

为 Apple Ads API 实施 OAuth

管理对 Apple Ads 账号的安全访问

概览

借助 OAuth 2, Apple Ads 广告系列管理 API 用户使用凭证进行身份验证, 以换取访问Token, 进而向 Apple Ads API 发出经过身份验证的请求。OAuth 2 取代了以前版本的 API 中的密钥和证书凭证身份验证。Apple Ads 广告系列管理 API 3 的用户可以继续使用密钥和证书凭证, 直到其证书过期, 但您无法为 API 3 生成新的证书。

使用最新版本 的API 和OAuth 2 的直接优势之一是无需提供或者分享用户登录名和密码凭证即可管理账号的访问权限

此文档将引导您完成以下步骤:

- 邀请具有 API 权限的用户
- 生成一个私钥 - 公钥对
- 从私钥中提取公钥
- 上传公钥
- 创建 client secret
- 请求访问 Token

邀请用户

账号管理员使用以下流程邀请具有 API 权限的用户：

1. 在 [Apple Ads UI](#) 中, 选择“登录”>“Advanced”并以账号管理员身份登录。
2. 在右上角的“用户”菜单中, 选择要邀请加入的用户账号。
3. 选择“账号设置”>“用户管理”。
4. 点按“邀请用户”, 邀请用户加入您的 Apple Ads 组织。
5. 在“用户详细信息”部分中, 输入用户的名字、姓氏和 Apple ID。
6. 在“用户访问”和“角色”部分中, 选择 API 用户角色。对于非 API 角色, 请参阅 [Invite users to your account](#)。
7. 点按“发送邀请”。受邀用户会收到一封包含安全代码的电子邮件。用户登录电子邮件中的安全 Apple URL 并输入所提供的代码, 从而激活用户账号。

生成私钥

API 用户需要创建一个私钥。

如果您使用的是 macOS 或类似 UNIX 的操作系统, 可以直接使用系统原生的 OpenSSL 来创建。

如果您使用的是 Windows 平台, 则需要下载 [OpenSSL](#)。

```
openssl ecparam -genkey -name prime256v1 -noout -out private-key.pem
```

参数	描述
----	----



-name	prime256v1 — Elliptic Curve Digital Signature 算法 (ECDSA) 文件名。
-out	生成存储密钥对的 .pem 文件名。

生成的 private-key.pem 文件示例：

```
-----BEGIN EC PRIVATE KEY-----  
MHcCAQEEIKtnxllRY8nbndBQwT9we4pEULtjpW6O5iwvzLlKcBq4oAoGCCqGSM49  
AwEHoUQDQgAEY58v74eQFyLtu5rtCpeU4NggVSUQSOcHhN744tOgWGc/xXkCSusz  
LaZriCQnnqq4Vx+IscLFcrjBj+ulZzKlUQ==  
-----END EC PRIVATE KEY-----
```

重要说明

请始终确保私钥的安全，切勿共享私钥。如果您的私钥遭泄露，则需要重新创建私钥和client secret，并将其重新上传到您的 [Apple Ads](#) 账号。

提取公钥

使用以下命令从您的私钥中提取公钥：

```
openssl ec -in private-key.pem -pubout -out public-key.pem
```

参数	描述
-in	私钥文件名：private-key.pem
-out	生成存储公钥的 ec256-public-key 文件。

在文本编辑器中打开 public-key.pem 文件并复制公钥，包括开始行和结束行。



上传公钥

请按照以下步骤上传公钥：

1. 在 Apple Ads 用户界面中，选择“账号设置”>“API”。将以上部分中创建的密钥粘贴到公钥字段中。
2. 点击“保存”，一组凭证将会显示在公钥字段上方，之后您就可以使用您的 `clientId`、`teamId` 和 `keyId` 创建 `client secret`。

```
clientId SEARCHADS.aeb3ef5f-0c5a-4f2a-99c8-fca83f25a9
teamId SEARCHADS.hgw3ef3p-0w7a-8a2n-77c8-scv83f25a7
keyId a273d0d3-4d9e-458c-a173-0db8619ca7d7
```

您可以选择“账号设置”>“API”>“编辑”以编辑公钥。

创建 `client secret`

`Client secret`是您使用私钥创建并签署的 JSON web token ([JWT](#))。您的 `client secret` 将被用于验证发送给授权服务器的 Token 请求。只有您和授权服务器知道 `client secret`。

