# 算法API调用说明书

2016秋季产品

|  |  |  |
| --- | --- | --- |
| 修改时间 | 修改人 | 备注 |
| 2016-11-07 | 汪彪 | 添加先行测试+能力值计算接口 |
| 2016-11-17 | 汪彪 | 添加算法环境说明 |
| 2016-01-13 | 汪彪 | 添加时间维度先行测试 |

## 环境说明(URL)

|  |  |  |
| --- | --- | --- |
| **环境** | **IP地址** | **说明** |
| 学习系统生产环境 | 139.196.75.14 | 秋季学习系统、学习系统所需算法服务器 |
| 招生抓手生产环境 | 139.196.136.134 | 各类招生抓手所需算法服务器 |
| 测试环境(All) | 139.196.92.195 | 所有应用算法系统测试服务器 |

使用说明：

所有算法已进行封装，开发团队不需要了解具体算法内容，只需要根据算法简称和所提供的接口便可创建相应算法对象并调用相应程序获取算法结果。

**Algorithm**类是所有算法继承的接口

**AlgoCreator**类的静态类函数***CreateAlgorithm***()用于创建算法类的实例

***CreateAlgorithm***()只带有一个参数，表示算法的名称，比如：”**Ability**”表示能力估计算法

***getResult***函数是所有算法类都有的通用函数，输入**json**格式的参数，将会返回算法所产生的结果，比如：能力估计算法将会返回该学生的能力**latent ability value**

算法使用样本代码：

// 使用算法选择器创建某个算法对象

Algorithm myAlgo = AlgoCreator.CreateAlgorithm(“**Ability**”);

// 执行算法并获取结果

String sJSONOuputData = myAlgo.getResult(**sJSONInputData**);

**红色**代表不同的算法和相应的参数

## 先行测试算法接口v1

Name: KSTMode

API地址： http:// [IP地址](#_环境说明(URL)):8080/AlgoService/algo/kst

使用语法算法时，所需的参数包含六个方面

1. 用户编号，< usr\_id > String 必要
2. 学生所选自评水平，< init\_kstatus >int(1,2,3,4,5) 必要
3. 知识图谱编码，< kmap\_code >String 必要
4. 前一个知识点编码，< pre\_knode >String 必要
5. 学生测试问题的答案编号(对-1，错-0),< usr\_ans > String 必要

**返回值：**

返回的JSON字符串中包含 6个方面的值：

**usr\_id：用户编号**

**init\_kstatus：用户对专题掌握的程度**

**kmap\_code：知识图谱编号**

**knode\_toaskq：所得知识点**

**weak\_elems：薄弱知识点列表**

**sErrors ： 错误信息**

## 先行测试算法接口

Name: KSTMode

API地址： http:// [IP地址](#_环境说明(URL)):8080/AlgoService/algo/kstmode

使用语法算法时，所需的参数包含六个方面

1. 用户编号，< usr\_id > String 必要
2. 学生所选自评水平，< init\_kstatus >int(1,2,3,4,5) 必要
3. 知识图谱编码，< kmap\_code >String 必要
4. 前一个知识点编码，< pre\_knode >String 必要
5. 学生测试问题的答案编号(对-1，错-0),< usr\_ans > String 必要
6. 知识点难度级别，< level\_mode > int(1-易(1道题),5-易(2道题)，2-基础(2=<t<=3),3-中(2=<t<=3)) 必要

**返回值：**

返回的JSON字符串中包含 6个方面的值：

**usr\_id：用户编号**

**init\_kstatus：用户对专题掌握的程度**

**kmap\_code：知识图谱编号**

**knode\_toaskq：所得知识点**

**weak\_elems：薄弱知识点列表**

**sErrors ： 错误信息**

## 先行测试算法接口(时间维度)

Name: KSTTime

API地址： http:// [IP地址](#_环境说明(URL)):8080/AlgoService/algo/kstime

使用语法算法时，所需的参数包含9个方面

1. 用户编号，< usr\_id > String 必要
2. 学生所选自评水平，< init\_kstatus >int(1,2,3,4,5) 必要
3. 知识图谱编码，< kmap\_code >String 必要
4. 前一个知识点编码，< pre\_knode >String 必要
5. 学生测试问题的答案编号(对-1，错-0),< usr\_ans > String 必要
6. 知识点难度级别，< level\_mode > int(1-易(1道题),5-易(2道题)，2-基础(2=<t<=3),3-中(2=<t<=3)) 必要
7. 做当前题目所需时间(单位:秒)，<need\_time> int 必要
8. 学生做当前题目所化时间(单位:秒) ，<take\_time > int 必要
9. 当前系统编号，<sys\_code > int，[系统编码表](#_附1：系统编号表)  必要

