腾讯云智绘

开发指南

腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】

腾讯云

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

开发指南
- 通用 API
  - 签名 sig 计算方法
  - 通过 appid 获取 access_token
  - 通过 secretId 获取 access_token
  - 获取模板分类 API
  - 获取素材分类 API
  - 多尺寸 PSD 解析
  - 获取用户草稿列表 API
  - 公共错误码
- 编辑器 SDK（PC 端）接入文档
- 编辑器 SDK（移动端）接入文档
- 获取模板 API
- 获取素材 API
- 人像抠图 API
- 物品抠图 API
- 智能消除笔 API
- 图片合成 API
- 智能封面 API
- 多尺寸 API
- 智能漫改 API
- 图像复原 API

版权所有：腾讯云计算（北京）有限责任公司

开发指南
通用 API
签名 sig 计算方法

最近更新时间：2024-10-24 22:06:32

签名 sig 计算流程

1. 业务方首先确定待签名参数以及私钥。
2. 业务方将待签名参数进行计算（注意一定要在后台进行计算，避免私钥泄露）

后台计算示例代码

```
const crypto = require('crypto')
const urlencode = require('urlencode')

// ***********************************
// params 为请求加密参数
// key 为加密密钥，比如：secretKey
// ***********************************
function sigCode(params: any, key: string) {
  try {
    let queryDataTmp = Object.assign({}, params)
    delete queryDataTmp.sig
    delete queryDataTmp.host
    // 第一步：将所有参数按key进行字典升序排列
    let paramtmp = Object.keys(queryDataTmp).sort()
    let parmaSort = ''
    // 第二步：将重新排序后的参数(key=value)用&拼接起来，得到源串
    for (const item of paramtmp) {
      // 对象和数组进行字符串化
      let itemValue = queryDataTmp[item]
      let changeType = ['object', 'array']
      if (changeType.indexOf(typeof itemValue) >= 0) {
        itemValue = JSON.stringify(itemValue)
      }
      parmaSort = parmaSort + item + '=' + itemValue + '&'
    }
    parmaSort = urlencode(parmaSort.substr(0, parmaSort.length - 1))
    // 第三步：使用HMAC-SHA256加密算法，通过密钥（secretKey或appkey）对源串加密
```

```
    const sig = crypto.createHmac('sha256', key).update(parmaSort)
    return sig.digest('hex')
  } catch (error) {
    console.log('codeSig error:', error)
    return false
  }
}
```

签名 sig 计算示例过程

请求参数：

```
{
    "id": "611f6b5b01af259c18954808",
    "secretId": "ca355ab0019211ecaed0ed79aa4a58cf",
    "users": [{"name": "javinz", "age": 18}],
}
```

排序并且字符串格式化后得到（用&连接每一个参数）：

```
id=611f6b5b01af259c18954808&secretId=ca355ab0019211ecaed0ed79aa4a58cf&us
ers=[{"name":"javinz","age":18}]
```

urlencode 编码后得到：

```
id%3D611f6b5b01af259c18954808%26secretId%3Dca355ab0019211ecaed0ed79aa4a5
8cf%26users%3D%5B%7B%22name%22%3A%22javinz%22%2C%22age%22%3A18%7D%5D
```

假设加密密钥 key 为：taidc，那么最后得到的 sig 为：

```
1c55d09921cad5a44de7cc983fe3de4e8ef6d7f98fd054b5335d729eb9ee3534
```

在线验证签名：验证签名

> **注意：**
> 您通过哪些参数获得的 sig，在使用 sig 时应传入相同的参数，例如您通过 id、secretId 参数得到了 sig，那么在请求“https://zhihui.qq.com/account/api/auth/token”等接口使用 sig 时也应只传入 id、secretId 参数，不可以多或少传入参数。

通过 appid 获取 access_token

最近更新时间：2024-10-24 22:06:32

通过 appid 获取 access_token 请求调用

- 调用地址：https://api.zhihui.qq.com/account/api/auth/token
- 请求方式：POST

请求 Header 说明

无请求 Header 说明。

请求 Body 说明

名称	类型	必填	示例值	描述
appid	string	是	commonIframe	管理员分配appid
channel	string	是	commonIframe	管理员分配channel
stamp	string	是	testUserId	业务方userId
timestamp	number	是	1629711353224	13位时间戳
sig	string	是	bf82618bc009d0a1cb62f045afccfe24cd50ea0235e32b95400a514dd44b5466	业务方签名，云智绘平台校验签名通过后颁发 token，业务方应使用appid、channel、stamp、timestamp获取sig，获取 sig

返回内容说明

名称	类型	示例值	描述
statusCode	number	200	状态码
data	object	{.....}	返回数据

返回数据说明

```
{
    "access_token": "xxxxxxx"
}
```

错误码说明

您可参见文档：公共错误码 。

通过 secretId 获取 access_token

最近更新时间：2024-10-24 22:06:32

通过 secretId 获取 access_token 请求调用

- 调用地址：https://api.zhihui.qq.com/account/api/auth/secret
- 请求方式：POST

请求 Header 说明

无请求 Header 说明。

请求 Body 说明

首先您需要通过腾讯云获取 userId、secretId、secretKey 。

名称	类型	必填	示例值	描述
id	string	是	61249df4fc4f470139c19612	通过腾讯云获得的 userId
secretId	string	是	e399a9e004ab11ecb4b77946aae1c3e1	通过腾讯云获得的 secretId
sig	string	是	bde6ff26fe9fd755923e165671f9bc57227167d4d9bfe0ebd774c23a69450dbe	业务方签名，云智绘平台校验签名通过后颁发 token，业务方应使用 id、secretId 获取 sig，获取sig，其中私钥为 secretKey

返回内容说明

名称	类型	示例值	描述
statusCode	number	200	状态码
data	object	{.....}	返回数据

返回数据说明

```
{
    "_id": "61249df4fc4f470139c19612",
    "pack": [], // 用户权益，用户暂时不需要关注
```

```
    "contact": [
        "61249df4fc4f470139c19615",
        "6167db4ed295b701584742cc",
        "6167db4ed295b701584742d5"
    ], // 用户关联信息，用户暂时不需要关注
    "team": [
        "6167db4ed295b701584742de"
    ], // 用户所属团队id，用户暂时不需要关注
    "company": [
        "615157d0d5493402184fceef"
        ], // 用户所属公司id，用户暂时不需要关注
    "scope": [], // 作用域，用户暂时不需要关注
    "name": "qcloud100007265271", // 名称
    "avatar": "", // 头像
    "country": "", // 国家
    "province": "", // 省份
    "city": "", // 城市
    "sex": 0, // 性别，0: 1:男 2:女
    "state": true, // 当前用户状态
    "pid": "611a392d15cfa40130f27165", // 应用appId，用户暂时不需要关注
    "secretId": "e399a9e004ab11ecb4b77946aae1c3e1",
    "createdAt": "2021-08-24T07:21:24.611Z",
    "updatedAt": "2023-02-09T07:50:54.853Z",
    "access_token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2MTI0OWRmNGZjNGY0NzAxMz
ljMTk2MTIiLCJuYW1lIjoicWNsb3VkMTAwMDA3MjY1MjcxIiwiYXZhdGFyIjoiIiwicGlkIj
oiNjExYTM5MmQxNWNmYTQwMTMwZjI3MTY1IiwiY29udGFjdCI6WyI2MTI0OWRmNGZjNGY0Nz
AxMzljMTk2MTUiLCI2MTY3ZGI0ZWQyOTViNzAxNTg0NzQyY2MiLCI2MTY3ZGI0ZWQyOTViNz
AxNTg0NzQyZDUiXSwiY3JlYXRlZEF0IjoiMjAyMS0wOC0yNFQwNzoyMToyNC42MTFaIiwidX
BkYXRlZEF0IjoiMjAyMy0wMi0wOVQwNzo1MDo1NC44NTNaIiwiaWF0IjoxNjc2MjcxNjIyLC
JleHAiOjE2Nzg4NjM2MjJ9.qAmC4EiL-y5Mkf7O9dUx0rQsdePZb66U7nJa4c3BJc4"
}
```


