|  |  |  |
| --- | --- | --- |
| 文档状态 | 文档编号 |  |
| [ ] 草稿 | 当前版本 | 1.0.0 |
| [√] 正式发布 | 作者 | 薛蕊 |
| [ ] 正在修改 | 完成日期 | 2016/11/18 |

**Ad Exchange广告交易平台的**

**设计与实现**

|  |  |  |  |
| --- | --- | --- | --- |
| 修改时间 | 作者 | 版本号 | 要点 |
| 2016/11/18 | 薛蕊 | 1.0.0 | 更新接口文档 |

**目录**

[1. 引言 3](#_Toc468285765)

[2. 系统相关技术简介 3](#_Toc468285766)

[2.1. B/S架构 3](#_Toc468285767)

[2.2. API接口 3](#_Toc468285768)

[2.3. MySQL数据库 4](#_Toc468285769)

[2.4. 编程语言 4](#_Toc468285770)

[2.4.1. JAVA 4](#_Toc468285771)

[2.4.2. HTML和JS 4](#_Toc468285772)

[2.4.3. C 4](#_Toc468285773)

[2.5. 硬件环境 4](#_Toc468285774)

[2.6. 软件环境 5](#_Toc468285775)

[2.7. 本章小结 5](#_Toc468285776)

[3. 系统需求分析 5](#_Toc468285777)

[3.1. 需求调研 5](#_Toc468285778)

[3.2. RTB实时竞价流程分析 5](#_Toc468285779)

[3.3. 用户用例说明分析 7](#_Toc468285780)

[3.3.1. 运营人员用例分析 7](#_Toc468285781)

[3.3.2. DSP公司用例分析 8](#_Toc468285782)

[3.3.3. 财务人员用例分析 9](#_Toc468285783)

[3.3.4. 媒介人员用例分析 9](#_Toc468285784)

[3.4. 系统功能性需求分析 9](#_Toc468285785)

[3.4.1. 管理员系统 9](#_Toc468285786)

[3.4.2. WEB主系统 9](#_Toc468285787)

[3.4.3. 报表系统 9](#_Toc468285788)

[3.4.4. 财务系统 10](#_Toc468285789)

[3.5. 系统非功能性需求分析 10](#_Toc468285790)

[3.5.1. 可靠性 10](#_Toc468285791)

[3.5.2. 易用性 10](#_Toc468285792)

[3.5.3. 安全性 11](#_Toc468285793)

[4. 系统概要设计 11](#_Toc468285794)

[4.1. 检索端 11](#_Toc468285795)

[4.2. 业务端 12](#_Toc468285796)

[4.3. 系统API 12](#_Toc468285797)

[5. 数据库设计 13](#_Toc468285798)

[5.1. one\_main主系统架构 13](#_Toc468285799)

[5.2. one\_dict数据字典库 13](#_Toc468285800)

[5.3. one\_final财务账单 13](#_Toc468285801)

[5.4. one\_cost消费 13](#_Toc468285802)

[5.5. 本章小结 13](#_Toc468285803)

[6. 系统详细设计与实现 13](#_Toc468285804)

[6.1. 检索端 13](#_Toc468285805)

[6.1.1. 流量分配逻辑 13](#_Toc468285806)

[6.1.2. Bid Request组装 13](#_Toc468285807)

[6.1.3. ADX平台Call out逻辑 14](#_Toc468285808)

[6.1.4. Bid Response接收 14](#_Toc468285809)

[6.1.5. 开始竞价的逻辑 15](#_Toc468285810)

[6.1.6. 竞价排序逻辑及计费规则 16](#_Toc468285811)

[6.1.7. Win notice通知规则 16](#_Toc468285812)

[6.1.8. ADX平台系统管理界面 17](#_Toc468285813)

[6.1.9. DSP公司的UI界面 17](#_Toc468285814)

[7. 系统测试 17](#_Toc468285815)

