facebook

Facebook广告API
入门到精通?

Yu LI
Solutions Engineer
APAC

了解广告API

https://developers.facebook.com/docs/marketing-apis

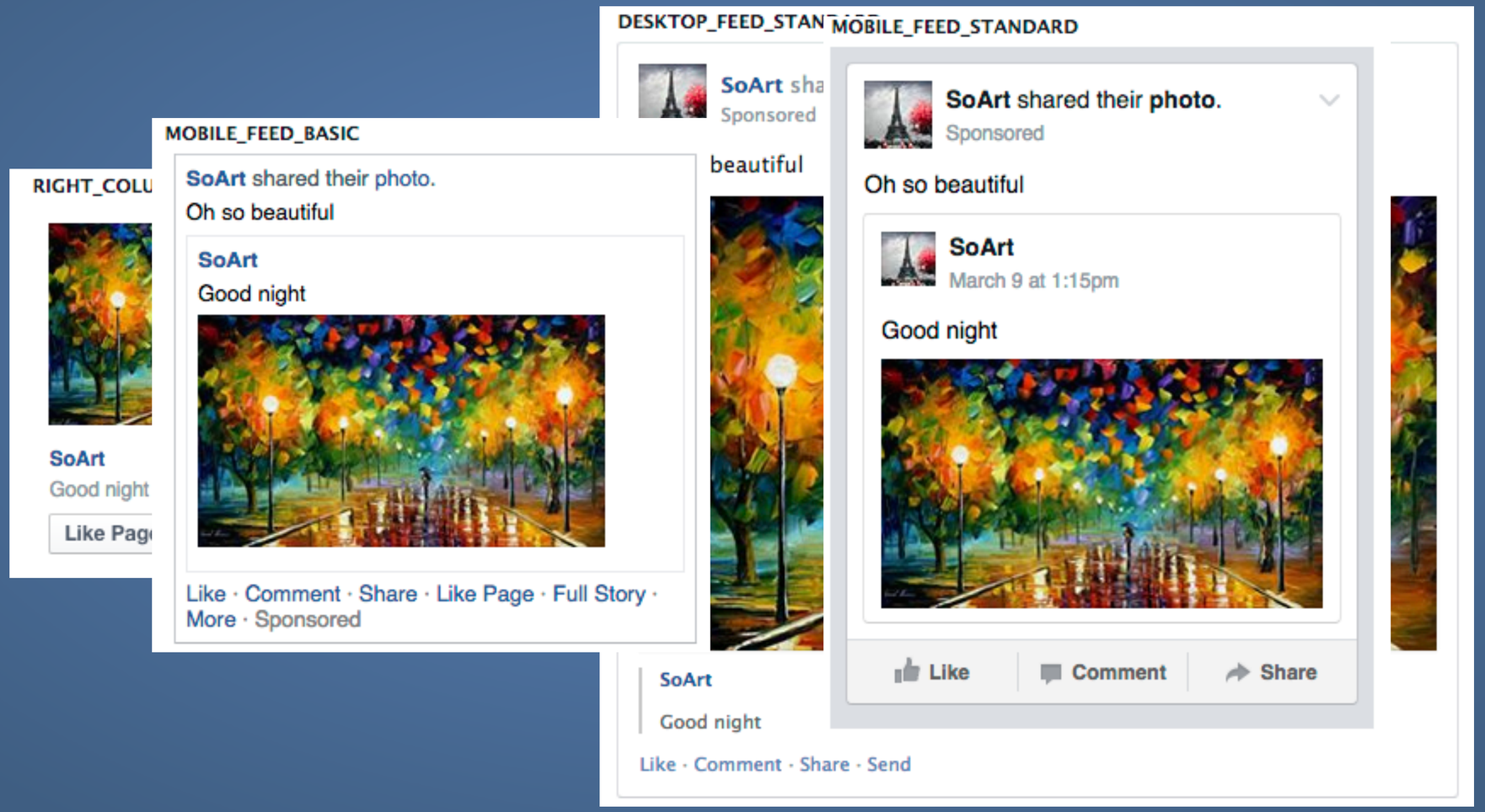
technology to help

Automate

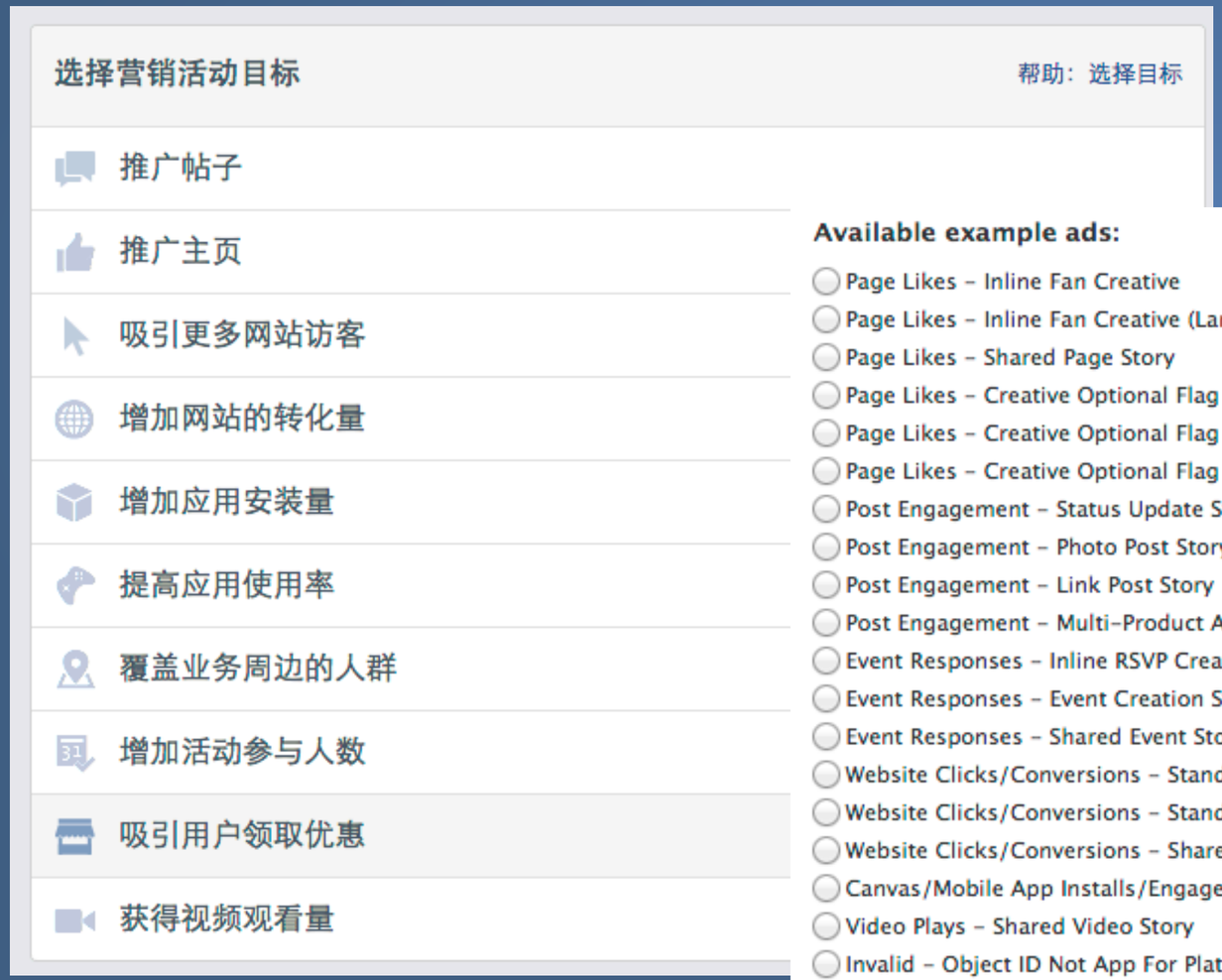
Integrate

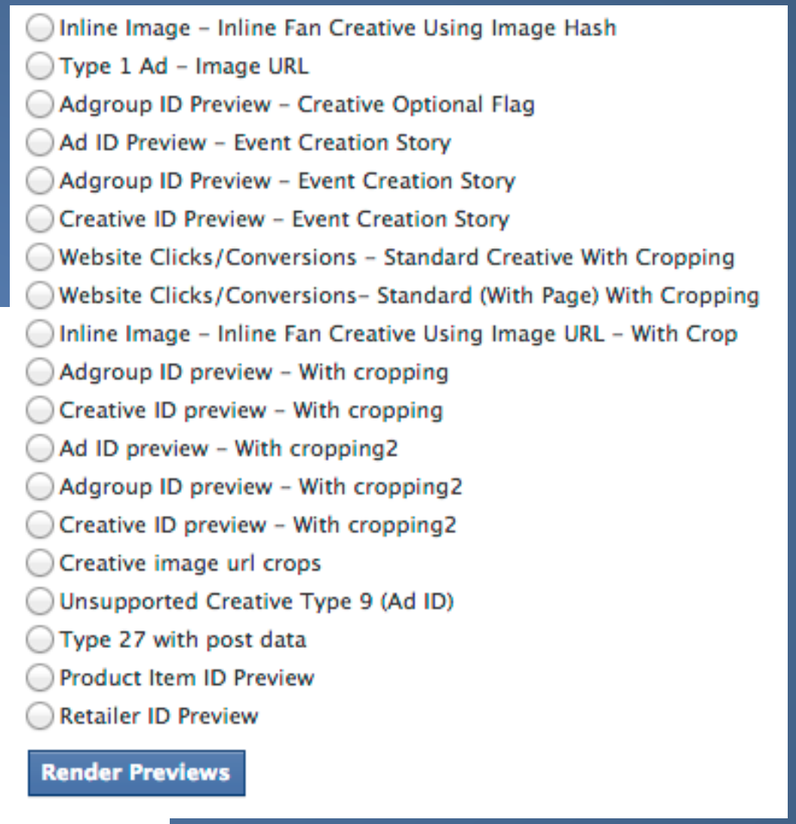
Scale

