Contents

[1 Account 账户信息相关api 1](#_Toc51168897)

[1.1 用户登录 POST /Account/login 1](#_Toc51168898)

[1.2 用户注册 POST /Account/register 2](#_Toc51168899)

[1.3 获取id 用户的详细信息（个人信息页面） GET /Account/getUserInfo 3](#_Toc51168900)

[1.4 保存用户填写后详细信息 POST /Account/saveUserInfo 4](#_Toc51168901)

[1.5 获取用户个人图片主页信息(个人图片主页) GET /Account/getProfileInfo 5](#_Toc51168902)

[1.6获取用户个人图片主页图片信息(个人图片主页) GET //Account/getProfilePicture 6](#_Toc51168903)

[1.7 用户的关注和取消关注 followUser POST /Account/followUser 7](#_Toc51168904)

[2 Picture 图片相关api 8](#_Toc51168905)

[2.1 赞和取消赞 POST /Picture/likePicture 8](#_Toc51168906)

[2.2 收藏和取消收藏 POST /Picture/favoritePicture 8](#_Toc51168907)

[2.3 评论和在原评论上进行修改 POST /Picture/comment 9](#_Toc51168908)

[2.4 获取当前图片所有评论 GET /Picture/getAllComment 10](#_Toc51168909)

[2.5 获取用户图片页面相关信息 GET /Picture/getPicViewInfo 11](#_Toc51168910)

[2.6 上传图片 POST /Account/Upload 13](#_Toc51168911)

[2.7 下载图片 Download GET /Picture/Download 14](#_Toc51168912)

[2.8 根据tag搜索图片 GET /Picture/searchByTag 15](#_Toc51168913)

[2.9 获得三张图片信息 GET /Picture/get3Pic 17](#_Toc51168914)

目前的所有接口如果前端同学传送的数据比如用户id,name之类的，除了登陆注册，其他的接口没有做意外处理，所以尝试的时候要去看一看数据库里面是否有相关数据。

# 1 Account 账户信息相关api

## 1.1 用户登录 POST /Account/login

**介绍：**

通过参数用户名密码可以登录，如果登陆成功会返回Success:true 以及附带的Token令牌；如果登陆失败则返回Success：false

**参数** [Body] {

用户名 userName:string,

用户密码 userPassword:string

}

**返回数据：**

{

    "success": true,

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiI3MjhiOTEyYi1lYjUzLTQ0YmMtODU3Ny1hM2UyNzMwODJiNWEiLCJuYW1lIjoiMTIzMTIzIiwibmJmIjoxNTk5OTk2ODUzLCJleHAiOjE1OTk5OTc0NTMsImlzcyI6Imp3dElzc3VlcnRlc3QiLCJhdWQiOiJqd3RBdWRpZW5jZXRlc3QifQ.-wtmH9Dn8G\_vBfFkDzuqp8vO-i23OCC-A1f9l2oAe48",

    "type": "Bearer"

}

Success：操作成功状态

Token：token信息

Type：Token类型

**参考代码：**

var settings = {

  "url": "https://localhost:44362/Account/login",

  "method": "POST",

  "timeout": 0,

  "headers": {

    "Content-Type": "application/json"

  },

  "data": JSON.stringify({"userName":"123123","userPassword":"123123"}),

};

$.ajax(settings).done(function (response) {

  console.log(response);

});

## 1.2 用户注册 POST /Account/register

**介绍：**

通过参数用户名密码以及邮箱可以注册，如果没有完全一摸一样的用户名和密码，则会返回Success:true 以及附带的Token令牌；如果注册失败则返回Success：false

**参数**：[Body]{

用户名 userName : string,

邮箱 email : string,

用户密码 userPassword : string

}

**返回数据：**

{

    "success": true,

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiJkNzlkZDg0ZS1lYjJjLTRlZGEtYTliOS04NzM4MDBlN2ExN2MiLCJuYW1lIjoibmV3T25lIiwibmJmIjoxNTk5OTk3MTE1LCJleHAiOjE1OTk5OTc3MTUsImlzcyI6Imp3dElzc3VlcnRlc3QiLCJhdWQiOiJqd3RBdWRpZW5jZXRlc3QifQ.LVHUU25hbYJfXZMTUDrplb8qqm1aeOqtafcat78BPNM",

    "msg": "New User Registered"

