

课堂考勤管理系统 接口设计文档*

软件工程1305班 田劲锋 201316920311

2015 年 5 月 25 日

目录

1 服务端	2	2.1.4 声音类 AudioModel .	8
1.1 服务端前台	2	2.2 视图 ui	8
1.1.1 课程管理模块	2	2.2.1 主界面类 MainActivity	8
1.1.2 班级管理模块	3	2.2.2 登录类 LoginActivity	9
1.1.3 学生管理模块	4	2.2.3 签到类 SignInActivity	9
1.1.4 考勤管理模块	5	2.2.4 验证类 VerifyActivity	9
1.2 服务端后台	5	2.3 控制器 logic	9
1.2.1 课程类 Subjects	5	2.3.1 主控类 MainServices .	9
1.2.2 班级类 Classes	6	2.3.2 通信类 Communication	10
1.2.3 学生类 Students	6	2.3.3 图 像 处 理 类 Im-	
1.2.4 签到类 Signin	7	ageProcessor	10
1.2.5 识别类 Identification .	7	2.3.4 音 频 处 理 类 Au-	
1.2.6 通信类 Communication	7	dioProcessor	10
1.2.7 日志类 Log	8	3 数据处理约定	11
2 客户端	8	3.1 课程表格式	11
2.1 模型 model	8	3.2 班级表格式	12
2.1.1 学生类 StudentModel	8	3.3 学生表格式	12
2.1.2 设备类 DeviceModel .	8	3.4 消息格式	12
2.1.3 图像类 ImageModel .	8	3.5 音频数据格式	13
		3.6 图像数据格式	13
		3.7 通信协议	13

*本文档托管在 GitHub 上, PDF 文件: <https://github.com/kingfree/haut/raw/master/course/se/attd.pdf>,
TeX源文件: <https://github.com/kingfree/haut/blob/master/course/se/attd.tex>.

本文档只描述对外的公开接口，不描述其具体实现细节（包括私有方法、私有类、私有属性，以及内部处理流程）。

考虑到服务端和客户端的统一性，以Java语言为基准，因为Java不仅可以有效开发Web应用（服务端前台）、系统应用（服务端后台），还可以开发基于Android的移动客户端应用。

由于时间不足和能力不够，不能做到十分完善，还请见谅。

1 服务端

1.1 服务端前台

服务端提供给教师可操作的前台界面，通过响应HTTP的GET/POST请求来实现交互。这实际上是一个Web服务器。

对于Web服务器而言，我们要提供的接口是URI，以及相应的请求类型和参数，为了简明起见，分成几个模块。

1.1.1 课程管理模块

/subject/index/?	<i>GET</i>
-------------------------	------------

参数	<i>page</i>	页码
	<i>search</i>	搜索关键字
	<i>order</i>	排序方式
	<i>asc</i>	排序顺序

描述 显示课程列表。按照指定的页码和排序方式显示，如果有搜索关键字传入，则显示匹配的搜索结果。

跳转 本页

/subject/import/?file	<i>(multipart)POST</i>
------------------------------	------------------------

参数	<i>type</i>	文件类型
	<i>file</i>	待导入的文件

描述 解析文件中的课程信息，跳转到自定义导入信息页面。

跳转 /subject/import/?data

/subject/import/?data	<i>POST</i>
------------------------------	-------------

参数	<i>data</i>	解析后的表格或映射表
----	-------------	------------

描述 导入课程。提供用户界面，允许选择表格行列与待导入字段对应。调用服务端后台接口写入数据库，跳转到导入状态页面。

跳转 /subject/import/?status

/subject/import/?status	<i>POST</i>
--------------------------------	-------------

参数	<i>status</i>	导入状态
----	---------------	------

data 已导入的课程表
描述 提示导入是否成功，显示成功导入的课程表。
跳转 本页

/subject/bind/?data *POST*

参数 *data* 待绑定的课程和班级、学生
描述 绑定课头。用户自行编辑课程与班级或学生之间的对应关系。系统解析后调用服务端后台接口写入数据库，跳转到绑定状态页面。该界面也提供复选框来解绑课头。
跳转 /subject/bind/?status

