**朝花夕拾**

**接口文档**

版本号：1.0

院     系：       软件学院

专     业： 软件工程

小 组： h5项目小组朝花夕拾队

编 写： 高予蜀、周弘

小组成员： 高予蜀、周弘、牛玉欣、孙童

指导教师： 祁乐、刘冠军、王勇

目录

[朝花夕拾 - 前台接口文档 4](#_Toc28368808)

[接口更新 4](#_Toc28368809)

[登录 4](#_Toc28368810)

[我的-退出 4](#_Toc28368811)

[日记-新增/提交 5](#_Toc28368812)

[交换信息 5](#_Toc28368813)

[申请加好友 5](#_Toc28368814)

[选择日记申请交换 5](#_Toc28368815)

[请求图片 5](#_Toc28368816)

[注册 register 6](#_Toc28368817)

[发送验证码 6](#_Toc28368818)

[注册 6](#_Toc28368819)

[登录 login 7](#_Toc28368820)

[广场 square 8](#_Toc28368821)

[最新 8](#_Toc28368822)

[最热 10](#_Toc28368823)

[关注 12](#_Toc28368824)

[点赞状态更改 13](#_Toc28368825)

[评论 14](#_Toc28368826)

[个人主页（暂未实现主页自定义可见不可见） 14](#_Toc28368827)

[改变关注状态 16](#_Toc28368828)

[日记 diary 16](#_Toc28368829)

[一日一文 16](#_Toc28368830)

[我的日记列表 17](#_Toc28368831)

[日记详细信息 18](#_Toc28368832)

[新增日记（暂未实现加入图片功能） 20](#_Toc28368833)

[编辑日记（暂未实现加入图片功能） 21](#_Toc28368834)

[设置文章状态 22](#_Toc28368835)

[交换 change 22](#_Toc28368836)

[一句话列表 22](#_Toc28368837)

[获取一句话详细信息 23](#_Toc28368838)

[发布一句话 25](#_Toc28368839)

[一句话申请 26](#_Toc28368840)

[我的一句话 27](#_Toc28368841)

[匿名交换界面 changed 28](#_Toc28368842)

[交换信息 28](#_Toc28368843)

[申请加好友 30](#_Toc28368844)

[相遇信息 31](#_Toc28368845)

[选择日记界面 32](#_Toc28368846)

[选择日记申请交换 33](#_Toc28368847)

[回复好友申请 33](#_Toc28368848)

[我的 mine.js 34](#_Toc28368849)

[修改个人信息(用户名) 34](#_Toc28368850)

[获取我的tab页面信息 34](#_Toc28368851)

[退出 35](#_Toc28368852)

[好友/匿名 列表 chat 35](#_Toc28368853)

[匿名列表 35](#_Toc28368854)

[好友列表 36](#_Toc28368855)

[消息 news 38](#_Toc28368856)

[评论消息 38](#_Toc28368857)

[点赞消息 39](#_Toc28368858)

[好友申请（无用） 40](#_Toc28368859)

[交换申请 42](#_Toc28368860)

[交换申请回复 43](#_Toc28368861)

[我申请的交换 44](#_Toc28368862)

[图片 45](#_Toc28368863)

[请求图片 45](#_Toc28368864)

[朝花夕拾 - 后台接口文档 45](#_Toc28368865)

[登录 45](#_Toc28368866)

[首页 46](#_Toc28368867)

[系统管理 46](#_Toc28368868)

[用户管理 48](#_Toc28368869)

[日记管理 49](#_Toc28368870)

[获取日记 49](#_Toc28368871)

[删除日记 50](#_Toc28368872)

[评论管理 50](#_Toc28368873)

