# Teaching详细设计文档(客户端和服务器)

**重要说明：谁要违反，拖出去砍了！不合理的，不要找我，直接执行就行。**

**一级目录小四号加粗华文细黑，二级目录小四号华文细黑，正文五号华文细黑，其中正文中，需要强调的字体可以加粗。总之，全文都是华文细黑字体。**

**支持一级，二级目录和三级**

1. **一级**
   1. **二级**
      1. **三级**

**1四级**

**（1）五级**

# 修改&记录

|  |  |  |  |
| --- | --- | --- | --- |
| **时间** | **版本** | **作者** | **备注** |
| 2016/4/10 | v\_0.1.0 | 陆江 | 创建文档 |
| 2016/4/10 | v\_0.1.1 | 江屿，陆江 | 评审字段和接口设计 |
| 2016/4/10 | v\_0.2.0 | 江屿 | 增加用户方案 |
| 2016/4/11 | v\_0.3.0 | 陆江 | 规范文档、重排版 |
| 2016/4/13 | v\_0.4.0 | 江屿，陆江 | 引入混合编程和面向容器编程 |
| 2016/4/24 | v\_0.5.0 | 陆江 | 调整目录结构增加三级目录，增加用户安全方案，增加服务器后台web设计概述。 |
| 2016/5/15 | v\_0.5.1 | 陆江 | 用户方案增加token鉴权 |

目录

[**1.** **项目背景** 3](#_Toc449288925)

[1.1 屌丝的大“项目” 3](#_Toc449288926)

[1.2 屌丝Idea 3](#_Toc449288927)

[**2.** **项目启动** 3](#_Toc449288928)

[2.1 屌丝们 3](#_Toc449288929)

[2.2 屌丝技术情结 3](#_Toc449288930)

[2.3 屌丝开干 3](#_Toc449288931)

[**3.** **模块设计** 3](#_Toc449288932)

[3.1 后台管理TeachingWeb 3](#_Toc449288933)

[3.1.1 模块划分 4](#_Toc449288934)

[3.1.2页面布局 4](#_Toc449288935)

[3.1.3动态调整App前端数据展示规则 4](#_Toc449288936)

[3.1 客户端UI和GUI 4](#_Toc449288937)

[3.2 用户系统 5](#_Toc449288938)

[**4** **数据库设计** 6](#_Toc449288939)

[4.1 数据库的设计原则 7](#_Toc449288940)

[4.2 数据库选型 7](#_Toc449288941)

[4.3 数据库集合(表)设计 7](#_Toc449288942)

[**5** **混合编程和面向容器编程（骚气）** 9](#_Toc449288943)

[5.1 Why引入高大上的概念？ 9](#_Toc449288944)

[5.2 实施方案 9](#_Toc449288945)

[**6** **接口设计** 9](#_Toc449288946)

[6.1 接口规范（请注意） 9](#_Toc449288947)

[6.2 首页 10](#_Toc449288948)

[6.3 用户注册|登录|验证号码是否注册 13](#_Toc449288949)

[**7** **总结** 15](#_Toc449288950)

[7.1 错误集锦(感觉屌屌的) 15](#_Toc449288951)

[7.2 技术难点 15](#_Toc449288952)

[7.2.1客户端 15](#_Toc449288953)

[7.2.2服务器 15](#_Toc449288954)

1. **项目背景**

## 屌丝的大“项目”

屌丝有逆袭的天性，你不造吗？

## 屌丝Idea

屌丝总是没钱的，通过这个平台屌丝也可以听名师讲课，~~~~(>\_<)~~~~

1. **项目启动**

## 屌丝们

屌丝两枚：江屿（听说要教他的儿子写hello world，屌丝啊，还要坑下一代，尼玛），陆江

## 屌丝技术情结

要做技术中的高富帅。

引入混合式编程和面向容器编程等概念。

客户端：（看到，请补充）

服务器：Golang+MongoDB

代码管理工具：git+gitHub

git status

git add xxx

git commit xxx –m ‘’

git rm xxx

git push

## 屌丝开干

干，别瞎逼逼……

1. **模块设计**

## 后台管理TeachingWeb

服务器为了更好的支持App前端业务，现决定服务器开发一套web管理界面，用于可视化管理后台数据。技术选型暂定：Go1.5 template+HTML5+Bootstrap3.x+jQuery。