/subject/bind/?status *POST*

参数 *status* 导入状态
data 已导入的课程表
描述 提示导入是否成功，显示成功导入的课程表。
跳转 本页

1.1.2 班级管理模块

/class/index/? *GET*

参数 *page* 页码
search 搜索关键字
order 排序方式
asc 排序顺序
描述 显示班级列表。
跳转 本页

/class/import/?file *(multipart)POST*

参数 *type* 文件类型
file 待导入的文件
描述 解析文件中的班级信息，跳转到自定义导入信息页面。
跳转 /class/import/?data

/class/import/?data *POST*

参数 *data* 解析后的表格或映射表
描述 导入班级。提供用户界面，允许选择表格行列与待导入字段对应。调用服务端后台接口写入数据库，跳转到导入状态页面。
跳转 /class/import/?status

/class/import/?status *POST*

参数 *status* 导入状态
data 已导入的班级
描述 提示导入是否成功，显示成功导入的班级。

跳转 本页

1.1.3 学生管理模块

/student/index/? *GET*

参数 *page* 页码
search 搜索关键字
class 班级
order 排序方式
asc 排序顺序

描述 显示学生列表，可以指定班级。

跳转 本页

/student/import/?file *(multipart)POST*

参数 *type* 文件类型
file 待导入的文件

描述 解析文件中的学生信息，跳转到自定义导入信息页面。

跳转 /student/import/?data

/student/import/?data *POST*

参数 *data* 解析后的表格或映射表

描述 导入学生。提供用户界面，允许选择表格行列与待导入字段对应。调用服务端后台接口写入数据库，跳转到导入状态页面。

跳转 /student/import/?status

/student/import/?status *POST*

参数 *status* 导入状态
data 已导入的学生

描述 提示导入是否成功，显示成功导入的学生。

跳转 本页

/student/collect/? *GET*

参数 *none*

描述 采集学生特征信息。教师在该页面向服务端后台发送要求开始采集信息的请求，后台接受请求开始处理，并实时返回采集进度。

跳转 /student/collect/?status

/student/<id>/edit *GET*

参数 *id* 学生内部编号

描述 编辑学生信息。可以在此界面启动对该学生特征的重新采集。

跳转 /student/<id>/?status

1.1.4 考勤管理模块

<code>/signin/do/?start</code>	<i>GET</i>
--------------------------------	------------

参数 *none*

描述 点名。开始一个新的签到进度，教师通过该页面向服务端后台发送请求，后台开始处理签到事务，并实时返回签到进度。

跳转 本页

<code>/signin/record/?id</code>	<i>GET</i>
---------------------------------	------------

参数 *id* 考勤记录编号

描述 考勤记录。显示指定考勤具体数据。可以生成考勤报表。

跳转 本页

1.2 服务端后台

服务端后台是不可见的，服务端前台通过公共类的公共方法调用使用后台提供的公开接口，客户端应用程序则通过约定的通信协议（见第13页第3.7节）与服务端后台进行交互。这并不是一个Web服务器。

1.2.1 课程类 Subjects

<code>public static JSONArray import(JSONArray data)</code>

返回 导入成功的数据。

描述 导入课程表。解析传入的JSON数据为课程表，导入成功后写入数据库并返回导入成功的数据。导入失败则返回错误信息以供参考。

<code>public static JSONArray get()</code>
--

返回 课程表。

描述 获取全部课程表。没有参数，查询数据库中所有课程信息并返回。

<code>public static JSONArray get(int page, int pageSize, ... /* String order, bool asc */)</code>
--

返回 课程表。

描述 获取指定页课程表。传入页码和每页数量，省略的参数为排序方式（默认为按编号顺序），返回按参数指定的课程表。

<code>public static JSONObject get(int id)</code>

返回 课程信息。

描述 获取课程信息。传入该课程的唯一ID，在数据库中查询后返回该课程信息。失败返回空。

<code>public static JSONObject update(int id, JSONObject data)</code>

返回 更新后的课程信息。

描述 修改指定课程。传入该课程的唯一 ID 和要修改的内容，解析后在数据库中进行更新，返回更新成功后的该课程信息。失败则返回错误信息。

1.2.2 班级类 Classes

public static JSONArray import(JSONArray data)

返回 导入成功的数据。

描述 导入班级表。解析传入的 JSON 数据为班级表，导入成功后写入数据库并返回导入成功的数据。导入失败则返回错误信息以供参考。

public static JSONArray get()

返回 班级表。

描述 获取全部班级表。没有参数，查询数据库中所有班级信息并返回。

public static JSONArray get(int page, int pageSize, ... /* String order, bool asc */)

返回 班级表。

描述 获取指定页班级表。传入页码和每页数量，省略的参数为排序方式（默认为按编号顺序），返回按参数指定的班级表。

public static JSONObject get(int id)

返回 班级信息。

描述 获取班级信息。传入该班级的唯一 ID，在数据库中查询后返回该班级信息。失败返回空。

public static JSONObject update(int id, JSONObject data)