错误码说明

您可参见文档：公共错误码 。

获取模板分类 API

最近更新时间：2024-10-24 22:06:32

获取模板分类

- 调用地址：https://api.zhihui.qq.com/middle-material/terms/apollo/list
- 请求方式：GET

请求Header说明

参数名称	示例值
Authorization	Bearer XXXXXXX(access_token)

如何获取 access_token，通过 appid 获取 access_token 。

请求Body说明

无请求 Body 说明。

返回内容说明

名称	类型	示例值	描述
statusCode	number	200	状态码
data	object	{.....}	返回数据

返回数据说明

```
[
    {
        "_id": "5e69d9b2d86c5391887423c2",
        "name": "手机海报",
        "type": "template",
        "termtype": "2",
        "status": 1,
```

```
        "createdAt": "2022-08-01T08:40:48.156Z",
        "updatedAt": "2022-08-01T08:40:48.156Z",
        "__v": 0
    },
    ...
]
```

错误码说明

您可参见文档：公共错误码 。

获取素材分类 API

最近更新时间：2024-10-24 22:06:32

获取素材分类

- 调用地址：https://api.zhihui.qq.com/bll-apollo/material/getall?parentType=1,2,3
- 请求方式：GET

请求 Header 说明

参数名称	示例值
Authorization	Bearer XXXXXXX(access_token)

如何获取 access_token，通过 appid 获取 access_token 。

请求参数说明

名称	类型	必填	示例值	描述
parentType	string	是	1,2,3	1代表通用元素素材，2代表人物角色素材，3代表背景素材

返回内容说明

名称	类型	示例值	描述
statusCode	number	200	状态码
data	object	{.....}	返回数据

返回数据说明

```
[
    {
        "name": "像素风字母元素",
        "material": [
            {
                "_id": "6062e4eb22429b7ac61bb656",
                "thumb": "//static.taishan.qq.com/editor/users-upload/team-
```

```
5ea63428a839e332e381db1b/67093c423632606a41d7ac6f816dca70.png?
imageMogr2/thumbnail/!50p"
            },
            {
                "_id": "6062e4eb9ae607a7ca0385ab",
                "thumb": "//static.taishan.qq.com/editor/users-
upload/team-
5ea63428a839e332e381db1b/d0ee3b15251170f8398934b35ff54b01.png?
imageMogr2/thumbnail/!50p"
            },
            {
                "_id": "6062e4eb3ddf224f18b1151d",
                "thumb": "//static.taishan.qq.com/editor/users-
upload/team-
5ea63428a839e332e381db1b/eb3a45e0503ba35c9bef29b6df1d30ed.png?
imageMogr2/thumbnail/!50p"
            }
        ],
        "id": "6062e4a961a5e6ed69c054f0" // 此处为对应的分类id
    },
  ...
]
```

错误码说明

您可参见文档：公共错误码 。

多尺寸 PSD 解析

最近更新时间：2024-10-24 22:06:32

多尺寸 PSD 解析请求调用

- 调用地址：https://api.zhihui.qq.com/atom-parser/psd/upload-parse-only
- 请求方式：POST

请求 Header 说明

参数名称	示例值
Authorization	Bearer XXXXXXX(access_token)

如何获取 access_token，通过 appid 获取 access_token 。

请求 Body 说明

名称	类型	必填	示例值	描述
file	file	是	示例 psd 文件	待解析的 psd

> **注意：**
> 文件大小应控制在100M 以内，psd 的图层需根据相关性分组：背景及其相关元素必须放在[背景]组中，标题、文案和主要人物(或物品等)分别放置在[标题]、[文案]、[主体]组中。其他图层可以放置在任意名称的多个其他组中。组名是纯文字，不包含括号([])。

其他的解析规则请参见：解析规则 。

返回内容说明

名称	类型	示例值	描述
statusCode	number	201	状态码
data	object	{.....}	返回数据

返回数据说明

```
{
    "name": "4c8841f5a89e7b55b63392f6612f8b79", // 解析后的文件名称
    "width": 900,    // 解析后的psd的实际宽度
```

```
    "height": 383,  // 解析后的psd的实际高度
    "thumb":
"https://static.taishan.qq.com/xrdnest/psd/4540d0c764b53fe7d79fc2c82710a
ea3/514bd324-d48a-49af-8afd-7e66b424e15e.png?
imageMogr2/thumbnail/!50p/format/webp",
    // 预览图
    "objects": [ // psd中的层级信息
        {...},
        {...},
        {...}
    ]
}
```

> **说明：**
> 解析后得到的 data 数据您可以直接用作多尺寸 API 请求中的 content 数据源。

错误码说明

您可参见文档：公共错误码 。

获取用户草稿列表 API

最近更新时间：2024-10-24 22:06:32

获取用户草稿列表

- 调用地址：https://api.zhihui.qq.com/atom-record/edit-record/list
- 请求方式：POST

请求 Header 说明

参数名称	示例值
Authorization	Bearer XXXXXXX(access_token)

如何获取 access_token，通过 appid 获取 access_token 。

请求 Body 说明

无请求 Body 说明。

> **说明：**
> 此接口是通过用户的 token，获取对应用户的作图记录，因此您在生成 token 时，需要使用目标用户的信息生成。

返回内容说明：

名称	类型	示例值	描述
statusCode	number	200	状态码
data	object	{.....}	返回数据

返回数据说明

```
{
    "list": [
    {...}, // 用户的草稿记录信息，其中包含草稿的缩略图
    {...},
    {...}
  ]
}
```

错误码说明

您可参见文档：公共错误码 。

公共错误码

最近更新时间：2024-10-24 22:06:32

code	错误码	错误信息
400	BAD REQUEST	错误请求
401	UNAUTHORIZED	未经授权
402	PAYMENT REQUIRED	权益未付费
403	FORBIDDEN	访问被禁止
404	NOT FOUND	访问链接错误
405	METHOD NOT ALLOWED	服务器禁止了使用当前 HTTP 方法的请求
406	NOT ACCEPTABLE	服务器端无法提供与 Accept-Charset 以及 Accept-Language 消息头指定的值相匹配的响应
407	PROXY AUTHENTICATION REQUIRED	缺乏代理服务器的验证凭证
408	REQUEST TIMEOUT	服务器想要将没有在使用的连接关闭
409	CONFLICT	请求与服务器端目标资源的当前状态相冲突
410	GONE	目标资源在原服务器上不存在
411	LENGTH REQUIRED	由于缺少确定的Content-Length 首部字段，服务器拒绝客户端请求
412	PRECONDITION FAILED	对于目标资源的访问请求被拒绝

版权所有：腾讯云计算（北京）有限责任公司

413	PAYLOAD TOO LARGE	请求主体的大小超过了服务器处理限度
414	URI TOO LONG	客户端所请求的 URI 超过了服务器允许的范围
415	UNSUPPORTED MEDIA TYPE	服务器不支持的载荷格式
416	REQUESTED RANGE NOT SATISFIABLE	服务器无法处理所请求的数据区间
417	EXPECTATION FAILED	服务器无法满足 Expect 请求消息头中的期望条件
418	I AM A TEAPOT	服务器拒绝
421	MISDIRECTED	请求被定向到一个不能产生响应的服务器上
422	UNPROCESSABLE ENTITY	服务器无法处理所包含的指令
424	FAILED DEPENDENCY	由于之前的某个请求错误，导致当前请求错误
428	PRECONDITION REQUIRED	服务器端要求发送条件请求
429	TOO MANY REQUESTS	在一定的时间内用户发送了太多的请求
500	INTERNAL SERVER ERROR	服务器遇到意外的情况并阻止其执行请求
501	NOT IMPLEMENTED	请求的方法不被服务器支持
502	BAD GATEWAY	从上游服务器中接收到的响应是无效的
503	SERVICE UNAVAILABLE	服务器尚未处于可以接受请求的状态
50	GATEWAY	服务器无法在规定的时间内获得想要的响应

版权所有：腾讯云计算（北京）有限责任公司

4	TIMEOUT	
505	HTTP VERSION NOT SUPPORTED	服务器不支持请求所使用的 HTTP 版本

编辑器 SDK（PC 端）接入文档

最近更新时间：2024-10-24 22:06:32

如何接入?