### 3.1.1 模块划分

目前后台命名为TeachingWeb，提供基本用户登录，基本数据管理（包括：CRUD教师，CRUD课程，CRUD滚动信息，CRUD课程大类，CRUD搜索信息，CRUD广告栏），动态调整App前端数据展示规则。

目前急需CRUD课程功能，动态调整App前端数据展示规则。

### 3.1.2页面布局

页面采用经典的左侧菜单，右侧显示主体的布局方式，有页头和页脚。页面的主体基调偏灰黑系。

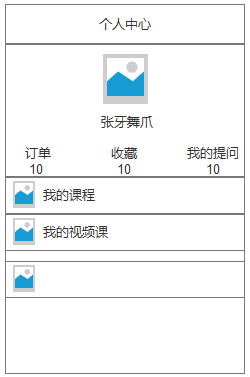
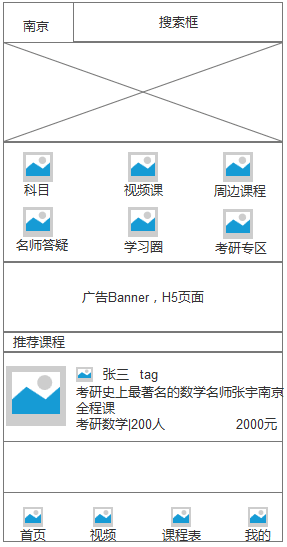
### 3.1.3动态调整App前端数据展示规则

首页承载了整个App很多重要信息，为了灵活控制，将客户端的控制权主动移到服务器。服务器需要对首页中的搜索框，滚动框，科目类别，广告栏，推荐课程进行控制。具体细节分解如下：

1. 对搜索框的控制
2. 滚动框控制
3. 科目类别控制
4. 广告栏控制
5. 推荐课程控制

## 客户端UI和GUI

UI和GUI主要抄袭“跟谁学”APP，当然也做了简单的“微创新”。

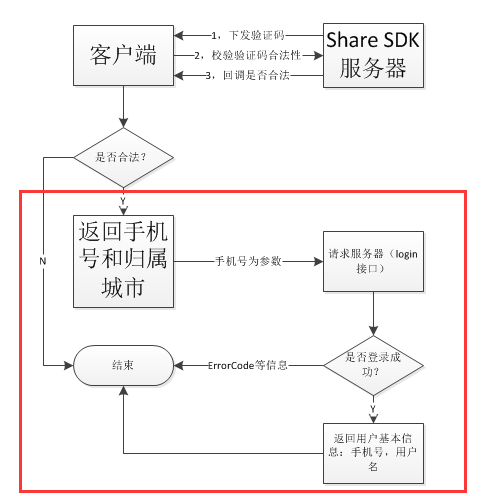


## 用户系统

### 3.3.1业务逻辑

只支持手机号注册并登录，客户端集成Share SDK短信验证功能，通过手机号获取验证码，与share sdk服务器交互验证验证码是否合法，如果验证合法，会回调对应的方法，返回验证通过的手机号，然后调用我们服务器login接口登录，登录成功默认用户名就是手机号。

逻辑图

****

接口：注册，登录，验证号码

用户系统安全方案

在每个系统中，几乎所有的用户系统对安全都有极高的要求。

对于模块需要保证两点

密码密文传输

请求防篡改

方案1：用户在注册成功时，为每个用户提供一对access key（以下简称ak）和private key（以下简称pk），可以理解为公私钥对，ak随接口返回给客户端。

同时所有客户端和服务器协商一个signature key（以下简称sk）。其中ak可公开；ak和sk客户端保存，但是sk不可公开；pk和sk服务器端保存均不可泄露。