# 引言

# 系统相关技术简介

考虑到DSP公司的使用的便捷性，我们使用了最方便实用的B/S架构，同时我们也会开放给DSP公司来访问ADX平台的API接口，即ADX平台的一些预先定义的函数，目的是为DSP的开发人员提供最便捷与ADX平台进行交互的能力，这样省去了DSP的开发人员理解ADX平台内部工作机制细节的时间，同时增强了系统的保密性，并可使DSP可以最大程度地把他们需要的功能集成到他们的系统中。

此外，ADX平台采用MYSQL作为数据库支持数据的存储和使用，后端使用JAVA作为开发语言，前端使用html及JS进行开发，减少开发人员学习新技术的时间和成本。

## B/S架构

B/S架构，即浏览器和服务器结构，在这种结构下，用户工作界面是通过WWW浏览器来实现，极少部分事务逻辑在前端（Browser）实现，几乎所有的主要事务逻辑在服务器端（Server）实现。ADX系统作为对外发布的开放平台，使用B/S架构可以使维护平台和之后对平台进行迭代升级的方式更加简单，而且用户的使用成本很低，不需要安装与自身系统匹配的客户端应用，只要用市面常见的浏览器即可方便的使用平台提供的各项服务，可使DSP很便捷的査阅ADX提供的各类数据报表。

## API接口

API接口其实是一些预先定义好参数和功能的函数，目的是提供给某个应用程序基于某软件或硬件设施而可以访问或使用另外一个应用程序或一组例程的能力，这种方式不要求访问应用程序的源代码或者理解该程序内部的工作机制的细节。例如Ａ程序开放了一个API接口给B程序．则B程序可以通过一定形式的传递方式（由A指定），仅仅将一个或几个参数交付给开放此API接口的A程序即可使用Ａ程序的相应功能。由于与ADX系统进行对接合同的DSP公司均有一定的技术背景，DSP公司也都会有自己的投放端平台，如果只是靠人力去ADX平台提供的业务界面上进行操作和设置会大大增加运营的人力和时间成本。此时，如果DSP利用ADX平台为DSP已经定义好规则的开放ＡＮ接口对自己的投放端系统进行一定程度的开发，将双方系统通过API进行连接，之后DSP的运营人员便可以不再额外登陆ADX的平台去进行某挫操作，取而代之的是使用DSP系统的相应模块通过API接口去实现对ADX平台的访问，从而使双方系统更好的进行协调合合作，在降低时间成本的同时也大大提高了效率。

## MySQL数据库

MySQL是由Sun Microsystem公司提供的一个多线程、多用户，快速且健壮的关系数据库管理系统（RDBMS，Relational Database Management System）。它使用了结构化査询语言（SQL，Structured Query Langurage），这是目前全球通用的标准数据库査询语言。

MySQL是目前在WEB应用方面相当常用的关联数据库管理系统之一，关联数据库是指将数据保存在不同的数据表之中的一种数据库。普通数据库是将所有数据放在一个大仓库內，査询使用和存储都千分的不方便，而关系型数据库由于在各个数据表之间均存在一定程度的关系，这样就増加了速度并提高了灵活性。

MySQL中使用的SQL语言是目前用于访问数据库的最常用的一种标准化语言，并且由于MySQL体积小、速度快、总体拥有成本低，在不断改进和升级的过程中日趋成热，稳定性很好，大部分软件开发人员都会在日常的学习和工作中使用MySQL作为系统的数据库选择，故在ADX平台开发时选用MySQL，也一定程度上降低了研发人员的学习成本。

## 编程语言

### JAVA

JAVA语言是目前来说最常用的编程语言之一，它是一种可撰写跨平台应用程序的面向对象的程序设计语言。由于JAVA拥有相当程度的通用性、安全性、平台移植性和商效性等等特质，被用于计算机和非计算机行业的各个领域，同时JAVA也拥有着全世界最大的开发者社交群体。基于以上原因，本系统采用JAVA作为后端实现的开发语言。

### HTML和JS

HTML，即超文本标记语言，是用来描述网页的一种语言。严格来说HTML不是一种编程语言，而是一种标记语言即一套标记标签，HTML使用标记标签来描述网页的相关内容，拥有简易性，可扩展性，平台无关性以及通用性等优点。HTML的结构包括”头”部分和”主体”部分，其中”头”部提供关于网页的信息，”主体”部分提供网页的具体内容。

