

接口:

APP 通过

<http://52.74.225.18/app2016/index.php?schoolid=eZlearn&method=GetServerURL>

来获得学校 ID 所对应的 Server Address:

如 schoolid 是 kudospark 对应的 url:

<http://52.74.225.18/app2016/index.php?schoolid=kudospark&method=CheckUser>

来判断用户信息

具体接口定义：

接口	输入	输出	备注
UpdateCharactersInfo(更新素材库)	UpdateTime:DateTime; 精确到秒 SchoolID:String	UpdateNeeded:0 不需要更新; 1 需要更新	上传本地数据库的最新的更新时间和对应的学校; 如果当前学校和服务器一致, 并且更新时间和服务器的更新时间一致, 则不用更新  当接口返回 1 的时候, 需要通过 DownloadCharactersInfo 来更新本地数据库
DownloadCharactersInfo	UpdateTime:DateTime	JSON 数组, 每一行对应一个素材表 [	得到新的素材库列表, 更新本地

		{ID, ChineseName, EnglishName, ChineseVoice, EnglishVoice, ImageURL, AnimationFolder, SceneID, UpdateTime}, {ID, ChineseName, EnglishName, ChineseVoice, EnglishVoice, ImageURL, AnimationFolder, SceneID, UpdateTime}}	数据库；下载新的素材 更新本地数据库的学校和更新时间信息
GetServerURL(获取学校地址 URL)	SchoolID: String IPAddress: String; IP地址	URL: String 返回学校对应的 URL	相关的数据表: School
CheckUser (检查学生 ID 是否正确)	StudentID: 学生 ID, 字符串 SchoolID: 学校 ID, 字符串	IsSuccess: 0 失败, 1 成功	相关的数据表: Student
UpdateUserID(随便玩玩时的用户信息获得)	UserName: 用户输入的名字 Email: 用户输入的 Email loginMachnie: 用户登录机器 device ID currentDateTime: 当前时间戳	IsSuccess: 0 失败, 1 成功 UserID : 用户 ID	后台根据当前时间和用户机器 deviceID 生成 UserID 相关数据表: User
GetAboutText (关于信息)	SchoolID: 学校的 ID 未登录状态为 kudospark APPStatus : 0 : 登录状态 ; 1 : 未登录状态	aboutText 返回分享的文字	SchoolInfo
GetShareText(用户分享信息)	UserID : 用户的 ID SchoolID: 学校的 ID; 未登录状态为空 APPStatus: 0 : 登录状态 ; 1 : 未登录状态	shareText 返回分享的文字	
GetRandomQuestions (随机获取题目接口)	StudentID : 字符串 SceneID : 0-1 (对应现有的 2 个场景) QuestionNumber : 获取题目的数据,	ReturnQestionNum : 返回的题目个数 GameID : 0-2 (对应现有的 3 款游	相关的数据表: Question Characters Scene 随机返回题目。

	<p>最大值 10, 超过 10 只返回 10 道题目</p> <p>Language: 0: 中文, 1: 英文, 2: 随机</p>	<p>戏, 具体在下面解释)</p> <p>SceneID 场景 ID</p> <p>Questions: 题目数组, 里面的每一项是对应游戏的一个题目:</p> <p>[ {QuestionID, Question}, {QuestionID, Question} ]</p> <p>QuestionID: 对应题目的 ID; int</p> <p>Question: 对应的题目, 具体每个游戏的题目在下面解释</p>	<p>题目语言 根据参数决定</p>
GetCorrectAnswer	<p>GameType: 1 语音; 2 写字</p> <p>Data: 语音宝 (学生发音); 写字宝 (学生写字的图片)</p> <p>Text: 对应的文字</p> <p>RecogText: 识别的文字</p> <p>RecogScore: 识别的分数</p>	<p>IsCorrect: 0 失败, 1 成功</p> <p>Points: 得分</p> <p>Text: 服务器识别的文字</p>	<p>判断学生的发音/写字是否准确</p>
GetQuizQuestions (获取题目接口)	<p>StudentID: 学生 ID, 字符串</p> <p>SceneID: 场景 ID</p> <p>QuestionPos: 当前和该学生相关的题目数组的获取位置, -1 代表从该学生最新的题目开始获取, 默认情况下从最新的题目获取</p> <p>QuestionNumber: 获取题目的数据, 最大值 50, 超过 50 只返回 50 道题目</p> <p>QuestionOrder: 0 (默认, 从当前题目往之前的题目获取),</p>	<p>CurrentQestionPos: 在所有返回的题目中的数组下标最大的; APP 端根据当前题目位置判断下次获取题目的起始位置。</p> <p>TotalQuestionSize: 所有分配给该学生的题目的总数</p> <p>ReturnQestionNum: 返回的题目个数</p> <p>SceneID: 场景 ID</p> <p>GameID: 0-2 (对应现有的 3 款游戏, 具体在下面解释)</p>	<p>相关的数据表: Quiz</p>