}

Success：操作成功状态

Token：Token信息

Msg：返回的文本你信息

**参考代码：**

var settings = {

  "url": "https://localhost:44362/Account/register",

  "method": "POST",

  "timeout": 0,

  "headers": {

    "Content-Type": "application/json"

  },

  "data": JSON.stringify({"userName":"newOne","email":"tongji@gmail","userPassword":"123123"}),

};

$.ajax(settings).done(function (response) {

  console.log(response);

});

## 1.3 [Authorize]获取id 用户的详细信息（个人信息页面） GET /Account/getUserInfo

**介绍：**

根据参数用户Id 返回用户的基本信息以及详细信息，用于个人信息界面

**参数：**[Param] 用户id userId : string

**返回数据:**

{

    "success": true,

    "userName": "123321",

    "name": "123321",

    "birthday": "1800-01-01T00:00:00",

    "email": "123123123@qq.com",

    "message": ""

}

Success ：表示操作成功状态

userName ：用户名

name : 用户本名

birthday ：用户生日

email : 用户邮箱

message : 用户个性签名留言

**参考代码：**

var settings = {

  "url": "https://localhost:44362/Account/getUserInfo?userId=2",

  "method": "GET",

  "timeout": 0,

};

$.ajax(settings).done(function (response) {

  console.log(response);

});

## 1.4 [Authorize]保存用户填写后详细信息 POST /Account/saveUserInfo

**介绍：**

保存参数信息至数据库 用于个人信息页面的保存

**参数：**[Body]{

用户ID userId : string

用户名 userName :string

用户真名 name : string

生日 birthday : DateTime

邮箱 email : string

个人留言 message : string

}

**返回数据：**

{

    "success": true,

    "msg": "Operation Done"

}

Success : 操作成功状态

Msg : 返回信息

**参考代码：**

var settings = {

  "url": "https://localhost:44362/Account/saveUserInfo",

  "method": "POST",

  "timeout": 0,

  "headers": {

    "Content-Type": "application/json"

  },

  "data": JSON.stringify({"userId":"2","userName":"newOne","name":"Zhang 3","birthday":"2020-09-13T11:50:53.104Z","email":"tongji@gmail","message":"I am happy"}),

};

$.ajax(settings).done(function (response) {

  console.log(response);

});

## 1.5 [Authorize]获取用户个人图片主页信息(个人图片主页) GET /Account/getProfileInfo

**介绍：**

根据参数用户id 能够获取其个人图片主页所需信息，用于个人图片主页

**参数：**[Param] 用户id userId : string

**返回数据：**

{

    "success": true,

    "starNum": 0,

    "likeNum": 0,

    "commentNum": 1,

    "followNum": 0

}

Success : 操作成功状态

starNum : 作品被收藏数

likeNum : 作品被点赞数

commentNum : 个人评论数

followNum : 个人粉丝数量

**参考代码：**

var settings = {

  "url": "https://localhost:44362/Account/getProfileInfo?userId=2",

  "method": "GET",

  "timeout": 0,

};

$.ajax(settings).done(function (response) {

  console.log(response);

});

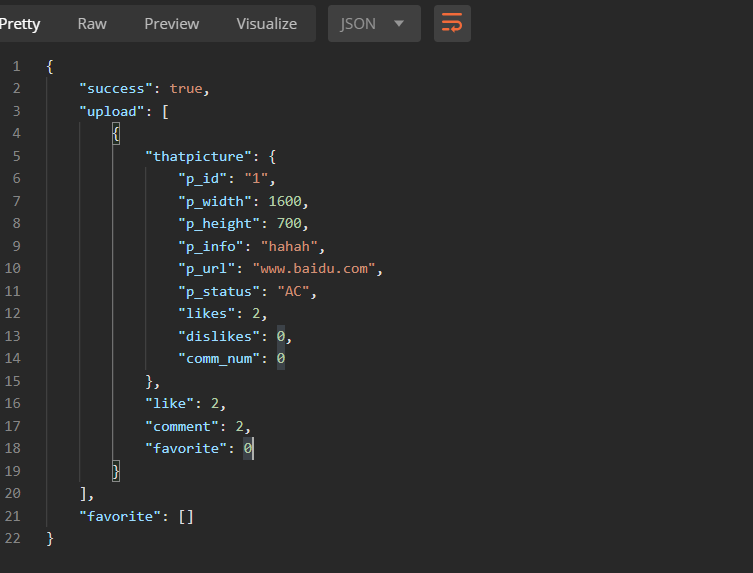
## 1.6[Authorize]获取用户个人图片主页图片信息(个人图片主页) GET /Account/getProfilePicture