JS即JavaScript．是一种直译式脚本语言，是在程序的运行过程中逐行进行加载解释，具有简单，动态性，跨平台性等优势，主要是用来给HTML网页增加动态功能，为用户提供更流畅美观的浏览效果。

如今，利用HTML，JS及JAVA相互配合进行WEB类平台开发己经是主流的开发方法，这种开发方法简单，学习成本低，且拥有相对成熟的技术和模式，故我们也采用了上述方法进行ADX平台系统的开发。

### C

## ****硬件环境****

|  |  |
| --- | --- |
| 硬件 | 描述 |
| cpu | 8核32线程， CPU E5-2630 v3 @ 2.40GHz |
| 内存 | 128G |

## ****软件环境****

|  |  |
| --- | --- |
| **软件** | **版本** |
| centos | 6.7 |
| tomcat | 7.0.73 |
| MySQL | 5.6.23 |
| flume | 1.6.0 |
| jdk | 1.7 |
| redis | 2.8.24 |
| supervisor | 3.2.1 |
| g++ | 4.4.7 |
| nginx | 1.8.1 |
| bandwidth | 500Mbps |
| protobuf | 3.0.0 |

## 本章小结

本章主要对系统所用到的相关技术进行简单介绍，考虑到开发人员学习新技术的时间和成本，本系统均使用了较为成熟普遍的技术和编程语言－包括本系统使用的B/S架构进行开发，使用API接口与内外部系统进行交互，并采用MySQL作为数据库支持数据的存储和使用，后端使用JAVA和C作为开发语言，前端使用html以及JS进行开发。

# 系统需求分析

## 需求调研

## RTB实时竞价流程分析

如图3-1-1所示，一次广告展现的RTB实时竞价流程如下：

（1）用户打开app；

（2）产生广告机会时，app将广告请求发送到Pxene ADX；

（3）Pxene ADX向多家想要对此次广告位进行竞价的DSP并行发起竞标请求；

（4）DSP估值后给出此次曝光的报价；

（5）Pxene ADX对有效时间内收到的报价进行竞标判定，对竞标成功的DSP发送获胜通知；

（6）Pxene ADX按照媒体广告模板进行样式渲染后，将获胜DSP的广告返回给用户展示。



图3-1-1 RTB实时竞价流程

Pxene ADX产生的背景比较特殊，因此会有一种独特的竞价流程见图3-1-2，具体流程如下：

（1）用户打开app；

（2）产生广告机会时，app将广告请求发送到ADX；

(3）ADX向Pxene ADX询价；

（4）Pxene ADX向多家想要对此次曝光进行竞价的DSP并行发起竞标请求；

（5）DSP估值后给出此次曝光的报价；

（6）Pxene ADX按照ADX接口文档返回response。

(7）ADX对有效时间内收到的报价进行竞标判定，给Pxene ADX发送获胜通知；

（8）Pxene ADX收到获胜通知后，对竞标成功的DSP发送获胜通知；

（9）ADX按照媒体广告模板进行样式渲染后，将获胜DSP的广告返回给用户展示。



图3-1-2 RTB实时竞价流程

## **用户用例说明分析**

如表3-1所示，系统涉及的角色有运营人员（PM），DSP账户，财务人员，各个角色使用系统时由于各自角色的特殊性会使用到不同的登录页。

|  |  |  |  |
| --- | --- | --- | --- |
| 角色名称 | 运营人员（PM） | DSP账户 | 财务人员 |
| 角色定位 | 管理员 | 直接用户 | 直接用户 |
| 登陆页 | 账号管理  审核列表 | 首页 | 消费报表 |

