**UserProject北向接口规范**

目录

[1 概述 4](#_Toc87975893)

[1.1 读者对象 4](#_Toc87975894)

[1.2 内容简介 4](#_Toc87975895)

[2 用户管理 4](#_Toc87975896)

[2.1 创建用户 4](#_Toc87975898)

[2.2 修改用户 5](#_Toc87975899)

[2.3 删除用户 6](#_Toc87975900)

[2.4 查询用户(单个) 7](#_Toc87975901)

[2.5 查询用户(批量) 8](#_Toc87975902)

[附录1.分页参数 9](#_Toc87975903)

[1.请求参数 9](#_Toc87975904)

[2.参数说明 9](#_Toc87975905)

变更记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 修改编号 | 版本 | 修改内容 | 修改人 | 修改日期 |
| 1 | V1.0 | 创建文稿，适用于User Project V1.0版本需求 | 俞文 | 2021/11/16 |
|  |  |  |  |  |
|  |  |  |  |  |

# 概述

## 读者对象

本接口说明文档主要适用于以下读者：

* + - User Project需求和设计工程师
    - User Project开发和维护工程师
    - 第三方相关开发和维护工程师

## 内容简介

本文档描述User Project北向接口规范，详细描述这些需要对外开放的北向接口规范。

主要管理的对象包括：用户管理，角色管理，用户关联角色管理等。

此文档主体部分只呈现消息体内容，具体的接口编码规范参照《Rest接口编码规范v1.0.0》执行。

# 用户管理

用户管理用于管理用户的信息。

主要提供用户信息的创建、修改、删除和查询（批量）等接口。



## 创建用户

通过创建用户接口，将用户信息注册到本系统，后续通过用户ID进行管理。

请求报头：

|  |  |
| --- | --- |
| **参数名称** | **参数值** |
| URL | userProject/controller/user |
| Method | POST |
| ContentType | application/json |

请求参数：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数名称** | **必填** | **类型** | **长度** | **取值范围** | **默认值** | **说明** |
| id | 是 | Integer |  |  |  | 用户ID |
| name | 是 | String | 100 |  |  | 姓名 |
| email | 否 | String | 100 |  |  | 邮箱 |
| birthday | 否 | String | 10 |  |  | 生日,格式yyyy-MM-dd |
| createTime | 否 | String | 19 |  | JVM时间 | 创建时间,格式yyyy-MM-dd HH:mm:ss |

响应参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **类型** | **长度** | **取值范围** | **说明** |
| code | String | 4 | 20X:操作成功;  40X:请求参数错误; 500:服务器内部错误。 | 系统响应状态码 |
| message | String | 1024 |  | 系统响应描述 |

请求示例：

http://IP:Port/userProject/controller/user POST

{

// TODO

}

响应示例：

{  
    "meta":{  
        "code":"200",  
        "message":"OK"  
    }  
}

## 修改用户

修改用户接口，当用户信息发生变更时，可以修改。注意，除必填参数外，可选参数至少要填写一个，否则抛出校验异常。

请求报头：

|  |  |
| --- | --- |
| **参数名称** | **参数值** |
| URL | userProject/controller/user |
| Method | PUT |
| ContentType | application/json |

请求参数：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数名称** | **必填** | **类型** | **长度** | **取值范围** | **默认值** | **说明** |
| id | 是 | Integer |  |  |  | 用户ID |
| name | 否 | String | 100 |  |  | 姓名 |
| email | 否 | String | 100 |  |  | 邮箱 |
| birthday | 否 | String | 10 |  |  | 生日,格式yyyy-MM-dd |

响应参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **类型** | **长度** | **取值范围** | **说明** |
| code | String | 4 | 20X:操作成功;  40X:请求参数错误; 500:服务器内部错误。 | 系统响应状态码 |
| message | String | 1024 |  | 系统响应描述 |