**介绍：**返回王佳铭那个个人主页的图片列表。

**参数：**

用户id userId

**返回数据：**



操作是否成功。

上传图片列表，

收藏图片列表。

这里只显示了一个。

**参考代码：**



## 1.7 [Authorize]用户的关注和取消关注 followUser POST /Account/followUser

**介绍：**

通过该接口可以实现某用户对另一用户的关注，接口将关注和取消关注整合为一体：若已经关注那么使用接口后则取消关注，若没有关注则使用后关注用户

**参数：**[Param]

粉丝ID fansID ：string

被关注者ID followId : string

**返回数据：**

{

    "success": true,

    "msg": "Operation Done"

}

Success : 操作成功状态

Msg : 返回信息

**参考代码：**

var settings = {

  "url": "https://localhost:44362/Account/followUser?fansId=3&followId=1",

  "method": "POST",

  "timeout": 0,

};

$.ajax(settings).done(function (response) {

  console.log(response);

});

# 2 Picture 图片相关api

## 2.1 [Authorize]赞和取消赞 POST /Picture/likePicture

**介绍：**

通过该接口可以实现某用户对图片的点赞，接口将点赞和取消点赞整合为一体：若已经点赞那么使用接口后则取消点赞，若没有点赞则使用后点赞图片

**参数：**[Param]

用户ID userId ：string

图片ID picId : string

点赞类型 type: string {LK : 点赞 , DL ：点踩} ps由于没有做点踩功能因此在请求时填写LK

**返回数据：**

{

    "success": true,

    "msg": "Operation Done"

}

Success : 操作成功状态

Msg : 返回信息

**参考代码：**

var settings = {

  "url": "https://localhost:44362/Picture/likePicture?userId=2&picId=1&type=LK",

  "method": "POST",

  "timeout": 0,

};

$.ajax(settings).done(function (response) {

  console.log(response);

});

## 2.2 [Authorize]收藏和取消收藏 POST /Picture/favoritePicture

**介绍：**

通过该接口可以实现某用户对图片的收藏，接口将收藏和取消收藏整合为一体：若已经收藏那么使用接口后则取消收藏，若没有收藏则使用后收藏图片

**参数：**[Param]

用户ID userId ：string

图片ID picId : string

**返回数据：**

{

    "success": true,

    "msg": "Operation Done"

}

Success : 操作成功状态

Msg : 返回信息

**参考代码：**

var settings = {

  "url": "https://localhost:44362/Picture/favoritePicture?userId=2&picId=1 ",

  "method": "POST",

  "timeout": 0,

};

$.ajax(settings).done(function (response) {

  console.log(response);

});

## 2.3 [Authorize]评论和在原评论上进行修改 POST /Picture/comment

**介绍：**

该接口实现用户对图片进行评论，如果当前用户已经对该图片进行评论，则修改为新的评论，如果用户未进行评论，则创建新的评论记录。

**参数:** [Body]{

用户ID userId : string

图片ID picId : string

评论内容 content : string

}

**返回数据：**

{

    "success": true,

    "msg": "Operation Done"

}

Success : 操作成功状态

Msg : 返回信息

**参考代码：**

var settings = {

  "url": "https://localhost:44362/Picture/comment",

  "method": "POST",

  "timeout": 0,

  "headers": {

    "Content-Type": "application/json"

  },

  "data": JSON.stringify({"userId":"3","picId":"1","content":"ShenMeLaJi"}),

};

$.ajax(settings).done(function (response) {

  console.log(response);

});

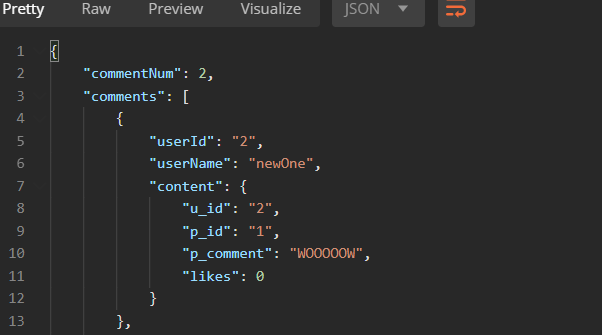
## 2.4 获取当前图片所有评论 GET /Picture/getAllComment