以下示例是一个 Python 3 脚本，它使用您的私钥生成、编码并签署 `client secret`。如果您决定使用其他的编程语言或开源库创建自己的 JWT，请确保您使用的库支持 `elliptic curve` 加密方法。

```
import jwt
import datetime as dt

client_id = 'SEARCHADS.27478e71-3bb0-4588-998c-182e2b405577'
team_id = 'SEARCHADS.27478e71-3bb0-4588-998c-182e2b405577'
key_id = 'bacaebda-e219-41ee-a907-e2c25b24d1b2'
audience = 'https://appleid.apple.com'
alg = 'ES256'

# Define issue timestamp.
issued_at_timestamp = int(dt.datetime.utcnow().timestamp())
# Define expiration timestamp. May not exceed 180 days from issue timestamp.
expiration_timestamp = issued_at_timestamp + 86400 * 180
```



```
# Define JWT headers.
headers = dict()
headers['alg'] = alg
headers['kid'] = key_id

# Define JWT payload.
payload = dict()
payload['sub'] = client_id
payload['aud'] = audience
payload['iat'] = issued_at_timestamp
payload['exp'] = expiration_timestamp
payload['iss'] = team_id

# Path to signed private key.
KEY_FILE = 'private-key.pem'

with open(KEY_FILE, 'r') as key_file:
    key = "".join(key_file.readlines())

client_secret = jwt.encode(
    payload=payload,
    headers=headers,
    algorithm=alg,
    key=key
)
with open('client_secret.txt', 'w') as output:
    output.write(client_secret.decode("utf-8"))
```

Client secret header 描述了Token的类型和其使用的哈希算法：

密钥	描述
alg	签署client secret的算法, 该值必须为 ES256。
kid	该值是您上传公钥时返回的 keyId。

Client secret 包含一个带有声明的payload：

声明	描述
----	----



请求访问Token

授权服务器将会验证您的凭证, 通过验证之后会返回一个访问Token。要接收访问Token, 请从Token端点向授权服务器发出 POST 请求。当访问Token过期时, 请使用同样的流程来获取一个新的Token。

```
curl -X POST \  
-H 'Host: appleid.apple.com' \  
-H 'Content-Type: application/x-www-form-urlencoded' \  
'https://appleid.apple.com/auth/oauth2/token?grant_type=client_credentials&  
client_id=SEARCHADS.27478e71-3bb0-4588-998c-182e2b405577&client_secret=eyJJO  
eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.zl1NiIsImprdSI6ImhOdHBzOi8vYXV0aC5kZXRYu  
YXB  
pLnJpY29oL3YxL2Rpc2NvdmVyeS9rZXlzIiwia2lkIjoiaMmlyZTgyMTA2NzIxZGM4ZmF  
kNzgxNW  
Q3ZmI1NDRhNjJmNzJjMTZmYSJ9.eyJpc3MiOiJodHRwczovL2F1dGguZGV2LmFwaS5  
yaWNvaC8iLCJhdWQiOiJodHRwczovL2lwcy5kZXRYuYXBpLnJpY29oLyIsImhhdCI6MTQ  
5MDglMjc0MSwiZXhwIjoxNDkwODU2MzQxLzJjbnRfaWQiOiI4ODQwMWU1MS05M  
zliLTQ3NzktYjdmNy03YzlmNGIzZjkyYzAiLCJzY29wZSI6ImhOdHBzOi8vaXBzLmRldi5  
hcGkuemlj2gvdjEiLCJyaWNvaF9tc3MiOmsibWVkaWEiOmsicXVvdGEiOjEwLzJhOaHJvd  
HRsZSI6eyJ2YWx1ZSI6MCwid2luZG93IjowfX19fQ.jVq_c_cTzgsLipkJKBjAHzm8KDehW  
4rFA1YgOEQRmqWmBDlEKtpRpDHZeF6ZSOfNH2OlrBWFbIVDV9Th091QFEYrZETZ1IE  
lkoAO14oj4kf8TCmhiG_CtJagvctvloW1wAdgMB1_Eubz9a8oimcODqL7_uTmA5jKFx3ez  
9uoqQrEKZ51g665jSI6NlyeLtj4LrxpI9jZ4zTx1yqqjQxOdoYqjBPhOB06Z5bdiVyhJDRpE  
8ksRCC3kDPS2nsvDAal28sMgyeP8sPvfKvp5sa2UsH78WJmTzeZWcJfX2C2ba3xwRMB  
5LaaVrQZlhj9xjumOMfDpIS1hJI6p5CHZ8w&scope=searchadsorg'
```