Facebook广告系统：跨设备平台，面向真实用户



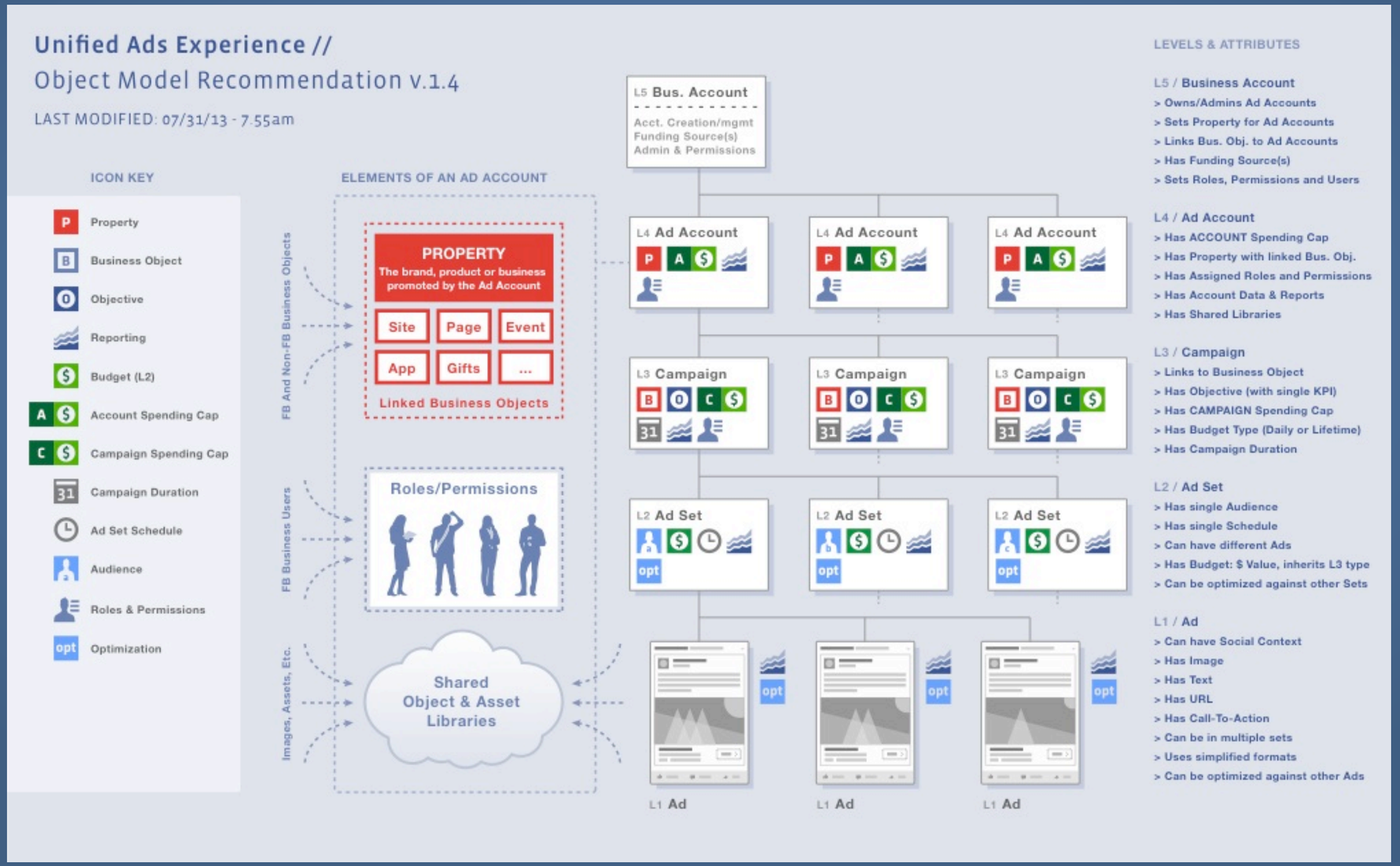
Facebook广告系统：丰富的营销目标与表现形式





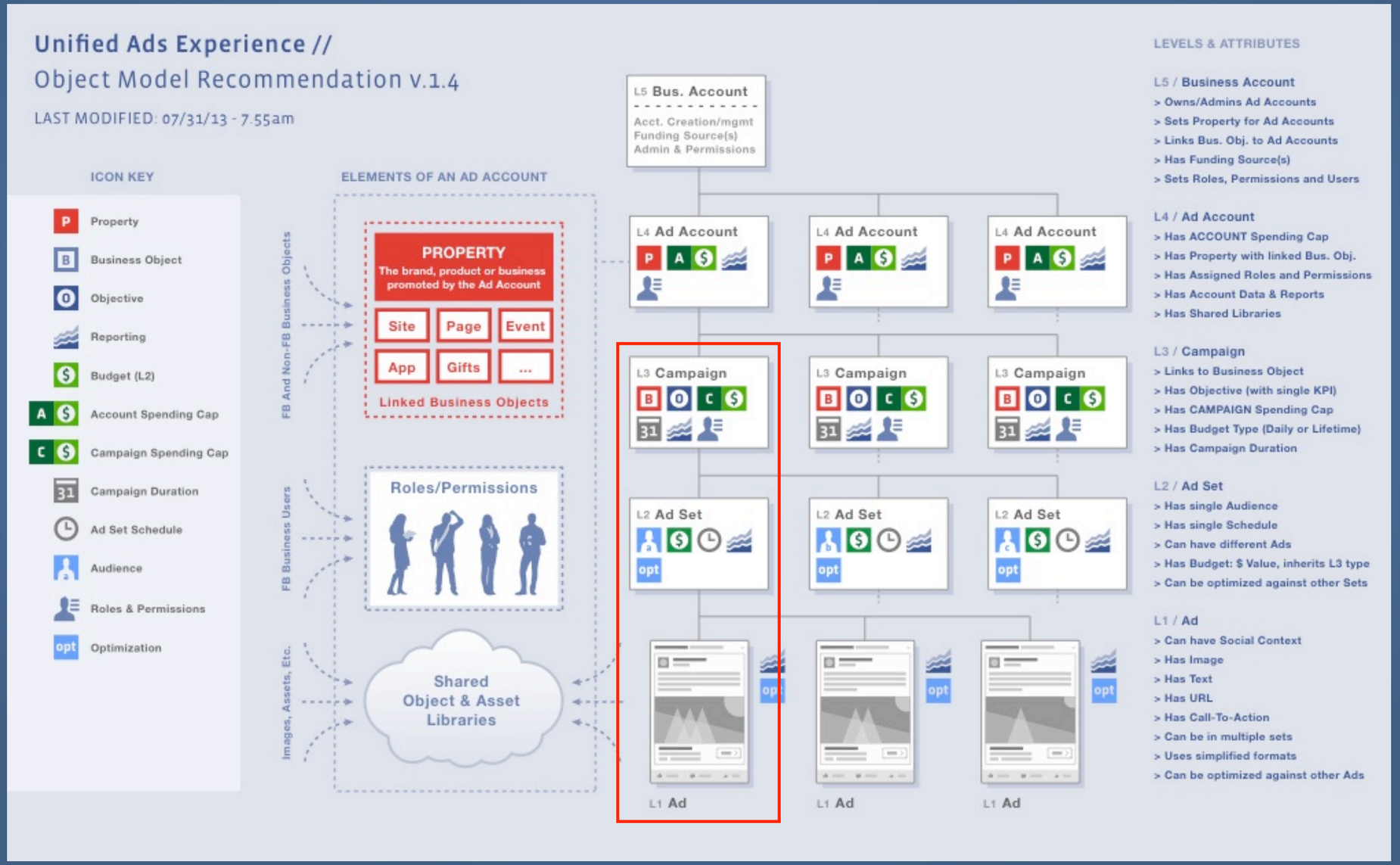
Facebook广告系统：基于真实用户投放系统

Facebook广告系统的架构



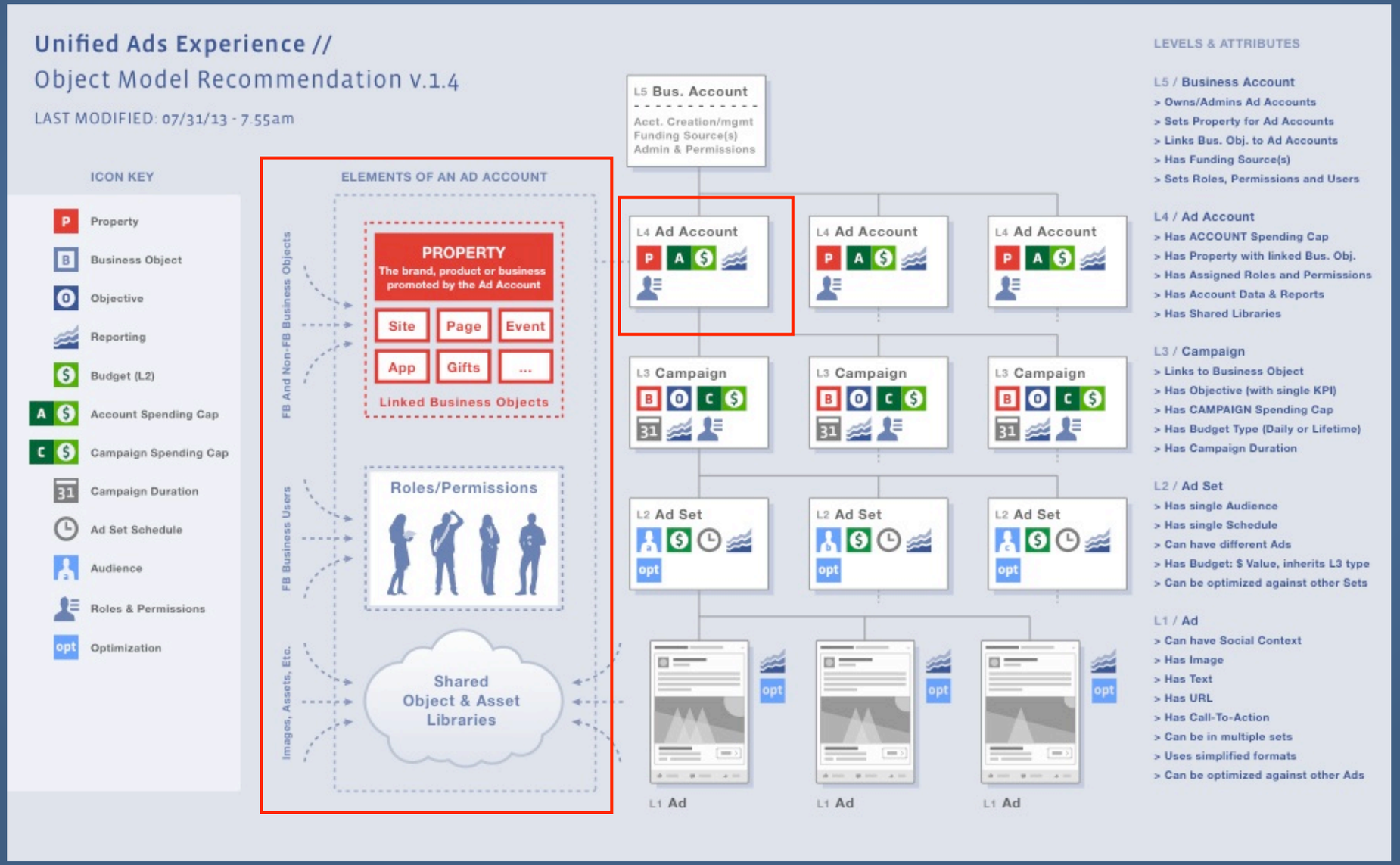
广告的数据结构类似于树

Campaign x Ad Set x Ad，以ID相连



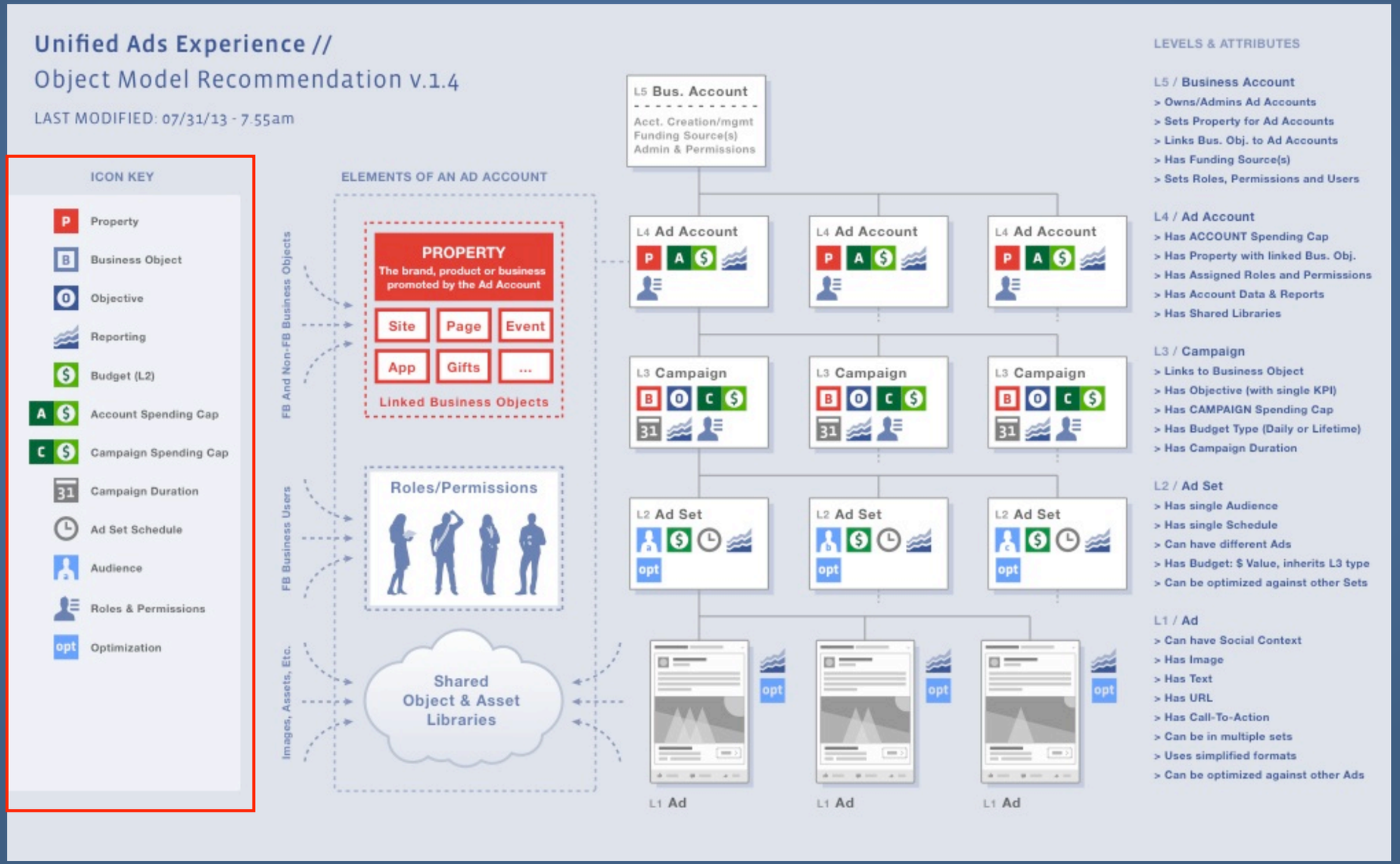
位于中心地位的广告账户

拥有图片等各类资产，关联Page，App，Events，为FB用户指定角色



注意各类要素存在的层级

Report? Budget? Objective? Audience?



Facebook的策略： API优先

- 我们是行动优先，API优先（mobile-first, API first）.
  - 产品开发是按照如下的顺序
    - 首先是API
    - 然后是Power Editor
    - 最后是Ads Manager
- "我可以等待Power Editor"
  - 最终不是所有的API都有UI支持
  - 早起的鸟儿有食吃

API能做什么?