1. 联系管理员索要，appid、channel、scope、appKey 您可以使用以下账户信息进行测试：

- appid: commonIframe
- channel: commonIframe
- scope: all
- appKey: bb6681972a3744488e9a8c7521635099

> **注意：**
> 其中 appKey 为私钥。

2. 通过 appid、channel、stamp、timestamp 获取 sig 。
3. 通过 appid 获取 access_token
4. 使用 npm 安装 zhihui-editor-sdk
5. 将 appid、channel、scope、stamp、timestamp、templateId、from、token 作为参数传入。
6. 调用 openIframe 方法打开即可

接入过程说明

> **说明：**
> 当您获取 access_token 后您需要执行以下步骤方可打开编辑器。

- 使用 npm 安装 zhihui-editor-sdk

```
npm i zhihui-editor-sdk
```

- 将 appid、channel、scope、stamp、timestamp、templateId、from、token 作为参数传入。

```
import ZhihuiEditor from 'zhihui-editor-sdk'
let option = {
  appid: 'commonIframe',
  channel: 'commonIframe',
  stamp: 'testUserId',
  timestamp: 1629711353224,
  scope: 'all', // 注意此处的scope是按照字符串传入
```

```
  templateId: '6254db7bccc34b01db8666ec',
  from: 'material',
  token: token.data.data.access_token,
}
let zhihuiEditor = new ZhihuiEditor(option)
```

- 调用 openIframe 方法打开即可

```
zhihuiEditor.openIframe()
```

实例化方法

1. openIframe — 打开 iframe

```
<button onClick={() => {zhihuiEditor.openIframe()}}>调起编辑器
SDK</button>
```

2. closeIframe — 关闭 iframe

```
<button onClick={() => {zhihuiEditor.closeIframe()}}>关闭编辑器
SDK</button>
```

打开编辑器 SDK 参数说明

参数	说明	类型	默认值	是否必须
appid	由管理员分配	string	—	是
channel	由管理员分配	string	—	是
scope	由管理员分配	string	—	是
teamId	由管理员分配，获取业务方团队内容时配置	string	—	否
companyId	由管理员分配，获取业务方公司内容时配置	string	—	否

stamp	业务方用户 id	string	—	是
timestamp	13位时间戳	number	—	是
token	请求 token	string	—	是
source	字体加载来源	string	—	否
templateId	模板 id，当 from 为 userConfig 时填写时间戳，templateId 说明	string	—	是
from	打开的模板来源，具体参见下方，from 字段有哪些，from 说明	string	—	是
loadingConfig	loading 配置，loadingConfig 说明	object	—	否
leftConfig	左侧配置，leftConfig 说明	object	—	否
headConfig	头部配置，headConfig 说明	object	—	否
templateDataConfig	自定义模板数据配置，templateDataConfig 说明	array	—	否
styleConfig	打开 iframe 样式配置，styleConfig 说明	object	—	否
commonConfig	通用的配置，commonConfig 说明	object	—	否
onOpen	初始化打开时候调用函数	function	—	否
onOpened	初始化打开完成时候调用函数	function	—	否

templateId 说明

1. 来源于智绘系统提供的模板 id，如果需要可以联系管理员。
2. 如果是用户自定义的模板数据，也就是 from 为 userConfig 时，那么此字段为时间戳。

from 说明

1. material — 当使用模板为智绘系统提供时，from 字段需要填写 material。
2. record — 当用户使用编辑器返回的 recordId 打开编辑器时，from 字段需要填写 record。或者当用户使用智绘封面 api 生成图片后，需要二次编辑时 from 字段需要填写 record。
3. userConfig — 当用户使用自定义模板数据时，from 需要填写 userConfig。
4. userPackageConfig — 针对性的配置，一般业务方不需要用到。

loadingConfig 说明

参数	说明	类型	默认值
logoUrl	加载时的logo	string	—

loadingConfig 配置 Demo

```
let option = {
  ...
  loadingConfig: {
    logoUrl: 'https://xxx.png'
  }
}
```

leftConfig 说明

参数	说明	类型	默认值
user	业务方自定义菜单配置	array	—

User 配置

用户可以配置三种自定义的素材列表，分别是无限滚动列表、无限级分类展示+无限滚动列表、按照分类展示+无限滚动列表。

无限滚动列表配置

```
user: [
    type: 'showAll', // 无限滚动列表的type为showall
    dataConfig: { // dataConfig为请求数据的配置
        url: 'https://xxxxx', // 获取无限滚动列表中数据的接口，获取方式为GET
        headers: { // 接口自定义token，为可选项
            'xxx-token': 'xxxx'
```

```
        },
    },
]
```

请求 Url 数据 API 需要提供的参数

参数	说明	类型	默认值
Offset	请求跳过数据数量	number	—
Limit	获取数据的数量，不可更改	number	50
Keyword	搜索关键词	string	—

需要注意参数首字母为大写，并且当 Keyword 为空时，应返回默认数据返回数据规定：

```
{
    items: [ // 无限分类列表的数据
        {
            thumbUrl: 'https://xxx.png', // 用户预览的缩略图
            url: 'https://xxx.png' // 实际添加到画布中的图片
        }
    ],
    total: 200 // 列表总数，如果返回错误无法正确计算加载更多
}
```

无限级分类展示+无限滚动列表

```
user: [
    type: 'unlimited',  // 无限级分类展示+无限滚动列表的type为unlimited
    dataConfig: { // dataConfig为请求数据的配置
      levelUrl: 'https://xxx', // 获取无限级分类的接口，获取方式为GET
      url: 'https://xxx', // 获取无限滚动列表中数据的接口，获取方式为GET
      headers: { // 接口自定义token，为可选项
        'xxx-token': 'xxxx'
      },
    },
]
```

请求 Url 数据 API 需要提供的参数

参数	说明	类型	默认值

Offset	请求跳过数据数量	number	—
Limit	获取数据的数量，不可更改	number	50
Keyword	搜索关键词	string	—
Category	分类参数	number	string

需要注意参数首字母为大写，并且当 Keyword 为空时，应返回默认数据返回数据规定：

```
{
    items: [ // 无限分类列表的数据
        {
            thumbUrl: 'https://xxx.png', // 用户预览的缩略图
            url: 'https://xxx.png' // 实际添加到画布中的图片
        }
    ],
    total: 200 // 列表总数，如果返回错误无法正确计算加载更多
}
```