**介绍：**

通过参数获取当前图片所有的评论列表List

**参数：**[Param] 图片ID picId : string

**返回数据：**



返回数据为一个列表

commentNum : 评论总数

userId : 评论者ID

username:用户名

p\_comment:用户评论

likes : 评论点赞数

**参考代码：**

var settings = {

  "url": "https://localhost:44362/Picture/getAllComment?picId=1",

  "method": "GET",

  "timeout": 0,

  "headers": {

    "Content-Type": "application/json"

  },

  "data": JSON.stringify({"userId":"3","picId":"1","content":"ShenMeLaJi"}),

};

$.ajax(settings).done(function (response) {

  console.log(response);

});

## 2.5 获取用户图片页面相关信息 GET /Picture/getPicViewInfo

**介绍：**

该接口用于获得某用户访问某图片界面时基本的信息。

需要指定用户因为不同用户看到的评论和是否关注发布者是不一样的。

**参数：**[Param]

用户ID userID

图片ID picID

**返回数据：**



Success : 操作成功状态

PicUrl : 图片来源Url

publisherFollow : 是否关注图片发布者

publisherId : 图片发布者的ID

picLike ：是否点赞该图片

picStar ：是否收藏该图片

picHeight/picWidth ：图片长/宽

picInfo : 图片信息介绍

picTags ：图片包含Tag

hasdownload:是否观看者已经下载

uploadtime:上传时间

uploadName:上传作者的昵称

price:价格。

nowComment : 当前用户对该图片的评论

后面新增downloads下载量

**参考代码：**

var settings = {

  "url": "https://localhost:44362/Picture/getPicViewInfo?userId=3&picId=1",

  "method": "GET",

  "timeout": 0,

  "headers": {

    "Content-Type": "application/json"

  },

  "data": JSON.stringify({"userId":"3","picId":"1","content":"ShenMeLaJi"}),

};

$.ajax(settings).done(function (response) {

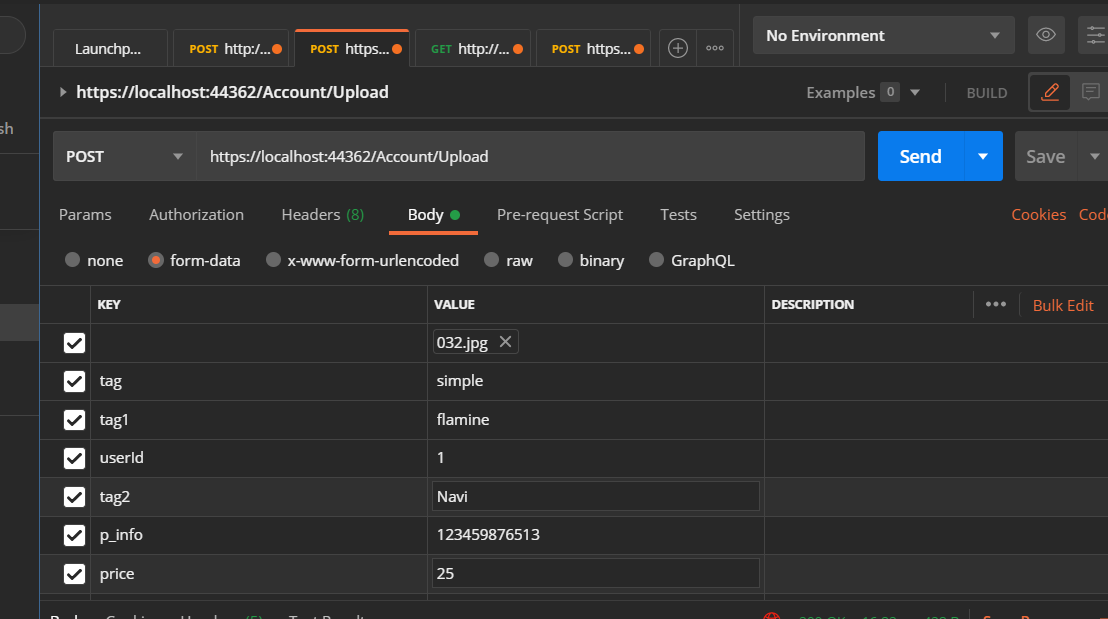
  console.log(response);