**朝花夕拾 - 前台接口文档**

**接口更新**

1.所有端口号为8666

2.所有端口不加zhaohuaxishi

举例：

原端口 http://116.62.14.0:8666/zhaohuaxishi/login

新端口 http://116.62.14.0:8666/user/login

3.以下端口是除了以上两条外需要特别关注更新的（可能有路径或传参的改变）

**登录**

**原接口** http://116.62.14.0:8666/zhaohuaxishi/login

**新接口** http://116.62.14.0:8666/user/login

**我的-退出**

**原接口** http://116.62.14.0:8666/zhaohuaxishi/mine/exit/:token

**新接口** http://116.62.14.0:8666/user/exit/:token

**日记-新增/提交**

**原接口** http://116.62.14.0:8666/zhaohuaxishi/diary/add

**新接口** http://116.62.14.0:8666/diary/edit

**交换信息**

**原接口** http://116.62.14.0:8666/zhaohuaxishi/changed/changedetail/:token/:oneid

**新接口** http://116.62.14.0:8666/changed/detail/:token/:oneid

**申请加好友**

**原接口** http://116.62.14.0:8666/zhaohuaxishi/changed/addFriend

**新接口** http://116.62.14.0:8666/changed/addfri

**选择日记申请交换**

**原接口** http://116.62.14.0:8666/zhaohuaxishi/changed/choose

**新接口** http://116.62.14.0:8666/changed/choose/:token

**请求图片**

**原接口** http://116.62.14.0:8666/zhaohuaxishi/image/:imgid

**新接口** http://116.62.14.0:8666/api/image/:imgid

**注册 register**

**发送验证码**

**后端：高予蜀**

前端 发送注册邮箱mail

后端 判断邮箱格式 相同邮箱 发送验证码到邮箱 返回发送结果

前端接收数据后 status为-1显示data中错误信息

post http://116.62.14.0:8666/sendcode

发送请求示例

fetch('http://116.62.14.0:8666/sendcode', {

method: 'POST',

headers: {

'Accept': 'application/json',

'Content-Type': 'application/json'

},

body: JSON.stringify({ mail:'1062945371@.com'})}).then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: -1, data: "邮箱格式错误！"}

{status: -1, data: "该邮箱已注册！"}

{status: 0, data: "验证码发送成功！"}

**注册**

**后端：高予蜀**

前端 发送注册邮箱mail 验证码code 密码password 重复密码repassword

后端 判断验证码 密码格式 两次密码 返回发送结果

前端接收数据后 status为-1显示data（后台已经设置好错误信息）status为0跳到登录页面

post http://116.62.14.0:8666/register

发送申请示例

fetch('http://116.62.14.0:8666/register', {

method: 'POST',

headers: {

'Accept': 'application/json',

'Content-Type': 'application/json'

},

body: JSON.stringify({ mail:'1062945371@qq.com',code:'357723', password:'gaoyushu11',repassword:'gaoyushu11'})}).then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: -1, data: "密码格式错误!"}

{status: -1, data: "两次密码不一致！"}

{status: 0, data: "注册成功！"}

**登录 login**

**后端：周弘**

前端 发送邮箱mail 密码password

后端 判断邮箱存在 判断密码 生成token

前端接收数据后 status为-1显示data中错误信息 status为0时前端保存token值 并进入程序

post http://116.62.14.0:8666/login

前端请求示例

fetch('http://116.62.14.0:8666/login', {

method: 'POST',

headers: {

'Accept': 'application/json',

'Content-Type': 'application/json'

},

body: JSON.stringify({ mail:'1062945371@qq.com', password:'gaoyushu11'})}).then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: -1, data: "邮箱或密码错误！"}

{status: 0, data: "登录成功！"}

**广场 square**

**最新**

**后端：周弘**

前端 发送get请求，携带参数token，收到返回值pstatus，通过判断pstatus来设置是否已点赞，若pstatus=true，表示已点赞，若pstatus=false，表示未点赞

后端 获取日记列表，按时间排序，获取点赞状态，判断praise表中当前用户是否对某文章的点赞过。

get http://116.62.14.0:8666/square/new/:token

返回数据示例

{"status":0,"data":[...]}

data为json数组 具体内容见下

"data":[{"uid":1,

"uname":"朝花夕拾",

"upassword":"gaoyushu11",

"uconcern":0,

"ufans":0,

"ucode":357723,

ufriend":0,

"uemail":"1062945371@qq.com",

"uintroduce":"暂无介绍",

"uimage":1,

ustatus":1,

"token":"12/11/2019, 5:14:40 PM",

"did":11,

"dtime":"2019/12/9 18:00",

"dtitle":"giao哥",

"dcontent":"一给窝里giaogiao",

"dtype":"public",

"dpraise":5,

"exchange":"false",

"dimage":null,

"comments":0,

"pstatus":"false"}

**最热**

**后端：周弘**

前端 发送get请求，携带参数token，收到返回值pstatus，通过判断pstatus来设置是否已点赞，若pstatus=true，表示已点赞，若pstatus=false，表示未点赞

