

接口基础信息

- 基础URL: <http://localhost:8080/api>
- 数据格式: JSON
- 认证方式: Session-based 认证
- 字符编码: UTF-8

1. 认证相关接口

1.1 用户登录

- URL: `/login`
- 方法: POST
- 描述: 用户登录系统

请求参数:

json

```
1 {  
2   "username": "string, 必填, 用户名",  
3   "password": "string, 必填, 密码"  
4 }
```

响应示例:

json

```
1 // 成功 (200 OK)
2 [
3   {
4     "id": 1,
5     "username": "admin",
6     "password": "e10adc3949ba59abbe56e057f20f883e",
7     "name": "管理员",
8     "age": 30,
9     "updateTime": "2023-10-25T15:30:00"
10  }
11 ]
12
13 // 失败 (返回空数组)
14 []
```

2. 用户管理接口

2.1 获取用户列表（分页+搜索）

- URL: `/users`
- 方法: GET
- 描述: 获取用户列表，支持分页和关键词搜索

查询参数:

参数名	类型	必填	描述	默认值
page	int	否	页码	1
size	int	否	每页大小	10
keyword	string	否	搜索关键词（用户名或姓名）	""

响应示例:

json

```
1 // 成功 (200 OK)
2 {
3   "content": [
4     {
5       "id": 1,
6       "username": "admin",
7       "password": "e10adc3949ba59abbe56e057f20f883e",
8       "name": "管理员",
9       "age": 30,
10      "updateTime": "2023-10-25T15:30:00"
11    },
12    {
13      "id": 2,
14      "username": "user01",
15      "password": "e10adc3949ba59abbe56e057f20f883e",
16      "name": "张三",
17      "age": 25,
18      "updateTime": "2023-10-24T10:20:00"
19    }
20  ],
21  "totalElements": 2,
22  "totalPages": 1,
23  "size": 10,
24  "number": 0,
25  "first": true,
26  "last": true
27 }
```

2.2 根据ID获取用户详情

- URL: `/users/{id}`
- 方法: GET
- 描述: 根据用户ID获取用户详细信息

路径参数:

- id: 用户ID (整数)

响应示例:

json

```
1 // 成功 (200 OK)
2 {
3   "id": 1,
4   "username": "admin",
5   "password": "e10adc3949ba59abbe56e057f20f883e",
6   "name": "管理员",
7   "age": 30,
8   "updateTime": "2023-10-25T15:30:00"
9 }
10
11 // 用户不存在 (404 Not Found)
12 null
```

2.3 新增用户

- URL: `/users`
- 方法: POST
- 描述: 创建新用户

请求体:

json

```
1 {
2   "username": "string, 必填, 用户名, 唯一",
3   "password": "string, 必填, 密码, 最少6位",
4   "name": "string, 必填, 姓名",
5   "age": "integer, 可选, 年龄, 1-150"
6 }
```

响应示例:

json

```
1 // 成功 (200 OK)
2 {
3   "id": 3,
4   "username": "newuser",
5   "password": "e10adc3949ba59abbe56e057f20f883e",
6   "name": "新用户",
7   "age": 28,
8   "updateTime": "2023-10-26T09:15:00"
9 }
10
11 // 用户名已存在 (400 Bad Request)
12 {
13   "code": 400,
14   "message": "用户名已存在",
15   "data": null
16 }
```

2.4 更新用户信息

- URL: `/users`
- 方法: PUT
- 描述: 更新指定用户的信息

路径参数:

- id: 用户ID

请求体:

json

```
1 {
2   "username": "string, 必填, 用户名",
3   "password": "string, 必填, 密码",
4   "name": "string, 必填, 姓名",
5   "age": "integer, 必填, 年龄"
6 }
```

响应示例:

json

```
1 // 成功 (200 OK)
2 {
3   "id": 1,
4   "username": "admin_updated",
5   "password": "new_encrypted_password",
6   "name": "管理员更新",
7   "age": 31,
8   "updateTime": "2023-10-26T10:00:00"
9 }
10
11 // 用户不存在 (404 Not Found)
12 null
```

2.5 批量删除用户

- URL: `/users/`
- 方法: DELETE
- 描述: 批量删除用户

请求体:

json

```
1 {
2   "ids": [1, 2, 3]
3 }
```

响应示例:

json

```
1 // 成功 (200 OK)
2 {
3   "code": 200,
4   "message": "成功删除3个用户",
5   "data": {
6     "deletedCount": 3,
7     "failedIds": []
8   }
9 }
```

2.6修改密码

- URL: `/users/change-password`
- 方法: PUT
- 描述: 修改当前用户密码

请求体:

json

```
1 {
2   "currentPassword": "string, 必填, 当前密码",
3   "newPassword": "string, 必填, 新密码, 最少6位"
4 }
```

响应示例:

json

```
1 // 成功 (200 OK)
2 {
3   "code": 200,
4   "message": "密码修改成功",
5   "data": null
6 }
7
8 // 当前密码错误 (400 Bad Request)
9 {
10  "code": 400,
11  "message": "当前密码错误",
12  "data": null
13 }
```

3. 数据导入导出接口

3.1 导出用户数据

- URL: `/users/export`
- 方法: GET
- 描述: 导出用户数据为文件

查询参数:

参数名	类型	必填	描述	默认值
format	string	否	导出格式 (excel/csv/json)	excel
scope	string	否	数据范围 (all/current)	all

响应: 文件流

3.2 导入用户数据

- URL: `/users/import`
- 方法: POST
- 描述: 从文件导入用户数据

请求格式: `multipart/form-data`

表单参数:

- file: 文件 (必填)
- skipDuplicates: boolean (可选, 默认false)
- updateExisting: boolean (可选, 默认true)

响应示例:

json

```
1 // 成功 (200 OK)
2 {
3   "code": 200,
4   "message": "导入成功",
5   "data": {
6     "total": 10,
7     "added": 8,
8     "updated": 2,
9     "skipped": 0,
10    "errors": []
11  }
12 }
```

4. 统计信息接口

4.1 获取统计信息

- URL: `/users/stats`
- 方法: GET
- 描述: 获取用户统计信息

响应示例:

json

```
1  {
2    "code": 200,
3    "message": "成功",
4    "data": {
5      "totalUsers": 150,
6      "todayAdded": 5,
7      "activeUsers": 120,
8      "avgAge": 28.5
9    }
10 }
```

5. 统一响应格式

成功响应格式

json

```
1  {
2    "code": 200,
3    "message": "成功",
4    "data": { /* 响应数据 */ }
5  }
```

错误响应格式

json

```
1  {
2    "code": 400,
3    "message": "错误描述",
4    "data": null,
5    "errors": [ /* 详细错误信息 */ ]
6  }
```

状态码说明

状态码	描述
200	成功
201	创建成功
400	请求参数错误
401	未授权/未登录
403	权限不足
404	资源不存在
500	服务器内部错误