Contents

[1 Account 账户信息相关api 2](#_Toc50922671)

[1.1 用户登录 POST /Account/login 2](#_Toc50922672)

[1.2 用户注册 POST /Account/register 3](#_Toc50922673)

[1.3 获取id 用户的详细信息（个人信息页面） GET /Account/getUserInfo 4](#_Toc50922674)

[1.4 保存用户填写后详细信息 POST /Account/saveUserInfo 4](#_Toc50922675)

[1.5 获取用户个人图片主页信息(个人图片主页) GET /Account/getProfileInfo 5](#_Toc50922676)

[1.6 用户的关注和取消关注 followUser POST /Account/followUser 6](#_Toc50922677)

[2 Picture 图片相关api 7](#_Toc50922678)

[2.1 赞和取消赞 POST /Picture/likePicture 7](#_Toc50922679)

[2.2 收藏和取消收藏 POST /Picture/favoritePicture 8](#_Toc50922680)

[2.3 评论和在原评论上进行修改 POST /Picture/comment 8](#_Toc50922681)

[2.4 获取当前图片所有评论 GET /Picture/getAllComment 9](#_Toc50922682)

[2.5 获取用户图片页面相关信息 GET /Picture/getPicViewInfo 10](#_Toc50922683)

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 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 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 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 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 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 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 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：**

{

    "commentNum": 3,

    "comments": [

        {

            "u\_id": "1",

            "p\_id": "1",

            "p\_comment": "What a great image!I like it!",

            "likes": 0

        },

        {

            "u\_id": "2",

            "p\_id": "1",

            "p\_comment": "WOOOOOW",

            "likes": 0

        },

        {

            "u\_id": "3",

            "p\_id": "1",

            "p\_comment": "ShenMeLaJi",

            "likes": 0

        }

    ]

}

返回数据为一个列表

commentNum : 评论总数

u\_id : 评论者ID

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": true,

    "picUrl": "url",

    "publisherFollow": false,

    "publisherId": "3",

    "picLike": false,

    "picStar": false,

    "picHeight": 700,

    "picWidth": 1600,

    "picInfo": "hahah",

    "picTags": [],

    "nowComment": "ShenMeLaJi"

}

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

nowComment : 当前用户对该图片的评论

**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) {

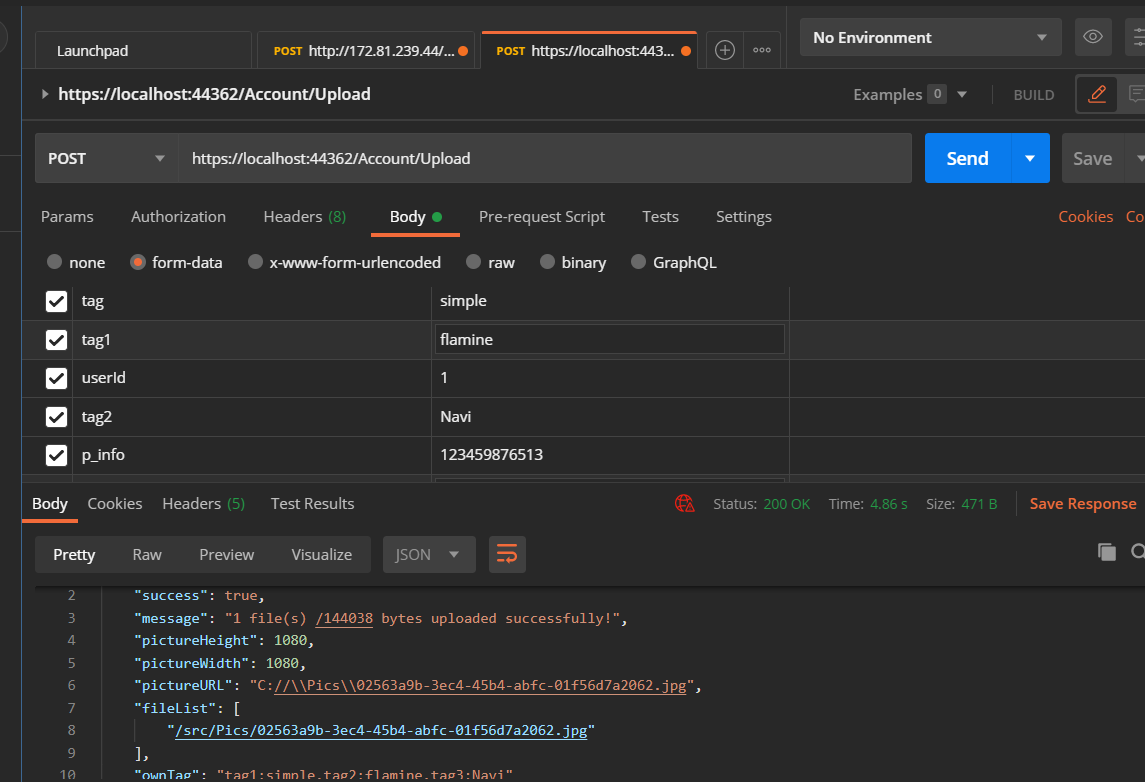
  console.log(response);

});

## 2.6 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]**

**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：**

