消息推送相关接口

Message push API

说明 instructions

目录 catalogue

1.1添加一条通知 add a notification

1.2修改一条通知 modify a notification

1.3删除一条通知delete a notification

1.4查看通知列表check notification list

1.5查看通知详情 check notification details

1.6发布通知post a notification

1.7用户实时获取消息 user get message in real time

1.8用户获取所有收到的消息列表 user get all received message list

1.9用户阅读一条消息 user read a message

1.10用户阅读所有的消息user read all messages

1.11用户删除一条消息user delete a message

1.12用户删除所有消息user delete all messages

添加一条通知

Add a notification

url:"/v1/cms/notify/add"

method:"post"

是否需要携带令牌：是

Need token or not: Need

前端请求的json数据

Front end requested json data

​

{

"title":"platform has update",

"body":"update some new functions",

"send\_method":"web", // web push //mobile short message //email email notification

"message\_type":"",

"send\_to":["xxxx","xxxxx"] //userId

}

​

后端返回的json数据

Back end returned json data

​

{

"error\_code":"1002",

"msg":"add success",

}

​

​

修改一条通知

Modify a notification

url:"/v1/cms/notify/:id/update"

method:"put"

是否需要携带令牌：是

Need token or not: Need

前端请求的json

Front end requested json

{

"title":"platform has update",

"body":"update some new functions",

"send\_method":"web", // web push //mobile short message //email email notification

"message\_type":"xxx", //update message

"send\_to":["xxxx","xxxxx"] //userId

}

​

数据后端返回的json数据

Data back end returned json data

​

{

"error\_code":"1003",

"msg":"modify success",

}

删除一条通知

Delete a notification

url:"/v1/cms/notify/:id/delete"

method:"delete"

是否需要携带令牌：是

Need token or not:Need

数据后端返回的json数据

Data back end returned json data

{

"error\_code":"1004",

"msg":"delete success",

}

查看通知列表

Check notification list

url:"/v1/cms/notify/list"

method:"get"

是否需要携带令牌：是

Need token or not: Need

前端请求的json

Front end requested json

数据后端返回的json数据

Data back end returned json data

{

"error\_code": 10000,

"msg": "query success",

"totalCount": 1,

"data": [

{

"id":"xxxxx",

"title":"platform push new content” ,//title

"message\_body"："xxxxx", //message body

"message\_type":"xxxx", //message type

"send\_method":"web", //push type

"recipients":[ //recipients

{"id":"xxx","username":"cxh"}

{"id":"xxx","username":"cxh2"}

],

"has\_read":[ //has read

{"id":"xxx","username":"cxh"}

],

"unread":[ //unread

{"id":"xxx","username":"cxh2"}

],

"has\_send":true, //has send or not

"date\_created": "2020-08-12T09:44:28.476+08:00",

"send\_at"："2020-08-12T09:44:28.476+08:00" //wen has\_send is false, this should be empty }

],

}

​

​

查看通知详情

Check notification details

url:"/v1/cms/notify/:id/detail"

method:"get"

是否需要携带令牌：是

Need token or not: Need

数据后端返回的json数据

Data back end returned json data

​

{

"data": {

"id": "6049ac3ba51408141461e499",

"title": "platform updated",

"message\_body": "update some new functions",

"message\_type": "update message",

"send\_method": "web",

"recipients": [

{

"id": "5f4777cfa51408189cb64d05",

"username": "1226465969@qq.com"

}

],

"has\_read": [],

"unread": [

{

"id": "5f4777cfa51408189cb64d05",

"username": "1226465969@qq.com"

}

],

"has\_send": false,

"date\_created": "2021-03-11T13:43:39.063+08:00",

"send\_at": null,

"use\_web\_when\_miss": false,

"lng": ""

},

"error\_code": 1001,

"msg": "get data success"

}

​

发布通知

Post notification

url:"/v1/cms/notify/:id/send"

method:"get"

是否需要携带令牌：是

Need token or not:Need

前端请求的json

Front end requested json

数据后端返回的json数据

Data back end returned json data

{

"error\_code": 10002,

"msg": "send message success"

}

用户实时获取消息

User get message in real time

url:"/v1/cms/getMessageRealTime"

method:"websocket persistent connection"

是否需要携带令牌：将令牌拼接到url中作为查询参数auth\_token=xxxx

Need token or not: take token as query parameter auth\_token=xxxx and joint it in url

是否服务器主动断开：否

Whether the server initiate to disconnect: No

服务器单次返回的数据实例

Server return data in one time, example:

[

{

"body":"updated some new functions ",

"has\_read":false,

"id":"6049ac3ba51408141461e499",

"title":"platform updated",

"type":"update message"

}

]

// 这里有可能是一个数组也有可能是一个单个的对象 it could be an array or a single object

//需要判断一下 need to judge

​

用户获取所有收到的消息列表

User get all received message list

url:"/v1/cms/message/list"

method:"get"

是否需要携带令牌：是

Need token or not:Need

数据后端返回的json数据

Data back end returned json data

{

"data": [

{

"body": "updated some new functions",

"has\_read": false,

"id": "6049ac3ba51408141461e499",

"title": "platform updated",

"type": "update message"

}

],

"error\_code": 1000,

"msg": "get list data success",

"totalCount": 1

}

用户阅读一条消息

User read a message

url:"/v1/cms/message/:id/readOneNotifyMessage"

method:"put"

是否需要携带令牌：是

Need token or not:Need

数据后端返回的json数据

Data back end returned json data

​

{

"error\_code": 10000,

"msg": "read notify message success"

}

​

用户阅读所有的消息

User read all messages

url:"/v1/cms/message/readAllNotifyMessage"

method:"put"

是否需要携带令牌：是

Need token or not:Need

数据后端返回的json数据

Data back end returned json data

​

{

"error\_code": 10000,

"msg": "read notify message success"

}

​

用户删除一条消息

User delete a message

url:"/v1/cms/message/:id/deleteOne"

method:"delete"

是否需要携带令牌：是

Need token or not:Need

数据后端返回的json数据

Data back end returned json data

{

"error\_code": 10001,

"msg": "clear notify message success"

}

用户删除所有消息

User delete all messages

url:"/v1/cms/message/deleteAll"

method:"delete"

是否需要携带令牌：是

Need token or not:need

数据后端返回的json数据

Data back end returned json data

{

"error\_code": 10001,

"msg": "clear notify message success"

}