- API能做到任何UI能做的事情
  - 创建广告，更改竞价，调整受众等等
- API还能完成已发布但是未出现在UI中的功能
  - 例如DPA for mobile广告，例如精细化受众定义
- 需要使用API的情况
  - 大规模自动化的广告购买与调整
  - 精细化的广告策略
  - 使用新的广告产品，抢占先机
  - 程序化获取广告报表数据（insights）

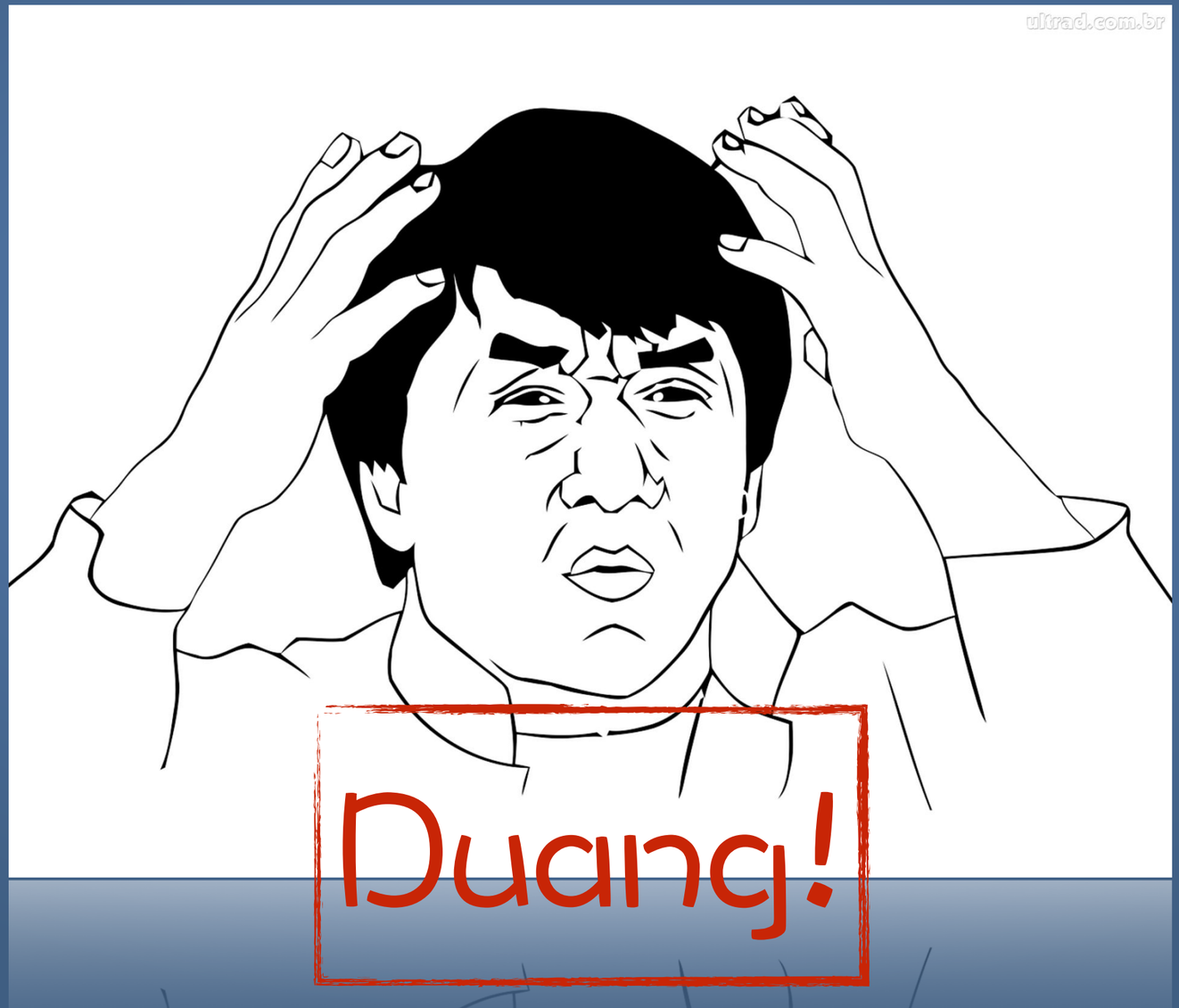
获得广告API的使用权限

https://developers.facebook.com/docs/marketing-api/using-the-api

Facebook广告API现已开放

默认级别	基本级别	高级别
Development Access	**Basic Access**	**Standard Access**
• 所有开发者 • 最多能同时管理5个账号，并且开发者至少是这些账号的advertiser • 每分钟对API的调用次数有很大限制	• 花费超过250美金 • 已超过1,000次API 调用 • 通过API调用的质量调查 • 最多能同时管理25个账号 • 每分钟对API的调用次数有一定限制	• 通过商业背景调查 • 对管理账号数没有限制 • 每分钟对API的调用次数基本没有限制

我们需要立刻升级到最高级么？



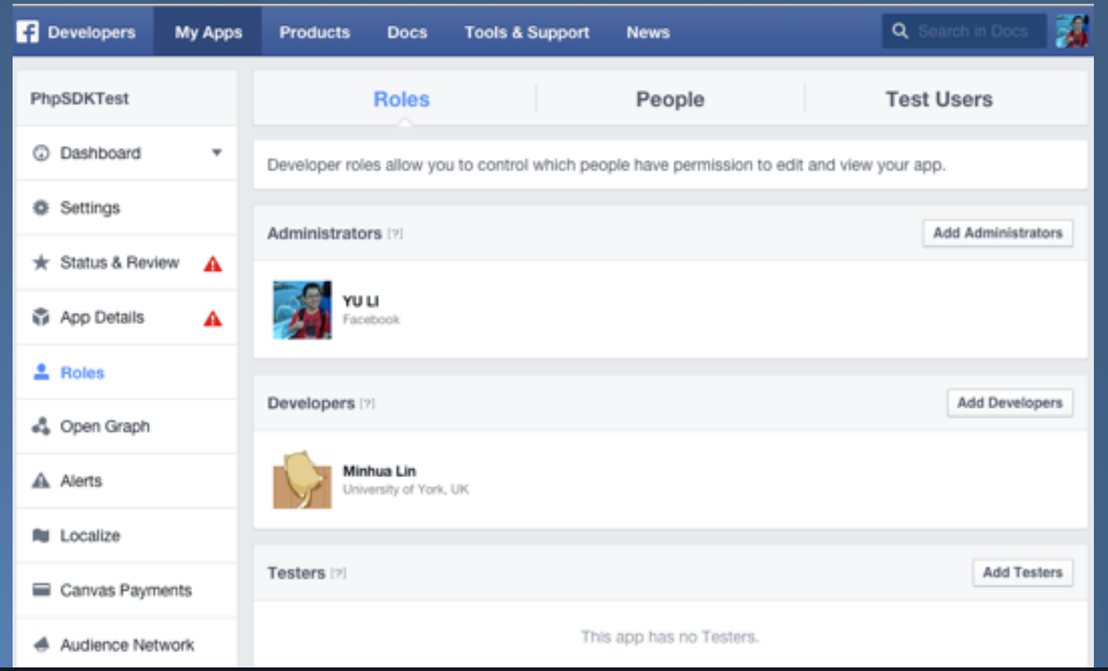
答案是：没有必要

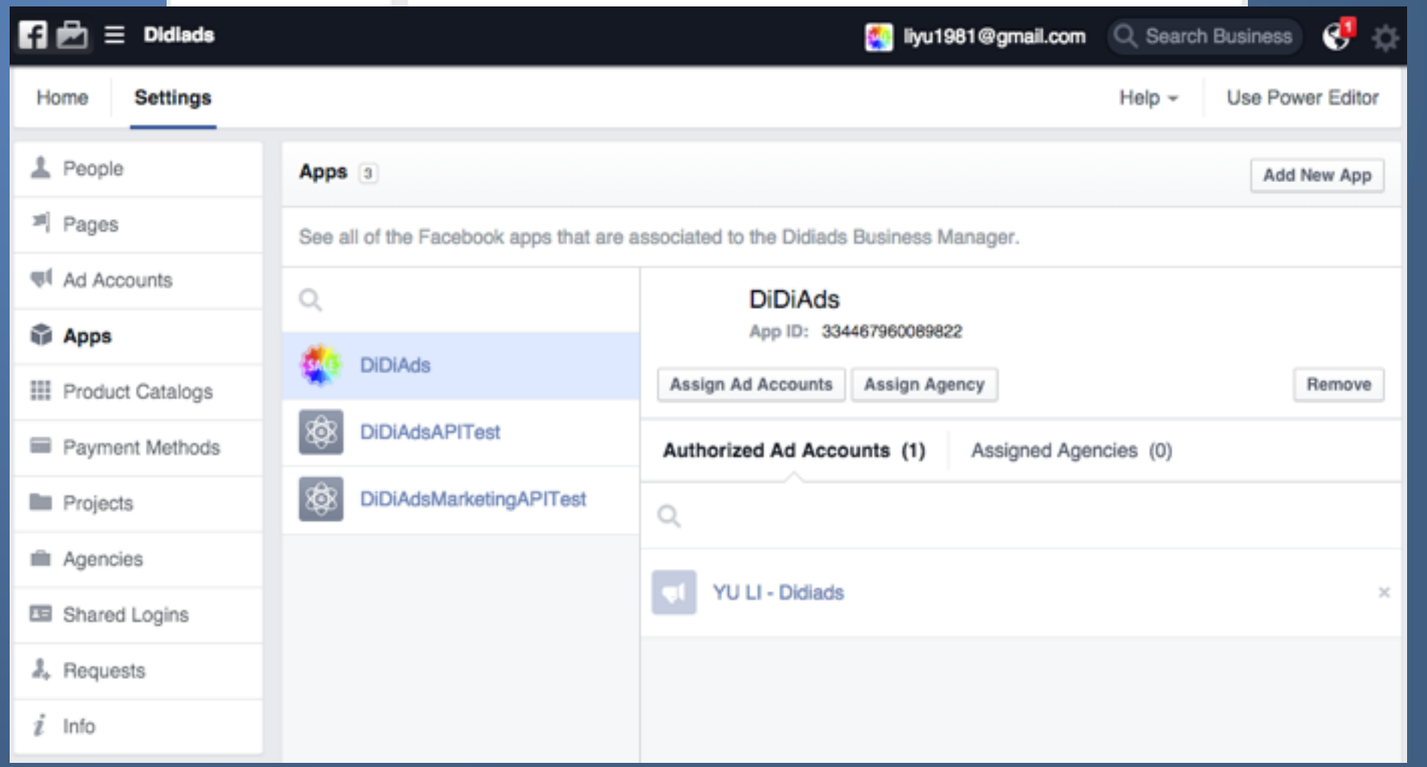
- Developer级别已经足够开发使用
  - 广告账户的限制5个
  - 调用频率限制足够工具类使用
  - API调用频率限制对双方都是很好的保护措施
- 升级到Basic Access?
  - 开发阶段没有必要
  - 需要提交人工Review，提供清晰的描述与UI
- 升级到Standard Access
  - 需要提交申请等待审核
  - https://www.facebook.com/business/standardadsapi