表3-1 角色表

### **运营人员用例分析**

图3-3-1

运营人员（PM），如图3-3-1所示，ADX平台的直接使用者，统筹管理整个平台的所有DSP用户，主要的职能为：

1. 建立DSP账号：使用管理员账户建立DSP公司账号，设置相关参数，包括DSP的公司基本信息，合同基本信息，技术参数，流量发送设置等等；
2. 附身DSP账号：以管理员权限进入DSP账号，获取与DSP账号相间权限，可以对DSP账号进行参数设置，并查看DSP维度的报表和数据等等；
3. 修改参数：开启/关闭DSP在WEB，APP，VIDEO三个维度的流量，修改DSP账号IP，RTB Url，曝光监测域，点击监测域等参数；
4. 查看操作记录：查看所有用户操作记录，包括管理员以及DSP用户；
5. 査询物料信息：可以根据广告主ID和创意ID进行物料査询，査看广告主名称，Url，资质等信息，査看创意图片，尺寸规格，创意行业等信息；
6. 审核物料信息：运营人员根据审核规范对物料和广告主信息进行审核，审核通过后变更物料状态为审核通过，审核不通过则变更物料状态为审核拒绝，并附上审核拒绝的理由；
7. 复审物料信息：审核员可二次对物料的审核状态进行变更，此时无需客户申请再次审核。

### **DSP公司用例分析**



图3-3-2

DSP公司，如图3-3-2所示，ADX平台的主要用户，通过技术方式与ADX平台完成对接，对接成功后可以使用ADX平台提供的广告流量及各类服务，主要的职能为：

1. 签订合同：与ADX平台线下签订交易合同，生成唯一对接DSP公司的合同ID，用于额度充值等财务服务；
2. 额度充值：DSP发起额度充值申请，款项由财务系统确认后自动同步到ADX平台，同步为DSP用户的额度中；
3. 登录：DSP不能自主注册账号信息，需要对接完成后由运营同学提供账号密码等登录信息，DSP可登录到ADX平台进行操作和查看；
4. 账户设置：DSP用户登录平台后可以设置接收的流量大小，接入广告尺寸，地域划分等等参数，并可以新増/修改用于API接口调用的IP地址；
5. 査看报表及信息：DSP用户可以从多个维度査看自己在ADX平台参与竞价的情况，包括竞价消耗报告，分网站报告，分APP报告，分尺寸报告，分广告主报告，RTB竞价参数报告，分创意竞价报表，全面了解自己在平台的各类数据，并可以下载到本地进行査看。DSP还可以看到自身的财务信息，账单，及汇款记录；
6. API巧用：DSP用户可以使用ADX平台提供的API接口进行开发，与DSP系统实现自动的数据交互已经功能的简单化，主要包括广告主相关API，广告主新増、修改、查询全部、按ID查询、查询广告主资质检査状态、查询开放域名状态；创意相关API，创意新增、修改、按ID查询、查询静态创意检查状态、查询ADX平台修正过的创意及行业、查询指定时段内被检查拒绝的动态创意；报告报表相关API，RTB竞价报告、消费报告、广告主报告；常用文件下载相关API，开发者瓜和名称对应表文件下载（仅Mobile流量使用）|媒体过滤设置文件下载；
7. 查看媒体推荐列表：DSP用户可以查看ADX平台提供的所有媒体网站的流量列表，包括广告尺寸，域名，流量量级等维度；
8. 查看操作记录：DSP用户可查看自身账号的所有操作记录，包括账号操作，管理员操作二者的所有记录。

### **财务人员用例分析**

财务人员，如图3-2-2所示，ADX平台中负责DSP财务相关服务的人员，主要职能为：

1. 新签终止合同：DSP确认合同后在合同系统新建DSP信息，生成合同ID化并作为DSP的一个重要参数添加到ADX平台的DSP账号中，用以同步DSP账号的生效时间（与合同时间一致）以及账户额度；
2. 收款及款项录入：在DSP公司发起额度充值申请后，负责确认款项到账并录入财务系统，由系统同步充值合同到ADX平台。