**返回值：**

返回的JSON字符串中包含 6个方面的值：

**usr\_id：用户编号**

**init\_kstatus：用户对专题掌握的程度**

**kmap\_code：知识图谱编号**

**knode\_toaskq：所得知识点**

**weak\_elems：薄弱知识点列表**

**sErrors ： 错误信息**

## 测试算法接口(Pioneer)

Name: KSTOnline

API地址： http:// [IP地址](#_环境说明(URL)):8080/AlgoService/algo/pioneerkst

使用语法算法时，所需的参数包含六个方面

1. 用户编号，< usr\_id > String 必要
2. 学生所选自评水平，< init\_kstatus >int(1,2,3,4,5) 必要
3. 知识图谱编码，< kmap\_code >String 必要
4. 前一个知识点编码，< pre\_knode >String 必要
5. 学生测试问题的答案编号(对-1，错-0),< usr\_ans > String 必要
6. 不需要学习的知识点编号集合 < excludes > String [] 必要
7. 知识点难度级别，< level\_mode > int(1-易(1道题),5-易(2道题)，2-基础(2=<t<=3),3-中(2=<t<=3)) 必要

**返回值：**

返回的JSON字符串中包含 7个方面的值：

**usr\_id：用户编号**

**init\_kstatus：用户对专题掌握的程度**

**kmap\_code：知识图谱编号**

**knode\_toaskq：所得知识点**

**weak\_elems：薄弱知识点列表**

**nkmap\_code：当前薄弱知识点重组的新知识图谱编号**

**sErrors ： 错误信息**

## 先行测试算法接口(Online&BigMap)

Name: KSTOnline

API地址： http:// [IP地址](#_环境说明(URL)):8080/AlgoService/algo/kstpioneer

使用语法算法时，所需的参数包含六个方面

1. 用户编号，< usr\_id > String 必要
2. 学生所选自评水平，< init\_kstatus >int(1,2,3,4,5) 必要
3. 知识图谱编码，< kmap\_code >String 必要
4. 前一个知识点编码，< pre\_knode >String 必要
5. 学生测试问题的答案编号(对-1，错-0),< usr\_ans > String 必要
6. 知识点难度级别，< level\_mode > int(1-易(1道题),5-易(2道题)，2-基础(2=<t<=3),3-中(2=<t<=3)) 必要

**返回值：**

返回的JSON字符串中包含 7个方面的值：

**usr\_id：用户编号**

**init\_kstatus：用户对专题掌握的程度**

**kmap\_code：知识图谱编号**

**knode\_toaskq：所得知识点**

**weak\_elems：薄弱知识点列表**

**nkmap\_code：当前薄弱知识点重组的新知识图谱编号**

**sErrors ： 错误信息**

## 先行测试+能力值接口

Name: KSTAbility

API地址： http:// [IP地址](#_环境说明(URL)):8080/AlgoService/algo/kstability

使用语法算法时，所需的参数包含10个方面

1. 用户编号，< usr\_id > String 必要
2. 学生所选自评水平，< init\_kstatus >int(1,2,3,4,5) 必要
3. 知识图谱编码，< kmap\_code >String 必要
4. 前一个知识点编码，< pre\_knode >String 必要
5. 学生测试问题的答案编号(对-1，错-0),< usr\_ans > String 必要
6. 知识点难度级别，< level\_mode > int(1-易(1道题),5-易(2道题)，2-基础(2=<t<=3),3-中(2=<t<=3))

4-(t=3) 必要

1. 学生做过的题目(**pre\_knode对应知识点的题目**)的难度，< difficulty > double [] 必要
2. 学生做过的题目(**pre\_knode对应知识点的题目的答案**)对应的得分, < score> String [] 必要
3. 学生的能力(**pre\_knode对应的能力**)估计对应的最大似然函数概率值，< likelihood> [99] 必要
4. 能力评估类型(1-assessment,2-practice)( **pre\_knode对应类型**)，< type > double [] 必要
5. 难度级别数，<total\_level> int (只有3或9两种值，默认为3) 可选 默认是3