准备条件

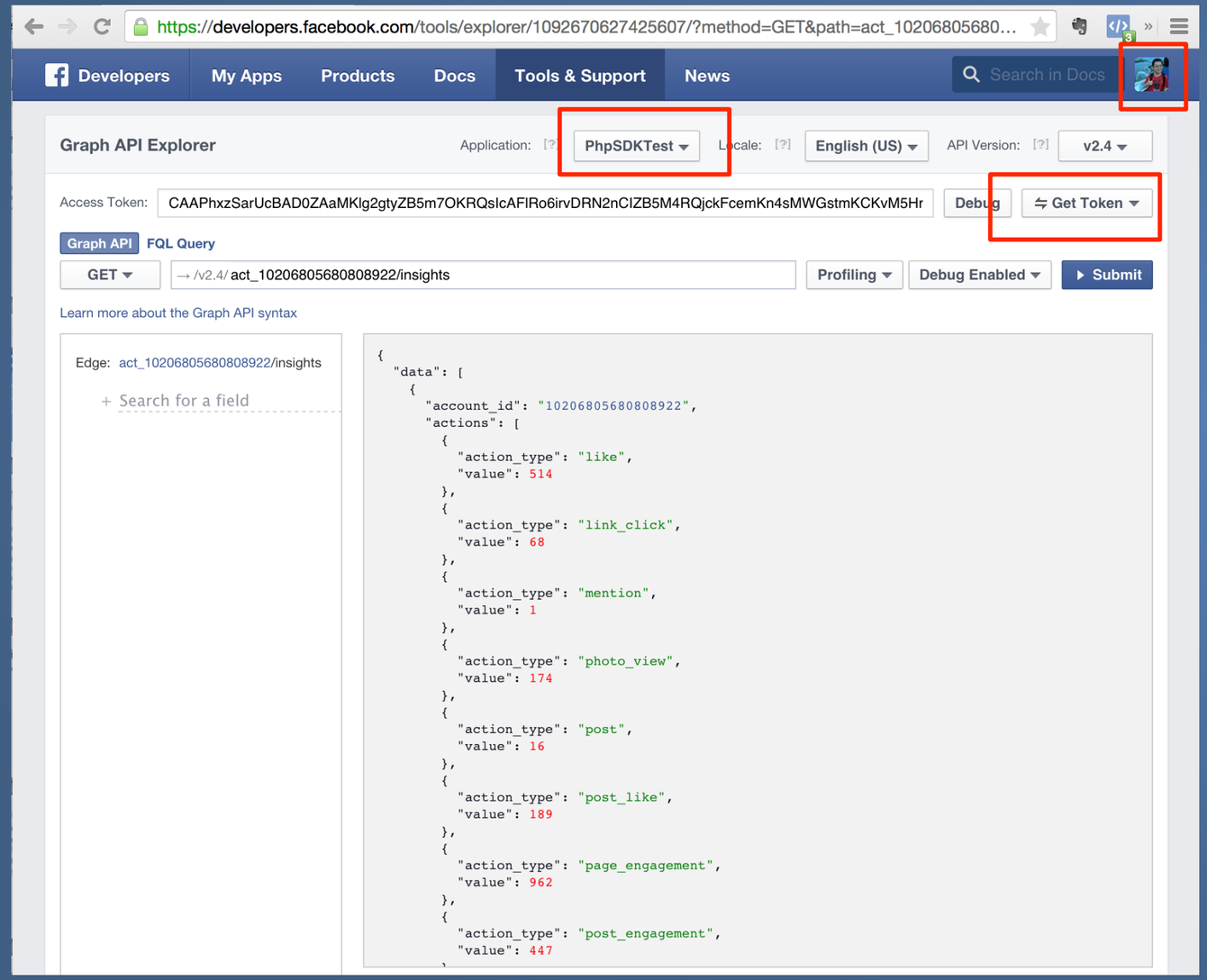
- 广告账户（Ad Account）
- Facebook App
- Facebook个人账号

- 授权
  - Facebook个人账号至少成为广告账户的Advertiser，Facebook App的Developer
  - Facebook App关联广告账户

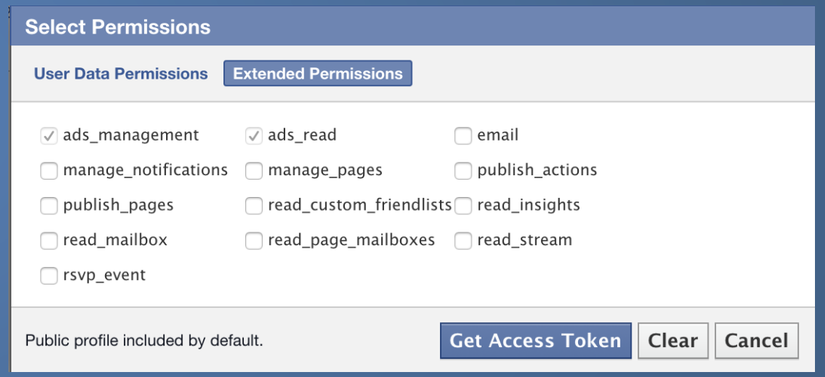




立即尝试广告API

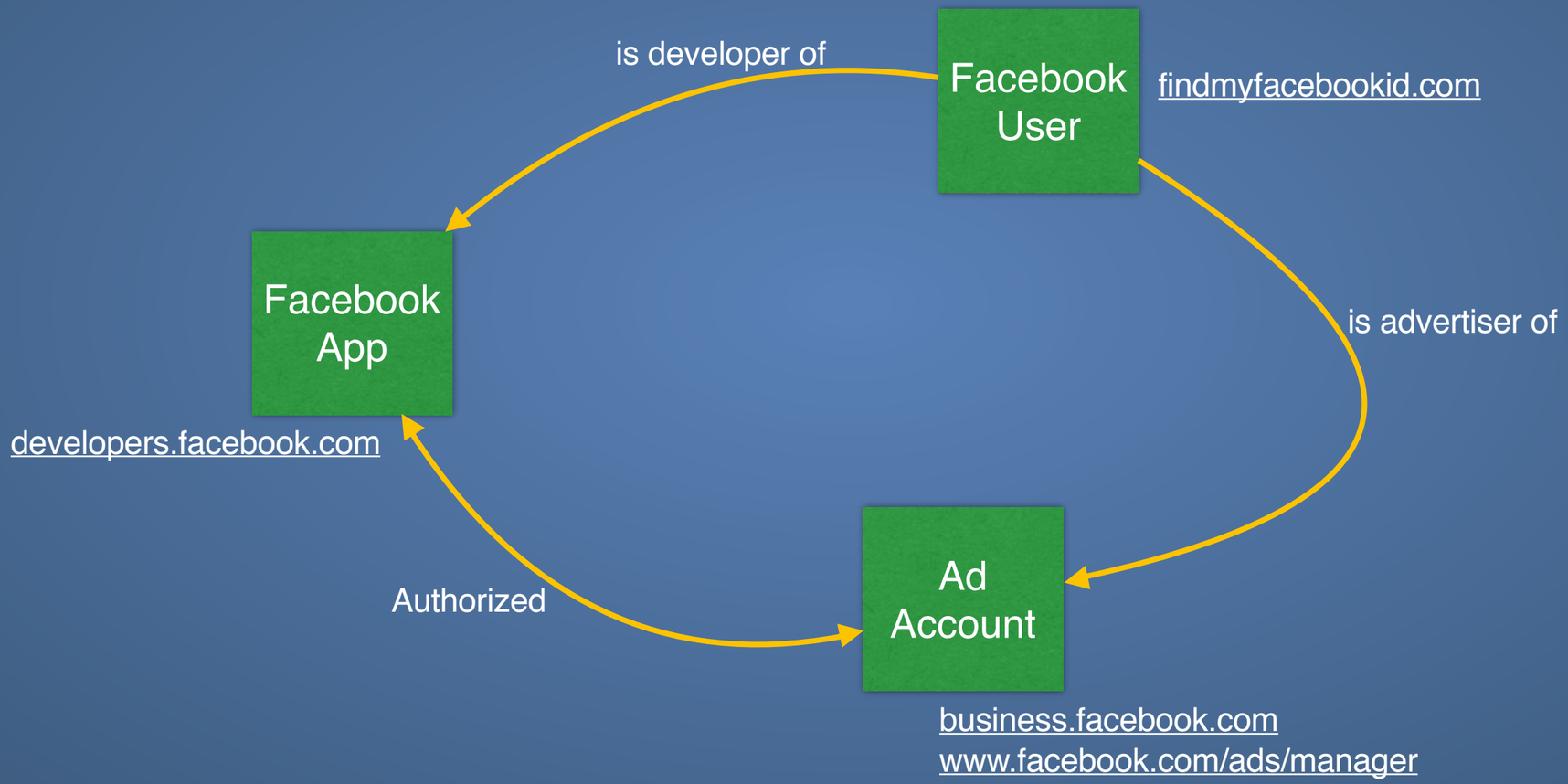


- 注意使用正确的FB账户
- 注意选择您的FB APP
- 注意Get Token时获取Ads Manage和Ads Read权限



OAuth error? Can not find ad account error?