请求 levelUrl 数据 API 需要提供的参数–无请求 levelUrl 返回数据规定：返回对象需要包含 id、parentId、name、rank 属性，其中 id 属性为当用户点击分类以后将此 id 传入获取图片素材 url 的 Category 字段。

```
[
  { id: 1, parentId: 0, name: '一级菜单', rank: 1 },
  { id: 2, parentId: 0, name: '一级菜单', rank: 1 },
  { id: 3, parentId: 0, name: '一级菜单', rank: 1 },
  { id: 4, parentId: 1, name: '二级菜单1', rank: 2 },
  { id: 5, parentId: 1, name: '二级菜单2', rank: 2 },
  { id: 6, parentId: 2, name: '二级菜单3', rank: 2 },
  { id: 7, parentId: 4, name: '三级菜单', rank: 3 },
  { id: 8, parentId: 7, name: '四级菜单', rank: 4 },
  { id: 9, parentId: 8, name: '五级菜单', rank: 5 },
  { id: 10, parentId: 9, name: '六级菜单', rank: 6 },
  { id: 11, parentId: 10, name: '七级菜单', rank: 7 },
  { id: 12, parentId: 11, name: '八级菜单', rank: 8 },
  { id: 13, parentId: 12, name: '九级菜单', rank: 9 },
  { id: 14, parentId: 13, name: '十级菜单', rank: 10 }
]
```

按照分类展示+无限滚动列表

```
user: {
    type: 'showChild',  // 按照分类展示+无限滚动列表的type为showChild
    dataConfig: { // dataConfig为请求数据的配置
      childUrl: 'https://xxx', // 获取无限滚动列表中数据的接口，获取方式为GET
      url: 'https://xxx', // 获取按照分类显示数据的接口，获取方式为GET
      tagList: ['测试1', '测试2', '测试3', '测试4', '测试5', '测试6', '测试
7', '测试8', '测试9'],
      // 在搜索栏下方显示的tag列表，当用户点击以后会将tag内容作为关键词请求
childUrl接口，此配置为可选项
      headers: { // 接口自定义token，为可选项
        'xxx-token': 'xxxx'
      },
    },
}
```

请求 Url 数据 API 需要提供的参数–无

返回数据规定：

```
[
    id: 'xxx', // 分类id，当用户点击分类以后将此id传入childUrl的Category字段。
    name: 'xxx', // 分类名称，显示作用
    material: [ // 返回的每一种分类的显示图片，每一个分类显示3个
        {
            url: 'https://xxx.png', // 实际添加到画布中的图片
            thumbUrl: 'https://xxx.png' // 用户预览的缩略图
        }
    ]
]
```

请求 childUrl 数据 API 需要提供的参数

参数	说明	类型	默认值
Offset	请求跳过数据数量	number	—
Limit	获取数据的数量，不可更改	number	50
Keyword	搜索关键词	string	—
Category	分类参数	number	string

需要注意参数首字母为大写，并且当 Keyword 为空时，应返回默认数据。

返回数据规定：

```
{
    items: [ // 无限列表的数据
        {
            thumbUrl: 'https://xxx.png', // 用户预览的缩略图
            url: 'https://xxx.png' // 实际添加到画布中的图片
        }
    ],
    total: 200 // 列表总数，如果返回错误无法正确计算加载更多
}
```

User 配置 Demo

```
let header = {} // 想要正确运行的话可以联系管理员获取header后进行请求

let option = {
  ...
  leftConfig: {
    user: [
      {
        menuName: '测试',
        menuIcon:
'https://static.taishan.qq.com/editor/assets/01b2269953df053926f7045d0a6
70655.png',
        menuIconActive:
'https://static.taishan.qq.com/editor/assets/280272c691fbdadb36701a48340
c5d36.png',
        type: 'showChild',
        dataConfig: {
          url: 'https://taishan.qq.com/api/category/getall?
parentType=2',
          tagList: ['测试1', '测试2', '测试3', '测试4', '测试5', '测试6', '测
试7', '测试8', '测试9'],
          headers: header,
          childUrl: 'https://taishan.qq.com/api/ape/search',
        },
      },
      {
        menuName: '测试1',
        menuIcon:
'https://static.taishan.qq.com/editor/assets/01b2269953df053926f7045d0a6
```

```
70655.png',
        menuIconActive:
'https://static.taishan.qq.com/editor/assets/280272c691fbdadb36701a48340
c5d36.png',
        type: 'showAll',
        dataConfig: {
          url: 'https://taishan.qq.com/api/ape/search',
          headers: header,
        }
      },
      {
        menuName: '测试3',
        menuIcon:
'https://static.taishan.qq.com/editor/assets/01b2269953df053926f7045d0a6
70655.png',
        menuIconActive:
'https://static.taishan.qq.com/editor/assets/280272c691fbdadb36701a48340
c5d36.png',
        type: 'unlimited',
        dataConfig: {
          levelUrl: 'https://taishan.qq.com/api/category',
          url: 'https://taishan.qq.com/api/ape/search',
          headers: header,
        },
      },
    ]
  }
}
```

headConfig 说明

参数	说明	类型	默认值
logoUrl	头部的logo	string	—
downloadName	下载按钮的名称	string	—
dropDownloadName	下拉框下载按钮的名称	string	—
onClickDownload	用户点击下载按钮callback	function	—
isDownloadImg	点击下载按钮后是否下载图片	boolean	true

selectQuality	是否需要选择图片质量	boolean	true
saveRecord	是否自动触发保存记录	boolean	true
updateTemplate	是否只更新模板（为true时只返回更新状态，不返回其他信息）	boolean	false

headConfig 配置 Demo

```
let option = {
  ...
  headConfig: {
    logoUrl: 'https://xxx.png',
    downloadName: '完成',
    onClickDownload: (res) => {
      console.log(res)
    },
    isDownloadImg: false,
  },
}
```

onClickDownload 返回

```
{
  "name": "56a4419401b62dcda242595e4f32114a.png",
  "status": "done",
  "url":
"//static.taishan.qq.com/editor/material/56a4419401b62dcda242595e4f32114
a.png",
  "thumbUrl":
"//static.taishan.qq.com/editor/material/56a4419401b62dcda242595e4f32114
a.png?imageMogr2/thumbnail/!50p",
  "width": 720,
  "height": 400,
  "recordId": 'xxxxxx'
}
```

参数	说明	类型	默认值
name	图片名称	string	—
status	图片导出状态	string	done

url	图片真实 url	string	—
thumbUrl	图片缩略图	string	—
width	图片宽度	number	—
height	图片高度	number	—
recordId	用户完成内容的 id，可以在二次编辑时使用	string	—

templateDataConfig 说明

templateDataConfig 字段为数组，数组中可以包含多个图片、文字类型对象。

> **注意：**
> 需要注意，因为画布本身没有默认大小，因此默认必须传入一层 workarea 用来定义画布的基础信息。

图片对象属性

参数	说明	类型	默认值
angle	图片旋转角度	number	—
height	图片高度	number	—
left	left距离	number	—
opacity	透明度	number	—
originX	对象变换的水平原点，可能值为"left", "right", "center"	string	—
originY	对象变换的垂直原点，可能值为"top", "bottom", "center"	string	—
src	图片的src	string	—
top	top 距离	number	—
type	类型，必须填写：image	string	—
scaleX	图片水平方向缩放大小	number	—
scaleY	图片垂直方向缩放大小	number	—
width	图片宽度	number	—