	1(从当前的题目往之后的题目获取)	<p>Questions:题目数组，里面的每一项是对应游戏的一个题目：</p> <pre>[ {QuizID, Question}, { QuizID, Question} ]</pre> <p>QuizID: 对应 Quiz 的 ID; int Question: 对应的题目，具体每个游戏的题目在下面解释</p>	
SubmitStudentLogs (返回学生做题信息,保存到数据库中)	<p>StudentID: 学生 ID, 字符串 GameID: 0-2 （对应现有的 3 款游戏，具体在下面解释） SceneID : 0-1 QuizID: Quiz 的 ID, int 在测试状态；非测试状态为-1 Question:完整的题目 SubmitDate: 上传的日期和时间(2013-01-01 20:11:45 年-月-日 小时:分钟：秒钟) Click No.:点击的次数 SpentTime: 花费的时间（秒）</p> <p>IsCorrect: 是否正确，0:错误；1 正确</p> <p>TotalLogID: APP 第一次进入场景的时候生成： StudentID+DeviceID+DateTime QuizStatus: 0:测试状态；1：随便做题状态</p>	<p>IsSuccess: 0 失败，1 成功</p>	<p>相关的数据表：</p> <p>Log Student Question Quiz Teacher</p> <p>把学生，题目，完成的统计情况，分配题目的老师，出题老师的情况保存到 Log 中</p> <p>学生多次完成 视为不正确。[Game 为连连看是 需要 4 次；其他两种游戏均为 1 次]</p> <p>Question 题目中素材数组，只需上传素材 ID [CharacterID, CharacterID, CharacterID, CharacterID]</p>
SubmitStudentTotalLogs(返回学生完成一次游戏的总体情况)	<p>StudentID: 用户 ID, 字符串 SceneID : 0-1</p>	<p>IsSuccess: 0 失败，1 成功</p>	<p>相关的数据表: TotalLog</p>

	<p>SubmitDate: 上传的日期和时间(2013-01-01 20:11:45 年-月-日 小时:分钟 : 秒钟)</p> <p>StateTime:开始时间</p> <p>EndTime:结束时间</p> <p>SpentTime: 总花费的时间 (秒)</p> <p>TotalQuestions: 总做过的题目数</p> <p>TotalCorrectQuestions:总做对的题目数</p> <p>TotalLogID:APP 生成</p>		<p>把学生, 题目, 完成的统计情况保存到 TotalLog 中</p>
SubmitUserLogs (返回学生每道题的完成情况)	<p>UserID: 用户 ID,字符串</p> <p>SceneID : 0-1</p> <p>GameID: 0-2 (对应现有的 3 款游戏, 具体在下面解释)</p> <p>Question: 题目 (具体在下面解释)</p> <p>SubmitDate: 上传的日期和时间(2013-01-01 20:11:45 年-月-日 小时:分钟 : 秒钟)</p> <p>Click No.:点击的次数</p> <p>SpentTime: 花费的时间 (秒)</p> <p>IsCorrect: 是否正确, 0:错误 ; 1 正确</p> <p>TotalLogID: APP 第一次进入场景的时候生成 : DeviceID+当前时间</p>	IsSuccess: 0 失败, 1 成功	<p>相关的数据表: User UserLog</p> <p>把用户, 题目, 完成的统计情况保存到 UserLog 中</p> <p>Question 题目中素材数组, 只需上传素材 ID [CharacterID, CharacterID, CharacterID, CharacterID]</p>
SubmitUserTotalLogs(返回学生完成一次游戏的总体情况)	<p>UserID: 用户 ID,字符串</p> <p>SceneID : 0-1</p> <p>SubmitDate: 上传的日期和时间(2013-01-01 20:11:45 年-月-日 小时:分钟 : 秒钟)</p>	IsSuccess: 0 失败, 1 成功	<p>相关的数据表: UserTotalLog</p> <p>把用户, 题目, 完成的统计情况保存到 TotalLog 中</p>

	SpentTime: 总花费的时间 (秒) TotalQuestions: 总做过的题目数 TotalCorrectQuestions: 总做对的题目数 TotalLogID: APP 生成		
--	--	--	--

## 1. 题目接口具体定义：

游戏：

### 1.1 看图识字

游戏 ID: 0

题目表现形式：如下：

```
{
  Characters: [CharacterID, CharacterID, CharacterID, CharacterID] 素材数组
  IsEnglish: 0:中文, 1 英文
  Choice: 1-4 （选项对应的, 素材数组 1 到 4）
  DisplayText: 0 不显示；1 显示 （显示文字）
  Type 选择类型（1. 选择文字；2. 发音；3. 写字）[随机只返回 1]
}
```

### 1.2 听音识字

游戏 ID: 1

题目表现形式：如下：

```
{
  Characters: [CharacterID, CharacterID, CharacterID, CharacterID] 素材数组
  IsEnglish: 0:中文, 1 英文
  Choice: 1-4 （选项对应的, 素材数组 1 到 4）
  DisplayText: 0 不显示；1 显示 （显示另外一种文字, 当 isEnglish=1 的时候, 是否显示中文）
}
```

### 1.3 连连看

游戏 ID: 2

题目表现形式：如下：

```
{Characters: [CharacterID, CharacterID, CharacterID, CharacterID] 素材数组
  IsEnglish: 0:中文, 1 英文
}
```

素材数组的每个元素：

```
{CharacterID: ChineseName: EnglishName: PictureURL: ChineseVoice: EnglishVoice:
  AnimationNumber: AnimationFolder}
```