- 检查授权关系



访问API的途径

https://developers.facebook.com/docs/marketing-api/reference

https://github.com/facebook/facebook-php-ads-sdk

https://github.com/facebook/facebook-python-ads-sdk

广告API入门

https://developers.facebook.com/docs/marketing-api/using-the-api

欲速则不达
Facebook Graph API使用入门

https://developers.facebook.com/docs/graph-api/using-graph-api

Facebook Graph API的结构

- Graph API是基于HTTP/HTTPS协议的API
- 版本号 vX.X
  - https://developers.facebook.com/docs/apps/changelog
- 组成要素
  - 节点node: http://graph.facebook.com/v2.3/<fbid>
  - 边Edge: http://graph.facebook.com/v2.3/<fbid>/feed
  - 域Field: http://graph.facebook.com/v2.3/<fbid>?fields=birthday
- 返回值
  - JSON文档
  - 错误也是JSON文档形式

Access Token

- https://developers.facebook.com/docs/facebook-login/access-tokens
- 每一次API访问都必须提交Access Token/User Access Token
- 怎样获得?
  - 测试? 使用 https://developers.facebook.com/tools/explorer/ 是最快的办法
    - 注意选择正确的App
  - 程序使用
    - 通过Javascript SDK与Facebook SDK见 https://developers.facebook.com/docs/facebook-login/access-tokens#usertokens
    - 通过http request

http://www.mywebsite.com发起请求

客户端

```
https://www.facebook.com/dialog/oauth?
client_id=<appid>&redirect_uri=http://www.mywebsite.com/fbauth
```

从结果中获得code(as c)，发起请求

服务器端

```
https://graph.facebook.com/v<X.X>/oauth/access_token?
client_id=<appid>&redirect_uri=http://www.mywebsite.com/
fbauth&client_secret=<appsecret>&code=<c>
```

从结果中获得access token

服务器端

案例：第1步，发起请求

```
var FBLogin = React.createClass({
  render: function() {
    return (
      <div>
        <ButtonToolbar>
          <Button bsStyle='primary' bsSize='large' onClick={this.login}>
            Login into your Facebook Account
          </Button>
        </ButtonToolbar>
      </div>
    );
  },

  login: function() {
    window.open('https://www.facebook.com/dialog/oauth?' +
                'client_id=' + this.props.appid +
                '&redirect_uri=' + window.location.origin + '/fblogin-cb',
                'FB Login', 'width=600,height=400');
  }
});
```

案例：第2-3步，获得Code后发起请求Access Token

```
@route('/fblogin-cb')
def fblogin_cb():
    if not request.params.code:
        return 'Ooops, do not find code in callback URL, please close this.'
    import urllib2
    url = fblogin_at_url_tpl.format(setting['appid'], fblogin_cb_url,
                                    setting['appsecret'], request.params.code)
    resp = urllib2.urlopen(url)
    respcontent = resp.read()
    with open('frontend/oauth_ok.tpl', 'r') as f:
        oauth_ok_page_tpl = f.read()
    return template(oauth_ok_page_tpl, respcontent=respcontent)
```

```
<!DOCTYPE html>
<html><head>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<p>FB Auth OK, will go back soon ...</p>
<script>
setTimeout(function() {
  window.opener.loginOk('{{!respcontent}}');
  window.close();
}, 1000);
</script>
</body></html>
```

案例：获得Access Token之后使用Graph API

```
curl -G -d
"access_token=CAAPhxzSarUcBAB42GF4y5qGMUygy2QNCGZAqMHWZBN0Y5
SUYORDKgl4wTlVOnQwnW2jsKAW8NN4npAhrkMJZAAf0IxWp4aIpzUOdBH7CX
Qd8d2IMU2QufdCeG5I5sz3YZA7uqDK6YlxMRIs8Cu0cX2ZAO9NlVHlMUUi6K
S1NHIlzV3XiqKFCouZBeJA93418GPOxEaOu514oCk8Vd3mmoA" -s
https://graph.facebook.com/v2.3/me?fields=id,name,adaccounts
```

使用广告API

http://www.mywebsite.com发起请求

```
https://www.facebook.com/dialog/oauth?
client_id=<appid>&redirect_uri=http://www.mywebsite.com/
fbauth&scope=ads_management,ads_read
```

客户端

从结果中获得code(as c)，发起请求

服务器端

```
https://graph.facebook.com/v<X.X>/oauth/access_token?
client_id=<appid>&redirect_uri=http://www.mywebsite.com/
fbauth&client_secret=<appsecret>&code=<c>
```

从结果中获得access token

服务器端

获得Access Token之后使用Ads API

```
curl -G -d
"access_token=CAAPhxzSarUcBAB42GF4y5qGMUygy2QNCGZAqMHWZBN0Y5SUYORDKgl4
wTlVOnQwnW2jsKAW8NN4npAhrkMJZAAf0IxWp4aIpzUOdBH7CXQd8d2IMU2QufdCeG5I5s
z3YZA7uqDK6YlxMRIs8Cu0cX2ZAO9NlVHlMUUi6KS1NHIlzV3XiqKFCouZBeJA93418GPO
xEaOu514oCk8Vd3mmoA" -s
https://graph.facebook.com/v2.3/act_1020680567676821
```

诀窍

https://graph.facebook.com/v2.4/<id>?metadata=1

怎样使用SDK

```php
use FacebookAds\Api;
use FacebookAds\Object\AdAccount;
use FacebookAds\Object\AdAccountFields;

Api::init($app_id, $app_secret, $access_token);

$account = new AdAccount('act_<ACCOUNT_ID>');
$fields = array(
  AdAccountFields::NAME,
  AdAccountFields::BALANCE,
);

$params = array(
  AdCampaignFields::STATUS => array(AdCampaign::STATUS_PAUSED),
);

$account->read($fields);

// Output account name
echo $account->{AdAccountFields::NAME}.PHP_EOL;

// Output account balance
echo $account->{AdAccountFields::BALANCE}.PHP_EOL;
```

```python
from facebookads.api import FacebookAdsApi
from facebookads.objects import AdAccount

FacebookAdsApi.init(
    app_id,
    app_secret,
    access_token
)

account = AdAccount('act_<ACCOUNT_ID>')

account.remote_read(fields=[
    AdAccount.Field.name,
    AdAccount.Field.balance
])
print(account[AdAccount.Field.name])
print(account[AdAccount.Field.balance])
```

创建广告四部曲

1.2.3.4

1. 创建campaign

   https://developers.facebook.com/docs/marketing-api/adcampaign

2. 创建ad set

   https://developers.facebook.com/docs/marketing-api/adset

3. 创建ad creative

   https://developers.facebook.com/docs/marketing-api/adcreative

4. 创建ad

   https://developers.facebook.com/docs/marketing-api/adgroup

步骤0: 上传image，创建page post，etc

- 例如上传image
  - https://developers.facebook.com/docs/marketing-api/adimage

```
curl \
  -F "test.zip=@test.zip" \
  -F "access_token=<ACCESS_TOKEN>" \
"https://graph.facebook.com/<API_VERSION>/act_<AD_ACCOUNT_ID>/adimages"
```