客户端通过使用非对称加密算法和ak对密码加密并密文传输到服务器，服务器通过pk对密文解密。

对于防窜改，可以对每个用户的请求和相关参数通过sk做签名。即使sk泄露了也无妨，因为对于请求参数，sk，签名之后的字符串，任何一个被修改，服务器验证都会失败。

缺点，实现复杂，sk一旦泄露非法用户可以模拟客户端请求。

方案2：所有客户端和服务器协商一个key，这个key客户端既可以通过对称加密算法加密密码，服务器解密；也可以做为请求签名的key。要求key决不可泄露，一旦泄露风险很大，用户密码泄露。实现简单，但是风险大。

总结：个人觉得方案1太复杂，而且实现麻烦；方案2，安全系数较低。如果从安全考虑，方案2不太建议。

密码保存：md5（密码明文+pk）

### 3.3.2引入Token

1. Token是什么？

Token是服务器端生成的一串字符串，以作客户端进行请求的一个令牌，当第一次登陆后，服务器生成一个Token便将此Token返回给客户端，以后客户端只需带上这个Token前来请求数据即可，无需再次带上用户名和密码。

1. Token的引入

Token是客户端频繁向服务器端请求数据，服务器端频繁的去数据库查询用户名和密码并进行对比，判断用户名和密码正确与否，并作出相应提示，在这样的背景下，Token便应运而生。

1. 使用Token的目的

Token的目的是为了减轻服务器的压力，减少频繁的插叙数据库，使服务器端更加健壮。

1. 如何使用Token？
2. 使用设备号/设备mac地址作为Token

客户端：客户端在登录的时候获取设备的设备号/mac地址，并将其作为参数传递到服务器。

服务器：服务器接收到该参数后，便用一个变量来接收同时将其作为Token保存在数据库，并将该Token设置到Session中，客户端每次请求的时候都要统一拦截，并将客户端传递的Token和服务器端session中的Token进行对比，如果相同则放行，不同则拒绝。

分析：此刻客户端和服务器就统一了一个唯一的标示token，而且保证了每一个设备拥有了一个唯一的会话。该方法的缺点是客户端需要带设备号/mac地址作为参数传递，而且服务器端号还需要保存；优点是客户端不需重新登录，只要登录一次以后一直可以使用，至于超时的问题是由客户端这边来处理。若服务器的token超时后，服务器只需将客户端传递的token向数据库中查询，同时并赋值给变量token，如此token的超时又重新计时。

1. 使用Session值作为Token

客户端：客户端只需携带用户名和密码登录即可。

服务器：服务器收到用户名和密码后并判断，如果正确了就将本地获取sessionID作为token返回给客户端，客户端以后只需带上请求数据即可。

分析：这种方式使用的好处是方便，不用存储数据，但是缺点就是当session过期后，客户端必须重新登录才能进行访问数据。

1. 实现方案

采用第2种方案，客户端携带用户名和密码登录，服务器先校验用户名和密码，成功则生成一个uuid+expire的字符串，然后保存在session当中，请求返回body中对uuid+expire做AES-128加密，然后base64.StdEncoding.EncodeToString(token)。

1. **数据库设计**

## 数据库的设计原则

经验就是原则！

## 数据库选型

采用mongodb

数据库：teaching

集合：course,teacher,item,user,banner

启动:mongod.exe –f ../conf/mongodb.conf

登录：mongo.exe localhost:20822

## 数据库集合(表)设计

（1）Course集合数据结构

|  |  |  |
| --- | --- | --- |
| course | | |
| \_id | ObjectId | Mongodb自生成 |
| cid | string | 课程编号 |
| name | string | 名称 |
| titile | string | 宣传语 |
| decription | string | 描述 |
| location | string | 位置 |
| teacherId | int | 老师标号,同teacher.tid |
| image | string | 原图 |
| thumbnail | string | 缩略图 |
| sign | int | 已经报名的人数 |
| total | int | 报名上限 |
| itemId | int | 类型ID,同item.iid |
| isTop | bool | 是否是滚动条目，默认false |
| timestamp | long | 创建课程的时间，用于排序分页 |
| price | double | 该门课程的价格 |
| address | string | 该门课程的授课地点 |
| startTime | long | 授课开始时间（时间戳） |
| endTime | long | 授课结束时间（时间戳） |
| days | int | 授课天数 |