文字对象属性

参数	说明	类型	默认值
angle	旋转角度	number	—
fill	颜色	string	—
fontFamily	字体	string	—
fontSize	字体大小	number	—
fontStyle	字体样式，可能的值为"normal", "italic", "oblique"	string	—
fontWeight	字体粗细，可能的值为 bold, normal, 400, 600, 800	number	string
height	字体高度	number	—
left	left距离	number	—
opacity	透明度	number	—
originX	对象变换的水平原点，可能值为"left", "right", "center"	string	—
originY	对象变换的垂直原点，可能值为"top", "bottom", "center"	string	—
scaleX	水平方向缩放大小	number	—
scaleY	垂直方向缩放大小	number	—
text	文本内容	string	—
textAlign	文字对齐方式，可能的值为"left", "center", "right", "justify", "justify-left", "justify-center" or "justify-right"	string	—
top	top距离	number	—
type	必须填写：textbox	string	—
width	字体宽度	number	—

templateDataConfig 配置 Demo

```
[
```

```
  {
    id: 'workarea', // 必须包含此id
    top: 0,
    left: 0,
    width: 1242,
    height: 2208,
  },
  {
    type: 'image',
    top: 0,
    left: 0,
    src:
'https://static.taishan.qq.com/editor/assets/2997b0c920be29c12c16ae3a2b2
bc378.png',
  },
]
```

styleConfig 说明

参数	说明	类型	默认值
hasBackground	是否显示 iframe 打开的默认背景遮罩	boolean	true
hasQuitButton	是否显示 iframe 的退出按钮	boolean	true
parentDomId	iframe 打开的父节点 id	string	—
iframeStyle	打开的 iframe 样式，可以按照 react 中 css 方式写入	object	—

```
let option = {
  ......
  styleConfig: {
    hasBackground: false,
    hasQuitButton: false,
    parentDomId: 'test',
    iframeStyle: {
      position: 'absolute',
      top: 0,
      left: 0,
      zIndex: 10000,
      display: 'block',
      width: '100%',
```

```
        height: '100%',
      },
    }
  }
```

commonConfig 说明

因为会有一些业务方提出个性化的定制化开发需求，因此将某些定制化的非通用性功能放在此配置中，如果您也有定制化功能的诉求，也可以随时联系我们的管理员。

编辑器 SDK（移动端）接入文档

最近更新时间：2024-10-24 22:06:32

如何接入?

1. 联系管理员索要，appid、channel、scope、appKey 您可以使用以下账户信息进行测试：
- appid: commonIframe
- channel: commonIframe
- scope: all
- appKey: bb6681972a3744488e9a8c7521635099

> **注意：**
> 其中 appKey 为私钥。

2. 通过 appid、channel、stamp、timestamp 获取 sig 。
3. 通过 appid 获取 access_token 。
4. 使用 npm 安装 zhihui-editor-mobile-sdk
5. 将 appid、channel、scope、stamp、timestamp、templateId、from、token 作为参数传入。
6. 调用 openIframe 方法打开即可

接入过程说明

当您获取 access_token 后您需要执行以下步骤方可打开编辑器。

- 使用 npm 安装 zhihui-editor-mobile-sdk：

```
npm i zhihui-editor-mobile-sdk
```

- 将 appid、channel、scope、stamp、timestamp、templateId、from、token 作为参数传入：

```
import ZhihuiEditor from 'zhihui-editor-mobile-sdk'
let option = {
  appid: 'commonIframe',
  channel: 'commonIframe',
  timestamp: 1629711353224,
  scope: 'all', // 注意此处的scope是按照字符串传入
  templateId: '62c2571219c76e620670584c',
  from: 'material',
  token: token.data.data.access_token,
}
```

```
let zhihuiEditor = new ZhihuiEditor(option)
```

- 调用 openIframe 方法打开即可：

```
zhihuiEditor.openIframe()
```

实例化方法

1. openIframe：打开 iframe

```
  <button onClick={() => {zhihuiEditor.openIframe()}}>调起智绘
iframe</button>
```

2. closeIframe：关闭 iframe

```
  <button onClick={() => {zhihuiEditor.closeIframe()}}>关闭智绘
iframe</button>
```

打开 Iframe 配置参数说明

参数	说明	类型	默认值	是否必须
appid	业务方唯一 id，由管理员分配	string	—	是
channel	业务方渠道，由管理员分配	string	—	是
scope	作用域配置，由管理员分配	string	—	是
teamId	团队 id，获取业务方团队内容时配置，由管理员分配	string	—	否
companyId	企业 id，获取业务方公司内容时配置，由管理员分配	string	—	否
timestamp	13位时间戳	number	—	是
token	请求 token	string	—	是

templateId	模板 id，templateId 说明	string	—	是
from	打开的模板来源，具体参见下方，from 字段有哪些，from 说明	string	—	是
headConfig	头部配置，headConfig 说明	object	—	是
bottomConfig	底部配置，bottomConfig 说明	object	—	是
templateDataConfig	自定义模板数据配置，templateDataConfig 说明	array	—	否
styleConfig	打开 iframe 样式配置，styleConfig 说明	object	—	否
commonConfig	通用的配置，commonConfig 说明	object	—	否

templateId 说明

1. 来源于智绘系统提供的模板 id，如果需要可以联系管理员。
2. 如果是用户自定义的模板数据，也就是 from 为 userConfig 时，那么此字段为时间戳。

from 说明

1. material — 当使用模板为智绘系统提供时，from 字段需要填写 material。
2. record — 当用户使用编辑器返回的 recordId 打开编辑器时，from 字段需要填写 record。
3. userConfig — 当用户使用自定义模板数据时，from 需要填写 userConfig。
4. userPackageConfig — 针对性的配置，一般业务方不需要用到。

headConfig 说明

参数	说明	类型	默认值
onClickDownload	用户点击下载按钮callback	function	—

headConfig 配置Demo

```
let option = {
  ...
  headConfig: {
```

```
    onClickDownload: (res) => {
      console.log(res)
    },
  },
}
```

onClickDownload 返回

```
{
  "name": "56a4419401b62dcda242595e4f32114a.png",
  "status": "done",
  "url":
"//static.taishan.qq.com/editor/material/56a4419401b62dcda242595e4f32114
a.png",
  "thumbUrl":
"//static.taishan.qq.com/editor/material/56a4419401b62dcda242595e4f32114
a.png?imageMogr2/thumbnail/!50p",
  "width": 720,
  "height": 400,
  "recordId": 'xxxxxx'
}
```

参数	说明	类型	默认值
name	图片名称	string	—
status	图片导出状态	string	done
url	图片真实 url	string	—
thumbUrl	图片缩略图	string	—
width	图片宽度	number	—
height	图片高度	number	—
recordId	用户完成内容的 id，可以在二次编辑时使用	string	—

bottomConfig 说明

参数	说明	类型	默认值
user	用户配置的底部自定义素材	object	—

版权所有：腾讯云计算（北京）有限责任公司

user 配置

用户可以配置三种自定义的素材列表，分别是无限滚动列表、无限级分类展示+无限滚动列表、按照分类展示+无限滚动列表。

无限滚动列表配置

```
user: {
    type: 'showAll', // 无限滚动列表的type为showall
    dataConfig: { // dataConfig为请求数据的配置
      url: 'https://xxxxx', // 获取无限滚动列表中数据的接口，获取方式为GET
      headers: { // 接口自定义token，为可选项
            'xxx-token': 'xxxx'
        },
    },
}
```

请求 url 数据 API 需要提供的参数：

参数	说明	类型	默认值
Offset	请求跳过数据数量	number	—
Limit	获取数据的数量，不可更改	number	50
Keyword	搜索关键词	string	—

需要注意参数首字母为大写，并且当 Keyword 为空时，应返回默认数据。
返回数据规定：

```
{
    items: [ // 无限分类列表的数据
        {
            thumbUrl: 'https://xxx.png', // 用户预览的缩略图
            url: 'https://xxx.png' // 实际添加到画布中的图片
        }
    ],
    total: 200 // 列表总数，如果返回错误无法正确计算加载更多
}
```

无限级分类展示+无限滚动列表

```
user: {
```

```
    type: 'unlimited',  // 无限级分类展示+无限滚动列表的type为unlimited
    dataConfig: { // dataConfig为请求数据的配置
      levelUrl: 'https://xxx', // 获取无限级分类的接口，获取方式为GET
      url: 'https://xxx', // 获取无限滚动列表中数据的接口，获取方式为GET
      headers: { // 接口自定义token，为可选项
        'xxx-token': 'xxxx'
      },
    },
}
```