### **媒介人员用例分析**

1. 签订合同：与DSP公司签订合同（线下），获得与DSP账号财务相关的合同ID信息。

## 系统功能性需求分析

系统功能性需求主要包括管理员系统．WEB主系统，报表系统和财务系统，其中WEB主系统是ADX系统中的核也部分，其他的系统模块均为此模块进行服务。

### 管理员系统

管理员系统是管理员用户主要使用的系统模块，用于管理所有DSP的账户，宏观把控整个ADX平台的使用情况，并在DSP用户使用出现异常的时候可较快定位并解决问题，管理员有权限在管理员系统中采取应急措施封禁/开启DSP账户。本模块面向日常运菅人员和PM，运营人员可以利用管理员身份建立和管理DSP账号，宏观査看所有广告主和创意的审核状态以及相关信息，且每个管理员可以査看所有管理员的操作记录。

### WEB主系统

WEB主系统是DSP用户使用的系统模块，也是整个业务端平台的主要组成部分，其他系统模块为此模块服务，是ADX平台业务端的核心。DSP可以使用WEB主系统査看帐户基本信息和状态，设置流量接入的类型和大小，査看各类报告，创建/修改广告主信息，上传/预览创意，进行流里投放设置，管理账号信息，査看操作记录等。管理员系统和WEB主系统对应账户状态应如图3-5所示。

### 报表系统

报表系统，DSP和管理员所査看的所有报表组成的系统模块，主要包括竞价消费报告，分网站报告，分APP报告，分尺寸报告，分广吿主报告，分创意报告，竞价参数报告。各个报毎反映DSP的投放表现数据，是DSP和管理员分析DSP投放情况的重要依据，DSP可以利用这部分数据对投放模型进行优化提升投放效果。报告中应该包含以下数据：

1. 实际发送PV:广告流量交易平台发送给当前账户所代表的DSP的bid request次数；
2. 放弃竞价次数：DSP平台放弃竞价的次数（通过”放弃参数”识别）；
3. 响应超时次数：DSP平台在规定的时间没有返回bid response的次数；
4. 解析错误次数：DSP平台发送的bid response不符合规范，无法解析的次数；
5. 放弃竞价率：放弃竞价次数/实际发送PV；
6. 响应超时率：响应超时次数/实际发送PV；
7. 解析错误率：解析错误次数/实际发送PV；
8. 参与竞价次数：DSP发送Response成功的次数，参与竞价次数＋放弃竞价次数＋响应超时次数＋解析错误次数＝实际发送PV；
9. 参与竞价比率：参与竞价次数/实际发送PV；
10. 竞价成功次数：DSP平台竞得广告展示机会的次数；
11. 展现次数：DSP通过ADX平台展现的物料次数；
12. 点击次数：展现出去的物料被点击的次数；
13. 消费: DSP在广告流量交易平台的消费；
14. 千次展现价格（千次展现价格）：消费/展现次数\*1000；
15. 平均点击价格：消费/点击次数。

### 财务系统

财务系统是ADX平台的计费系统，DSP在成功竞得广告位后会产生计费，此时财务系统会记录每一笔交易消耗并实时反馈到DSP账户状态中进行扣费。DSP在广告流量交易平台的消耗额度小于0元时会被锁定账户，此时ADX平台将停止流量服务，需要DSP充值额度后重新恢复账户开启状态继续投放。DSP在自身余额较少或己经为负的情况下可以发起充值申请，待ADX平台的财务确认收帐后会由ADX平台相关的财务系统同步数据到广告流量交易平台系统，完成款项录。如果DSP此时充值前余额己为负，则额度到账后会改写账户状态，由锁定变为有效开启。

## 系统非功能性需求分析

ADX平台系统在完成功能性需求的条件下，还需要考虑可靠性、易用性和安全性等非功能性需求，使得系统更加健壮和完善，减少系统上线运行期间可能出现的问题。

### 可靠性

可靠性即系统在一定的时间和条件下无故障地执行指定功能的能力或可能性，作为连接DSP和SSP的中间平台，如何增加ADX平台的可靠性是重中之重。为防止线上使用和迭代升级造成的系统信息丢失或死锁等错误，平台应具有定期备份，自动回滚，手动回滚等相关功能。

### 易用性

作为对外开放的交易平台，系统的易用性也是千分重要的，也正是由此我们采用了WEB界面与API接口结合的模式，在为用户提供简单易操作WEB UI的同时也给用户提供可以用于系统整合的API接口，用户可以自由选择使用何种方式与ADX平台系统进行交互。

