软件接口文档

# 约定

所有API接口通过HTTP协议进行访问，所有请求和响应参数默认为JSON格式(特殊声明除外)。

http返回码是2XX的，表示API调用成功。如果API调用成功，返回结果（如果有结果的话）中不会有error字段。

http返回码不是2XX的，表示API调用失败。如果API调用失败，会返回固定格式的错误结果，该格式如下表所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 可选 | 定义 | 描述 |
| error |  | 错误的字符串标识 | HTTP状态码对应的字符串描述 |
| message |  | 错误的描述信息 |  |
| exception | 可选 | 错误的异常类型 |  |
| status |  | 错误的数字标识 | HTTP状态码 |
| timestamp |  | 错误发生的时间戳 |  |
| path | 可选 | 错误发生的API路径 |  |
| id | 可选 | 错误对应的资源ID | 发生NotExistException时存在该属性 |
|  |  |  |  |

API调用失败时返回的错误信息示例如下：

*HTTP STATUS: 404*

*HTTP RESPONSE BODY:*

*{*

*"error": "Not Found",*

*"message": "Product 12 is not exist!",*

*"exception": "com.ma.cm.exception.ProductNotExistException",*

*"status": 404,*

*"timestamp": 1549202539590,*

*"path": "/v1/products/12",*

*"id": 12,*

*}*

# 版面管理

## 产品管理

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| function | method | API | request Content-Type | request body | response body |
| 查 | GET | /v1/products | application/json |  | the list of $(product) |
| 增 | POST | /v1/products | application/json | $(product) | $(product) |
| 查 | GET | /v1/products/{pid} | application/json |  | $(product) |
| 改 | PUT | /v1/products/{pid} | application/json | $(product) | $(product) |
| 删 | DELETE | /v1/products/{pid} | application/json | $(product) |  |

上表使用$(product)指代产品的详细信息，其详细信息如下表所示。

|  |  |  |
| --- | --- | --- |
| 属性 | 定义 | 描述 |
| id |  | primary key  对于post接口如果不提供则会自动生成 |
| name |  |  |
| options |  |  |

$(product)的示例：

*{*

*“id”: 1,*

*“name”: “王者乐园”,*

*“options”: null*

*}*

## 栏目管理

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| function | method | API | request Content-Type | request body | response body |
| 查 | GET | /v1/products/{pid}/columns | application/json |  | the list of $(column) |
| 增 | POST | /v1/products/{pid}/columns | multipart/form-data | $(column\_file) | $(column) |
| 查 | GET | /v1/products/{pid}/columns/{cid} | application/json |  | $(column) |
| 改 | PUT | /v1/products/{pid}/columns/{cid} | multipart/form-data | $(column\_file) | $(column) |
| 删 | DELETE | /v1/products/{pid}/columns/{cid} | application/json | $(column) |  |

上表使用$(column)指代栏目的详细信息，其详细信息如下表所示。

|  |  |  |
| --- | --- | --- |
| 属性 | 定义 | 描述 |
| productId |  | primary key |
| columnId |  | primary key |
| name |  |  |
| type |  | 1: 游戏栏目  2: 视频栏目  0: 其他栏目 |
| poster |  |  |
| link |  |  |
| options |  |  |

$(column)的示例：

*{*

*“productId”: 1,*

*“columnId”: 1,*

*“name”: “塔防游戏”,*

*“type”: 1,*

*“poster”: “example.pic”,*

*“link”: “example.html”,*

*“options”: null*

*}*

上表使用$(column\_file)指代带海报信息的栏目的详细信息，POST和PUT接口使用该数据结构。其详细信息如下表所示。

|  |  |  |
| --- | --- | --- |
| 属性 | 定义 | 描述 |
| productId |  | primary key |
| columnId |  | primary key  对于post接口如果不提供则会自动生成 |
| name |  |  |
| type |  | 1: 游戏栏目  2: 视频栏目  0: 其他栏目 |
| posterFile | 海报文件 | 如果该字段不存在或者为空时，那么数据库中该字段不会被更新  如果文件存在，那么会被上传到文件服务器，数据库中记录其在文件服务器中的位置 |
| link |  |  |
| options |  |  |

## 内容管理

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| function | method | API | request Content-Type | request body | response body |
| 查 | GET | /v1/products/{pid}/contents | application/json |  | the list of $(content) |
| 增 | POST | /v1/products/{pid}/contents | multipart/form-data | $(content\_file) | $(content) |
| 查 | GET | /v1/products/{pid}/contents/{cid} | application/json |  | $(content) |
| 改 | PUT | /v1/products/{pid}/contents/{tid} | multipart/form-data | $(content\_file) | $(content) |
| 删 | DELETE | /v1/products/{pid}/contents/{tid} | application/json | $(content) |  |

上表使用$(content)指代内容的详细信息，其详细信息如下表所示。

|  |  |  |
| --- | --- | --- |
| 属性 | 定义 | 描述 |
| productId |  | primary key |
| contentId |  | primary key |
| name |  |  |
| poster |  |  |
| icon |  |  |
| screenshot |  |  |
| type |  | 1: 游戏  2: 视频  0: 其他 |
| link |  |  |
| tip |  |  |
| options |  |  |