- 创建page post
  - 多图广告（Carousal Ads）必须
  - https://developers.facebook.com/docs/marketing-api/guides/carousel-ads/v2.3#pageFirst

```
def create_campaign(self, accountid, name):
    campaign = AdCampaign(parent_id=accountid)
    campaign[AdCampaign.Field.name] = name
    campaign[AdCampaign.Field.objective] = (
        AdCampaign.Objective.mobile_app_engagement
    )
    campaign.remote_create()
    return campaign[AdCampaign.Field.id]
```

```
def create_ad_set(self, accountid, name, dailybudget, campaignid,
                  bidinfo, targeting, appinfo):
    logger.debug('bidinfo:' + str(bidinfo))
    pdata = {
        AdSet.Field.name: name,
        AdSet.Field.bid_type: AdSet.BidType.cpc,
        AdSet.Field.bid_info: bidinfo,
        AdSet.Field.status: AdSet.Status.active,
        AdSet.Field.daily_budget: dailybudget,
        AdSet.Field.campaign_group_id: campaignid,
        AdSet.Field.promoted_object: {
            'application_id': appinfo['fbapplication_id'],
            'object_store_url': appinfo['appstore_link']
        },
        AdSet.Field.targeting: targeting,
    }
    adset = AdSet(parent_id=accountid)
    adset.remote_create(params=pdata)
    return adset[AdSet.Field.id]
```

```
def create_creative(self, accountid, name, imagehash, message, appinfo):
    link_data = LinkData()
    link_data[LinkData.Field.link] = appinfo['appstore_link']
    link_data[LinkData.Field.message] = message
    link_data[LinkData.Field.image_hash] = imagehash
    call_to_action = {'type': 'USE_MOBILE_APP'}
    call_to_action['value'] = {
        'link': appinfo['appstore_link'],
        'app_link': appinfo['app_deep_link'],
        'application': appinfo['fbapplication_id'],
        'link_title': appinfo['app_name']
    }
    link_data[LinkData.Field.call_to_action] = call_to_action
    object_story_spec = ObjectStorySpec()
    object_story_spec[ObjectStorySpec.Field.page_id] = appinfo['fbpage_id']
    object_story_spec[ObjectStorySpec.Field.link_data] = link_data
    creative = AdCreative(parent_id=accountid)
    creative[AdCreative.Field.name] = name
    creative[AdCreative.Field.object_story_spec] = object_story_spec
    creative.remote_create()
    return creative[AdCreative.Field.id]
```

```
def create_ad(self, accountid, name, adsetid, creativeid, appinfo):
    adgroup = AdGroup(parent_id=accountid)
    adgroup[AdGroup.Field.name] = name
    adgroup[AdGroup.Field.campaign_id] = adsetid
    adgroup[AdGroup.Field.creative] = {'creative_id': str(creativeid)}
    adgroup[AdGroup.Field.objective] = \
        AdGroup.Objective.mobile_app_engagement
    tracking_specs = [
        {'action.type': ['mobile_app_install'],
         'application': [appinfo['fbapplication_id']]},
        {'action.type': ['app_custom_event'],
         'application': [appinfo['fbapplication_id']]}
    ]
    if not (appinfo['fboffsitepixel_id'].strip() == 'null'):
        tracking_specs.append({
            'action.type': ['offsite_conversion'],
            'offsite_pixel': [appinfo['fboffsitepixel_id']]
        })
    adgroup[AdGroup.Field.tracking_specs] = tracking_specs
    adgroup[AdGroup.Field.status] = AdGroup.Status.active
    adgroup.remote_create()
    return adgroup[AdGroup.Field.id]
```

object_story_spec

https://developers.facebook.com/docs/marketing-api/adcreative

object_story_spec	object	The page id and the content to create a new unpublished page post specified using one of `link_data`, `photo_data`, `video_data`, `offer_data`, `text_data` or `template_data`.
page_id	int	ID of a Facebook page. An unpublished page post will be created on this page. User must have Admin or Editor role for this page.
link_data	object	Content for creating a link page post or a multi product ad
photo_data	object	Content for creating a photo page post.
video_data	object	Content for creating a video page post.
offer_data	object	Content for creating an offer page post
text_data	object	Content for creating a status page post.
template_data	object	Content for creating a dynamic ad template creative

bid_info

https://developers.facebook.com/docs/marketing-api/adset

bid_info	JSON object	A dictionary of {objective}: {value} that you place on your bid, based on the `bid_type`. Values are defined in your currency's minimum denomination: For `CPM`, use `{'IMPRESSIONS': <value>}` For `CPC`, use `{'CLICKS': <value>}` For `ABSOLUTE_OCPM`, use `{'ACTIONS':<value>, 'REACH':<value>, 'CLICKS':<value>, 'SOCIAL':<value>}` For `CPA`, use `{'ACTIONS': <value>}` Impressions min value: 1c Reach min value: 2c Clicks min value: 1c Action min value: 1c Social min value: 1c	Yes if `is_autobid` not true
bid_type	string	The bid type of the set. All ad groups of this set use this bid type once it is set. Possible values: CPC CPM MULTI_PREMIUM ABSOLUTE_OCPM CPA	Yes

定义受众
Targeting Specs

https://developers.facebook.com/docs/marketing-api/targeting-specs

Facebook广告的特色功能：Targeting

- 不同于关键字广告，Facebook广告使用Targeting来定义受众
  - 是人的行为而不是关键词
- Facebook Targeting的能力
  - https://developers.facebook.com/docs/marketing-api/targeting
  - 人口学 Demographic
  - 用户兴趣 Interest
  - 用户行为 Behaviour
  - 社交联系 Connection

Targeting Spec

- https://developers.facebook.com/docs/marketing-api/targeting-specs
- JSON格式

```
curl \
-X POST \
-F "name=My AdSet" \
-F "bid_type=CPC" \
-F "bid_info={'CLICKS': 150}" \
-F "campaign_status=ACTIVE" \
-F "daily_budget=2000" \
-F "campaign_group_id=<CAMPAIGN_GROUP_ID>" \
-F "targeting= { \
          'geo_locations': { \
            'countries':['US'], \
          }, \
          'behaviors':[ \
            {'id':6004386044572,'name':'Android Owners (All)'}, \
            {'id':6007101597783,'name':'Business Travelers'}, \
          ], \
        }" \
-F "access_token=<ACCESS_TOKEN>" \
"https://graph.facebook.com/<API_VERSION>/act_<AD_ACCOUNT_ID>/adcampaigns"
```

怎样熟悉Targeting Spec?

- 练习，多加练习
- 创建广告时使用 adgroup_status = PAUSED
- 使用reach estimate
  - https://developers.facebook.com/docs/marketing-api/reachestimate

```
curl -G \
-d "currency=USD" \
-d "targeting_spec=___" \
-d "access_token=___" \
"https://graph.facebook.com/<API_VERSION>/act_<AD_ACCOUNT_ID>/reachestimate"
```

- 广告预览
  - https://developers.facebook.com/docs/marketing-api/ad-preview
  - https://developers.facebook.com/docs/marketing-api/ad-preview-plugin

