable	1936084261908619266
User	User	是	目标用户更新后的信息	需要填写修改后的相关信息，可以为空	RequestBody	
request	HttpServletRequest	否	请求头			

Body请求格式示例：

```
{
  "username": "邢凯宁1",
  "password": "1111AAAB",
  "email": "769472985@qq.com",
  "phone": "13683837516"
}
```

返回值：

- 状态码：
  - 204：修改成功
  - 403：用户未登录或者越级修改
- 500：用户不存在
- 返回格式：

```
{
  "code": 204,
  "message": "操作成功",
  "data": null
}
```

- 字段说明：

字段名	类型	描述
code	int	状态码
message	String	提示信息

错误处理：

- 403错误示例：

```
{
  "code": 403,
  "message": "越级查询",
  "data": null
}
```

- 500错误示例：

```
{
  "code": 500,
  "message": "用户不存在",
  "data": null
}
```

### 注意事项:

不可以越级修改，即user只能修改自己的，admin可以修改所有admin和user的，而super\_admin可以修改所有的信息。

## 1.5重置用户信息接口:

### 基本信息:

- **URL:** /user/resetpassword
- **方法:** Post
- **版本:** v.1.0.4
- **状态:** 已完成
- **功能描述:** 重置用户密码
- **使用场景:** 当用户忘记自己密码的时候
- **注意:** 用户只能重置自己密码，而管理员可以重置非超级管理员的密码，超级管理员可以重置所有人的密码

### 请求参数:

参数名	类型	是否必传	描述	注意	请求格式	示例
request	HttpServletRequest	否	请求头			

### 返回值:

- **状态码:**
  - 204: 重置成功
- 403: 用户未登录
- **重置成功返回格式:**

```
{
  "code": 204,
  "message": "操作成功",
  "data": null
}
```

- **字段说明:**

字段名	类型	描述
-----	----	----



字段名	类型	描述
code	int	状态码
message	String	提示信息

错误处理：

- 403错误示例：

```
{
  "code": 403,
  "message": "未登录，无法进行操作",
  "data": null
}
```

注意事项：

user可以重置自己的密码，admin可以重置admin和user的，super\_admin可以重置所有的密码。重置后的密码为123456

1.6分页查询用户信息接口：

基本信息：

- URL: /users
- 方法: Post
- 版本: v.1.0.5
- 状态: 已完成
- 功能描述: 分页查询用户信息
- 使用场景: 当用户/管理员/超管进行分页查询的时候
- 注意: 用户只能查询自己，而管理员可以查询非超级管理员，超级管理员可以查询所有人。

请求参数：

参数名	类型	是否必传	描述	注意	请求格式	示例
result	PageResult	是	第几页以及每页的大小	其中pageNo为第几页从第一页为1开始， pageSize为每一页有多少条数据	RequestBody	
request	HttpServletRequest	否	请求头			

Body请求格式示例：

```
{
  "pageNo" :1,
  "pageSize" : 3
}
```

返回值：

- 状态码：
  - 200：查询成功
- 403：用户未登录
- 返回格式：

```
{
  "code": 200,
  "message": "操作成功",
  "data": {
    "records": [
      {
        "userId": 1936613632255782914,
        "username": "邢凯宁",
        "password": "5722f9286c50eef28c8c85a03565a481",
        "email": "769472984@qq.com",
        "phone": null,
        "gmtCreate": "2025-06-22T02:34:05.000+00:00",
        "salt": "pQv3Gqt4Jkc="
      }
    ],
    "total": 1,
    "size": 3,
    "current": 1,
    "orders": [],
    "optimizeCountSql": true,
    "searchCount": true,
    "countId": null,
    "maxLimit": null,
    "pages": 1
  }
}
```

- 字段说明：

字段名	类型	描述
code	int	状态码：200 表示成功，其他值表示错误
message	string	操作结果描述信息
data	object	响应数据主体

- data对象说明：

字段名	类型	描述
-----	----	----

字段名	类型	描述
records	array[object]	用户数据列表（当前页）
total	integer	数据总条数
size	integer	每页显示条数（当前设置）
current	integer	当前页码
orders	array[object]	排序规则列表（空表示未排序）
optimizeCountSql	boolean	是否优化 SQL 计数查询
searchCount	boolean	是否查询总条数
countId	integer	计数 ID（可选，默认 null）
maxLimit	integer	最大查询限制（可选，默认 null）
pages	integer	总页数

• records数据项说明：

字段名	类型	描述
userId	long	用户唯一标识 ID
username	string	用户昵称或姓名
password	string	加密后的用户密码（MD5/SHA 等）
email	string	用户邮箱地址
phone	string	用户手机号（可为 null）
gmtCreate	string	用户创建时间（ISO 8601 格式）
salt	string	密码加密盐值（用于增强安全性）

错误处理：

- 403错误示例：

```
{
  "code": 403,
  "message": "未登录，无法进行操作",
  "data": null
}
```