注意：difficulty 、score 的每个元素一一对应

**返回值：**

返回的JSON字符串中包含 6个方面的值：

**usr\_id：用户编号**

**init\_kstatus：用户对专题掌握的程度**

**kmap\_code：知识图谱编号**

**knode\_toaskq：所得知识点**

**weak\_elems：薄弱知识点列表**

**error ： 错误信息**

**ability** ：**pre\_knode对应的能力值能力值**

**likelihood：** **该学生目前的能力估计对应的最大似然函数概率值(99个)**

**abilityprob：对pre\_knode整体评估能力可能性值**

## 能力评估算法接口

Name: Abilityx

API地址： http://[IP地址](#_环境说明(URL)):8080/AlgoService/algo/abilityx

使用语法算法时，所需的参数包含4个方面

1. 学生做过的题目的难度，< difficulty > double [] 必要
2. 学生做过的题目对应的得分, < score> String [] 必要
3. 该学生目前的能力估计对应的最大似然函数概率值(**99**个)，< likelihood> double [99] 必要
4. 能力评估类型(1-assessment,2-practice)，< type > double [] 必要

**返回值(4个值)：**

**ability** ：能力值

**likelihood：** 该学生目前的能力估计对应的最大似然函数概率值(99个)

**abilityprob：**整体评估能力可能性值

**error ：** 错误信息

## 下一个学习知识点算法接口

Name: NLIModel

API地址： http:// [IP地址](#_环境说明(URL)):8080/AlgoService/algo/nlix

使用语法算法时，所需的参数包含6个方面

1. 知识图谱编码，< kmap\_code >String 必要
2. 所有知识点编码, < elements\_codes> String [] 必要
3. 学生对所有知识点掌握的能力，< elements\_abilities> double [] 必要
4. 每个知识点，学习的次数，< learning\_counts> int [] 必要
5. 所有薄弱知识点，< weak\_elements> String [] 必要
6. 已学习过的知识点，< learned\_elements> String [] 必要

**返回值(2个值)：**

**next\_element ： 下一个题目的编号**

**error： 错误信息**

## 边学边练推题算法接口

Name: Assessment

API地址：

(A).http:// [IP地址](#_环境说明(URL)):8080/AlgoService/algo/assessmentn

~~(B). http://~~[~~IP地址~~](#_环境说明(URL))~~:8080/AlgoService/algo/assessmentm （使用于题目ID为字符串）~~

使用语法算法时，所需的参数包含4个方面

1. 学生能力值，<ability> double 必要
2. 题目ID集合, < question\_ids> String[] 必要
3. 题目难度值集合，< question\_difficulties > double [] 必要
4. 推题的数量，< assessment\_size > int (默认1即可) 必要
5. 难度级别数，<total\_level> int (只有3或9两种值，默认为3) 可选
6. 知识图谱编号 kmap\_code String 可选
7. 知识点编号 node\_code 可选

**返回值(2个值)：**

**questions ： 题目ID集合 (int[])**

**error： 错误信息**

## 堂堂清推题算法接口(目前没有使用-相似题)

Name: SimilarAssessment

API地址： http:// [IP地址](#_环境说明(URL)):8080/AlgoService/algo/similarquest

使用语法算法时，所需的参数格式如下(如下字符串为三个知识点的数据实例)：

String strm="[{"done\_quest\_map":{"61":"3","21":"3"},

"knode\_code":"key1","

quest\_diff\_map":{"1":"3","71":"2","51":"3","41":"2","31":"3"}

},

{"done\_quest\_map":{"2":"1","52":"3"},

"knode\_code":"key2",

"quest\_diff\_map":{"22":"3","12":"2","72":"2","62":"4","42":"1","32":"3"}

},

{"done\_quest\_map":{"23":"1","73":"5","43":"3"},

"knode\_code":"key3",

"quest\_diff\_map":{"33":"3","44":"1","13":"5","3":"1","63":"3","74":"3","53":"5"}

}]";