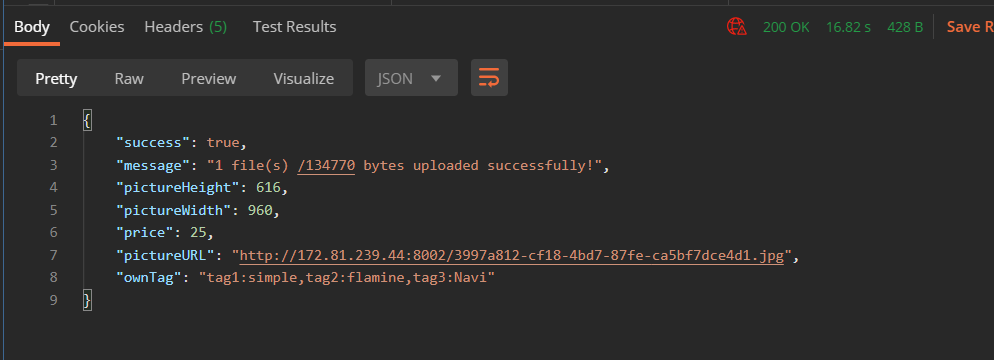
});

## 2.6 [Authorize]上传图片 POST /Account/Upload

**上传图片的时候附加用户输入的tag和图片的相关介绍。**

**大家在测试的时候不要上传高清图片，对于userId因为有外码约束，如果数据库里面没有你写的userId(可以去看看数据库)那么插入是不成功的，要确保userId是正确的是要前端的同学来保证。**





**参数：**

**图片文件**

**标签 [tag]**

**标签 [tag1]**

**标签 [tag2]**

**用户id [userId]**

**图片附加说明 [p\_info]**

**图片定价：price**

**返回值：**

**Success:操作是否成功**

**Message文件个数和文件大小**

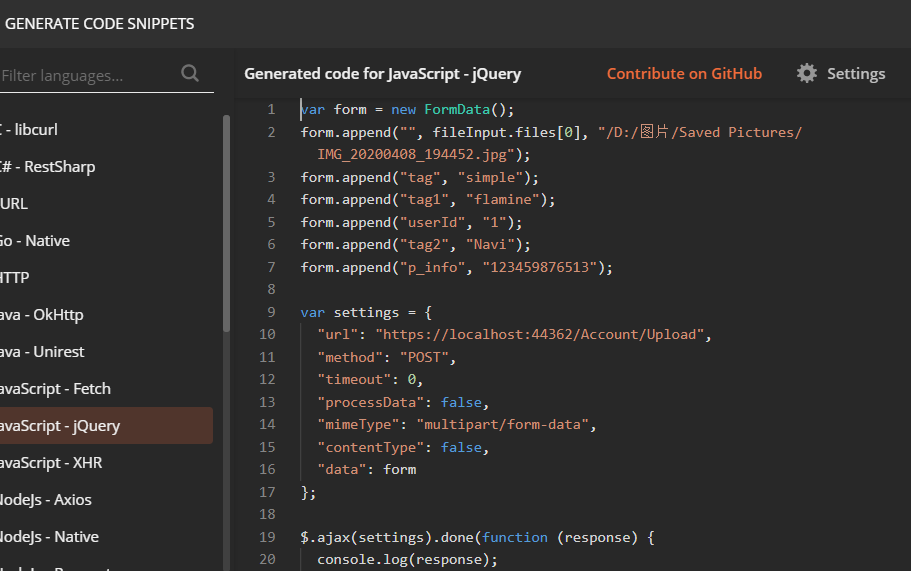
**pictureHeight:高**

**pictureWidth:宽**

**pictureURL：图片在服务器上的地址**

**ownTag:图片所有的tag.**

**参考代码：**



## 2.7 [Authorize]下载图片 Download GET /Picture/Download

**说明：通过传输参数 用户ID以及图片ID可以进行请求下载的操作，如果没有购买图片的记录，则进行购买图片，硬币数不足导致购买失败时不返回图片下载的url；如果是发布者下载或者钱包硬币能够购买图片，则消耗硬币返回图片下载的url，同时增加发布者的硬币数。**