后端 获取日记列表，按点赞数排序，获取点赞状态，判断praise表中当前用户是否对某文章的点赞过。

get http://116.62.14.0:8666/square/hot/:token

返回数据示例

{"status":0,"data":[...]}

data为json数组 具体内容见下

"data":[{"uid":1,

"uname":"朝花夕拾",

"upassword":"gaoyushu11",

"uconcern":0,

"ufans":0,

"ucode":357723,

ufriend":0,

"uemail":"1062945371@qq.com",

"uintroduce":"暂无介绍",

"uimage":1,

ustatus":1,

"token":"12/11/2019, 5:14:40 PM",

"did":11,

"dtime":"2019/12/9 18:00",

"dtitle":"giao哥",

"dcontent":"一给窝里giaogiao",

"dtype":"public",

"dpraise":5,

"exchange":"false",

"dimage":null,

"comments":0,

"pstatus":"false"}

**关注**

**后端：周弘**

前端 发送get请求，携带参数token，收到返回值pstatus，通过判断pstatus来设置是否已点赞，若pstatus=true，表示已点赞，若pstatus=false，表示未点赞

后端 获取当前用户所关注的人的日记列表，获取点赞状态，判断praise表中当前用户是否对某文章的点赞过。

get http://116.62.14.0:8666/square/follow/:token

返回数据示例

{"status":0,"data":[...]}

data为json数组 具体内容见下

"data":[{"uid":1,

"uname":"朝花夕拾",

"upassword":"gaoyushu11",

"uconcern":0,

"ufans":0,

"ucode":357723,

ufriend":0,

"uemail":"1062945371@qq.com",

"uintroduce":"暂无介绍",

"uimage":1,

ustatus":1,

"token":"12/11/2019, 5:14:40 PM",

"did":11,

"dtime":"2019/12/9 18:00",

"dtitle":"giao哥",

"dcontent":"一给窝里giaogiao",

"dtype":"public",

"dpraise":5,

"exchange":"false",

"dimage":null,

"comments":0,

"pstatus":"false"}

**点赞状态更改**

**后端：周弘**

前端 点赞后发送put请求，携带参数token和did，收到返回值pstatus，通过判断pstatus来设置是否已点赞，若pstatus=true，表示已点赞，若pstatus=false，表示未点赞

后端 更新点赞，先判断praise表中是否有当前用户对某文章的点赞记录，若没有则添加记录，pstatus默认为true，同时diary表中的dpraise加1，若有则将相应的pstatus设为false表示取消点赞 同时diary表中的dpriase减1.

put http://116.62.14.0:8666/square/praise/:token/:did

返回数据示例

{

"status":0,

"data":

{"pstatus":"true"}

}

**评论**

**后端：周弘**

前端 发送post请求，携带参数token

后端 添加评论

post http://116.62.14.0:8666/square/comments/:token

**个人主页（暂未实现主页自定义可见不可见）**

**后端：高予蜀**

前端 发送请求 携带参数token和当前用户id

后端 用户信息 公开日记信息 是否是自己 是否已经关注

get http://116.62.14.0:8666/square/userhome/:token/:userid

前端申请示例

fetch('http://116.62.14.0:8666/square/userhome/1/2').then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: 0, data: {…}}

data为一个对象 具体见下

uid: 2

uname: "朝花夕拾"

upassword: null

uconcern: 0

ufans: 0

ucode: 441054

ufriend: 0

uemail: null

uintroduce: "暂无介绍"

uimage: 1

ustatus: -1

token: null

isFollow: "myself" // 判断关注状态 string类型 有myself/false/true三个值

diary如果有数据为json数组 如果没有数据返回字符串

did: 2

uid: 2

dtime: "2019/12/8 17:13"

dtitle: "第1篇"

dcontent: "第1篇日记胡歌ivy我UN我尽量别弄 "

dtype: "public"

dpraise: 1

exchange: "false "