请求 url 数据 API 需要提供的参数：

参数	说明	类型	默认值
Offset	请求跳过数据数量	number	—
Limit	获取数据的数量，不可更改	number	50
Keyword	搜索关键词	string	—
Category	分类参数	number | string	—

需要注意参数首字母为大写，并且当Keyword为空时，应返回默认数据。
返回数据规定：

```
{
    items: [ // 无限分类列表的数据
        {
            thumbUrl: 'https://xxx.png', // 用户预览的缩略图
            url: 'https://xxx.png' // 实际添加到画布中的图片
        }
    ],
    total: 200 // 列表总数，如果返回错误无法正确计算加载更多
}
```

请求levelUrl数据API需要提供的参数-无
请求levelUrl返回数据规定：返回对象需要包含id、parentId、name、rank属性，其中id属性为当用户点击分类以后将此id传入获取图片素材url的Category字段。

```
[
  { id: 1, parentId: 0, name: '一级菜单', rank: 1 },
  { id: 2, parentId: 0, name: '一级菜单', rank: 1 },
  { id: 3, parentId: 0, name: '一级菜单', rank: 1 },
```

```
  { id: 4, parentId: 1, name: '二级菜单1', rank: 2 },
  { id: 5, parentId: 1, name: '二级菜单2', rank: 2 },
  { id: 6, parentId: 2, name: '二级菜单3', rank: 2 },
  { id: 7, parentId: 4, name: '三级菜单', rank: 3 },
  { id: 8, parentId: 7, name: '四级菜单', rank: 4 },
  { id: 9, parentId: 8, name: '五级菜单', rank: 5 },
  { id: 10, parentId: 9, name: '六级菜单', rank: 6 },
  { id: 11, parentId: 10, name: '七级菜单', rank: 7 },
  { id: 12, parentId: 11, name: '八级菜单', rank: 8 },
  { id: 13, parentId: 12, name: '九级菜单', rank: 9 },
  { id: 14, parentId: 13, name: '十级菜单', rank: 10 }
]
```

按照分类展示+无限滚动列表

```
user: {
    type: 'showChild',  // 按照分类展示+无限滚动列表的type为showChild
    dataConfig: { // dataConfig为请求数据的配置
      childUrl: 'https://xxx', // 获取无限滚动列表中数据的接口，获取方式为GET
      url: 'https://xxx', // 获取按照分类显示数据的接口，获取方式为GET
      tagList: ['测试1', '测试2', '测试3', '测试4', '测试5', '测试6', '测试
7', '测试8', '测试9'],
      // 在搜索栏下方显示的tag列表，当用户点击以后会将tag内容作为关键词请求
childUrl接口，此配置为可选项
      headers: { // 接口自定义token，为可选项
        'xxx-token': 'xxxx'
      },
    },
}
```

请求 url 数据 API 需要提供的参数–无

返回数据规定：

```
[
    id: 'xxx', // 分类id，当用户点击分类以后将此id传入childUrl的Category字段。
    name: 'xxx', // 分类名称，显示作用
    material: [ // 返回的每一种分类的显示图片，每一个分类显示3个
        {
            url: 'https://xxx.png', // 实际添加到画布中的图片
            thumbUrl: 'https://xxx.png' // 用户预览的缩略图
        }
    ]
```

```
]
```

请求 childUrl 数据API需要提供的参数

参数	说明	类型	默认值
Offset	请求跳过数据数量	number	—
Limit	获取数据的数量，不可更改	number	50
Keyword	搜索关键词	string	—
Category	分类参数	number | string	—

需要注意参数首字母为大写，并且当 Keyword 为空时，应返回默认数据。
返回数据规定：

```
{
    items: [ // 无限列表的数据
        {
            thumbUrl: 'https://xxx.png', // 用户预览的缩略图
            url: 'https://xxx.png' // 实际添加到画布中的图片
        }
    ],
    total: 200 // 列表总数，如果返回错误无法正确计算加载更多
}
```

templateDataConfig 说明

templateDataConfig 字段为数组，数组中可以包含多个图片、文字类型对象。

> **注意：**
> 需要注意，因为画布本身没有默认大小，因此默认必须传入一层 workarea 用来定义画布的基础信息。

图片对象属性

参数	说明	类型	默认值
angle	图片旋转角度	number	—
height	图片高度	number	—

left	left距离	number	—
opacity	透明度	number	—
originX	对象变换的水平原点，可能值为"left", "right", "center"	string	—
originY	对象变换的垂直原点，可能值为"top", "bottom", "center"	string	—
src	图片的src	string	—
top	top距离	number	—
type	类型，必须填写：image	string	—
scaleX	图片水平方向缩放大小	number	—
scaleY	图片垂直方向缩放大小	number	—
width	图片宽度	number	—

文字对象属性

参数	说明	类型	默认值
angle	旋转角度	number	—
fill	颜色	string	—
fontFamily	字体	string	—
fontSize	字体大小	number	—
fontStyle	字体样式，可能的值为"normal", "italic", "oblique"	string	—
fontWeight	字体粗细，可能的值为 bold, normal, 400, 600, 800	number	string
height	字体高度	number	—
left	left距离	number	—
opacity	透明度	number	—
originX	对象变换的水平原点，可能值为"left", "right", "center"	string	—

originY	对象变换的垂直原点，可能值为"top", "bottom", "center"	string	—
scaleX	水平方向缩放大小	number	—
scaleY	垂直方向缩放大小	number	—
text	文本内容	string	—
textAlign	文字对齐方式，可能的值为"left", "center", "right", "justify", "justify-left", "justify-center" or "justify-right"	string	—
top	top距离	number	—
type	必须填写：textbox	string	—
width	字体宽度	number	—

templateDataConfig 配置 Demo

```
[
  {
    id: 'workarea', // 必须包含此id
    top: 0,
    left: 0,
    width: 1242,
    height: 2208,
  },
  {
    type: 'image',
    top: 0,
    left: 0,
    src:
'https://static.taishan.qq.com/editor/assets/2997b0c920be29c12c16ae3a2b2
bc378.png',
  },
]
```

styleConfig 说明

参数	说明	类型	默认值
parentDomId	iframe 打开的父节点 id	string	—

iframeStyle	打开的 iframe 样式，可以按照 react 中 css 方式写入	object	—

styleConfig 配置 Demo

```
let option = {
  ......
  styleConfig: {
    parentDomId: 'test',
    iframeStyle: {
      position: 'absolute',
      top: 0,
      left: 0,
      zIndex: 10000,
      display: 'block',
      width: '100%',
      height: '100%',
    },
  }
}
```

commonConfig 说明

因为会有一些业务方提出个性化的定制化开发需求，因此将某些定制化的非通用性功能放在此配置中，如果您也有定制化功能的诉求，也可以随时联系我们的管理员。

获取模板 API

最近更新时间：2024-10-24 22:06:32

获取模板

- 调用地址：https://api.zhihui.qq.com/middle-material/template/apollo/list
- 请求方式：POST

请求 Header 说明

参数名称	示例值
Authorization	Bearer XXXXXXX(access_token)

如何获取 access_token，通过 appid 获取 access_token 。

请求 Body 说明

名称	类型	必填	示例值	描述
filter	object	是	{"state": "success", "type": "image"}	查询时的过滤条件，默认应填写示例值内容
hideObjects	boolean	否	true	查询结果是否隐藏模板 object 对象，影响返回内容体积
filterTermArr	string[]	否	["5f44c02a6bf4e1c95844d2d9"]	用来查询某个分类下的模板，获取模板分类 API
pageNumber	number	是	1	分页查询时页码从1开始
pageSize	number	是	50	分页查询数据量大小
sort	object	否	{"updatedAt": "desc"}	查询时候的排序条件，可以填写示例值来控制查询结果的排序