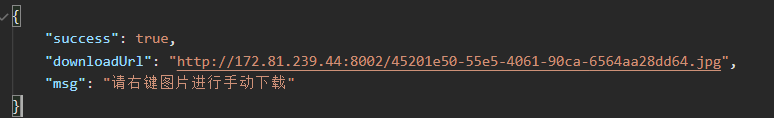
**参数：[Params]**

**用户ID userId ：string**

**图片ID picId ：string**

**返回数据**

**成功时：**

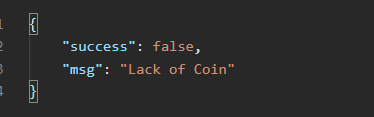


**Success 操作成功状态**

**DownloadUrl 下载图片url**

**Msg 返回信息**

**失败时**



**Success 操作成功状态**

**Msg 返回信息**

**参考代码：**

var settings = {

  "url": "https://localhost:44362/Picture/Download?userId=5&picId=34",

  "method": "GET",

  "timeout": 0,

  "headers": {

    "Authorization": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiIzZGI2NzIyOS0yOGNhLTQ3NDgtYjg1Ny0yNGM1NDZkNzFiMDQiLCJuYW1lIjoiMTIzMTIzIiwibmJmIjoxNjAwMTUyNTc3LCJleHAiOjE2MDAxNTMxNzcsImlzcyI6Imp3dElzc3VlcnRlc3QiLCJhdWQiOiJqd3RBdWRpZW5jZXRlc3QifQ.1WYqLq9vL9DQ-eWzAqQ5dFrkYId8RwSMR11FIqkOdhA"

  },

};

$.ajax(settings).done(function (response) {

  console.log(response);

});

## 2.8根据tag搜索图片 GET /Picture/searchByTag

**说明：根据参数tag返回所有的拥有该tag的图片信息**

**参数：[Params]**

**图片Tag tag : string**

**返回数据：**



**Success: 操作成功状态**

**returnList: 返回图片信息**

**picId: 图片ID**

**picUrl：图片url**

**publisherId: 发布者ID**

**publisherName: 发布者昵称**

**info: 图片信息**

**likeNum: 图片获赞数**

**starNum: 图片获收藏数**

**commNum：图片评论数**

**示例代码：**

var settings = {

  "url": "https://localhost:44362/Picture/searchByTag?tag=autumn",

  "method": "GET",

  "timeout": 0,

  "headers": {

    "Authorization": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiIzZGI2NzIyOS0yOGNhLTQ3NDgtYjg1Ny0yNGM1NDZkNzFiMDQiLCJuYW1lIjoiMTIzMTIzIiwibmJmIjoxNjAwMTUyNTc3LCJleHAiOjE2MDAxNTMxNzcsImlzcyI6Imp3dElzc3VlcnRlc3QiLCJhdWQiOiJqd3RBdWRpZW5jZXRlc3QifQ.1WYqLq9vL9DQ-eWzAqQ5dFrkYId8RwSMR11FIqkOdhA"

  },

};

$.ajax(settings).done(function (response) {

  console.log(response);

});

## 2.9 获得三张图片信息 GET /Picture/get3Pic

**说明：需要传已经取过的次数信息，然后以时间正序返回最新的不同三张图片**

**参数：[Params]**

**已请求次数 requestTimes ：int**

**返回数据：**

**Success: 操作成功状态**

**returnList: 返回图片信息**

**picId: 图片ID**

**picUrl：图片url**

**publisherId: 发布者ID**

**publisherName: 发布者昵称**

**info: 图片信息**

**likeNum: 图片获赞数**

**starNum: 图片获收藏数**

**commNum：图片评论数**

**示例代码：**

var settings = {

  "url": "https://localhost:44362/Picture/get3Pic?requestTimes=3",

  "method": "GET",

  "timeout": 0,

  "headers": {

    "Authorization": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiIzZGI2NzIyOS0yOGNhLTQ3NDgtYjg1Ny0yNGM1NDZkNzFiMDQiLCJuYW1lIjoiMTIzMTIzIiwibmJmIjoxNjAwMTUyNTc3LCJleHAiOjE2MDAxNTMxNzcsImlzcyI6Imp3dElzc3VlcnRlc3QiLCJhdWQiOiJqd3RBdWRpZW5jZXRlc3QifQ.1WYqLq9vL9DQ-eWzAqQ5dFrkYId8RwSMR11FIqkOdhA"

  },

};

$.ajax(settings).done(function (response) {

  console.log(response);

});

## 2.10 [Authorize]第一回合上传 POST /Account/Upload1

前端需要传递的参数：

用户id:userId

图片文件.

这几项以form表单的形式提交

后端返回形式

{

Success: **操作是否成功**

Tags(tag数组一共十个).