dimage: null

comments: 0

pstatus: true // 判断日记点赞状态 布尔值

**改变关注状态**

**后端：周弘**

前端 发送put请求，携带参数token

后端 更新关注状态，当前用户关注其他用户后，当前用户关注数加1，被关注用户粉丝数加1，取消关注后，当前用户关注数减1，被关注用户粉丝数减1

put http://116.62.14.0:8666/square/concerns/:token

无返回值

**日记 diary**

**一日一文**

**后端：高予蜀**

前端 发送请求

后端 返回最新的一日一文

get http://116.62.14.0:8666/diary/metto

前端申请示例

fetch('http://116.62.14.0:8666/diary/metto').then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: 0, data: {…}}

status:0

data:

author: "匿名用户"

mid: 2

mtext: "2019/12/8 17:13"

mtime: "一个人承担了很多东西

**我的日记列表**

**后端：高予蜀**

前端 发送请求

后台 返回json数组 时间由近到远

get http://116.62.14.0:8666/diary/list/:token

前端申请示例

fetch('http://116.62.14.0:8666/diary/list/1575896364524').then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: 0, data: Array(10)}

{status: -1, data: "获取日记列表失败！"}

data为一个json数组 每个json中有数据库diary表中的所有属性

**日记详细信息**

**后端：高予蜀**

前端 发送请求 url中携带diaryid

后端 返回相应的用户信息和日记信息

get http://116.62.14.0:8083/diary/detail/:diaryid/:token

前端请求示例

fetch('http://116.62.14.0:8083/zhaohuaxishi/diary/detail/3/3').then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: 0, data: {…}}

data为json 详情见下

comments为data.comments 详情见下

comcontent: "111111"

comid: 1

comstatus: "false"

comtime: "2"

did: 5

token: "12/11/2019, 5:14:40 PM"

ucode: 357723

uconcern: 0

uemail: "1062945371@qq.com"

ufans: 0

ufriend: 0

uid: 1

uimage: 1

uintroduce: "暂无介绍"

uname: "朝花夕拾"

upassword: "gaoyushu11"

ustatus: 1

以下均为data的属性

dcontent: "第4篇日记"

did: 5

dimage: null

dpraise: 0

dtime: "2019/12/6 17:13"

dtitle: "第4篇"

dtype: "private"

exchange: "false "

isGood: false

isMe: true

token: "1"

ucode: 441054

uconcern: 0

uemail: "106294537@qq.com"

ufans: 0

ufriend: 0

uid: 2

uimage: 1

uintroduce: "暂无介绍"

uname: "朝花夕拾"

upassword: null

ustatus: -1

**新增日记（暂未实现加入图片功能）**

**后端：高予蜀**

实际上是前端get再post 此处只写post 标题title 内容content token值

post http://116.62.14.0:8666/diary/add

发送申请示例

fetch('http://116.62.14.0:8666/diary/add', {

method: 'POST',

headers: {

method: 'POST',

'Content-Type': 'application/json'

},

body: JSON.stringify({token:1, title:'nihao', content:'nihao edit'})}).then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: 0, data: 12}

data: 12 // 日记id

status: 0

**编辑日记（暂未实现加入图片功能）**

**后端：高予蜀**

实际上是前端get再post 此处只写post

put http://116.62.14.0:8666/diary/update

发送申请示例

fetch('http://116.62.14.0:8666/diary/update', {

method: 'PUT',

headers: {

'Accept': 'application/json',

'Content-Type': 'application/json'

},

body: JSON.stringify({ diaryid:12,title:'nihao', content:'nihao edit'})}).then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: 0, data: "编辑日记成功"}

**设置文章状态**

**后端：高予蜀**

点击前端按钮 可在两个状态间切换 public/private

前端 发送token 文章id

后端 返回status为-1前端显示错误信息 status为0按钮变色

put http://116.62.14.0:8666/diary/type/:diaryid

发送申请示例

fetch('http://116.62.14.0:8666/diary/type/7', {

method: 'PUT',

headers: {

'Accept': 'application/json',

'Content-Type': 'application/json'

},

body: JSON.stringify({ token:175896364524, diaryid:7})}).then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: 0, data: "更改日记成功"}