（2）Teacher集合数据结构

|  |  |  |
| --- | --- | --- |
| teacher | | |
| \_id | ObjectId | Mongo自生成 |
| tid | string | 老师编号 |
| name | string | 姓名 |
| avatar | string | 老师头像 |
| age | int | 老师年龄 |
| phone | string | 联系方式 |
| teachAge | int | 教龄 |
| experience | string | 过往经历 |
| sex | int | 性别1:男0:女 |
| isVerify | int | 是否认证1:认证0:未认证 |
|  |  |  |

（3）Item集合数据结构

|  |  |  |
| --- | --- | --- |
| item | | |
| \_id | ObjectId | Mongo自生成 |
| iid | int | 项目标号 |
| name | string | 名称 |
| sortBy | int | 排序依据,按照数值大小进行排序 |
| display | int | 是否显示  1:显示【默认】  0：不显示 |

（4）User集合数据结构

|  |  |  |
| --- | --- | --- |
| user | | |
| \_id | ObjectId | Mongo自生成 |
| uid | string | User id  uuid+timestamp |
| name | string | 真实姓名 |
| nickname | string | 昵称 |
| phone | string | 联系方式，默认是name和nickname的值 |
| pwd | string | 具体方案，再考虑。 |
| avatar | string | 头像 |
| sex | int | 性别，1：男，0：女 |
| city | string | 所在城市，如：南京市 |

（5）UserInfo集合数据结构

|  |  |  |
| --- | --- | --- |
| UserInfo | | |
| \_id | ObjectId | Mongo自生成 |
| userId | string | 同uid |
| comment:{  cid:"",  info:""  } | jsonobject | cid评论ID  info评论具体信息 |
| collection:{  cid:"",  info:""  } | jsonobject | cid 收藏ID  info收藏信息 |
|  |  |  |
|  |  |  |

（6）banner集合数据结构

|  |  |  |
| --- | --- | --- |
| item | | |
| \_id | ObjectId | Mongo自生成 |
| bid | string | 项目标号 |
| icon | string | logo |
| sortBy | int | 排序依据，同上 |
| display | int | 是否显示，同上 |

~~（6）menu集合数据结构~~

|  |  |  |
| --- | --- | --- |
| ~~item~~ | | |
| ~~\_id~~ | ~~ObjectId~~ | ~~Mongo自生成~~ |
| ~~mid~~ | ~~string~~ | ~~项目标号~~ |
| ~~icon~~ | ~~string~~ | ~~logo~~ |
| ~~sortBy~~ | ~~int~~ | ~~排序依据，同上~~ |
| ~~display~~ | ~~int~~ | ~~是否显示，同上~~ |

1. **混合编程和面向容器编程（骚气）**

## Why引入高大上的概念？

为了APP的灵活和更好的展示效果，决定客户端和服务器互相配合采用混合编程，即客户端混合Activity和HTML5来完成前段展示；服务器尽力为客户端的展示提供数据支撑。

为了灵活控制App，客户端的数据除布局方式之外，所有数据均由服务器统一派发和控制。

客户端通过webview与Activity之间的交互完成混合编程，取长补短。

## 实施方案

1. **接口设计**

## 接口规范（请注意）

总纲：接口未开，规范先行

接口返回字段名称：首字母小写，采用驼峰式。

200 OK 表示请求OK，并且服务器有数据返回，注意可能数据为空。

204 No Content，服务器没有数据返回，body为null。

400 Bad Request，客户端的请求有错误，具体信息请参考子码ret和描述信息msg。

500 Server Error，可以理解为服务器挂了。

举例：

（1）200 OK

body:

{

"data":[{

"id":1,

"name":"lujiang"

}]

}或

{

"data":[]

}

（2）204 No Content

body:null

（3）400 Bad Request

body:

{

"ret":400100,

"msg":"location is empty"

}

（4）500 Server Error

body：服务都挂了，你还想有body体？

字码及描述信息（服务器定期更新，不保证实时）

|  |  |  |
| --- | --- | --- |
| ret | msg | desc |
| 400100 | location is empty | 位置信息为空 |
|  |  |  |