**pictureHeight:高**

**pictureWidth:宽**

**Message文件个数和文件大小**

**pictureURL：图片在服务器上的地址**

pictureId(这个图片的id).

}

## 2.11 [Authorize]第二回合上传 POST /Account/Upload2

前端需要传递的数据

用户id:userId

图片id:pictureId

图片简介:p\_info

图片定价:price

Tag数组/(三个tag、tag1、tag2):

同样还是以form表单形式提交。

**返回值：**

**Success:操作是否成功**

**ownTag:用户选择的三个tag.**

**定价：price**

**简介：p\_info**

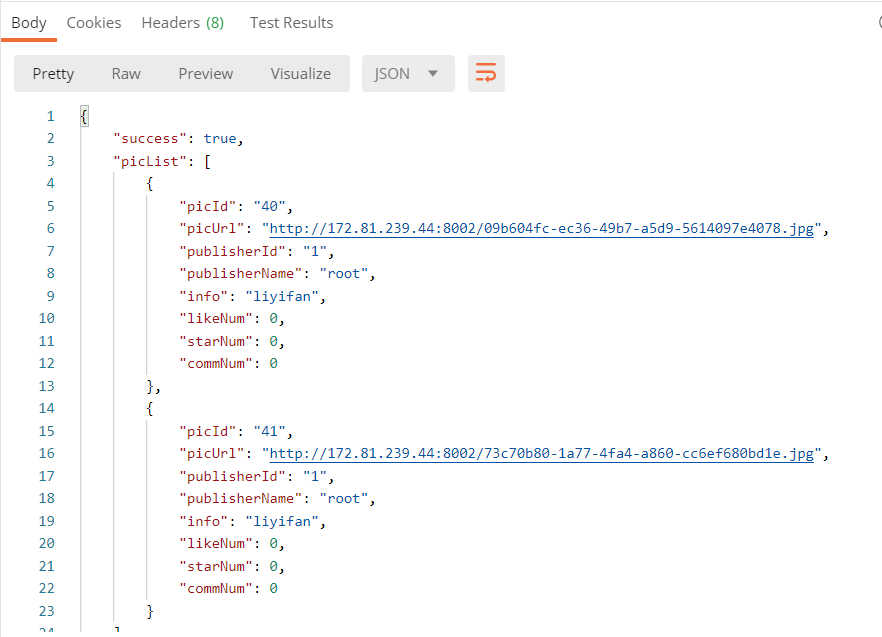
## 2.12 获得相似图片列表 POST /Account/SimilarPicture

**说明：该接口可根据传入图片获得数据库内匹配度最高的三个标签的全部图片**

**参数 Body form表单**

**图片文件**

**返回数据：**

****

**Success: 操作成功状态**

**Piclist:图片列表**

**picId：图片id**

**picUrl：图片url**

**publisherId：图片上传者id**

**publisherName：图片上传者名称**

**info：图片信息**

**likenum：喜欢人数**

**starnum：标注人数**

**commNum：评论数**

**参考代码：**

var form = new FormData();

form.append("", fileInput.files[0], "/D:/Entertainment/Entertainment/MechineLearning/Data/train/beach/084.jpg");

var settings = {

  "url": "http://172.81.239.44/Account/SimilarPicture",

  "method": "POST",

  "timeout": 0,

  "processData": false,

  "mimeType": "multipart/form-data",

  "contentType": false,

  "data": form

"headers": {

    "Authorization": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiIzZGI2NzIyOS0yOGNhLTQ3NDgtYjg1Ny0yNGM1NDZkNzFiMDQiLCJuYW1lIjoiMTIzMTIzIiwibmJmIjoxNjAwMTUyNTc3LCJleHAiOjE2MDAxNTMxNzcsImlzcyI6Imp3dElzc3VlcnRlc3QiLCJhdWQiOiJqd3RBdWRpZW5jZXRlc3QifQ.1WYqLq9vL9DQ-eWzAqQ5dFrkYId8RwSMR11FIqkOdhA"

  },

};

$.ajax(settings).done(function (response) {

  console.log(response);

});

# 3 Blog 博客相关api

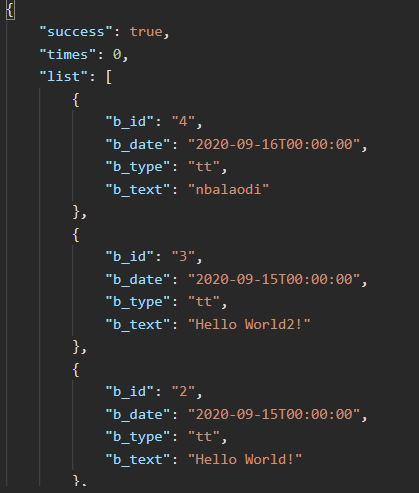
## 3.1 获取十个（不满十个则返回小于十） GET /Blog/getTen

