通用规则

（一）调用格式

HTTP GET {base-url}/api/{controller}/{action}/{id}

或

HTTP GET {base-url}/api/{controller}/{action}?{query-strings}

HTTP POST {base-url}/api/{controller}/{action}

POST-HEADER: contentType=application/json;charset=utf-8

POST-BODY: json(T) //T是需要POST给API的业务对象类型

T是业务对象，例如Project，User，Employee等。

（二）响应格式

无论是GET还是POST，返回类型都是ApiResult<T>的JSON，T是请求的业务对象

ApiResult<T>的定义如下：

{

“isSuccess” : true, //或false （仅在returnCode=0时返回true）

“returnCode : 0, //0: Success, -1: Exception

“message” : “”, //异常描述

“dataJson” : “{*json(T)*}” //正确返回的T类型业务对象的json串

}

**对于T的附加说明：**

如果返回的业务对象是一个记录数组的话，一般来说，会把结果进行封装后，返回一个PagedQuery<T>的业务对象，其定义如下：

{

“Total” : 20, //表示数据库中一共有符合条件的记录20条。

“Records” : [T1, T2, …] //T类型的数组

}

其中，Records数组的长度并非一定是Total的大小，这取决于查询API是否要求了分页，如果要求了分页，那么Records的长度是当前页包括的记录数量。

**对于响应处理的附加说明：**

客户端调用API的时候，服务器返回一个JSON，这个JSON可以反序列化为一个ApiResult对象，客户端应该进行如下处理：

var result := json.Deserialized(response\_json);

if(result.isSuccess == true)

{

var myObject := json.Deserialized(result.dataJson);

//… 后面就可以使用myObject了

}

（三）文档说明

1. 本文档按Controller分章节，章节标题即是{controller}；
2. 每一章节下面按Action分段，段落标题即是{action}或{action}/{id}，标题为{id}表示无需提供Action；
3. 每一段落下面是一个表格，表格内容说明了API的输入参数、返回值和注意事项；
4. 表格里的返回值里的业务类型指的是ApiResult<T>里面的T类型。
5. {base-url}根据应用部署的情况确定，不在本文档的描述范围之内。

# User

## GetUsers

|  |  |  |  |
| --- | --- | --- | --- |
| 路径模板 | /api/User/GetUsers | | |
| 控制器 | User | 操作 | GetUsers |
| 功能说明 | 返回系统中的所有用户 | | |
| HTTP Method | ■GET  □POST | 参数传递方法 | ■NA  □Route  □Query String  □Body |
| 输入参数列表 | 名称 | 类型 | Json格式 |
| 无 |  |  |
| 返回值 | 业务类型 | 内容说明 | |
| PagedQuery<User> |  | |
| 注意事项 |  | | |
| 调用示例 |  | | |

## GetUserById/{id}

|  |  |  |  |
| --- | --- | --- | --- |
| 路径模板 | /api/User/GetUserById/{id} | | |
| 控制器 | User | 操作 | GetUserById |
| 功能说明 | 根据用户的UserID返回系统中的用户 | | |
| HTTP Method | ■GET  □POST | 参数传递方法 | □NA  ■Route  □Query String  □Body |
| 输入参数列表 | 名称 | 类型 | Json格式 |
| id | int |  |
| 返回值 | 业务类型 | 内容说明 | |
| User | 如果没有找到用户，返回null | |
| 注意事项 |  | | |
| 调用示例 | http:/localhost:5000/api/User/GetUserById/5  返回一个ApiResult<User>的JSON，其中User.UserID=5. | | |

## Save

|  |  |  |  |
| --- | --- | --- | --- |
| 路径模板 | /api/User/Save | | |
| 控制器 | User | 操作 | Save |
| 功能说明 | 将POST到服务器的User对象保存到数据库，如果User.UserID<=0，新增User，否则修改User。 | | |
| HTTP Method | □GET  ■POST | 参数传递方法 | □NA  □Route  □Query String  ■Body |
| 输入参数列表 | 名称 | 类型 | Json格式 |
| User | User | {  “userID”: 0,  “login”: “admin”,  “name”: “管理员”,  “role”: 1  “activeStatus”: true  } |
| 返回值 | 业务类型 | 内容说明 | |
| int | 成功新增或修改后，返回UserID | |
| 注意事项 |  | | |
| 调用示例 | http:/localhost:5000/api/User/Save  POST：{user\_json}  返回一个ApiResult<int>的JSON，其中dataJson=json(id). | | |

