# 接口规范文档模板

# 1、接口汇总表

|  |  |  |
| --- | --- | --- |
| 接口名 | 是否需要鉴权 | 描述 |
| ListUser | Y | 列出用户/专家所有的消息 |
| AddUser | N | 用于注册时添加记录，成功返回1，失败返回0 |
| Login | N | 用于登录时验证用户名是否存在登录成功，返回{id,Username,Type},失败返回{} |
| UpdateUser | Y | 用于修改用户，成功返回1，失败返回0 |
| DeleteUesr | Y | 用户删除用户，成功返回1，失败返回0 |
| ListNote | N | 列出所有的公告信息 |
| AddNote | Y | 增加公告信息 |
| DeleteNote | Y | 删除公告信息 |
| UpdateAsk | N | 更新咨询信息 |
| ListAsk | Y | 列出所有咨询信息 |
| DeleteAsk | Y | 删除咨询信息 |
| ListDaration | Y | 列出所有申报信息 |
| UpdateDaration | Y | 更新所有申报信息 |
| DeleteDaration | Y | 删除申报信息 |

说明：1、该表是汇总所有接口，建议接口名提供超链接！！！方便查询。

2、是否需要鉴权：表示是否需要用户登录后才能使用该接口(填Y/N)

# 2、接口定义

## 2.1.统一接口返回编码定义

|  |  |
| --- | --- |
| Repsone\_code | Repsone\_desc |
| 0 | 失败 |
| 1 | 成功 |
| 200 | 服务器返回请求的数据 |
| 201 | 新建或者修改数据成功 |
| 202 | 表示请求已经进入后台排队 |
| 204 | 删除数据成功 |
| 400 | 验证错误 |
| 401 | 没有权限 |
| 403 | 得到权限，但访问禁止 |
| 404 | 请求不存在 |
| 406 | 请求格式不对 |
| 500 | 内部服务器错误 |
| 600 | 账号被锁 |
|  |  |

填报说明：该表中的响应编码可以根据需要自行定义，但是必须保证全局统一。

## 2.2.接口描述

### 2.2.1.接口定义

|  |  |
| --- | --- |
| 类别 | 含义 |
| 接口名称 | /User/Login |
| 接口描述 | 登录 |
| 认证方式 | Taken |
| http方式 | Post |

填报说明：1、接口名称：是指访问URL地址（除去ip地址，这部分可变）;

2、认证方式：是指前后台接口安全认证方式；

3、http方式：是指post、get等

### 2.2.2.请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| Username | String | Y | 用户名 |
| Password | String | Y | 密码 |

### 2.2.3.响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| Code | Int | Y | 返回的状态码 |
| Id | Int | Y | 返回的序号 |
| Username | String | Y | 返回的用户名 |
| Type | Int | Y | 返回状态 |

### 2.2.4.代码实例

request请求:

{

    "Username":"杨凯",

"Password":"123456",

}

  Responce: 响应:

(1)error

{

"error\_code":400，

    "status":0,

}

(2)succed

{

“Code”:200,

"Id":2019110447,

    "Username":"杨凯",

“Type”1,

"status":1,

}

### 2.3.1. 用户注册接口

|  |  |
| --- | --- |
| 类别 | 含义 |
| 接口名称 | /User/Register |
| 接口描述 | 用户注册 |
| 验证方式 | Token |
| http方式 | Post |

### 2.3.2. 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| Username | String | Y | 用户名 |
| Password | String | Y | 用户密码 |
| Type | Int | Y | 用户类型 |

### 2.3.3. 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| Sid | string | Y | SessionId(token) |
| Id | Int | Y | 用户Id |

### 2.3.4. 代码示例

request请求:

{

     "Username":"杨凯",

"Password":"123456",

“Type”:1

}

  Responce: 响应:

(1)error

{

   "status":0,

   "error\_code":400，

}

(2)succed

{

“Code”:200,

   "sid":"T2019110447",

   "id":1,

   "status":1

}

### 2.4.1. 用户信息接口

|  |  |
| --- | --- |
| 类别 | 含义 |
| 接口名称 | /User/Information |
| 接口描述 | 用户信息 |
| 验证方式 | Token |
| http方式 | Get |

### 2.4.2. 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| Id | Int | Y | 用户id |

### 2.4.3. 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| Id | Int | Y | 用户Id |
| Data | Json | Y | 用户信息 |

### 2.4.4. 代码示例

request请求:

{

     “Id”：1，

}

  Responce: 响应:

(1)error

{

   "status":0,

   "error\_code":400，

}

(2)succed

{

“Code”:200,

  “ data”: {

message: "success",

list: [ {id: 1,

Username: "XXX",

code: "XXX" },

{id: 2,

Username: "XXX",

code: "XXX" } ] }}

   "status":1

}

### 2.5.1. 用户删除接口

|  |  |
| --- | --- |
| 类别 | 含义 |
| 接口名称 | /User/deleteUser |
| 接口描述 | 删除用户 |
| 验证方式 | Token |
| http方式 | Delete |

