

Ez-Learn Web Server 1 需求说明

1. 和 APP 进行通信，数据格式使用 JSON
2. 学校版：每个学校对应一个数据库：根据学校的连接不同的数据库

个人版对应数据库 xx.xx.xx/kudospark/

3. 接口具体定义：

接口	输入	输出	备注
GetServerURL(获取题目 URL)	SchoolID: String IPAddress: String; IP 地址	URL: String 返回学校对应的 URL	
学校版			
GetQuestions (获取题目接口)	<p>StudentID: 学生 ID, 字符串</p> <p>GameID: 0-13 (对应现有的 14 款游戏，具体在下面解释)</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), 1(从当前的题目往之后的题目获取)</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>GameID: 0-13 (对应现有的 14 款游戏，具体在下面解释)</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>	相关的数据表：Quiz
SubmitLogs (返回学生)	StudentID: 学生 ID,	IsSuccess: 0 失败，	相关的数据表：

做题信息,保存到数据库)	字符串 GameID: 0-13 (对应现有的 14 款游戏, 具体在下面解释) QuestionID: 题目的 ID, int 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) ClickArray: 点击的游戏各个选项的次数, String “选项 1: 次数;选项 2: 次数...” IsCorrect: 是否正确, 0:错误; 1 正确	1 成功	Log Student Question Quiz Teacher 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
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
个人版-题目接口			
IND_GetQuestions (获取题目接口)	StudentID: 字符串 GameID: 0-13 (对应现有的 14 款游戏, 具体在下面解释) DifficultLevel: 题目难度, 0-6 (对应 7 个难度级别) QuestionNumber: 获取题目的数据, 最大值 10, 超过 10 只返回 10 道题目	DifficultLevel: 返回题目的难度, 0-6 (对应 7 个难度级别) ReturnQuestionNum: 返回的题目的个数 GameID: 0-13 (对应现有的 14 款游戏, 具体在下面解释) Questions: 题目数组, 里面的每一项是对应游戏的一个题目: [{QuestionID, Question}, {QuestionID, Question}]	相关的数据表: Question 根据题目难度随机返回题目。

		QuestionID: 对应题目的 ID; int Question: 对应的题目，具体每个游戏的题目在下面解释	
IND_SubmitLogs (返回学生每道题的完成情况)	StudentID: 学生 ID, 字符串 GameID: 0-13 (对应现有的 14 款游戏, 具体在下面解释) QuestionID: 题目的 ID, int 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) ClickArray: 点击的游戏选项序列, 字符串, 具体每个游戏的序列在下面解释 IsCorrect: 是否正确, 0:错误; 1 正确	IsSuccess: 0 失败, 1 成功	相关的数据表: Log Student Question Quiz 把学生, 题目, 完成的统计情况保存到 Log 中
IND_SubmitTotalLogs(返回学生完成一次游戏的总体情况)	UserID: 用户 ID, 字符串 GameID: 0-13 (对应现有的 14 款游戏, 具体在下面解释) SubmitDate: 上传的日期和时间(2013-01-01 20:11:45 年-月-日 小时:分钟: 秒钟) SpentTime: 总花费的时间 (秒) TotalQuestions: 总做过的题目数 TotalCorrectQuestions: 总做对的题目数	IsSuccess: 0 失败, 1 成功	相关的数据表: TotalLog 把学生, 题目, 完成的统计情况保存到 TotalLog 中
个人版-登录退出接口			
IND_GetStudentID (得到学生的 Student ID)	loginID: 用户登录信息(game center 账号)-可为空 loginName: 根据 game center ID 得	Student ID : 字符串 IsSuccess: 0 失败, 1 成功	相关的数据表: Student 根据 loginID 判断是否存在用户表

	到, 可为空 loginMachnie: 用户 登录机器 (Ipad/Iphone 的 UIUD)		<p>中, 如果不存 在, 则通过 loginMachine 判断 是否存在用户表 中; 如果存在: 返回 StudentID</p> <p>如果 loginID 不存 在, loginMachine 存在, 则添加 loginID 到对应的 loginMachine 表项 中, 返回 StudentID</p> <p>如果 loginID 和 loginMachine 都不 存在: 在数据表 中添加一条记 录, 并生成一个 唯一的 StudentID 返回。</p> <p>如果获取 loginName;把名字 保存到数据库表 中</p>
IND_SubmitUserActivit y(退出是提交用户使 用时间)	StudentID: 学生 ID, 字符串 TotalTime:总的使用 时间 login_date: 登录时间 (2015-11-11 年-月- 日)	IsSuccess: 0 失败, 1 成功	相关的数据表: Student User_Login_Activit y 添加 StudentID, login_date, totalTime 到 user_login_Activity 表中
IND_GetDifficultLevel(获取难度分级信息)	空	Levels: 难度数组 [{0, Easy} {1, 3}]	相关的数据表: DifficultLevel 从 DifficultLevel 中 读取 Level 难度返 回。
个人版-分享			