返回 更新后的班级信息。

描述 修改指定班级。传入该班级的唯一 ID 和要修改的内容，解析后在数据库中进行更新，返回更新成功后的该班级信息。失败则返回错误信息。

1.2.3 学生类 Students

public static JSONArray import(JSONArray data)

返回 导入成功的数据。

描述 导入学生表。解析传入的 JSON 数据为学生表，导入成功后写入数据库并返回导入成功的数据。导入失败则返回错误信息以供参考。

public static JSONArray get()

返回 学生表。

描述 获取全部学生表。没有参数，查询数据库中所有学生信息并返回。

public static JSONArray get(int page, int pageSize, ... /* String order, bool asc */)

返回 学生表。

描述 获取指定页学生表。传入页码和每页数量，省略的参数为排序方式（默认为按编号顺序），返回按参数指定的学生表。

public static JSONObject get(int id)

返回 学生信息。

描述 获取学生信息。传入该学生的唯一ID，在数据库中查询后返回该学生信息。失败返回空。

public static JSONObject update(int id, JSONObject data)

返回 更新后的学生信息。

描述 修改指定学生。传入该学生的唯一ID和要修改的内容，解析后在数据库中进行更新，返回更新成功后的该学生信息。失败则返回错误信息。

1.2.4 签到类 Signin

public static void do(int subjectId)

返回 无

描述 开始一个签到进程。传入带签到的课程编号，系统检测当前时间来判断当前待签到学生信息。系统启动新线程开始签到，新线程向客户端发送签到请求进行签到。没有返回值。

public static JSONObject status()

返回 签到状态。

描述 获取当前签到状态。由服务器前端轮询，返回当前已经点到的学生信息。

1.2.5 识别类 Identification

public static JSONObject init(int studentId, JSONObject data)

返回 初始化情况。

描述 初始化学生识别信息。传入参数为学生编号和采集到的数据，系统分析数据特征并存入数据库。返回初始化情况。

public static JSONObject update(int studentId, JSONObject data)

返回 更新情况。

描述 更新学生识别信息。传入参数为学生编号和采集到的数据，系统分析数据特征并更新数据库。返回更新情况。

1.2.6 通信类 Communication

通信类提供基于第13页第3.7所述通信协议的收发，启动监听进程和发送进程属于具体实现而并非接口设计，这里不再阐述。

public static void send(JSONObject *datagram*)

返回 无。

描述 发包。向客户端传送的数据包对象。

1.2.7 日志类 Log

public static void log(int *level*, String *message*)

返回 无。

描述 写入日志。传入日志等级和日志消息，写出带时间的日志到日志文件和数据库。

2 客户端

客户端与服务端后台进行通信，是C/S架构中的C。客户端本身采用MVC（模型-视图-控制器）的架构来设计，简述如下。由于没有实际的Android编程经验，该设计存在不可避免的错误。

2.1 模型 model

2.1.1 学生类 StudentModel

提供对学生访问和识别的方法。

2.1.2 设备类 DeviceModel

提供对设备唯一识别ID访问和确认的方法。

2.1.3 图像类 ImageModel

一个图像数据的包装类。

2.1.4 声音类 AudioModel

一个声音数据的包装类。

2.2 视图 ui

2.2.1 主界面类 MainActivity

MainActivity **extends** Activity

@Override public void onCreate(Bundle *savedInstanceState*)

返回 无。

描述 创建主界面。载入主控制器。显示欢迎信息并等待接收指令。

2.2.2 登录类 LoginActivity

LoginActivity **extends** Activity

@Override public void onCreate(Bundle savedInstanceState)

返回 无。

描述 创建登录界面。允许学生用户输入学号和密码以登录系统。

2.2.3 签到类 SignInActivity

SignInActivity **extends** Activity

@Override public void onCreate(Bundle savedInstanceState)

返回 无。

描述 创建签到界面。提示用户要通过验证，应该做的动作和发出的声音。

2.2.4 验证类 VerifyActivity

VerifyActivity **extends** Activity

@Override public void onCreate(Bundle savedInstanceState)

返回 无。

描述 创建验证界面。即时显示从前置摄像头获取的图像信息，动态录取话筒接收的声音波形图。

2.3 控制器 logic

2.3.1 主控类 MainServices

public void login(String id, String passwd)

