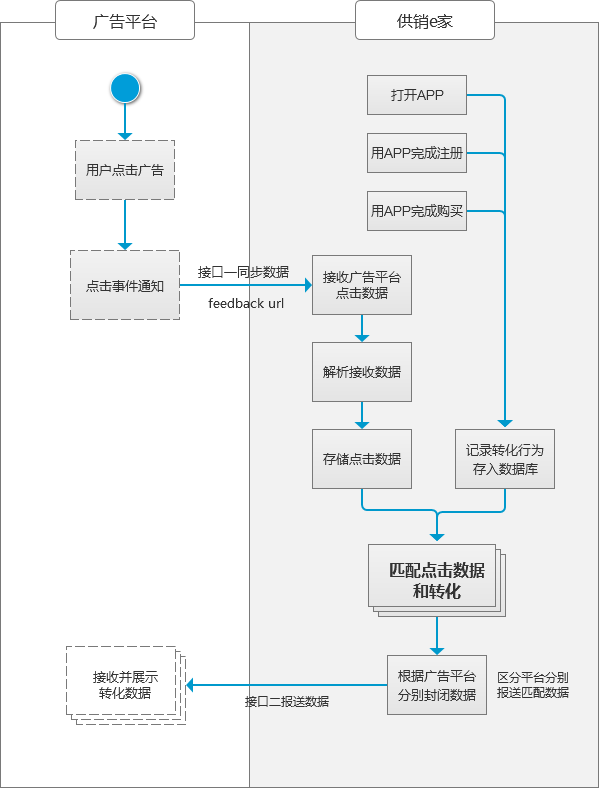
# 总体流程图



其中‘**存储点击数据**’我们用下表存储

广告平台点击表 ad\_click（示例）

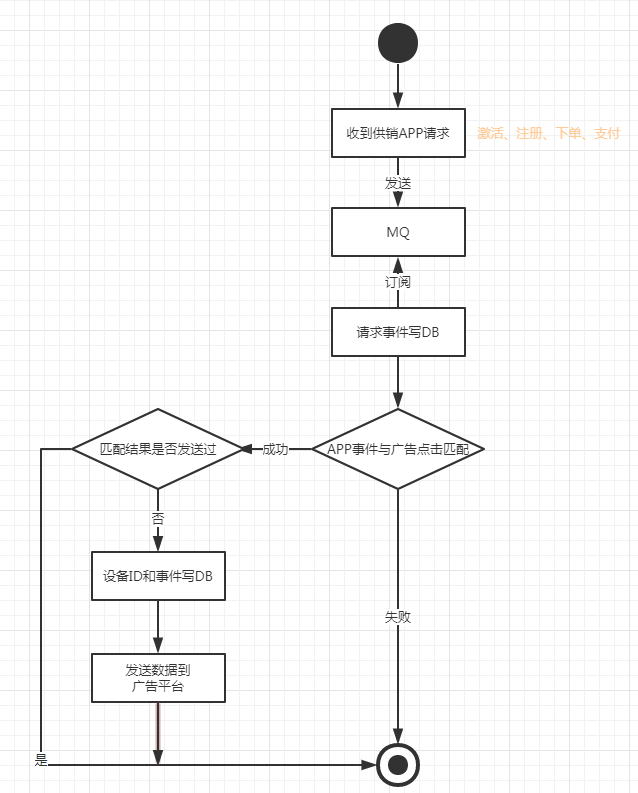
|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **类型** | **约束** | **说明** |
| ID | Varchar（32） | PK | 可用uuid |
| Divice\_id | Varchar(64) | Not null | 设备ID |
| source | tinyint | Not null | 点击来源  1 头条 2 广点通 |
| Click\_time | datatime | Not null | 广告点击时间 |

**‘记录转化行为存入数据库’**框用下表来表示

供销app事件表 device\_event（示例）

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **类型** | **约束** | **说明** |
| ID | Varchar（32） | PK | 可用uuid |
| Divice\_id | Varchar(64) | Not null | 设备ID |
| type | tinyint | Not null | 事件类型  1 激活 2 注册 3下单 4支付 |
| event\_time | datatime | Not null | 事件产生时间 |

# 供销APP开发流程图



大概流程如下

1. 供销app发送请求事件到服务器
2. 服务器将请求数据发送给MQ并返回
3. MQ订阅者收到请求数据入库并与广告平台数据匹配
4. 匹配成功将设备ID和请求事件等信息入库，后发送至广告平台
5. 匹配失败，直接结束

# APP事件与广告点击匹配算法

采用实时匹配，就是说收到app发来的事件后立即执行匹配

匹配算法很简单，如下

服务器收到app发送过来的数据，如下表device\_event

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **Device\_id(手机id)** | **type** | **Event\_time** |
| 123 | 1 | 注册 | 2017-11-11 11:30 |
|  |  |  |  |

那么我们会使用device\_id = 1 和 time=2017-11-11 11:30 这两个条件去 ad\_click 表查询

Ad\_click表数据如下

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **Device\_id(手机id)** | **source** | **Click\_time** |
| 456 | 1 | 头条 | 2017-11-10 11:30 |
| 457 | 1 | 广点通 | 2017-11-09 11:30 |
| 458 | 1 | 广点通 | 2017-11-15 11:30 |
| 459 | 2 | 头条 | 2017-11-15 11:35 |

查询sql如下（示例）

**SELECT device\_id**

**FROM ad\_click**

**WHERE device\_id = 1 AND click\_time between (‘2017-11-11 11:30 - 7天’) and ‘2017-11-11 11:30’**

**ORDER BY click\_time DESC**

**LIMIT 1**

可以看出最终查询结果是 id = 456 这条

这时我们将数据

Device\_id = 1 发送给头条接口

这样整个流程就处理完了

但是如果接下来我们又下单并支付了 如下表（device\_event） id=124

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **Device\_id(手机id)** | **type** | **Event\_time** |
| 123 | 1 | 注册 | 2017-11-11 11:30 |
| 124 | 1 | **支付** | 2017-11-11 11:32 |

这样按照上面的逻辑

Device\_id = 1 又会发给头条接口一次，会不会产生重复计费呢？

当然我们也可以做些工作来避免重复发送