请求头	描述
Host	必需。appleid.apple.com
Content-Type	必需。application/x-www-form-urlencoded



响应参数	类型	描述
access_token	字符串	您的 access_token 是调用 Apple Ads 广告系列管理 API 端点的必要条件。请参阅 Calling the Apple Ads API 。 您的 access_token 在 expires_in 指定的秒数内有效。
token_type	字符串	访问Token的类型。该值始终为 Bearer。
expires_in	整数	一小时 (3,600 秒) 的Token生命周期 (TTL)。
scope	字符串	您请求的访问权限, 限制您收到的访问Token的权限。

调用 Apple Ads API

在 HTTP 请求的授权标头中传递您的访问Token。

概览

在调用 API 之前, 您须完成 [Implementing OAuth for the Apple Ads API](#) 中的实施步骤。

调用 Apple Ads 广告系列管理 API时, 请将您的访问Token作为 Bearer 传递到 HTTP 请求的授权 header 中。通过 Bearer 值, API 会得知 Token的持有者拥有相应的权访问权限以执行指定的操作。以下是API 的调用示例:

```
curl "https://api.searchads.apple.com/api/v4/campaigns" \
-H "Authorization: Bearer {access_token}" \
-H "X-AP-Context: orgId={orgId}"
```

标头	描述
Authorization	必需。授权值始终为 Bearer。
X-AP-Context	必需。该值为您的 orgId。 注意: 调用 Get User ACL 和 Get Me Details 时非必需。

要返回 API 调用者的 userId 和 parentOrgId, 请使用 [Get Me Details](#)。

处理错误

HTTP 状态码	错误信息	描述
401	未经授权	Token无效或已过期。
403	禁止	该请求所需的权限高于访问Token提供的权限。



主题

访问控制列表

[Get User ACL](#)

获取 API 有权访问的角色和组织。

object [UserAcl](#)

对 ACL 请求的响应。

object [UserAclListResponse](#)

ACL 调用响应的容器。

[Get Me Details](#)

获取 API 调用者的详细信息。

object [MeDetail](#)

API 调用者标识符。

object [MeDetailResponse](#)

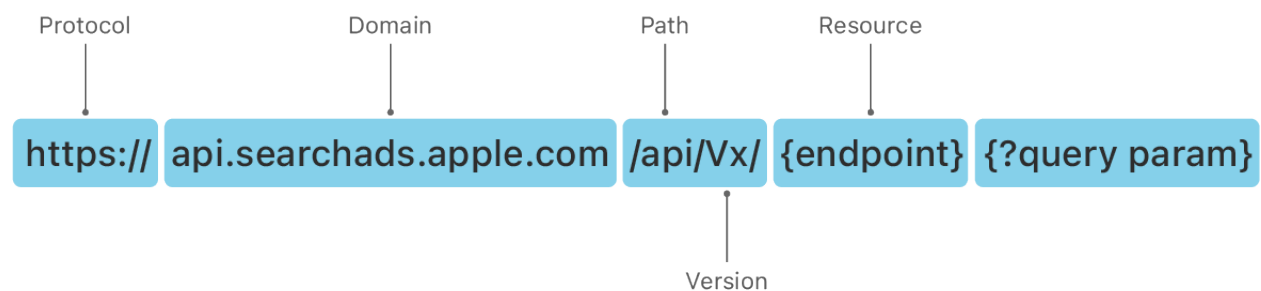
“Me Detail”调用的响应。

使用 Apple Ads API 功能

使用 CRUD 方法调用端点

概览

Apple Ads API 采用 REST 数据模型, 您可以使用 CRUD (创建、读取、更新、删除) 功能来调用端点资源。



Apple Ads API 中的 CRUD 方法包括：

POST

创建新资源、查找资源以及提取报告。

API 支持在 find 调用中使用的 selectors 来过滤资源数据。请参阅 [Selector](#) 对象。

GET

读取某个资源。一些调用通过使用标识符和 limit 和 offset 查询参数来获取一个资源的集合。

相关的关查询参数, 请参阅下面的“使用部分获取”小节。

PUT

按标识符更新一个或多个资源。

DELETE

按标识符删除某个资源。



格式化请求和响应

API 使用 JSON 来表示请求和响应 payload。使用 API 提供的字段、属性和值来创建 JSON 格式的 payload。请注意必填字段。

以下是对 [Create a Campaign](#) 的 POST 请求示例：

```
{
  "orgId": 40669820,
  "adChannelType": "SEARCH",
  "supplySources": ["APPSTORE_SEARCH_RESULTS"],
  "billingEvent": "TAPS",
  "name": "campaign 1",
  "budgetAmount": {
    "amount": "2000",
    "currency": "USD"
  },
  "dailyBudgetAmount": {
    "amount": "300",
    "currency": "USD"
  },
  "adamId": 427916203,
  "countriesOrRegions": ["US","AU"]
}
```