返回 无。

描述 执行登录。将用户名和密码（加密后）打包，调用通信控制器向服务端发送数据包，等待服务端返回登录状态。如果成功则跳转到登陆成功页面，失败则提示继续登录。

public interface signInListener {}

描述 签到监听器。创建与服务端链接的套接字，并监听服务器的签到请求，跳转到签到界面。

public void signIn()

返回 无。

描述 执行签到。获取当前环境的学生信息，向服务端发送待签到学生的识别信息，等待服务端返回签到内容，进入签到验证逻辑。

public void verify()

返回 无。

描述 执行验证。获取用户的声音和图像信息，检测到可识别的图像和声音之后，与服务端建立连接并上传数据，等待服务端返回验证信息。如果验证成功，提示签到成功；如果验证失败，返回验证逻辑继续采集并签到。

2.3.2 通信类 Communication

虽然作为客户端，但是我们需要监听服务端发送来的指令，所以也应提供相应接口。当然，说是自定义的通信类，实际上是对套接字类做了一个封装而已。

public Communication()

返回 无。

描述 创建与服务端连接的套接字。从配置文件中读入服务器地址，建立相应连接。失败则抛出异常，由调用者捕获。

public JSONObject recv()

返回 数据包。

描述 接收数据。从服务器接收数据，监听服务器发送来的指令，当没有监听到时返回空，监听到时返回获得的数据。该方法需要轮询调用。

public void send(JSONObject 数据包)

返回 无。

描述 传送数据。向服务器发送该数据包，该方法将一直阻塞直到服务器确认数据已经收到。

public void close()

返回 无。

描述 关闭连接。

2.3.3 图像处理类 ImageProcessor

public bool isPortrait(Images 图像)

返回 是否为人像。

描述 检查图像是否为人像。程序将图像处理为矩阵进行计算，判断是否为人的形象。当吻合率达到设定比率时返回真。注意，判断是否为本人的逻辑应在服务端完成而不是客户端！

2.3.4 音频处理类 AudioProcessor

public bool isSentence(Audios 声音, String 句子)

返回 是否为句子。

描述 检查声音是否为指定句子。程序将声音分解为语音元素，判断语音是否与句子类似。当吻合率达到设定比率时返回真。同样注意，判断是否为本人的逻辑应在服务端完成而不是客户端！

3 数据处理约定

为了简化格式解析处理流程，所有以字符串形式交换数据的格式，均采用以UTF-8编码的JSON表述。数据在传输过程中可以取消缩进和压缩以节省传输成本，但这里用缩进来明晰其嵌套层次。

3.1 课程表格式

课程表是一个数组，其中每个元素为一个课程对象。课程对象包含数个键值对以表示该课程。数组可以为空，表示没有相应数据，这样就可以在没有班级和学生的情况下导入单纯的课程。

```
[
  {
    id:      1,
    name:    "软件工程概论",
    teacher: "王珂",
    course:  [
      {
        classes:  [2, 3], // 班级ID
        students: [], // 学生ID, 如重修跟班的学生
        weeks:    [1, 2, 3, 4, 5, 6, 7, 8],
        times:    [
          {
            weekday: 2,
            start:   "8:30",
            end:     "10:05"
          },
        ],
        classroom: "4538"
      }
    ]
  }
]
```

3.2 班级表格式

班级表是一个数组，其中每个元素为一个班级对象。班级对象包含数个键值对以表示该班级。

```
[
  {
    id:          3,
    name:         "软件工程1305班",
    shortName:    "软件1305",
    college:      "信息科学与工程学院",
    grade:        2013,
    students:     [1, 2, 3, 1098, 1099, 1100], // 学生编号
    memo:         "备注"
  }
]
```

3.3 学生表格式

学生表是一个数组，其中每个元素为一个学生对象。学生对象包含数个键值对以表示该学生，这里只给出基本键值对，如有需求可以随时添加。

```
[
  {
    id:          1100,
    sid:          "201316920311"
    name:         "田劲锋",
    memo:         "备注"
  }
]
```

3.4 消息格式

消息格式约定了调用公共方法后，出现异常情况时返回值的格式。

```
{
  level:         "error",
  message:        "出错具体信息",
  timestamp:     1431416795,
  from:          "出错模块"
}
```

数组形式为：

```
[  
  "error",  
  "出错具体信息",  
  1431416795,  
  "出错模块"  
]
```

3.5 音频数据格式

OGG。

3.6 图像数据格式

PNG。

3.7 通信协议

系统通信协议基于经典的TCP/IP协议，其中IP是IPv4。数据的拆解包可以由系统套接字接口实现，数据包中传递压缩的JSON格式字符串。关于TCP/IP的具体实现不再赘述。