**交换 change**

**一句话列表**

**后端：高予蜀**

前端 发送请求

后端 时间倒序 返回一句话信息 json数组

get http://116.62.14.0:8666/change/list

前端请求示例

fetch('http://116.62.14.0:8799/zhaohuaxishi/change/list').then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: 0, data: Array(4)}

data为json数组 每个json的键值对见下

aid: null

ddid: null

did: null

dnum: 0

dstatus: "true"

dtime: null

longdes: "3333333"

shortdes: "333"

shortdes\_id: 3

uid: 2

**获取一句话详细信息**

**后端：高予蜀**

前端 发送请求 url携带一句话id oneid

后端 判断自己是否申请 返回一句话信息json

get http://116.64.14.0:8666/change/detail/:oneid/:token

前端申请示例

fetch('http://116.62.14.0:8083/zhaohuaxishi/change/detail/3/3').then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: 0, data: {…}}

data为json 具体见下

aid: null

ddid: null

did: null

dnum: 0

dstatus: "true"

dtime: null

isRequest: false

isMe:false

longdes: "3333333"

shortdes: "333"

shortdes\_id: 3

token: "1"

ucode: 441054

uconcern: 0

uemail: "106294537@qq.com"

ufans: 0

ufriend: 0

uid: 2

uimage: 1

uintroduce: "暂无介绍"

uname: "朝花夕拾"

upassword: null

ustatus: -1

**发布一句话**

**后端：高予蜀**

前端 发送post请求 token值token 一句话标题title 详情content 交换日记diaryid

后端 判断用户 获取时间和id 插入数据库 日记变为返回发布的一句话的id

前端收到数据后 若status值为-1 显示data中错误信息 若status为0 跳转到一句话详情页

post http://116.64.14.0:8666/change/add

前端请求示例

fetch('http://116.62.14.0:8799/zhaohuaxishi/change/add', {

method: 'POST',

headers: {

'Accept': 'application/json',

'Content-Type': 'application/json'

},

body: JSON.stringify({token:'1',title:'test',content:'detail',diaryid:5})}).then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: 0, data: 7}

data: 7 // 相应的一句话的id

status: 0

**一句话申请**

**后端：高予蜀**

前端 发送post请求 token值为token 一句话id oneid 申请理由 content 选择日记id diaryid

后端 判断用户 获取时间和id 插入apply表 改变一句话申请数量

前端收到数据后 若status值为-1 显示data中错误信息 停留在申请页 若status为0 跳转到一句话详情页

post http://116.64.14.0:8666/change/request

前端申请示例

fetch('http://116.62.14.0:8799/zhaohuaxishi/change/request', {

method: 'POST',

headers: {

'Accept': 'application/json',

'Content-Type': 'application/json'

},

body: JSON.stringify({token:'1',title:'test',content:'detail',oneid:2})}).then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: 1, data: "申请成功！"}

**我的一句话**

**后端：高予蜀**

前端 发送请求 url携带token

后端 判断用户 获取我的一句话 时间倒序 返回json数组 数组中0为未关闭 1为已关闭

get http://116.64.14.0:8666/change/mine/:token

前端请求示例

fetch('http://116.62.14.0:8799/zhaohuaxishi/change/mine/1').then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: 0, data: {…}}

data为json close和open有数据时为数组 没有数据时为“为空”信息

close: "已经关闭的一句话为空！" // 没有相应数据的情况

open: Array(4) // 有数据为数组 数组内容见下

aid: null

ddid: null

did: null

dnum: 0

dstatus: "true"

dtime: null

longdes: "3333333"

shortdes: "333"

shortdes\_id: 3

uid: 2

**匿名交换界面 changed**

**交换信息**

**后端：高予蜀**

前端 发送请求 url携带token 一句话id oneid

后端 返回交换日记信息 type isMe isRead title etime 加好友请求 type isMe isReply（true false）reply content atime（根据oneid找所有exchange 和apply-friend 循环插入chat 如果是自己isMe=true type均为exchange 循环判断好友时间插入chat）

前端收到数据后 直接map 如果type为exchange 则交换框 如果type为friend 则好友框

get http://116.62.14.0:8666/changed/changedetail/:token/:oneid

前端申请示例

fetch('http://116.62.14.0:8666/changed/changedetail/1/2').then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: 0, data: Array(5)}

data: Array(5)

0:

diaryid: 5

eid: 2

isMe: "true"

isRead: "true"

time: "12/12/2019, 7:33:28 PM"

title: "第4篇"

type: "exchange"

\_\_proto\_\_: Object

1: {type: "exchange", isRead: "true", diaryid: 5, title: "第4篇", time: "12/12/2019, 7:36:49 PM", …}

2: {type: "exchange", isRead: "true", diaryid: 5, title: "第4篇", time: "12/12/2019, 7:39:04 PM", …}