### 安全性

ADX平台是商用的广告流量交易平台，系统的保密性和安全性就千分重要，因此无论是数据库系统的安全系数，用户信息的安全性和保密性，以及用户权限的管理等等都是系统安全性必须关注的重点。

# 系统概要设计

系统主要分为检索端和业务端，检索端负责维护系统的整套RTB逻辑，业务端面向DSP客户，平台的主要运营人员和PM，用于DSP查看自己的消费情况以及各类报表，并且给运营人员和PM的日常维护提供服务。

## 检索端

ADX平台在整个RTB系统中的整体架构如图4-1所示。



图4-1 AD Exchange整体架构图

整体架构包含以下几个模块：

1. ADX adpater：ADX适配器部分，根据不同的ADX编写对应的ADX adpater，主要功能是将不同的ADX request转化为common request，同时将ADX Module返回的common response根据ADX平台接口协议转化为对应的ADX response返回；

2. ADX Module：控制部分，将ADX common request进行处理，选择所有符合要求的买方，通过DSP Agent进行分发；从DSP Agent获取各个DSP返回的广告信息后，对DSP返回的bid进行排序，找出出价最高的，加密并通过win-notice server告诉给dsp最终出价；

3.DSP Agent：第三方dsp的接口模块，用于接受广告请求，进行格式转化，验证广告合法性，并进行广告拼装和返回广告；

4.Pixel Connector：接收各种通知解析并封装成内部格式，发送给Pixel Module；将Pixel Module的处理结果返回给流量方；

5.Pixel Module：将处理结果返回给Pixel Connector，将部分处理结果实时写入Redis；

5.redis：业务端的数据以及Pixel部分处理结果实时写入Redis。

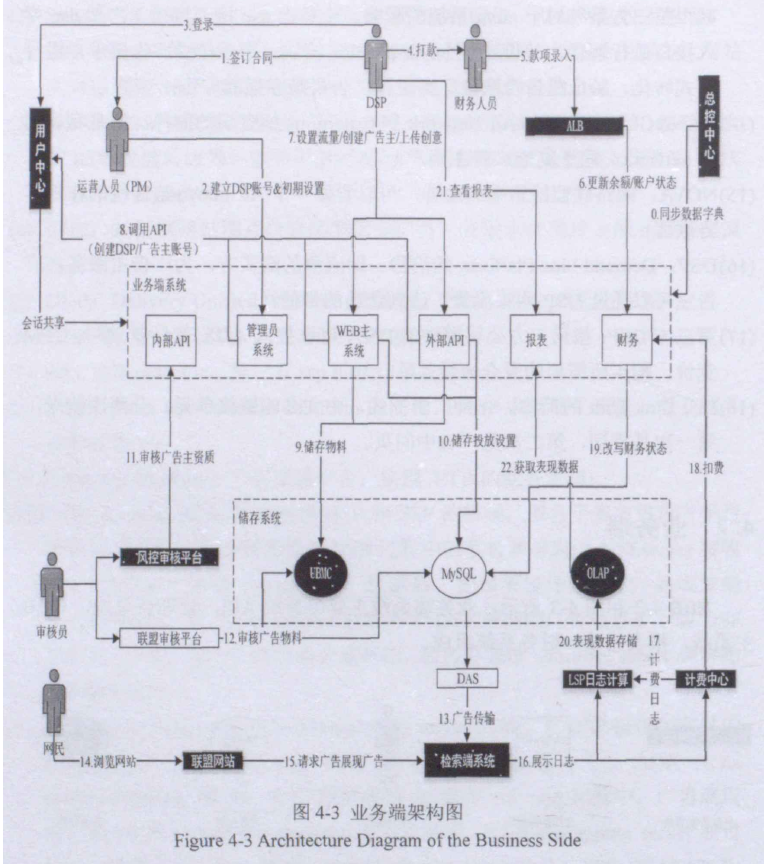
## 业务端

如图4-2和图4-3所示，业务端系统主要由系统API，管理员系统，WEB主系统，报表系统，财务系统组成。



图4-2业务端架构图图例说明

## 系统API



系统API主要包括2个部分，内部系统API以及DSP使用的外部API。內部系统API是业务端系统与其他相关系统交互的主要途径，是整个系统架构的桥梁。内部系统API将流量交易平台业务端与风控审核平台，联盟审核平台，用户中心，总控中心，存储系统相联系，实现内部数据交互，使系统模块化，简化整体架构。