### 2.5.2. 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| Id | Int | Y | 用户id |
| Username | String | Y | 用户名 |
| Password | String | Y | 用户密码 |
| Type | Int | Y | 用户类型 |
| … | … | Y | … |

### 2.5.3. 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| Id | Int | Y | 用户Id |
| Username | String | Y | 用户名 |
| Password | String | Y | 用户密码 |
| Type | Int | Y | 用户类型 |
| … | … | Y | … |

### 2.5.4. 代码示例

request请求:

{

     “Id”：1，

     "Username":"杨凯",

"Password":"123456",

“Type”:1

…

}

Responce: 响应:

(1)error

{

   "status":0,

   "error\_code":400，

}

(2)succed

{

“Code”:200,

   “Id”：1，

     "Username":"杨凯",

"Password":"123456",

“Type”:1

…

   "status":1

}

### 2.6.1. 用户修改接口

|  |  |
| --- | --- |
| 类别 | 含义 |
| 接口名称 | /User/updateUser |
| 接口描述 | 修改用户 |
| 验证方式 | Token |
| http方式 | Put |

### 2.6.2. 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| Id | Int | Y | 用户id |
| Username | String | Y | 用户名 |
| Password | String | Y | 用户密码 |
| Type | Int | Y | 用户类型 |
| … | … | Y | … |

### 2.6.3. 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| Id | Int | Y | 用户Id |
| Username | String | Y | 用户名 |
| Password | String | Y | 用户密码 |
| Type | Int | Y | 用户类型 |
| … | … | Y | … |

### 2.6.4. 代码示例

request请求:

{

     “Id”：1，

     "Username":"杨凯",

"Password":"123456",

“Type”:1

…

}

  Responce: 响应:

(1)error

{

   "status":0,

   "error\_code":400，

}

(2)succed

{

“Code”:200,

   “Id”：1，

     "Username":"杨凯",

"Password":"123456",

“Type”:1

…

    "status":1

}

### 2.7.1.增加公告接口

|  |  |
| --- | --- |
| 类别 | 含义 |
| 接口名称 | /Note/addNote |
| 接口描述 | 增加公告 |
| 验证方式 | Token |
| http方式 | Post |

### 2.7.2. 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| Id | Int | Y | 公告 |
| Notice | String | Y | 公告内容 |
| Type | Int | Y | 公告类型 |

### 2.7.3. 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| Sid | string | Y | SessionId(token) |
| Id | Int | Y | 公告 |

### 2.7.4. 代码示例

request请求:

{

     “Id”：1，

     "Notice“:“职称申报要求……”,

“Type”:1

}

  Responce: 响应:

(1)error

{

   "status":0,

   "error\_code":400，

}

(2)succed

{

“Code”:200,

“Sid“：“T2019110447“，

   “Id”：1，

    "status":1,

}

### 2.8.1. 用户删除接口

|  |  |
| --- | --- |
| 类别 | 含义 |
| 接口名称 | /Note/deleteNote |
| 接口描述 | 删除公告 |
| 验证方式 | Token |
| http方式 | Delete |

### 2.8.2. 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| Id | Int | Y | 用户id |
| Notice | String | Y | 公告内容 |
| Type | Int | Y | 公告类型 |

### 2.8.3. 响应参数

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 参数名 | 类型 | | 是否必须 | | 描述 | |
| Id | | Int | | Y | | 用户id |
| Notice | | String | | Y | | 公告内容 |
| Type | | Int | | Y | | 公告类型 |

### 2.8.4. 代码示例

request请求:

{

     “Id”：1，

     "Notice“:“职称申报要求……”,

“Type”:1

}

Responce: 响应:

(1)error

{

   "status":0,

   "error\_code":400，

}

(2)succed

{

“Code”:200,

   “Id”：1，

     "Notice“:“职称申报要求……”,

“Type”:1

"status":1

}

### 2.9.1.公告信息接口

|  |  |
| --- | --- |
| 类别 | 含义 |
| 接口名称 | /Note/Information |
| 接口描述 | 公告信息 |
| 验证方式 | Token |
| http方式 | Get |

### 2.9.2. 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| Id | Int | Y | 公告id |

### 2.9.3. 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必须 | 描述 |
| Id | Int | Y | 公告id |
| Data | Json | Y | 公告信息 |

### 2.9.4. 代码示例

request请求:

{

     “Id”：1，

}

  Responce: 响应:

(1)error

{

   "status":0,

   "error\_code":400，

}

(2)succed

{

“Code”:200,

  “ data”: {

message: "success",

list: [ {id: 1,

Username: "XXX",

code: "XXX" },

{id: 2,

Username: "XXX",

code: "XXX" } ] }}

   "status":1

}