3: {aid: 9, type: "friend", isReply: false, content: "nihao", time: "12/12/2019, 9:44:50 PM", …}

4:

aid: 1

content: "111111"

isMe: "false"

isReply: false //reply:true/false

time: "5"

type: "friend"

**申请加好友**

**后端：高予蜀**

前端 发送请求 token 一句话id oneid

后端 判断自己 是否已经申请加好友 apply插入 一句话状态request

post http://116.62.14.0:8666/changed/addFriend

前端申请示例

fetch('http://116.62.14.0:8666/changed/addFriend', {

method: 'POST',

headers: {

'Accept': 'application/json',

'Content-Type':'application/json'

},body: JSON.stringify({token:'6',oneid:1,text:'nihao'})}).then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: 0, data: "申请成功！"}

{status: -1, data: "您已经申请过加好友！请勿重复申请！"}

**相遇信息**

**后端：高予蜀**

前端 发送请求 一句话id oneid

后端 一句话表和申请表找一句话详情和申请消息

get http://116.62.14.0:8666/changed/meet/:token/:oneid

前端申请示例

fetch('http://116.62.14.0:8666/changed/meet/1/2').then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: 0, data: {…}}

data:

acontent: "detail"

aid: 8

astatus: "0"

atime: "12/12/2019, 9:13:12 AM"

atype: "exchange"

ddid: null

did: null

dnum: 0

dstatus: "false"

dtime: null

longdes: "5555555"

shortdes: "555"

shortdes\_id: 2

uid: 2

**选择日记界面**

**后端：高予蜀**

前端 发送请求 url携带token

后端 判断自己 返回private 时间倒序日记列表

get http://116.62.14.0:8666/changed/choose/:token

前端申请示例

fetch('http://116.62.14.0:8666/changed/choose/1').then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: -1, data: "您没有可选日记"}

{status: 0, data: Array(8)}

data为json数组 每个json具体见下

did: 3

uid: 2

dtime: "2019/12/9 17:13"

dtitle: "第2篇"

dcontent: "第2篇日记"

dtype: "private"

dpraise: 2

exchange: "false "

dimage: null

comments: 0

**选择日记申请交换**

**后端：高予蜀**

前端 发送申请 token 一句话id oneid 日记id diaryid

后端 判断自己 判断解锁estatus为true 插入申请表exchange

post http://116.62.14.0:8666/changed/choose

前端申请示例

返回数据申请

**回复好友申请**

**后端：高予蜀**

前端 发送申请 url携带token 申请id aid 用户态度 reply 同意为1 不同意为-1

后端 根据token和id apply 和describe变状态 添加好友列表

put http://116.62.14.0:8666/changed/replyFriend

前台申请示例

返回数据示例返回数据示例

**我的 mine.js**

**修改个人信息(用户名)**

**后端：周弘**

前端 发送put请求，body携带修改后的用户名、token、uintroduce

后端 更新users表中的当前用户的用户名

put http://116.62.14.0:8666/mine/name/

返回数据示例

data:'ok'

**获取我的tab页面信息**

**后端：周弘**

前端 发送get请求，携带参数token。

后端 获取用户名、个人介绍、邮箱、获取用户关注数、获取用户粉丝数、获取用户好友数

get http://116.62.14.0:8666/mine/:token

返回数据示例

{"status":0,

"dataone":{"uname":"用户1","uintroduce":"ok","uemail":"mail"},

"datatwo":{"count":"0"},

"datathree":{"count":"2"},

"datafour":{"count":"2"}

}

**退出**

**后端：高予蜀**

前端 发送请求 携带token

后端 更改ustatus为2

get http://116.62.14.0:8666/mine/exit/:token

前端申请示例

fetch('http://116.62.14.0:8666/mine/exit/3').then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: -1, data: "退出失败！"}

{status: 0, data: "退出成功！"}

**好友/匿名 列表 chat**

**匿名列表**

**后端：高予蜀**

前端 发送请求 url携带token

后端 判断token 根据一句话表得到交换方 json数组

get http://116.62.14.0:8666/chat/exchange/:token

前端申请示例

fetch('http://116.62.14.0:8666/chat/exchange/3').then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: -1, data: "匿名列表为空或获取好友失败，后端错误，请联系管理员！"}