## 首页

1. 业务逻辑

该接口支持客户端请求首页数据包括：

topCourses滚动条数据[默认5条]；

items课程类别信息[默认数据库中所有]；

items规则如下：

约定一下items 的url：

|  |  |  |
| --- | --- | --- |
| 名称 | url | 排序 |
| 考研数学 | aixue://a=maths | 1 |
| 考研英语 | aixue://a=english | 2 |
| 考研政治 | aixue://a=politics | 3 |
| 专业课 | aixue://a=special | 4 |
| 视频 | aixue://a=video | 5 |
| 历年真题 | aixue://a=exam | 6 |
| 经验分享 | aixue://a=experience\_share | 7 |
| 名师答疑 | aixue://a=forasked | 8 |

timeOrbanner广告[默认1条]；或者倒计时，

datas具体课程信息[默认10条]）；

~~menus底部的菜单[默认4个]；~~

支持按照时间戳分页，支持按照位置信息查询数据。

1. 接口定义

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **接口定义：**http://42.96.149.208:10029/teaching/index | | | | | |
| **METHOD** | | | GET | |  |
| **请求参数：** | | | | | |
| **参数名(是否必填)** | | **类型** | | **备注** | |
| location(yes) | | string | | 位置信息，只支持市级，如：南京市 | |
| timestamp(no) | | long | | 分页，默认timestamp=0,表示首页  timestamp!=0表示查询小于timestamp之后10条数据 | |
| **返回值：** | | | | | |
| 200 OK | 当timestamp=0  {  topCourses:[],  items:[],  timeOrbanner:[],  datas:{},  ~~menus:[],~~  timestamp:  }  当timestamp!=0  {  data:[],  timestamp:  } | | | timestamp=0表示请求首页数据。timestamp!=0表示请求小于timestamp之后10条数据。  topCourses:[]表示请求5条滚动数据。数据库中只能有5条滚动数据，如果接口返回大于5条数据，请客户端默认只处理前5条。如果接口返回小于5条数据，请客户端自行处理异常，保证返回的小于5条的数据显示正常。  items:[]表示目前支持的类别，如：科目，考研等  timeOrbanner:{}目前准备放个图片用于链接HTML5广告页面或者直接倒计时。  datas:[]默认按照时间戳降序的10条数据  ~~menus:[]菜单项~~  timestamp:每次请求的data中最后一条数据的时间戳。  具体topCourses,items,timeOrbanner,datas，~~menus~~数据结构如下表 | |
| 400,500 | {  “ret”:”xxx”,  “msg”:”xxx”  } | | |  | |
| **字段** | **数据结构** | | | **备注** | |
| topCourses | topCourses:[  {  cid:string  name:string  title:string  description:string  location:string  teacher:{  tid:int  name:string  avatar:string  }  image:string  thumbnail:string  sign:int  total:int  price:double  address:string  url:string  }  ] | | | cid:课程编号  name:课程名称  title:课程的宣传语  description:描述  location:位置信息  teacher.tid:老师编号  teacher.name:老师姓名  teacher.avatar:老师头像（小图）  image:课程原图  thumbnail:课程缩略图  sign:已经报名  total:报名上限  price:价格  address:该门课程的授课地点  url:点击跳转的url | |
| items | items:[  {  iid:string  hotIcon:string  icon:string  name:string  url:string  }  ] | | | iid:项目编号  hotIcon:热图图标  icon:图标  name:项目名称  url:点击该条目链接的地址，以http[https]开头的直接用Html5展示，以aixue开头的直接用Activity布局。 | |
| timeOrbanner | timeOrbanner:{  type:int  base:{}  }或  timeOrbanner:{  type:int  banner:{  bid:string  icon:string  url:string  }  } | | | type:类型，0：倒计时【默认】  1：广告  base:倒计时的基准日期  banner:广告详细信息  bid:广告标号  icon：广告的图片  url：点击广告的url | |
| datas | 同topCourses | | |  | |
| ~~menus~~ | ~~menus:[~~  ~~{~~  ~~mid:string~~  ~~icon:string~~  ~~url:string~~  ~~}~~ | | | ~~mid:菜单ID~~  ~~icon:菜单图标~~  ~~url:点击菜单图标跳转的url~~ | |

