

mini-adwall 后端接口文档

本文档基于当前 `apps/api` 目录内的 Express 路由与数据库结构自动整理。版本：v1.0.0

总览

- 基础 URL: `http://<host>:<port>` (默认端口 `3000`)
- 所有业务接口前缀: `/api`
- 静态资源: 上传后的视频与预览图通过 `/uploads` 目录暴露, 例如 `/uploads/videos/<filename>`
- 认证: 当前系统无认证/鉴权逻辑, 所有接口默认公开访问 (生产环境需补充认证中间件)。
- 数据库: MySQL (`adwall`), 主要表: `ad_type, form_config, ad, video`
- 返回格式: 统一使用 JSON `{ code: <number>, data: <any>, message?: <string> }`
 - `code = 200` 成功; `4xx` 为客户端请求错误; `500` 为服务器错误。

通用错误结构

```
{  
  "code": 400,  
  "message": "错误提示"  
}
```

健康检查

GET `/api/health`

说明: 服务健康与连通性检查。 **响应示例:**

```
{ "code": 200, "data": "ok" }
```

广告类型 (Ad Types)

GET `/api/ad-types`

说明: 获取启用状态(`status=1`)的广告类型列表。 **响应字段:** `id, type_code, type_name` **响应示例:**

```
{  
  "code": 200,  
  "data": [  
    { "id": 1, "type_code": "short_video", "type_name": "短视频广告" },  
    { "id": 2, "type_code": "brand", "type_name": "品牌广告" }  
  ]  
}
```

可能错误: 500 查询广告类型失败

表单配置 (Form Config)

GET /api/form-config?type_code=<code>&config_key=<key>

必填 Query: type_code **可选 Query:** config_key (默认 ad_create_form) **说明:** 查询某广告类型对应的表单配置；若数据库中缺失则返回内置默认配置。 **成功响应示例 (存在配置) :**

```
{
  "code": 200,
  "data": {
    "formTitle": "创建短视频广告",
    "fields": [{ "name": "title", "type": "input" }]
  }
}
```

默认响应示例 (类型不存在或无配置) :

```
{
  "code": 200,
  "data": { "formTitle": "默认广告表单", "fields": [] },
  "message": "默认配置"
}
```

错误:

- 缺少 type_code → 400 type_code required
- 服务器异常 → 500 查询配置失败

视频上传 (Videos)

POST /api/videos/upload

说明: 上传单个视频文件并解析其基础信息（时长、分辨率），生成首帧预览图。可根据 type_id 进行类型特定校验规则（时长 / 格式 / 文件大小）。 **请求类型:** multipart/form-data **表单字段:**

- video (File, 必填) 视频文件
- type_id (Number, 可选) 若传则根据对应 ad_type.type_code 选用校验规则

类型内置规则 (按 type_code) :

type_code	时长范围(秒)	最大大小(Byte)	允许扩展名
short_video	5 - 60	104857600 (100MB)	.mp4
brand	15 - 120	209715200 (200MB)	.mp4, .avi

type_code	时长范围(秒)	最大大小(Byte)	允许扩展名
effect	10 - 90	157286400 (150MB)	.mp4

成功响应:

```
{
  "code": 200,
  "data": {
    "videoId": "<uuid>",
    "url": "http://host/uploads/videos/<file>",
    "duration": 35,
    "resolution": "1080x1920",
    "previewUrl": "http://host/uploads/previews/<preview>.jpg"
  }
}
```

失败场景:

- 未上传文件: 400 视频文件未上传, 请使用字段名 video
- 类型规则校验失败: 时长/大小/格式 → 400 对应提示
- 解析/预览生成失败: 仍可能返回基本成功 (无预览) 或 500 视频解析失败

广告 (Ads)

数据结构 (表 ad 关键字段)

字段	说明
id	广告唯一 UUID
type_id	广告类型外键
publisher	投放方/发布者
title	标题
content	文本内容 (可为空)
heat	热度值 (INT)
price	出价 (DECIMAL)
landing_url	跳转落地页 URL
video_ids	逗号分隔的视频ID列表 (冗余, 用于快速检索)
ext_info	JSON 扩展信息
created_at / updated_at	时间戳

1. 创建广告

POST /api/ads 请求体 (JSON):

```
{
  "publisher": "公司A",
  "title": "新品推广",
  "content": "内容文本",
  "heat": 10,
  "price": 99.5,
  "landing_url": "https://example.com",
  "type_id": 1,
  "video_ids": ["vid-1", "vid-2"],
  "ext_info": { "tags": ["新品", "短视频"] }
}
```