DSP使用的外部API接口是DSP公司的内部系统与ADX平台进行数据交互的桥梁，DSP可以按照API接口文档将ADX平台提供的接口集成到DSP自己的系统中，实现双方系统的高效连接，为DSP使用ADX平台提供的各项服务提供了便利。外部接口主要包括广告主相关API，创意相关API。报告报表相关API，常用文件下载相关API四个类别。

外部API使用Json消息格式，HTTP请求Content-Type头部值为application/json。通过HTTPS协议交换信息。ADX平台通过API提供了便捷一致的数据请求和修改方法，DSP通过HTTPS请求URL来查询、新增、修改数据。API从资源的角度来使用数据，资源是指具有唯一标识符的单个数据实体。API方法的操作对象就是资源。例如，创意资源代表广告物料，同时也是流量交易服务API的数据类型。

API通过用户ID（dspid）以及授权令牌（token）来做权限验证，授权信息需要包含在API请求头中。参数编码使用UTF-8格式，通过HTTP POST方式、使用加密的HTTPS协议进行交互，増强了调用API服务的安全性。

API URL格式为：

<https://api.adx.pxene.com/>${apiName}/${resourceAction}

其中apiName表示调用的资源，resourceAction表示资源的操作方法。

基本的rescourceAtion包含四个动作，如表4-1所示，不支持册除。

|  |  |
| --- | --- |
| 操作 | 类型 |
| add | 新增 |
| update | 更新 |
| getAll | 获取所有 |
| get | 根据ID获取 |

例如：https://api.adx.pxene.com/creative/add 表示新增创意。

# 数据库设计

数据库共分one\_main主系统架构，one\_dict数据字典库，one\_final财务账单，one\_cost消费四个模块，数据库中各数据表逻辑关系如图5-1所示。

## one\_main主系统架构

## one\_dict数据字典库

## one\_final财务账单

## one\_cost消费

## 本章小结

本章主要介绍系统使用的数据库情况，主要分为主系统架构，数据字典库，财务账单，消费四个模块，并对每个模块的主要数据表进行详细介绍和说明，主要包括每个字段的属性定义以及特殊情况的备注说明内容。

# 系统详细设计与实现

## 检索端

检索端负责处理整个ADX平台内部RTB的逻辑，功能主要包括流量分配、Bid Request组装、流量Call out、Bid Response接收、DSP竞价和排序、Win Notice通知，以下为各个部分的详细逻辑设计。

### 流量分配逻辑

1. 所有的流量都会经过ADX平台，由ADX平台的检索端分发流量的去向。
2. 当前流量未超出DSP设置的QPS都会分发。

### Bid Request组装

Bid Request中的字段及来源如下表6-1所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 信息分类 | 详细信息 | 来源 | 数据价值 | 备注 |
| 用户信息 | IP | HTTP协议 | 基本数据 |  |
| 用户信息 | userAgent | HTTP协议 | 基本数据 |  |
| 用户信息 | 用户的基本信息 | 暂无 | 增值数据 | 未来从DMP系统获得 |
| 用户信息 | 性别 | 暂无 | 增值数据 | 未来从DMP系统获得 |
| 用户信息 | 年龄 | 暂无 | 增值数据 | 未来从DMP系统获得 |
| 广告位 | 展示类型 | ADX平台提供 | 基本数据 |  |
| 广告位 | 宽\*高 | ADX平台提供 | 基本数据 |  |
| 广告位 | 展示位置 | ADX平台提供 | 基本数据 |  |
| 广告位 | 发布商允许的创意类型 | ADX平台提供 | 基本数据 |  |
| 广告位 | 发布商允许的广告行业 | ADX平台提供 | 基本数据 |  |
| 广告位 | 发布商不允许的广告行业 | ADX平台提供 | 基本数据 |  |
| 广告位 | 发布商设置的底价 | ADX平台提供 | 基本数据 |  |
| 应用信息 | 应用id | HTTP协议 | 基本数据 |  |
| 应用信息 | 应该名称 | ADX平台提供 | 基本数据 |  |
| 应用信息 | 应用分类 | ADX平台提供 | 基本数据 |  |
| 设备信息 | 设备id | HTTP协议 | 基本数据 |  |
| 设备信息 | MAC | HTTP协议 | 基本数据 |  |