Example:

GET

当http://42.96.149.208:10029/teaching/index?location=南京timestamp=0

200 OK

Body:

{

topCourses:[],

items:[],

timeOrbanner:{},

datas:[],

~~menus:[],~~

timestamp:

}

当http://42.96.149.208:10029/teaching/index?location=南京timestamp=12

200 OK

Body:

{

datas:[],

timestamp:

}

当http://42.96.149.208:10029/teaching/index?location=南京timestamp=12

200 OK，并且是最后一页时，返回timestamp为0，datas数组为空

Body:

{

datas:[],

timestamp:0

}

## 用户注册|登录|验证号码是否注册

1. 业务逻辑

支持用户利用手机号注册和登录，之前需要客户端通过验证码验证手机，对于已经注册的号码，不允许重复注册。

1. 接口定义

（a）用户注册

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **接口定义：**http://42.96.149.208:10029/teaching/user/base/\_reg | | | | | |
| **METHOD** | | | POST | |  |
| **请求参数：** | | | | | |
| **参数名(是否必填)** | | **类型** | | **备注** | |
| phone(true) | | string | | 手机号码，默认也作为用户名 | |
| pwd(true) | | string | | 注册上传密码（客户端用ak加密传输，服务器pk解密） | |
| **返回值：** | | | | | |
| 204 No Content | null | | | 注册成功，保存用户相关信息。 | |
| 400,500 | {  “ret”:”xxx”,  “msg”:”xxx”  } | | |  | |

（b）用户登录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **接口定义：**http://42.96.149.208:10029/teaching/user/base/\_login | | | | | |
| **METHOD** | | | POST | |  |
| **请求参数：** | | | | | |
| **参数名(是否必填)** | | **类型** | | **备注** | |
| phone(true) | | string | | 手机号码，默认也作为用户名 | |
| pwd(true) | | string | | 注册上次密码（客户端用ak加密传输，服务器pk解密） | |
| **返回值：** | | | | | |
| 200 OK | {  name：string  nickname:string  phone:string  ……  token:string  } | | | 返回用户相关信息。 | |
| 400,500 | {  “ret”:”xxx”,  “msg”:”xxx”  } | | |  | |

1. 验证号码是否被注册

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **接口定义：**http://42.96.149.208:10029/teaching/user/base/phone/\_check | | | | | |
| **METHOD** | | | POST | |  |
| **请求参数：** | | | | | |
| **参数名(是否必填)** | | **类型** | | **备注** | |
| phone(true) | | string | | 手机号码，默认也作为用户名 | |
| **返回值：** | | | | | |
| 200 OK | {  valid:0  } | | | valid:是否被注册过。  0：无效，被注册过。  1：有效，未注册过。 | |
| 400,500 | {  “ret”:”xxx”,  “msg”:”xxx”  } | | |  | |

1. **总结**

## 错误集锦(感觉屌屌的)

1. go在机器内存不足的情况下会：

runtime: VirtualAlloc of 1048576 bytes failed with errno=1455

fatal error: runtime: cannot map pages in arena address space

解决方案：无

## 技术难点

### 7.2.1客户端

（1）客户端视频模块，视频下载等技术

（2）客户端混合式开发，如果无缝对接服务器动态配置以及给用户无感知是H5还是Native

（3）客户端hotfix技术，希望能够不发版本，可以修复客户端bug

现在最流行的客户端新技术。

（4）客户端模块式开发技术，以最小代价开发不同模块

现在最流行的插件化开发技术

### 7.2.2服务器

（1）后端搜索算法

（2）后端推荐算法

根据考研学生在平台的出错情况，推荐给他同类题型以及复习意见

（3）真题或者每日一练的接口设计（是否要图片）

（4）接口加签和密文传输

接口请求防窜改，密码传输和保存。

## 7.3 技术点

### 7.3.1 AES高级加密标准