返回内容说明

名称	类型	示例值	描述

statusCode	number	201	状态码
data	object	{.....}	返回数据

返回数据说明

```
{
    "info": {
            "page":1,
            "total": 1280
        },
    "items": [
    {...}, // 模板信息
    {...}
  ]
}
```

错误码说明

您可参见文档：公共错误码 。

获取素材 API

最近更新时间：2024-10-24 22:06:32

获取素材

- 调用地址：https://api.zhihui.qq.com/bll-apollo/material/oneCategory
- 请求方式：POST

请求 Header 说明

参数名称	示例值
Authorization	Bearer XXXXXXX(access_token)

如何获取 access_token，通过 appid 获取 access_token 。

请求 Body 说明

名称	类型	必填	示例值	描述
categoryId	string	是	6194b4a0056b8000ed4f64b2	需要获取的元素分类 id，获取素材分类 API
page	number	是	0	分页查询时页码从0开始
limit	number	是	50	分页查询数据量大小

返回内容说明

名称	类型	示例值	描述
statusCode	number	200	状态码
data	object	{.....}	返回数据

返回数据说明

```
{
    "list": [
        {
            "_id": "619751aabc29e501163a504f",
            "thumb": "//static.taishan.qq.com/editor/users-upload/team-6126127b2c396400e5390356/2973c1a91ff2b1b12bb575917f36b4a7.png?
```

```
imageMogr2/thumbnail/!50p"
      // 返回的图片有可能未携带协议头
        },
    ...
  ],
  "total": 70
}
```

错误码说明

您可参见文档：公共错误码 。

人像抠图 API

最近更新时间：2024-10-24 22:06:32

人像抠图

- 调用地址：https://api.zhihui.qq.com/bll-smart-image/atom/human-sega
- 请求方式：POST

请求Header说明

参数名称	示例值
Authorization	Bearer XXXXXXX(access_token)

如何获取 access_token，通过 appid 获取 access_token 。

请求 Body 说明

名称	类型	必填	示例值	描述
url	string	是	https://static.taishan.qq.com/test/zzj.jpeg	目标图像 url，**url 地址需要找商务或管理员添加白名单**
needFaceRect	boolean	否	true	是否需要返回人脸位置信息

返回内容说明

名称	类型	示例值	描述
statusCode	number	200	状态码
url	string	https://xxx.png	结果图片 url
faceReact	object[]	`[ { "x": 308, "y": 92, "width": 120, "height": 191 }`	人脸位置信息

		]	

错误码说明

您可参见文档：公共错误码 。

物品抠图 API

最近更新时间：2024-10-24 22:06:32

物品抠图

- 调用地址：https://api.zhihui.qq.com/bll-smart-image/atom/object-sega
- 请求方式：POST

请求 Header 说明

参数名称	示例值
Authorization	Bearer XXXXXXX(access_token)

如何获取 access_token，通过 appid 获取 access_token 。

请求 Body 说明

名称	类型	必填	示例值	描述
url	string	是	https://static.taishan.qq.com/editor/74bcb53aee9f1df3cd0d5ac191c21743.jpeg	目标图像 url，**url 地址需要找商务或管理员添加白名单**

返回内容说明

名称	类型	示例值	描述
statusCode	number	200	状态码
url	string	https://xxx.png	结果图片 url

错误码说明

您可参见文档：公共错误码 。

智能消除笔 API

最近更新时间：2024-11-20 21:44:22

图片整体消除水印

- 调用地址：https://api.zhihui.qq.com/bll-smart-image/atom/text-remove
- 请求方式：POST

请求 Header 说明

参数名称	示例值
Authorization	Bearer XXXXXXX(access_token)

如何获取 access_token，通过 appid 获取 access_token 。

请求 Body 说明

名称	类型	必填	示例值	描述
url	string	是	https://static.taishan.qq.com/editor/74bcb53aee9f1df3cd0d5ac191c21743.jpeg	目标图像 url，**url 地址需要找商务或管理员添加白名单**

返回内容说明

名称	类型	示例值	描述
statusCode	number	200	状态码
data	object[]	[{.....}]	返回数据

返回数据说明

```
[
    {
    "image_base64": "xxxxxx", // 返回的base64结果数据
    "image_name": "", // 图片名称，可忽略
    "image_url": "", // 图片url，可忽略
  }
]
```

错误码说明

您可参见文档：公共错误码 。

图片局部消除

- 调用地址：https://api.zhihui.qq.com/bll-smart-image/atom/image-inpaint
- 请求方式：POST

请求 Header 说明

参数名称	示例值
Authorization	Bearer XXXXXXX(access_token)

如何获取 access_token，通过 appid 获取 access_token 。

请求 Body 说明

名称	类型	必填	示例值	描述
url	string	是	https://static.taishan.qq.com/editor/image/467d94ad6ba79019c1c0bea5e252dbad.png	目标图像 url，**url 地址需要找商务或管理员添加白名单**
maskUrl	string	是	https://static.taishan.qq.com/editor/dc7727be71e843f736005d46dcf5a147/335752b50a5825fccf043f2d95337d18.png	需要消除区域的遮罩图片，**url 地址需要找商务或管理员添加白名单**

返回内容说明

名称	类型	示例值	描述
statusCode	number	200	状态码
url	string	http://50058.gzc.svp.tencent-cloud.com/0b53kud7yaahbqae6ixoejry2vod7rkqp7aa.f0.png?dis_k=d11759f104fc7ce957bd787907d52cd4&dis_t=1676260185	结果图片 url

错误码说明

您可参见文档：公共错误码 。

图片合成 API

最近更新时间：2024-10-24 22:06:32

图片合成

- 调用地址：https://api.zhihui.qq.com/bll-smart-image/atom/image-render
- 请求方式：POST

请求 Header 说明

参数名称	值
Authorization	Bearer XXXXXXX(access_token)

如何获取 access_token，通过 appid 获取 access_token 。

请求 Body 说明

名称	类型	必填	示例值	描述
content	object	是	{.....}	待生成 json 信息

待生成 json 信息说明

```
{
    "width": 900, // 生成图片的宽度
  "height": 383, // 生成图片的高度
  "objects": [
    {
        "type": "image",
        "top": 0,
        "left": 0,
        "src":
"https://static.taishan.qq.com/editor/assets/2997b0c920be29c12c16ae3a2b2
bc378.png"
      },  // 生成图片的每层数据
      {
        "type": "textbox",
        "fill": "rgba(255, 255, 255, 1)",
        "angle": 0,
        "text": "测试文字",
        "fontSize": 80,
```

```
        "fontFamily": "SourceHanSansCN-Heavy"
      }
    ]
  }
```

注意：

- json 信息中 objects 的图片对象url地址需要管理员开通白名单方可正确渲染。
- json 信息中 objects 的每层信息是需要符合云智绘平台的数据结构，如您已有数据结构需要转换或者需要从无生成数据结构，可以联系我们的商务沟通具体的解决方案。

返回内容说明

名称	类型	示例值	描述
statusCode	number	200	状态码
url	string	https://xxx.png	生成的结果图
content	object	{.....}	生成时使用的数据信息

错误码说明：

您可参见文档：公共错误码 。

智能封面 API

最近更新时间：2024-11-20 21:44:22

文章封面图合成

- 调用地址：https://api.zhihui.qq.com/bll-smart-image/smartcover/article-tmp
- 请求方式：POST

请求 Header 说明

参数名称	示例值
Authorization	Bearer XXXXXXX(access_token)

如何获取 access_token，通过 appid 获取 access_token 。

请求 Body 说明

名称	类型	必填	示例值	描述
title	string	是	文章封面合成	生成封面的标题
num	number	否	6	单次生成的封面图数量，默认为4
projectId	string	是	62f38041389e7cd8fe1e1fa0	由管理员统一分类