## ChangeCredential

|  |  |  |  |
| --- | --- | --- | --- |
| 路径模板 | /api/User/ChangeCredential | | |
| 控制器 | User | 操作 | ChangeCredential |
| 功能说明 | 根据POST到服务器的ChangeCredential对象修改用户的登录凭据（即Password）。 | | |
| HTTP Method | □GET  ■POST | 参数传递方法 | □NA  □Route  □Query String  ■Body |
| 输入参数列表 | 名称 | 类型 | Json格式 |
| changeCredential | ChangeCredential | {  “userID”: 0,  “oldCredential”: “xxxxxxxxx”,  “newCredential”: “xxxxxxx”  } |
| 返回值 | 业务类型 | 内容说明 | |
| bool | 修改成功返回true，失败的话，  ApiResult.isSuccess=false,  ApiResult.message包含失败原因。 | |
| 注意事项 | oldCredential和newCredential都是根据AES算法加密的串。  具体加密算法可与API开发人员协商。 | | |
| 调用示例 | http:/localhost:5000/api/User/ChangeCredential  POST：{changeCredential\_json}  返回一个ApiResult<bool>的JSON，其中dataJson=json(true/false). | | |

## Authenticate

|  |  |  |  |
| --- | --- | --- | --- |
| 路径模板 | /api/User/Authenticate | | |
| 控制器 | User | 操作 | Authenticate |
| 功能说明 | 根据POST到服务器的ChangeCredential对象修改用户的登录凭据（即Password）。 | | |
| HTTP Method | □GET  ■POST | 参数传递方法 | □NA  □Route  □Query String  ■Body |
| 输入参数列表 | 名称 | 类型 | Json格式 |
| authenticate | Authenticate | {  “login”: “admin”,  “credential”: “xxxxxxxxx”  } |
| 返回值 | 业务类型 | 内容说明 | |
| User | 身份验证成功返回User对象，失败的话，  ApiResult.isSuccess=false,  ApiResult.message包含失败原因。 | |
| 注意事项 | Credential是根据AES算法加密的串。  具体加密算法可与API开发人员协商。 | | |
| 调用示例 | http:/localhost:5000/api/User/ Authenticate  POST：{authenticate\_json}  返回一个ApiResult<User>的JSON，其中dataJson=json(user). | | |

# Reference

## GetSpecialties

## GetQualifications

## GetDuties

## Save

## Delete

## Get

## {id}

# OrganizationController

## Save

## Get

## {id}

## Delete

# EmployeeController

## {id}

## InOrganization/{id}

## InProject/{id}

## Save

## Delete

# CertificationController

## Employee/{id}

## {id}

## Save

## Delete

# ProjectController

## Status/{id}

## {id}

## Save

## Delete

## Verify

## Transfer

## Inlog

## Outlog

## RequiredQualifications/{id}

## AddQualification

## RemoveQualification

api/Project/Status/1

api/Project/5

POST api/Project/Save json(Project)

POST api/Project/Delete json(id)

POST api/Project/Verify json({EmployeeId, ProjectId, DutyId})

return: ApiResult.ReturnCode = 0 /// OK

= -1 /// EXCEPTION

= 1 /// WARNING (ApiResult.Message 包含 警告项

POST api/Project/Transfer json({EmployeeId, ProjectId, DutyId})

return: ApiResult.ReturnCode = 0 /// OK

= -1 /// EXCEPTION

= 1 /// WARNING (ApiResult.Message 包含 警告项

api/Project/InLog?employeeId=&projectId=

api/Project/OutLog?employeeId=&projectId=

api/Project/RequiredQualifications/5

POST api/Project/AddQualification json({QualificationID, DutyID})

POST api/Project/RemoveQualification json({QualificationID, DutyID})