必填: publisher, title, content, heat, price, landing_url, type_id (**video_ids** 可数组或逗号字符串, **ext_info** 可选 JSON) **成功响应:**

```
{ "code": 200, "data": { "ad_id": "<uuid>", "message": "广告创建成功" } }
```

错误:

- 缺字段 → 400 publisher、title、content、heat、price、landing_url 为必填 / 400 type_id 必填
- 服务器异常 → 500 创建广告失败

2. 分页获取广告列表

GET /api/ads?page=<page>&size=<size> **说明:** 按广告类型的 sort_rule 中定义的字段排序 (当前实现: 若 sort_rule.field == 'price' 则优先按价格降序, 再按创建时间降序)。并返回每条广告对应的视频 URL 列表。默认分页: page=1, size=10 **响应示例:**

```
{
  "code": 200,
  "data": {
    "page": 1,
    "size": 10,
    "list": [
      {
        "id": "<uuid>",
        "type_id": 1,
        "publisher": "公司A",
        "title": "新品推广",
        "price": 99.5,
        "video_ids": "vid-1,vid-2",
        "video_urls": [ "/uploads/videos/<file1>", "/uploads/videos/<file2>" ]
      }
    ]
}
```

```

    }
}

```

错误: 500 获取广告列表失败

3. 更新广告出价

PUT /api/ads/:id/price **请求体:** { "price": 120.00 } **说明:** 更新指定广告出价，并返回更新前后索引位置变化及受影响的相邻广告ID。 **成功响应示例:**

```

{
  "code": 200,
  "data": {
    "adId": "<uuid>",
    "oldIndex": 5,
    "newIndex": 2,
    "affectedAdIds": ["neighbor1", "neighbor2"],
    "message": "出价更新成功，位置变化 3"
  }
}

```

错误:

- 缺 price → 400 price required
- 广告不存在 → 404 广告不存在
- 服务器异常 → 500 更新出价失败

4. 部分字段更新

PUT /api/ads/:id **请求体:** 仅包含需要更新的字段（任意子集）：

publisher, title, content, heat, price, landing_url, ext_info, video_ids 若无字段传入 → 400 无更新字段 **成功:** { code:200, data:{ adId: "<uuid>", message:"更新成功" } } **失败:** 500 更新失败

5. 复制广告

POST /api/ads/:id/copy **说明:** 复制指定广告生成新广告（标题附加（副本））。 **成功:** { code:200, data:{ ad_id:<newUuid>, message:"复制成功" } } **失败:**

- 源不存在 → 404 源广告不存在
- 服务器异常 → 500 复制失败

6. 删除广告

DELETE /api/ads/:id **说明:** 删除广告并解除其关联的视频（video.ad_id 置 NULL，文件不删除）。 **成功:** { code:200, data:{ message:"删除成功" } } **失败:** 500 删除失败

视频记录 (Video Table)

上传后会插入 `video` 表，部分字段：`id`, `ad_id`(初始NULL), `file_name`, `file_path`, `file_size`, `file_type`, `duration`, `resolution`。关联逻辑：在广告创建时若传 `video_ids`，后端会为这些视频更新 `ad_id`。

排序规则 (ad_type.sort_rule JSON)

示例：

```
{"priority":2,"field":"price","order":"desc","secondField":"created_at","secondOrder":"desc"}
```

当前列表查询实现仅使用 `field` 与可选 `secondField` 排序（降序或升序）。

示例快速流程

1. 上传视频: `POST /api/videos/upload` → 获得 `videoId`
 2. 创建广告: `POST /api/ads` 带上 `video_ids: ["videoId"]`
 3. 查询列表: `GET /api/ads?page=1&size=10`
 4. 更新价格: `PUT /api/ads/<id>/price` 调整排序位置
 5. 复制或删除广告: `POST /api/ads/<id>/copy` / `DELETE /api/ads/<id>`
-

统一响应与错误规范建议 (可扩展)

未来可扩展字段：

- `trace_id`: 用于链路追踪
 - `error_code`: 细粒度错误码
 - `meta`: 分页或统计信息容器
-

后续改进建议

- 添加鉴权（如 JWT 或 API Key）
 - 增加 OpenAPI 自动生成与版本化
 - 视频转码与异步处理队列（防止阻塞上传）
 - 更丰富的排序与过滤（价格区间、发布者、`type_code`）
 - 列表接口拆分：基础信息与详情信息
-

文档到此结束。