$(content)的示例：

*{*

*“productId”: 1,*

*“contentId”: 1,*

*“name”: “example”,*

*“poster”: “example.pic”,*

*“icon”: “example.pic”,*

*“screenShot”: “example.pic”,*

*“type”: 1,*

*“link”: “example.html”,*

*“tip”: “this is a helping message”,*

*“options”: null*

*}*

上表使用$(content\_file)指代带海报信息的内容的详细信息，POST和PUT接口使用该数据结果。其详细信息如下表所示。

|  |  |  |
| --- | --- | --- |
| 属性 | 定义 | 描述 |
| productId |  | primary key |
| contentId |  | primary key  对于post接口如果不提供则会自动生成 |
| name |  |  |
| posterFile | 海报文件 | 如果该字段不存在或者为空时，那么数据库中该字段不会被更新  如果文件存在，那么会被上传到文件服务器，数据库中记录其在文件服务器中的位置 |
| iconFile | 图标文件 | 如果该字段不存在或者为空时，那么数据库中该字段不会被更新  如果文件存在，那么会被上传到文件服务器，数据库中记录其在文件服务器中的位置 |
| screenshotFile | 截图文件 | 如果该字段不存在或者为空时，那么数据库中该字段不会被更新  如果文件存在，那么会被上传到文件服务器，数据库中记录其在文件服务器中的位置 |
| type |  | 1: 游戏  2: 视频  0: 其他 |
| link |  |  |
| tip |  |  |
| options |  |  |

## 栏目和内容的关系管理

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| function | method | API | request Content-Type | request body | response body |
| 查 | GET | /v1/products/{pid}/columns/{cid}/contents | application/json |  | the list of $(relation) |
| 查 | GET | /v1/products/{pid}/columns/{cid}/contents?detail | application/json |  | the list of $(detail-relation) |
| 增 | POST | /v1/products/{pid}/columns/{cid}/contents | application/json | $(relation) | $(relation) |
| 查 | GET | /v1/products/{pid}/columns/{cid}/contents/{tid}?type=0 | application/json |  | $(relation) |
| 查 | GET | /v1/products/{pid}/columns/{cid}/contents/{tid}?type=1 | application/json |  | $(relation) |
| 改 | PUT | /v1/products/{pid}/columns/{cid}/contents/{tid} | application/json | $(relation) | $(relation) |
| 删 | DELETE | /v1/products/{pid}/columns/{cid}/contents/{tid}?type=0 | application/json | $(relation) |  |
| 删 | DELETE | /v1/products/{pid}/columns/{cid}/contents/{tid}?type=1 | application/json | $(relation) |  |

上表使用$(relation)指代栏目和内容关系的详细信息，其详细信息如下表所示。

|  |  |  |
| --- | --- | --- |
| 属性 | 定义 | 描述 |
| productId |  | primary key |
| columnId |  | primary key |
| type | 如果type=0，那么contentId表示内容的id  如果type=1，那么contentId表示子栏目的id | primary key |
| contentId |  | primary key |
| position |  |  |
|  |  |  |

$(relation)的示例：

*{*

*“productId”: 1,*

*“columnId“: 1,*

*“type”: 0,*

*“contentId”: 1,*

*“position”: 9*

*}*

上表使用$(detail-relation)指代栏目和内容关系的详细信息，其详细信息如下表所示。

|  |  |  |
| --- | --- | --- |
| 属性 | 定义 | 描述 |
| productId |  | primary key |
| columnId |  | primary key |
| type | 如果type=0，那么contentId表示内容的id  如果type=1，那么contentId表示子栏目的id | primary key |
| contentId |  | primary key |
| position |  |  |
| content | content描述了内容的详细信息，具体内容参见章节“内容管理”中关于$(content)的介绍 | 如果type=0则content有详细信息 |
| childColumn | childColumn描述了子栏目的详细信息，具体内容参见章节“栏目管理”中关于$(column)的介绍 | 如果type=1则childColumn有详细信息 |

$(detail-relation)的示例：

*{*

*“productId”: 1,*

*“columnId“: 1,*

*“type”: 0,*

*“contentId”: 1,*

*“position”: 9,*

*“content”: {*

*“productId”: 1,*

*“contentId”: 1,*

*“name”: “name”,*

*“poster”: “example.pic”,*

*“icon”: “example.pic”,*

*“screenShot”: “example.pic”,*

*“type”: 1,*

*“link”: “example.html”,*

*“tip”: “this is a helper message”,*

*“options”: null*

*},*

*“childColumn”: null*

*}*

## 文件服务器管理

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| function | method | API | request Content-Type | request body | response body |
| 查 | GET | /v1/ftps | application/json |  | the list of $(ftp) |
| 查 | GET | /v1/ftps/{pid} | application/json |  | $(ftp) |
| 改 | PUT | /v1/ftps/{pid} | application/json | $(ftp) | $(ftp) |

文件服务器管理接口只支持查、改，禁止增、删。并且数据库中存在且仅存在一条id=1的记录。

上表使用$(ftp)指代文件服务器的详细信息，其详细信息如下表所示。

|  |  |  |
| --- | --- | --- |
| 属性 | 定义 | 描述 |
| id |  | primary key |
| host |  |  |
| port |  |  |
| username |  | 该用户名密码必须具有文件服务器的读写权限 |
| password |  |  |

用户上传的海报会存储在文件服务器的某个目录中，详细信息见下表。

|  |  |  |
| --- | --- | --- |
| 海报属主 | 位置 | 举例 |
| 栏目 | /product-${pid}/column/uuid.png | /product-1/column/8d2d235b-8444-4a96-99bc-f51d68b515c5.png |
| 内容 | /product-${pid}/content/uuid.png | /product-1/content/c33b40ae-9e5f-4594-8d3e-1d6db48e6770.png |

数据库中只会记录海报在文件服务器中的位置，即上表第二列内容。前台程序拿到位置信息后需要拼接文件服务器地址信息作为前缀才能获取海报的文件内容。文件服务器地址信息可以通过/v1/apis/1接口获取。

# 统计分析

未完待续…

# API管理

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| function | method | API | request body | response body |
| 查 | GET | /v1/apis |  | the list of API |