返回内容说明

名称	类型	示例值	描述
statusCode	number	200	状态码
data	object[]	[{.....}]	返回数据

返回数据说明

```
[
    {
        "id": "63eaf2daddb734daa631c4cb",
        "tplId": "62fc8f88993ce41c30edb033",
```

```
        "url": "https://static.taishan.qq.com/attom-
render/taishan/0f25d399af43fadc24abb7214f8581ff.png" // 生成结果
  },
  {...},
  {...}
]
```

错误码说明

您可参见文档：公共错误码 。

电商运营图合成

- 调用地址：https://api.zhihui.qq.com/bll-smart-image/smartcover/engine-v2-tmp
- 请求方式：POST

请求Header说明

参数名称	示例值
Authorization	Bearer XXXXXXX(access_token)

如何获取 access_token，通过 appid 获取 access_token 。

请求Body说明

名称	类型	必填	示例值
inputer	object	是	`{ "main": "https://static.taishan.qq.com/editor/dc7727be71e843f736005d46dc` // 商品主体图片 `"text_price": "12"` // 商品价格 `"text_shop": "快乐购"` // 商铺名称 `"text_text": "良心产品"` // 主文案 `}`
num	num	否	6

	ber		
projectId	string	是	63be85792299c6012dd35578

返回内容说明

名称	类型	示例值	描述
statusCode	number	201	状态码
data	object[]	[{.....}]	返回数据

返回数据说明

```
[
    {
        "id": "63eaf2daddb734daa631c4cb",
        "tplId": "62fc8f88993ce41c30edb033",
        "url": "https://static.taishan.qq.com/attom-
render/taishan/0f25d399af43fadc24abb7214f8581ff.png" // 生成结果
  },
  {...},
  {...}
]
```

错误码说明

您可参见文档：公共错误码 。

人物封面图合成

- 调用地址：https://api.zhihui.qq.com/bll-smart-image/smartcover/video-cover-tmp
- 请求方式：POST

请求 Header 说明

参数名称	值
Authorization	Bearer XXXXXXX(access_token)

如何获取 access_token，通过 appid 获取 access_token 。

请求 Body 说明

名称	类型	必填	示例值	描述
title	string	是	人物封面	生成时的标题
desc	string	是	描述性文案	生成时的描述文案
mainUrl	string	否	https://xxx.png	生成时使用的主体图片，建议传入主体图，否则会因主体图缺失导致无内容生成。
num	number	否	6	单次生成的封面图数量，默认为4
projectId	string	是	63199c45e381c6761f1e6399	由管理员统一分类

返回内容说明

名称	类型	示例值	描述
statusCode	number	201	状态码
data	object[]	[{.....}]	返回数据

返回数据说明

```
[
    {
        "id": "63eaf2daddb734daa631c4cb",
```

```
        "tplId": "62fc8f88993ce41c30edb033",
        "url": "https://static.taishan.qq.com/attom-
render/taishan/0f25d399af43fadc24abb7214f8581ff.png" // 生成结果
  },
  {...},
  {...}
]
```

错误码说明

您可参见文档：公共错误码 。

多尺寸 API

最近更新时间：2024-10-24 22:06:32

多尺寸

- 调用地址：https://api.zhihui.qq.com/bll-smart-image/multisize/common
- 请求方式：POST

请求 Header 说明

参数名称	示例值
Authorization	Bearer XXXXXXX(access_token)

如何获取 access_token，通过 appid 获取 access_token 。

请求 Body 说明

名称	类型	必填	示例值	描述
content	object	是	`{ "width": 900, "height": 383, "objects": [ {...} ] }`	需要多尺寸生成的数据源
target	object[]	是	`[ { "width": 800, "height":`	生成的目标尺寸

```
400
    },
    {

"width":
800,

"height":
460
    }
]
```

> **说明：**
> 需要多尺寸生成的数据源如何获取?
> 您需要通过调用多尺寸 PSD 解析接口，通过解析psd获取数据源，多尺寸 PSD 解析 。
> - 如果您使用的 PSD 模板为横版，自定义尺寸需宽>高, 且宽/高需在1.2-2.0之间。
> - 如果您使用的 PSD 模板为竖版，自定义尺寸需宽<高, 且高/宽需在1.5-2.5之间。

返回内容说明

名称	类型	示例值	描述
statusCode	number	201	状态码
data	object	{.....}	返回数据

返回数据说明

```
{
    "status": 200, // 生成结果状态
    "message": "模板结构正确", // 生成结果信息
    "target": [ // 生成的目标尺寸信息
        {
            "width": 800, // 生成目标的宽度
            "height": 400, // 生成目标的高度
            "status": 200,  // 生成目标的状态
            "message": "尺寸符合要求", // 生成目标的信息
            "url": "https://static.taishan.qq.com/attom-
render/apollo/224f99e26929cf38b031e7eef538f3fd.png", // 生成的目标图片
            "content": { // 生成目标的结构信息
```

```
                "width": 800,
                "height": 400,
                "objects": [
                    {...}
                ]
            }
        },
        {
            "width": 800,
            "height": 460,
            "status": 200,
            "message": "尺寸符合要求",
            "url": "https://static.taishan.qq.com/attom-
render/apollo/a431b1375ddb93a28a9227f49cde10a3.png",
            "content": {
                "width": 800,
                "height": 460,
                "objects": [
                    {...}
                ]
            }
        }
    ]
}
```

错误码说明

您可参见文档：公共错误码 。

智能漫改 API

最近更新时间：2024-10-24 22:06:32

智能漫改

- 调用地址：https://api.zhihui.qq.com/bll-smart-image/atom/comics
- 请求方式：POST

请求Header说明

参数名称	示例值
Authorization	Bearer XXXXXXX(access_token)

如何获取 access_token，通过 appid 获取 access_token 。

请求Body说明

名称	类型	必填	示例值	描述
url	string	是	https://static.taishan.qq.com/test/zzj.jpeg	目标图像url，**url地址需要找商务或管理员添加白名单**

返回内容说明

名称	类型	示例值	描述
statusCode	number	201	状态码
data	object	{.....}	返回数据

返回数据说明

```
{
    "output_url": "http://50058.gzc.svp.tencent-cloud.com/0b53oub5uaadvqajlcyd7zsis5od3j2qhwqa.f0.jpeg?dis_k=a41759d5c2acacfa81ed2d051684d2f3&dis_t=1677742041"
  // 漫改后的结果
}
```

错误码说明

您可参见文档：公共错误码 。

版权所有：腾讯云计算（北京）有限责任公司

图像复原 API

最近更新时间：2024-10-24 22:06:32

图像复原

- 调用地址：https://api.zhihui.qq.com/bll-smart-image/atom/image-restore
- 请求方式：POST

请求Header说明

参数名称	示例值
Authorization	Bearer XXXXXXX(access_token)

如何获取 access_token，通过 appid 获取 access_token 。

请求Body说明

名称	类型	必填	示例值	描述
url	string	是	https://static.taishan.qq.com/test/zzj.jpeg	目标图像url，**url地址需要找商务或管理员添加白名单**

返回内容说明

名称	类型	示例值	描述
statusCode	number	201	状态码
data	object	{.....}	返回数据

返回数据说明

```
{
    "image_name": "xxxx", // 复原后的图片名称
  "image_base64": ".....", // 复原后的图片base64数据
  "image_url": "http://50058.gzc.svp.tencent-cloud.com/0b535yapoaaasmakkcqaw5siv3wd67xab5ya.f0.jpg?dis_k=86d4720dd13bce936ad01a25e3bdd6c7&dis_t=1677742717"
  // 复原后的图片url
```

```
}
```

错误码说明

您可参见文档：公共错误码 。