

## Adms

Assist Of Adx

pypi v1.0.0rc2

downloads 3M/month

linux passing

windows passing

coverage 100%

```
1. http http://restful.adleida.com:8008/
2.
3. HTTP/1.0 200 OK
4. Content-Length: 41
5. Content-Type: application/json
6. Date: Tue, 09 Jun 2015 09:09:44 GMT
7. Server: Werkzeug/0.10.4 Python/2.7.6
8.
9. "Welcome to adleida restful API service!"
```

## 特性

- 成熟的数据管理系统 妥善保存 dsp 信息
- 使用高性能 gridfs 完整存储物料
- 物料经由审核流程确保安全合法
- cdn 大范围分发 大大降低等待时延
- 支持高并发 HTTP 请求

## 客户端准备工作

```
1. sudo apt install python
2. wget https://bootstrap.pypa.io/get-pip.py
3. sudo python get-pip.py
4. sudo pip install httpie
```

## 使用指南

### Dsp 相关操作

#### 1. 注册 Dsp 基本信息

## 1. 数据准备

注册信息采用 json 数据格式，示例模型如下所示，`name` 为 Dsp 方的名称，`burl` 为 Dsp 方竞价的服务器地址

```
1. {  
2.   "name": "dspname",  
3.   "burl": "http://example.dsp.com/path/to/bid/"  
4. }
```

2. 调用 `RESTful-API` 进行注册（假设现在你已经完成一份 `json` 格式的 `Dsp` 数据信息，这里暂时命名为 `Dsp-info.json`）

```
1. http POST http://restful.adleida.com:8008/v1/dsp/ < dsp-info.json
```

## 3. 确认并保留返回信息

您将会接收到如同下列形式的返回信息。其中 `id` 是在我方服务端自行生成的唯一标识，`access_token` 作为以后对该注册信息操作的凭据，请自行妥善保存

```
1. HTTP/1.0 200 OK  
2. Content-Length: 103  
3. Content-Type: application/json  
4. Date: Tue, 09 Jun 2015 10:44:01 GMT  
5. Server: Werkzeug/0.10.4 Python/2.7.6  
6.  
7. {  
8.   "id": "5576c371c44b1a5756805130",  
9.   "access_token": "d19a1398-ccf5-4c47-868c-a4abaf24e011",  
10.  "message": "successfully create your dsp information to adexchange"  
11. }
```

## 2. 查询 Dsp 信息

### 1. 通过返回的 id 查询信息

```
1. http GET http://restful.adleida.com:8008/v1/dsp/<id>
```

其中 `<id>` 为注册返回的 id 值 下面的例子直接使用了上述返回的 id 结果

http GET

<http://restful.adleida.com:8008/v1/dsp/5576c371c44b1a5756805130>

## 2. 你会接受到如同下列形式的返回信息

```
1. HTTP/1.0 200 OK
2. Content-Length: 98
3. Content-Type: application/json
4. Date: Tue, 09 Jun 2015 10:52:58 GMT
5. Server: Werkzeug/0.10.4 Python/2.7.6
6.
7. {
8.   "burl": "http://example.dsp.com/path/to/bid/",
9.   "id": "5576c371c44b1a5756805130",
10.  "name": "dspname"
11. }
```

## 3. 修改已注册的 Dsp 信息

### 1. 发送请求参数，提交 PUT 请求

修改只需要将你想改变的字段作为参数交给请求一并发送至服务器即可，这里依然使用上述返回的 `id` 作为演示，请注意修改操作是需要提供 `access_token` 作为凭据的。下方示例修改了 `name` 字段的值

```
1. http PUT http://restful.adleida.com:8008/v1/dsp/ access_token:d19a1398-ccf5-4c47-86
8c-a4abaf24e011 id=5576c371c44b1a5756805130 name=newdspname
```

### 2. 返回提示

修改操作成功后返回提示信息如下所示

```
1. HTTP/1.0 200 OK
2. Content-Length: 38
3. Content-Type: application/json
4. Date: Wed, 10 Jun 2015 02:45:32 GMT
5. Server: Werkzeug/0.10.4 Python/2.7.6
6.
7. "successfully update your information"
```

### 3. 查询确认

成功更新字段后查询进行确认

```
1. http GET http://restful.adleida.com:8008/v1/dsp/5576c371c44b1a5756805130
2.
3. HTTP/1.0 200 OK
4. Content-Length: 98
```

```
5. Content-Type: application/json
6. Date: Wed, 10 Jun 2015 02:45:42 GMT
7. Server: Werkzeug/0.10.4 Python/2.7.6
8.
9. {
10.  "burl": "http://example.dsp.com/path/to/bid/",
11.  "id": "5576c371c44b1a5756805130",
12.  "name": "newdspname"
13. }
```

## 4. 删除记录

### 1. 发起 DELETE 请求

删除动作只需要提供 `id` 和 `access_token` 即可完成操作，如下例所示

```
1. http DELETE http://restful.adleida.com:8008/v1/dsp/ access_token:d19a1398-ccf5-4c4
  7-868c-a4abaf24e011 id=5576c371c44b1a5756805130
```

### 2. 确认返回结果

成功删除该条记录后会接收到如下提示信息

```
1. HTTP/1.0 200 OK
2. Content-Length: 38
3. Content-Type: application/json
4. Date: Wed, 10 Jun 2015 03:32:29 GMT
5. Server: Werkzeug/0.10.4 Python/2.7.6
6.
7. "successfully delete your information"
```

### 3. 查询验证

此时便无法查询到该条记录

```
1. http GET http://restful.adleida.com:8008/v1/dsp/5576c371c44b1a5756805130
2.
3. HTTP/1.0 404 NOT FOUND
4. Content-Length: 39
5. Content-Type: application/json
6. Date: Wed, 10 Jun 2015 03:32:34 GMT
7. Server: Werkzeug/0.10.4 Python/2.7.6
8.
9. {
10.  "message": "Not Found",
11.  "status": 404
```

# License

MIT