### ADX平台Call out逻辑

当bid request组装完毕后，就需要将request的信息向已经对接完毕的DSP进行call out（广播）。由于每个DSP的不同设置，因此并不是对所有的DSP都进行call out，因设置不同，有可能每个DSP获得的request信息也不相同，目前能够影响到call out的设置如下表6-2和表6-3所示。

1. call out次数只能小于等于QPS的设置，否则不进行call out。QPS的值取ADX平台与DSP厂商设置最小值。

|  |  |  |
| --- | --- | --- |
| 设置 | 说明 | Call out规则 |
| DSP的状态 | 状态有3种：开启/关闭/锁定 | 只有在开启状态时才进行call out；其它两种状态不发送 |
| QPS | Query Per Second | 根据数值大小进行发送 |
| 消费是否超额 | 超额计算公式见说明 | 如果消费超额，则不进行call out；否则进行call out |

表6- 2 ADX平台设置

|  |  |  |
| --- | --- | --- |
| 设置 | 说明 | Call out规则 |
| QPS | Query Per Second | 根据数值大小进行分发 |

表6-3 DSP设置

### Bid Response接收

经过call out过滤，向DSP发送bid request后，对bid response的接收需要设置timeout（超时时间）。

Bid Response需要接收的字段如下表6-5所示。

|  |  |  |
| --- | --- | --- |
| 详细信息 | 来源 | 数据价值 |
| 最高竞价 | DSP | 基础数据 |
| 创意ID | DSP | 基础数据 |
| DSP自定义id | DSP | 扩展数据 |
| DSP需要传递的额外自定义数据 | DSP | 扩展数据 |

表6-5Bid response接收字段

1）因为DSP的素材需要提前注批到ADX平台的服务器，所投放中需要的素材URL、点击跳转URL、曝光监测URL等内容不需要DSP在response中传递，只需要传递此次参与竞价的广告创意id。竞价成功后，ADX平台需要在ADX平台服务器找到对应此id的创意进行投放。

2）传递的价格信息需要进行加密处理（加密解密方法由ADX平台提供）。

### 开始竞价的逻辑

⑴开始竞价的触发条件有两种情况：一是在time out时间内全部的bid request均得到对应的bid response；二是达到了time out时间，此时对收到的bid response进行竞价。

⑵开始竞价前需要对收到的bid response进行有效性过滤，以符合媒体的品牌保护要求以及运营需求。属于品牌保护要求的有发布商不允许的广告主、发布商不允许的创意类型、发布商不允许的广告行业、竞价价格大于等于发布商设置的底价；属于运营需求的有竞价价格大于等于保留价。

竞价逻辑关系如下图6-5所示

图中相关说明如下：

1）如果媒体没有设置底价，则认为媒体的底价为０。

2）对每个流量默认有保留价。

3）对于因违反媒体品牌保护要求规定，不参与竞价的，统一记为无效竞价；对于因违反运营需求规定，不参与竞价的，记为竞价失败。



图6-5 竞价逻辑

### 竞价排序逻辑及计费规则

竞价排序的规则是以DSP的出价进行排序，报价最高的 DSP获取广告展示机会，并以第二高价加一分钱作为成交价，如果只有一个DSP 出价，那么以底价加一分钱作为成交价。

### Win notice通知规则

Win notice通知由server端发送。计费价格（CPM计费，单位为分），传递的价格信息需要进行加密处理。

## 业务端

### ADX平台系统管理界面

待补充

### DSP公司的UI界面

待补充

# 系统测试