IND_GetGameShareInfo(用户游戏 Game Over 的时候弹出)	StudentID: 学生 ID, GameID: 游戏 ID	ShareText:分享的文字	相关的数据表: Student TotalLogs 根据最新和上一次的学生的 TotalLog 信息生成 ShareText: “我这次在 xx 游戏中坚持了 xx 秒, 完成了 xx 道题目, 获得了 xx 积分, 比上次[多/少]做了 xx 道题目 (和上次做的题目一样), 在所有的小朋友中排名第 xx,欢迎来 EzLearn 易学岛挑战我把” APP 端根据 ShareText 生成分享图片, 在 ShareText 后面添加 APP 下载链接
IND_GetAPPShareInfo (用户点击游戏红心的时候弹出)	StudentID: 学生 ID, HeartNumber:红心的数目 TotalTime:玩的时间	ShareText:分享的文字	ShareText: “我在 EzLearn 易学岛中一共玩了 xx 分钟, 积累了 xx 颗红心。你也一起来和我玩吧” APP 端根据 ShareText 生成分享图片, 在 ShareText 后面添加 APP 下载链接
IND_GetAwardShareInfo (用户点击兑换礼品的时候弹出, 用户不分享不能兑换礼品)	StudentID: AwardID: shareID: [facebook, qq, twitter,email]	ShareText: 分享的文字	相关的数据表: Award AwardList 从 Award 表中读取分享的文字返回
IND_GetTaskShareInfo (用户点击兑换礼品的时候弹出, 用户不分享不能兑换礼品)	StudentID: TaskID: shareID: [facebook, qq,twitter, email]	ShareText: 分享的文字	相关的数据表: Task TaskList 从 Task 表中读取分享的文字返回

个人版-任务奖品			
IND_GetTasks(得到任务列表, 用户兑换任务后要更新列表)	StudentID:	TaskDescription: 在顶端显示的描述 Tasks: 任务列表: [{TaskID, TaskDetail, Pic, Redeemable}] TaskDetail:任务描述 Pic, Url,可为空 Redeemable:是否可兑换	相关的数据表: Task 读取所有 Tasks;判断是否可以兑换返回
IND_GetAwards(得到奖品列表, 用户兑换积分后要更新列表)	StudentID:	AwardsDescription: 在顶端显示的描述 Awards: 奖品列表: [{AwardID, AwardDetail, Pic, Credits, Redeemable}] AwardsDetail:奖品描述 Pic: Url,可为空 Redeemable: 是否可兑换	相关的数据表: Awards Students 读取所有 Awards;根据积分判断是否可以兑换返回
IND_RedeemAwards(用户分享后调用, 调用更新奖品列表)	StudentID: AwardID: shareID: [facebook, qq, twitter, email]	IsSuccess: 0 失败, 1 成功	相关的数据表: Awards Students AwardList Student 减去相应的积分 把 StudentID, Award Detail; ShareID 存入 AwardList 表中。
IND_RedeemTasks(用户分享后调用, 调用更新任务列表)	StudentID: TaskID: shareID: [facebook, qq, twitter, email]	IsSuccess: 0 失败, 1 成功	相关的数据表: Awards Students TaskList 需要做相应的操

			作 把 StudentID, Task Detail ShareID 存 入 AwardList 表 中。
个人版-其他			
IND_SubmitSchoolInfo	StudentID: SchoolName	IsSuccess: 0 失败, 1 成功	相关的数据表: SchoolInfo 保存到数据表中

4. 游戏接口具体定义

1. 字母宝

游戏 ID: 0

题目的表现方式: 无

2. 字母锤

游戏 ID: 1

题目的表现方式:

题目表现方式: JSON 数组: 如下

```
{
  "pinyin": "a",
  "audio_path": "a.mp3",
  "choiceA": "a",
  "choiceB": "b",
  "choiceC": "c",
  "choiceD": "d",
  "answer": 1
}
```

pinyin: 字母

audio_path: 音频文件名 //路径在 APP 预先设置好

choiceA/B/C/D: 1,2,3,4 四个选择

answer: 正确答案, 1, 2, 3, 4

3. 字母筐

游戏 ID: 2

题目的表现方式:

题目表现方式: JSON 数组: 如下

```
{
  "pinyin": "a",
  "audio_path": "a.mp3",
  "choiceA": "a",
  "choiceB": "b",
```

```
"choiceC": "c",  
"choiceD": "d",  
"answer": 1  
}  
pinyin: 字母  
audio_path: 音频文件名 //路径在 APP 预先设置好  
choiceA/B/C/D: 1,2,3,4 四个选择  
answer: 正确答案, 1, 2, 3, 4
```