{status: 0, data: Array(2)}

data为json数组 具体键值对见下

dstatus: "false"

duid: 2

longdes: "1234567"

shortdes: "111"

shortdes\_id: 1

uid: 3

**好友列表**

**后端：高予蜀**

前端 发送请求 url携带token

后端 判断token 根据好友表获得好友 好友id为fuid 自己id为uid

get http://116.62.14.0:8666/chat/friends/:token

前端申请示例

fetch('http://116.62.14.0:8666/chat/friends/3').then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: -1, data: "好友列表为空或获取好友失败，后端错误，请联系管理员！"}

{status: 0, data: Array(2)}

data为json数组 具体键值对见下

fuid: 4

token: "4"

ucode: 0

uconcern: 5

uemail: "mail4"

ufans: 2

ufriend: 2

uid: 3

uimage: 1

uintroduce: "ok"

uname: "用户2"

upassword: "1"

ustatus: 1

**消息 news**

**评论消息**

**后端：高予蜀**

前端 发送请求 url携带token

后端 判断token 按时间排序 评论人的信息 判断匿名者 日记信息

get http://116.62.14.0:8666/news/comment/:token

前端申请示例

fetch('http://116.62.14.0:8666/news/comment/1').then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: -1, data: "您没有评论消息！"}

{status: 0, data: Array(2)}

data为json数组 每个json的键值对见下

0:

comcontent: "333333"

comid: 4

comments: 0

comstatus: "false"

comtime: "5"

dcontent: "第6篇日记"

did: 7

dimage: null

dpraise: 0

dtime: "2019/12/4 17:13"

dtitle: "第6篇"

dtype: "public"

exchange: "false "

uid: 5

uimage: 1

uintroduce: "ok"

uname: "用户3"

ustatus: 1

**点赞消息**

**后端：高予蜀**

前端 发送请求 url携带token

后端 判断token 按时间排序 点赞人的信息 日记信息

get http://116.62.14.0:8666/news/good/:token

前端申请示例

fetch('http://116.62.14.0:8666/news/good/1').then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: -1, data: "您没有点赞消息！"}

{status: 0, data: Array(2)}

data为json数组 每个json的键值对见下

comments: 0

dcontent: "第2篇日记"

did: 3

dimage: null

dpraise: 2

dtime: "2019/12/9 17:13"

dtitle: "第2篇"

dtype: "private"

exchange: "false "

pid: 3

pstatus: "true"

ptime: null

uid: 2

uimage: 1

uintroduce: "暂无介绍"

uname: "朝花夕拾"

ustatus: -1

**好友申请（无用）**

**后端：高予蜀**

前端 发送请求 url携带token

后端 判断token 按时间排序 申请人信息 一句话信息

get http://116.62.14.0:8666/news/friends/:token

前端申请示例

fetch('http://116.62.14.0:8666/news/friends/1').then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: -1, data: "您没有申请！"}

{status: 0, data: Array(4)}

data为json数组 每个json的键值对见下

acontent: "111111"

aid: 1

astatus: "0"

atime: "5"

atype: "friend"

dstatus: "true"

dtime: null

duid: null

longdes: "3333333"

shortdes: "333"

shortdes\_id: 3

uid: 5

uimage: 1

uintroduce: "ok"

uname: "用户3"

ustatus: 1

**交换申请**

**后端：高予蜀**

前端 发送请求 url携带token

后端 判断token 按时间排序 申请人信息 一句话信息

get http://116.62.14.0:8666/news/exchange/:token

前端申请示例

fetch('http://116.62.14.0:8666/news/exchange/21').then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: -1, data: "您没有申请！"}

{status: 0, data: Array(1)}

data为json数组 每个json的键值对见下

acontent: "333333"

aid: 3

astatus: "0"

atime: "3"

atype: "exchange"

dstatus: "true"

dtime: null

duid: null

longdes: "3333333"

shortdes: "333"

shortdes\_id: 3

uid: 1

uimage: 1

uintroduce: "暂无介绍"

uname: "朝花夕拾"

ustatus: 1

**交换申请回复**

**后端：高予蜀**

前端：发送申请 申请id aid 选择状态 reply（同意1或拒绝-1）

后端：找到申请 更改申请 一句话 日记 交换 返回成功与否

put http://116.62.14.0:8666/news/replyreq

前台申请示例

fetch('http://116.62.14.0:8666/news/replyreq', {

method: 'PUT',

headers: {

'Accept': 'application/json',

'Content-Type': 'application/json'