**说明：该接口可根据参数获得对应参数的最新十个博客信息并返回，如times=0则是最新的第1-10条，如times=1则是最新的第11-20条……依次类推，如果剩余没有十条，则返回小于十条博客信息**

**参数 [Params]**

**请求的第几次 times : int (如果第一次请求，times=0)**

**返回数据：**



**Success: 操作成功状态**

**Times : 当前是第几次请求，下一次请求请+1**

**b\_id : 博客ID**

**b\_date: 博客编写日期**

**b\_type : 博客类型 tt表示文本**

**b\_text : 博客内容**

**参考代码：**

var settings = {

  "url": "https://localhost:44362/Blog/getTen?times=0",

  "method": "GET",

  "timeout": 0,

  "headers": {

    "Authorization": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiIzZGI2NzIyOS0yOGNhLTQ3NDgtYjg1Ny0yNGM1NDZkNzFiMDQiLCJuYW1lIjoiMTIzMTIzIiwibmJmIjoxNjAwMTUyNTc3LCJleHAiOjE2MDAxNTMxNzcsImlzcyI6Imp3dElzc3VlcnRlc3QiLCJhdWQiOiJqd3RBdWRpZW5jZXRlc3QifQ.1WYqLq9vL9DQ-eWzAqQ5dFrkYId8RwSMR11FIqkOdhA"

  },

};

$.ajax(settings).done(function (response) {

  console.log(response);

});

## 3.2 [Authorize]编写博客 POST /Blog/writeBlog

**说明：通过该接口可实现用户写博客操作**

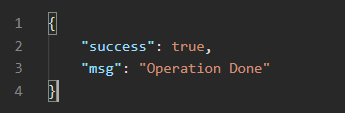
**参数：[Body]{**

**用户ID userId : string**

**博客内容 content : string**

**}**

**返回数据：**



**Success: 操作成功状态**

**Msg : 返回消息**

**参考代码：**

var settings = {

  "url": "http://172.81.239.44/Blog/writeBlog",

  "method": "POST",

  "timeout": 0,

  "headers": {

    "Content-Type": "application/json"

  },

  "data": JSON.stringify({"userId":"2","content":"DUckDUck"}),

};

$.ajax(settings).done(function (response) {

  console.log(response);

});

# 4 Wallet 钱包相关api

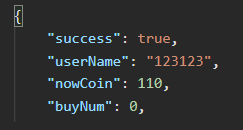
## 4.1 [Authorize]获取钱包信息 GET /Wallet/getWalletInfo

**说明：通过参数userId 可以获取响应用户的钱包帐户信息**

**参数：[Params]**

**用户ID userId : string**

**返回数据：**



**Success: 操作成功状态**

**username : 用户名**

**nowCoin : 用户硬币数**

**buyNum : 用户购买图片数**

**参考代码：**

var settings = {

  "url": "http://172.81.239.44/Wallet/getWalletInfo?userId=1",

  "method": "GET",

  "timeout": 0,

  "headers": {

    "Content-Type": "application/json"

  },

  "data": JSON.stringify({"userId":"2","content":"DUckDUck"}),

};

$.ajax(settings).done(function (response) {

  console.log(response);

});

## 4.2 [Authorize]向钱包进行充值 POST /Wallet/depositWallet

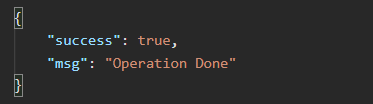
**说明：通过参数向对应用户钱包充值相应硬币数**

**参数：[Params]**

**用户ID userId : string**

**金额 amount : int**

**返回数据：**



**Success: 操作成功状态**

**Msg : 返回消息**

**参考代码：**

var settings = {

  "url": "http://172.81.239.44/Wallet/depositWallet?userId=4&amount=20",

  "method": "POST",

  "timeout": 0,

  "headers": {

    "Content-Type": "application/json"

  },

  "data": JSON.stringify({"userId":"2","content":"DUckDUck"}),

};

$.ajax(settings).done(function (response) {

  console.log(response);

});