API 响应类似于以下内容：

```
{
  "data": {
    "id": 542317095,
    "orgId": 40669820,
    "name": "campaign 1",
    "budgetAmount": {
      "amount": "2000",
      "currency": "USD"
    },
    "dailyBudgetAmount": {
      "amount": "300",
      "currency": "USD"
    },
    "adamId": 427916203,
    "paymentModel": "PAYG",
    "locInvoiceDetails": null,
    "budgetOrders": [],
    "displayStatus": "ON_HOLD",
    "adChannelType": "SEARCH",
    "supplySources": ["APPSTORE_SEARCH_RESULTS"],
    "billingEvent": "TAPS",
```



```
"startTime": "2020-04-08T10:33:31.650",
"endTime": "2020-04-09T10:33:31.650",
"status": "ENABLED",
"servingStatus": "NOT_RUNNING",
"servingStateReasons": [
  "AD_GROUP_MISSING"
],
"modificationTime": "2020-04-08T18:21:35.133",
"deleted": false,
"sapinLawResponse": "NOT_ANSWERED",
"countriesOrRegions": [
  "AU",
  "US"
],
"countryOrRegionServingStateReasons": {}
},
"pagination": null,
"error": null
}
```

您可以使用端点 URI 响应中返回的 id 来定位某个资源。在上面的示例中, 每当调用该广告系列的资源时, 该 id 都会在整个 API 调用中充当 campaignId。当您使用 [Create an Ad Group](#) 时, 同样的逻辑也适用。API 返回的 id 即为广告组标识符 (adgroupId)。

在其他示例中, 当您使用 [Create Targeting Keywords](#) 和 [Create Ad Group Negative Keywords](#) 端点时, id 来自 URI 中的 campaignId 和 adgroupId。当您使用 [Create Campaign Negative Keywords](#) 时, id 来自 campaignId 资源。

使用部分更新

部分更新是更新对象子集的属性, 而不是整个对象集的属性。API 支持大多数 PUT 调用的部分更新。在您的广告系列请求 payload 中使用广告系列对象信封。其他对象不需要 JSON 信封。

例如, 以下 payload 是对广告系列名称 countriesOrRegions 的更新:

PUT <https://api.searchads.apple.com/api/v4/campaigns/<campaignId>>

```
{
  "campaign": {
    "name": "new campaign name",
    "countriesOrRegions": [
      "US",
      "CA"
    ]
  }
}
```



使用“部分获取”

在获取数据时，“部分获取”用于指明需要 API 返回哪些属性。使用 [Selector](#) 来选取每条返回记录中的字段，通过在 field 查询参数中指定相应的字段，可以对某些 GET 端点使用“部分获取”。例如，您可以选择仅返回每个广告系列的 id，如以下示例所示：

```
GET https://api.searchads.apple.com/api/v4/campaigns?
fields=id,name,countriesOrRegions
```

```
{
  "data": [
    {
      "id": 542370539,
      "name": "Multicountry campaign example 1",
      "countriesOrRegions": [
        "FR"
      ]
    },
    {
      "id": 542370549,
      "name": "Multicountry campaign example 2",
      "countriesOrRegions": [
        "GB"
      ]
    },
    {
      "id": 542370559,
      "name": "Multicountry campaign example 3",
      "countriesOrRegions": [
        "CA",
        "JP",
        "NZ",
        "AU"
      ]
    }
  ],
  "pagination": {
    "totalResults": 3,
    "startIndex": 0,
    "itemsPerPage": 20
  }
}
```



```
},  
  "error": null  
}
```

使用查询参数

通过在方法调用中将查询参数附加到 URI, 您可以在调用 [Search for iOS Apps](#)、[Search for Geolocations](#) 和 [Get a List of Geolocations](#) 端点时使用各类查询参数。

GET 调用使用查询参数来控制 [Pagination](#) 并为返回的响应分配限制。您可以将这些查询参数与要获取的数据结合起来, 如下示例所示:

```
GET https://api.searchads.apple.com/api/v4/search/geo?entity=Locality&limit=1000
```

主题

响应

[Usability and Errors](#)

其他用于参考文档。