},

body: JSON.stringify({ aid:3,reply:1})}).then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: 0, data: "回复成功！"}

**我申请的交换**

**后端：高予蜀**

前台 发送申请 携带token

后台 找到自己的交换申请 返回状态

get http://116.62.14.0:8666/news/myreq/:token

前台申请示例

fetch('http://116.62.14.0:8666/news/myreq/1').then(res =>{ return res.json() }).then(d =>{ console.log(d); });

返回数据示例

{status: -1, data: "暂无申请！"}

{status: 0, data: Array(2)}

data为json数组 每个json见下

aid: 8

uid: 2

shortdes\_id: 2

astatus: "0"

acontent: "detail"

atime: "12/12/2019, 9:13:12 AM"

shortdes: "555"

**图片**

**请求图片**

**后端：高予蜀**

前台 发送请求 携带imgid

后端 根据images找到图片路径显示图片

get http://116.62.14.0:8666/image/:imgid

## 朝花夕拾 - 后台接口文档

### 登录

前端发送post请求

后端 判断是否存在用户，若返回值为ok则存在

接口：zhaohuaxishi/register

返回数据示例

data：'ok

### 首页

前端发送一个get请求

后端 获取后台管理首页的各个数据（用户上线量，总用户量，日记量，一句话量，交换日记量）

接口：http://116.62.14.0:8666/zhaohuaxishi/all

返回数据实示例

{"status":0,

"dataone":{"count":"6"},

"datatwo":{"count":"11"},

"datathree":{"count":"19"},

"datafour":{"count":"10"},

"datafive":{"count":"3"}

}

### 系统管理

前端发送一个get请求

后端获取管理员表数据

接口：zhaohuaxishi/administrator

返回数据示例

{"status":0,

"data":[

{

id":1,

"aname":"高予蜀",

"aemail":"66666666666@qq.com",

"anumber":"123456",

"aposition":"开发兼测试"},

{

"aid":2,"aname":"牛玉欣",

"aemail":"88888888888@qq.com",

"anumber":"222222",

"aposition":"开发兼测试"},

{

"aid":3,"aname":"孙童",

"aemail":"55555555555@qq.com",

"anumber":"333333",

"aposition":"开发兼测试"},

{

"aid":4,

"aname":"周弘",

"aemail":"77777777777@qq.com",

"anumber":"999999",

"aposition":"开发兼测试"

}]}

### 用户管理

前端发送一个get请求

后端获取用户表数据

接口：zhaohuaxishi/users

返回数据示例

{"status":0,

"data":[

{

"uid":8,

"uname":"朝花夕拾",

"uimage":1,

"uemail":"1120733074@qq.com",

"uintroduce":"暂无介绍",

"upassword":"niuyuxin666",

"ustatus":1

},

{

"uid":4,

"uname":"用户2",

"uimage":1

"uemail":"mail4",

"uintroduce":"ok",

"upassword":"1",

"ustatus":1

}]}

### 日记管理

##### 获取日记

前端发送一个get请求

后端获取日记表数据

接口：zhaohuaxishi/diary

返回数据示例

{"status":0,

"data":[

{

"did":6,

"uid":2,

"dtitle":"第5篇",

"dtime":"2019/12/5 17:13",

"dtype":"private",

"dpraise":0,

"comments":0,

"exchange":"false "

},

{

"did":1,

"uid":1,

"dtitle":"奥利给",

"dtime":"2019/12/9 17:13",

"dtype":"public",

"dpraise":1,

"comments":0,

"exchange":"false"

}]}

##### 删除日记

前端发送一个delete请求,body里传did

后端 对日记表进行数据删除

接口：zhaohuaxishi/diary/delete

返回数据示例

data:'ok'

### 评论管理

前端发送一个get请求

后端 获取评论表数据

接口：zhaohuaxishi/comments

返回数据示例

{"status":0,

"data":[

{

"comid":1,

"did":5,

"uid":1,

"comcontent":"111111",

"comtime":"2",

"comstatus":"false"

},

{

"comid":2,

"did":4,

"uid":7,

"comcontent":"222222",

"comtime":"1",

"comstatus":"false"

}]}