4. 听音识字

游戏 ID: 3

题目的表现方式: JSON 数组: 如下

```
{  
  "pinyin": "zai",  
  "audio_path": "zai.mp3",  
  "choiceA": "在",  
  "choiceB": "和",  
  "choiceC": "一",  
  "choiceD": "的",  
  "answer": 1  
}
```

pinyin: 字符串

audio_path: 音频文件名 //路径在 APP 预先设置好

choiceA/B/C/D: 1,2,3,4 四个选择

answer: 正确答案, 1, 2, 3, 4

5. 碰碰识字

游戏 ID: 4

题目表现方式: JSON 数组: 如下

```
[  
  {"pinyin": "zai", "audio_path": "zai.mp3", "word": "在"},  
  {"pinyin": "he", "audio_path": "he.mp3", "word": "和"},  
  {"pinyin": "yi", "audio_path": "yi.mp3", "word": "一"},  
  {"pinyin": "de", "audio_path": "de.mp3", "word": "的"}  
]
```

pinyin: 拼音

audio_path: 音频文件

word: 字

//三者是一一对应的

6. 识字锤

游戏 ID: 5

题目表现方式: JSON 数组: 如下

```
{
```



```
"pinyin": "zai",  
"audio_path": "zai.mp3",  
"choiceA": "在",  
"choiceB": "和",  
"choiceC": "一",  
"choiceD": "的",  
"answer": 1  
}
```

pinyin: 字符串

audio_path: 音频文件名 //路径在 APP 预先设置好

choiceA/B/C/D: 1,2,3,4 四个选择

answer: 正确答案, 1, 2, 3, 4

7. 笔画宝

游戏 ID:6

题目的表现方式: JSON 数组: 如下

```
{  
  strokeNo: 8 //选择正确的 strokeNo  
  [  
    {"strokeNo": 6, "audio_path": "zai.mp3", "word": "在"},  
    {"strokeNo": 8, "audio_path": "he.mp3", "word": "和"},  
    {"strokeNo": 1, "audio_path": "yi.mp3", "word": "一"},  
    {"strokeNo": 8, "audio_path": "de.mp3", "word": "的"}  
  ]  
}
```

strokeNo: 笔画数

audio_path: 音频文件

word: 字

//三者是一一对应的

8. 拼字宝

游戏 ID: 7

题目的表现方式: JSON:

```
{  
  "part": "土",  
  "audio_path": "di.mp3",  
  "choiceA": "也",  
  "choiceB": "夜",  
  "choiceC": "口",  
  "choiceD": "飞",  
  "answer": 1  
}
```

part: 部首

audio_path: 正确汉字的音频文件名 //路径在 APP 预先设置好

choiceA/B/C/D: 1,2,3,4 四个选择

answer: 正确答案, 1, 2, 3, 4

9. 拼字锤

游戏 ID: 8

题目的表现方式: JSON:

```
{
  "part": "土",
  "audio_path": "di.mp3",
  "choiceA": "也",
  "choiceB": "夜",
  "choiceC": "口",
  "choiceD": "飞",
  "answer": 1
}
```

part: 部首

audio_path: 正确汉字的音频文件名 //路径在 APP 预先设置好

choiceA/B/C/D: 1,2,3,4 四个选择

answer: 正确答案, 1, 2, 3, 4

10. 拼字框

游戏 ID: 9

题目的表现方式: JSON 数组:

```
{
  "part": "土",
  [
    {"part": "也", "audio_path": "di.mp3", "correct": 1},
    {"part": "夜", "audio_path": "", "correct": 0},
    {"part": "申", "audio_path": "kun.mp3", "correct": 1},
    {"part": "飞", "audio_path": "", "correct": 0}
  ],
  ....
}
```

Part:部首

Audio_path:正确汉字的音频文件名, 错误选项为空

Correct: 是否可以组成汉字

11. 组词宝

游戏 ID: 10

题目的表现方式: JSON 数组: 如下

```
{
  {character: "天", "audio_path": "tian.mp3"} //题目的汉字
  [
    {"correct": 1, "audio_path": "kong.mp3", "word": "空"},
    {"correct": 0, "audio_path": "wo.mp3", "word": "我"},
    {"correct": 1, "audio_path": "lan.mp3", "word": "蓝"},
    {"correct": 1, "audio_path": "xia.mp3", "word": "下"}
  ]
}
```

Correct:是否可以组词

audio_path: 音频文件

word: 字

//三者是一一对应的

12. 碰碰搭词

游戏 ID: 11

题目的表现方式: JSON 数组: 如下

```
[
  {"part_word1": "一个", "part_word2": "苹果"},
  {"part_word1": "一张", "part_word2": "桌子"},
  {"part_word1": "一片", "part_word2": "面包"},
  {"part_word1": "一匹", "part_word2": "小马"},
]
```

13. 句子宝

游戏 ID: 12

题目的表现形式: JSON 数组: 如下

```
{
  "word": "我们",
  "word": "祖国的",
  "word": "花朵"
}
```

Miss_position: 2

"choiceA": "是",

"choiceB": "夜",

"choiceC": "口",

"choiceD": "飞",

"answer": 1

}

Word: 组成句子的词语, 按顺序排列

Miss_position: 句子的第几个词语丢失了

choiceA/B/C/D 选择

answer 正确的答案

14. 句子桥

游戏 ID: 13

题目的表现形式: JSON 数组: 如下

```
{
  "word": "我们",
  "word": "是",
  "word": "祖国的",
  "word": "花朵"
}
```

"sentence": "我们是祖国的花朵"

}

Word: 组成句子的词语