**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。**

**支持一级和二级目录，三级以下按照如下顺序**

1. **一级**

**1.1 二级**

1. **三级，注意括号是中文输入法下的括号，下同。**
2. **四级**

**不允许出现五级目录。**

**修改&记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **时间** | **版本** | **作者** | **备注** |
| 2016/4/10 | 创建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 |
|  |  |  |  |
|  |  |  |  |

目录

[**1.** **项目背景** 3](#_Toc448264417)

[1.1 屌丝的大“项目” 3](#_Toc448264418)

[1.2 屌丝Idea 3](#_Toc448264419)

[**2.** **项目启动** 3](#_Toc448264420)

[2.1 屌丝们 3](#_Toc448264421)

[2.2 屌丝技术情结 3](#_Toc448264422)

[2.3 屌丝开干 3](#_Toc448264423)

[**3.** **模块设计** 3](#_Toc448264424)

[3.1 用户系统 3](#_Toc448264425)

[**4.** **数据库设计** 4](#_Toc448264426)

[4.1 数据库的设计原则 4](#_Toc448264427)

[4.2 数据库选型 5](#_Toc448264428)

[4.3 数据库集合(表)设计 5](#_Toc448264429)

[**5.** **接口设计** 7](#_Toc448264430)

[5.1 接口规范（请注意） 7](#_Toc448264431)

[5.2 获取课程信息 8](#_Toc448264432)

[5.3 用户登录 10](#_Toc448264433)

[**6.** **总结** 10](#_Toc448264434)

[6.1 错误集锦(感觉屌屌的) 10](#_Toc448264435)

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

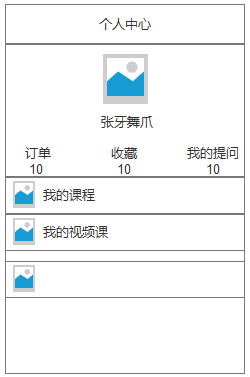
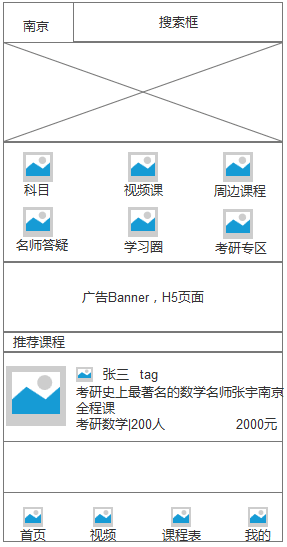
## 屌丝开干

干，别瞎逼逼……

1. **模块设计**

## 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”。



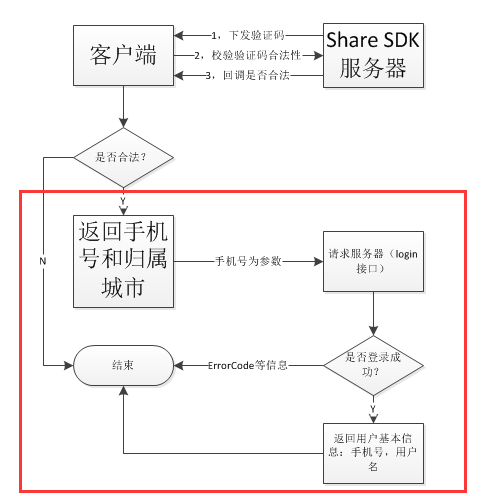
## 用户系统

（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。

补充：登录要分老师和普通用户（学生或者家长）登录？

1. 逻辑图



1. 登录接口

（a）登录成功后，服务器返回的数据样式

{

uid:用户唯一的id，根据这个id可以去查询用户的订单，收藏列表，关注列表等信息

phone:"151511860255",

avatar: "",

name:"15151860255",

sex:1,

city:南京市

}

（b）登录失败返回的数据样式

{

~~errCode~~ ret:400xxx,

~~errMsg~~ msg:""

}

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

## 数据库集合(表)设计

（1）Course集合数据结构

|  |  |  |
| --- | --- | --- |
| course | | |
| \_id | ObjectId | Mongodb自生成 |
| cid | int | 课程编号 |
| name | string | 名称 |
| titile | string | 宣传语 |
| ~~comment~~ | ~~string~~ | ~~评论~~ |
| 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 |
| categoryId | 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 | int | 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 |

（4）User集合数据结构

|  |  |  |
| --- | --- | --- |
| user | | |
| \_id | ObjectId | Mongo自生成 |
| uid | int | User id |
| name | string | 真实姓名 |
| nickname | string | 昵称 |
| phone | string | 联系方式，默认是name和nickname的值 |
| 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 | int | 项目标号 |
| icon | string | logo |
| sortBy | int | 排序依据 |
| display | int | 是否显示 |

（6）menu集合数据结构

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

（4）User集合数据结构

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。

## 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]；

banners广告[默认1条]；

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:[],  banners:[],  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  banners:[]目前准备放个图片用于链接HTML5页面  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,banners,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。 | |
| banners | banner:[  {  bid:string  icon:string  url:string  }  ] | | | 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:[],

banners:[],

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

}

## 用户登录

支持用户利用手机号登录，之前需要客户端通过验证码验证手机。

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