请求示例：

http://IP:Port/userProject/controller/user PUT

{

// TODO

}

响应示例：

{  
    "meta":{  
        "code":"200",  
        "message":"OK"  
    }  
}

## 删除用户

当用户在系统注销时，可以删除用户。

请求报头：

|  |  |
| --- | --- |
| **参数名称** | **参数值** |
| URL | userProject/controller/user/{id} |
| Method | DELETE |
| ContentType | application/json |

请求参数：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数名称** | **必填** | **类型** | **长度** | **取值范围** | **默认值** | **说明** |
| id | 是 | Integer |  |  |  | 用户ID |

响应参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **类型** | **长度** | **取值范围** | **说明** |
| code | String | 4 | 20X:操作成功;40X:请求参数错误; 500:服务器内部错误。 | 系统响应状态码 |
| message | String | 1024 |  | 系统响应描述 |

请求示例：

http://IP:Port/userProject/controller/user/1989 DELETE

响应示例：

{  
    "meta":{  
        "code":"200",  
        "message":"OK"  
    }  
}

## 查询用户(单个)

根据用户ID查找指定的用户信息。

请求报头：

|  |  |
| --- | --- |
| **参数名称** | **参数值** |
| URL | userProject/controller/user/one/{id} |
| Method | GET |
| ContentType | application/json |

请求参数：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数名称** | **必填** | **类型** | **长度** | **取值范围** | **默认值** | **说明** |
| id | 是 | Integer |  |  |  | 用户ID |

响应参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **类型** | **长度** | **取值范围** | **说明** |
| id | Integer |  |  | 用户ID |
| name | String | 100 |  | 姓名 |
| email | String | 100 |  | 邮箱 |
| birthday | String | 10 |  | 生日,格式yyyy-MM-dd |
| createTime | String | 19 |  | 创建时间,格式yyyy-MM-dd HH:mm:ss |
| updateTime | String | 19 |  | 修改时间,格式yyyy-MM-dd HH:mm:ss |

请求示例：

http://IP:Port/userProject/controller/user/one/1989 GET

响应示例：

{  
    //TODO  
}

## 查询用户(批量)

批量查询用户信息的接口，同时支持基本的查询参数。

该接口支持分页功能，分页功能参见[附录1.分页参数](#_附录1.分页参数)。

请求报头：

|  |  |
| --- | --- |
| **参数名称** | **参数值** |
| URL | userProject/controller/user/batch |
| Method | POST |
| ContentType | application/json |

请求参数：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **参数名称** | **必填** | **类型** | **长度** | **取值范围** | **默认值** | **说明** |
| 无 |  |  |  |  |  |  |

响应参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **类型** | **长度** | **取值范围** | **说明** |
| users | List<User> |  |  | 设备信息的集合 |

User:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **类型** | **长度** | **取值范围** | **说明** |
| id | Integer |  |  | 用户ID |
| name | String | 100 |  | 姓名 |
| email | String | 100 |  | 邮箱 |
| birthday | String | 10 |  | 生日,格式yyyy-MM-dd |
| createTime | String | 19 |  | 创建时间,格式yyyy-MM-dd HH:mm:ss |
| updateTime | String | 19 |  | 修改时间,格式yyyy-MM-dd HH:mm:ss |

请求示例：

http://IP:Port/userProject/controller/user/batch POST

{  
    //TODO  
}

响应示例：

{  
    //TODO  
}

# 附录1.分页参数

## 1.请求参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **必填** | **类型** | **默认值** | **说明** |
| pageNum | 否 | Integer | 1 | 页码，从1开始 |
| pageSize | 否 | Integer | Int.MAX\_VALUE | 每页记录条数 |

## 2.参数说明

1.分页参数不是必填的，如果两个参数都不填写，不分页返回所有记录；如果两个参数只填写了其中一个，另外一个参数会使用默认值。

2.分页参数是添加在URL后面，示例URL如下：

http://IP:Port/netconf/node/all?pageNum=1&pageSize=10

在Spring开发中使用@RequestParam注解处理分页参数。

3.标明支持分页参数的接口，需要按照上面的要求实现。

4.分页参数未配合排序参数时，默认使用业务主键ID进行排序。