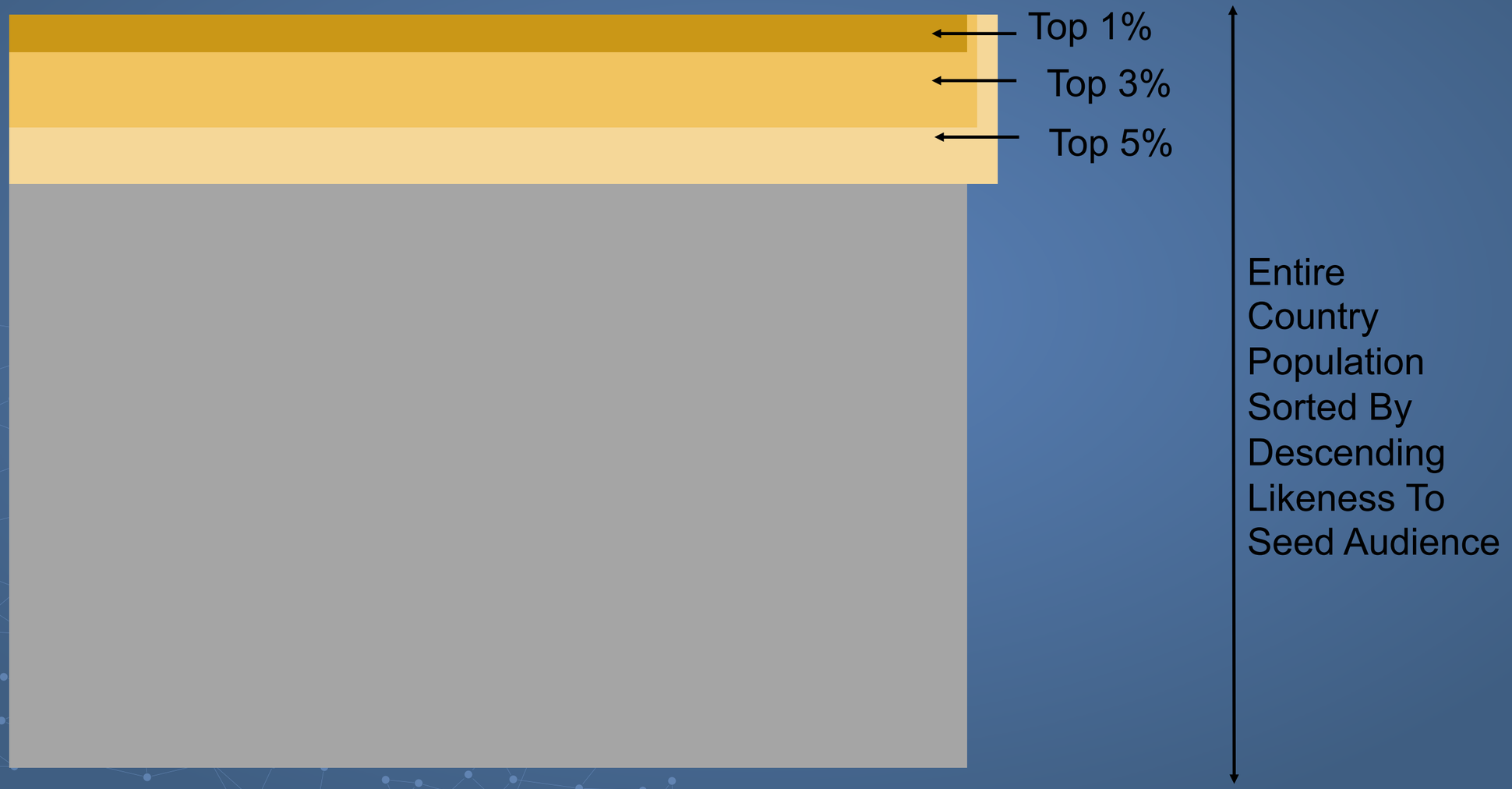
案例分析

http://a.url.can/go?here

Scale While Maintaining CPI

Increasing Reach While Maintaining CPI/ROI

Tiered Lookalike Strategy



Manual Steps

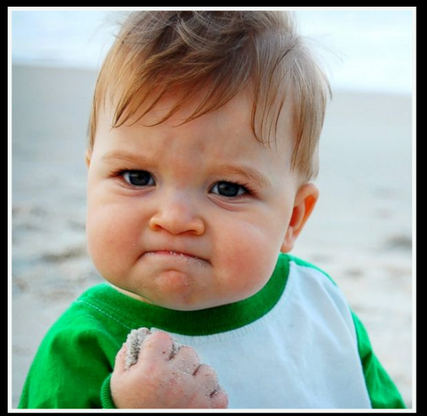
1. Create a custom audience
2. Create 1% lookalike
3. Repeat for 3% and 5%
4. Create adset using 1% lookalike
5. Create adset using 3% lookalike, exclude 1% lookalike, bid lower
6. Create adset using 5% lookalike, exclude 3% lookalike, bid lower
7. Don't forget cross-device tracking!
8. Repeat 1- 7 for other seed audiences
9. Repeat 1- 8 for other countries

Create Tired LAL

```
for tier in range(1, tiers + 1):
    lookalike = CustomAudience(parent_id=accountid)
    lookalike[LookalikeAudience.Field.name] = \
        '{0} LAL {1}'.format(name, tier)
    lookalike[LookalikeAudience.Field.origin_audience_id] = seed_id
    lookalike[LookalikeAudience.Field.lookalike_spec] = {
        'ratio': '{:.2f}'.format(tier / 100.0),
        LookalikeAudience.Field.LookalikeSpec.country: country,
    }
    lookalike.remote_create()
```

Create Ad Set

```
for tier in range(1, tiers + 1):
    ad_set = AdSet(parent_id=accountid)
    ad_set[AdSet.Field.bid_info] = tiered_bidinfos[tier - 1]
    audience = tiered_lookalikes[tier - 1]
    exclude_audience = None
    if tier != 1:
        exclude_audience = tiered_lookalikes[tier - 2]
    targeting = {
        TargetingSpecsField.custom_audiences: [{
            'id': audience[CustomAudience.Field.id],
            'name': audience[CustomAudience.Field.name],
        }]
    }
    if exclude_audience:
        targeting[TargetingSpecsField.excluded_custom_audiences] = [{
            'id': exclude_audience[CustomAudience.Field.id],
            'name': exclude_audience[CustomAudience.Field.name],
        }]
    ad_set[AdSet.Field.targeting] = targeting
    ad_set.remote_create()
```

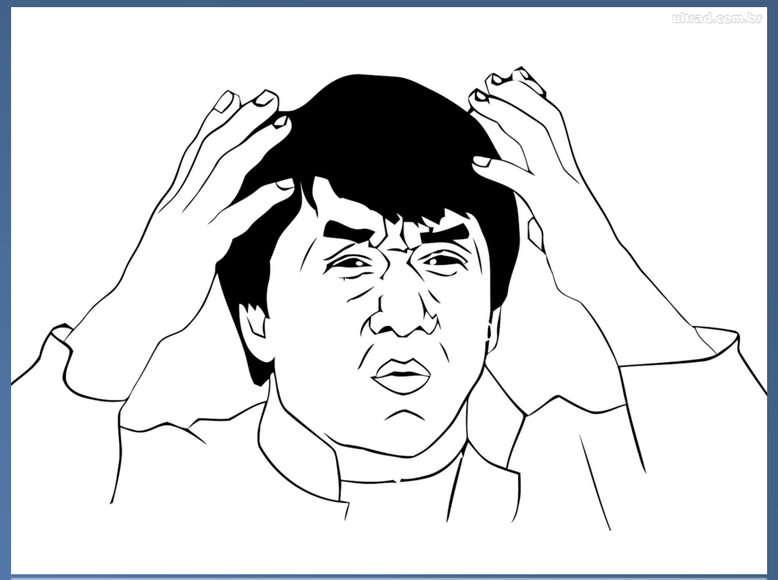



Pulling Ad Insights Into
Database/Excel

Manual Steps

1. Open saved report in Facebook ads reports
2. Refresh data
3. Export to file
4. Open Excel
5. Import file
6. Save spreadsheet
7. Repeat daily
8. Everyone does this!

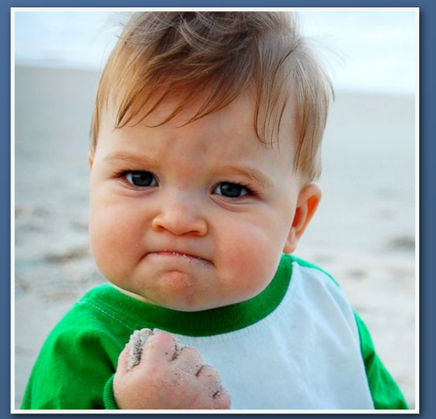
Not finish yet



- Big team will have hierarchy reporting
  - Each team member takes care of his own
  - Someone takes care of summary
- This can go up
  - Different team in different product category takes care of its summary
  - Someone takes care of teams' summary
- Then what if any one got the number wrong? (typo, copy/paste missing)

A Better Pipeline

1. ~~Open saved report in Facebook ads reports~~ One command to retrieve insights to mysql
2. ~~Refresh data~~
3. ~~Export to file~~
4. Open Excel
5. ~~Import file~~
6. ~~Save spreadsheet~~ Hit refresh button to refresh from mysql
7. Repeat daily
8. ~~Everyone~~ Build a robot to do~~es~~ this!



```
######################### main #########################
# get all ad campaign from an ad account
if not db_unix_socket == '':
  con = mdb.connect(host='localhost', user=db_username,
                    db=db_name, unix_socket=db_unix_socket, charset='utf8')
else:
  con = mdb.connect(host=db_host, port=db_port,
                    user=db_username, passwd=db_password,
                    db=db_name, charset='utf8')

with con:
    deleteAdSetsInsight(con, report_date)
    for my_ad_account_id in my_ad_account_ids:
        ad_account = AdAccount(my_ad_account_id)
        fields = [
            'campaign_group_name',
            'campaign_name',
            'campaign_id',
            'impressions',
            'clicks',
            'spend',
            'reach',
            'actions',
            'action_values'
        ]
        params = {
            'time_range': {
                'since': report_date,
                'until': report_date
            },
            'action_attribution_windows': ['28d_click'],
            'breakdowns': ['impression_device', 'placement'],
            'level': 'campaign',
            'limit': max_records
        }
        ad_insights = ad_account.get_insights(fields, params)
        print('Number of insight records:' + str(len(ad_insights)))
        count = 0
        for idx in range(0,min(max_records,len(ad_insights))):
            ad_insight = ad_insights[idx]
            print('Writing record number: ' + str(idx))
            writeAdInsight(ad_insight, con, report_date)
```

```
def writeAdInsight(ad_insight, con, report_date):
    impression_device = 0
    if 'impression_device' in ad_insight:
        impression_device = ad_insight['impression_device']

    action_device = 0
    if 'action_device' in ad_insight:
        action_device = ad_insight['action_device']


    key_value = {
        'start_date': report_date,
        'end_date': report_date,
        'campaign': ad_insight['campaign_group_name'],
        'adset': ad_insight['campaign_name'],
        'adset_id': ad_insight['campaign_id'],
        'impressions': ad_insight['impressions'],
        'clicks': ad_insight['clicks'],
        'spend': ad_insight['spend'],
        'reach': ad_insight['reach'],
        'impression_device': impression_device,
        #'conversion_device': action_device
    }
    if 'actions' in ad_insight:
        actions = ad_insight['actions']
        for action in actions:
            t = action['action_type']
            if t in action_type_columns:
                key_value[action_type_columns[t]] = action['28d_click']


    if 'action_values' in ad_insight:
        action_values = ad_insight['action_values']
        for action_value in action_values:
            t = action['action_type']
            if t in action_value_columns:
                key_value[action_value_columns[t]] = action['28d_click']

    key_str = ""
    value_str = ""
    count = 0
    for (key, value) in key_value.items():
        if count > 0:
            key_str += ", "
            value_str += ", "
        key_str += key
        value_str += "\"" + unicode(value) + "\""
        count += 1

    stat = "INSERT INTO ad_set_insight (" + key_str + ") VALUES (" + value_str + ")";
    print stat

    cur = con.cursor()
    cur.execute(stat)
```

SOLUTIONS+ENGINEERING

https://developers.facebook.com/docs/marketing-api