1. 学生所学的单个知识点编号，< knode\_code > String 必要
2. 学生做错的题目及题目难度，< done\_quest\_map > Map<String,String> 键为题目编号，值为题目对应的难度值 必要
3. 对应知识点下的题目编号及题目难度(不包含边学边练中做过的题目),

< quest\_diff\_map> Map<String,String> 键为题目编号，值为题目对应的难度值 必要

**返回值(2个值)：**

**questions： 所有知识点及对应的问题编号集合 (Map<String,Integer[]>)**

**error： 错误信息（String）**

**返回结果字符串demo如下：**

**{"questions":"{"key1":["1","51"],"key2":["42","22"],"key3":["44","13","33"]}","error":""}**

## 获取知识图谱每个知识点的前置或后置知识点列表算法接口

Name: KnowledgeNode

API地址： http:// [IP地址](#_环境说明(URL)):8080/AlgoService/algo/knowledgenode

1. 知识图谱编号，<kmap\_code> String 必要
2. 知识点类型（PREREQ：前置，POSTREQ：后置），< kmap\_type >String 必要

**返回值(4个值)：**

**knodes：节点集合 <String,Array>**

**error： 错误信息 String**

**返回数据Demo如下：**

**{"error":"","knodes":"{"eg\_tb\_w\_8aqz1.6":[],"eg\_tb\_w\_8aqz1.5":[],"eg\_tb\_w\_8aqz1.4":[],"eg\_tb\_w\_8aqz1.3":[],"eg\_tb\_w\_8aqz1.2":[],"eg\_tb\_w\_8aqz1.1":["eg\_tb\_w\_8aqz1.3","eg\_tb\_w\_8aqz1.2"]}"}**

## 获取知识图谱每个知识点的前后知识点列表算法接口

Name: RelatedNode

API地址： http:// [IP地址](#_环境说明(URL)):8080/AlgoService/algo/relatednode

1. 知识图谱编号，<kmap\_code> String 必要

**返回值(4个值)：**

**knodes：节点集合 <String,<String,Array > >**

**error： 错误信息 String**

**返回数据Demo如下：**

**{"error":"","knodes":"{"postnodes":{"eg\_tb\_w\_8aqz1.6":[],"eg\_tb\_w\_8aqz1.5":[],"eg\_tb\_w\_8aqz1.4":[],"eg\_tb\_w\_8aqz1.3":[],"eg\_tb\_w\_8aqz1.2":[],"eg\_tb\_w\_8aqz1.1":["eg\_tb\_w\_8aqz1.3","eg\_tb\_w\_8aqz1.2"]},"prenodes":{"eg\_tb\_w\_8aqz1.6":[],"eg\_tb\_w\_8aqz1.5":[],"eg\_tb\_w\_8aqz1.4":[],"eg\_tb\_w\_8aqz1.3":["eg\_tb\_w\_8aqz1.1"],"eg\_tb\_w\_8aqz1.2":["eg\_tb\_w\_8aqz1.1"],"eg\_tb\_w\_8aqz1.1":[]}}"}**

**注意：postnodes：表示后置即可，prenodes：标识前置集合**

## 获取某个知识图谱中每个知识点的型号级别(报告)（目前没有使用）

Name:KSTAR

API地址： http://[IP地址](#_环境说明(URL)):8080/AlgoService/algo/kstar

使用语法算法时，所需的参数格式如下：

1. 用户编号，< usr\_id > String 必要
2. 知识图谱编码，< kmap\_code >String 必要

**返回值(4个值)：**

**usr\_id：用户编号 String**

**kmap\_code：知识图谱编号 String**

**knode\_toaskq：所得知识点 Map<String,String>**

**error： 错误信息 String**

**返回值Demo:**

**{"kmap\_code":"eg\_tb\_r\_7au1","usr\_id":"1000009","kmap\_stars":"{"ev\_tb\_r\_7au1.2":"3","eg\_tb\_r\_7au1.3":"3","ev\_tb\_r\_7au1.3":"0","ev\_tb\_r\_7au1.1":"2","eg\_tb\_r\_7au1.2":"2","eg\_tb\_r\_7au1.1":"1"}","error":""}**

**注意：**

|  |  |
| --- | --- |
| **0** | **无星** |
| **1** | **一星** |
| **2** | **二星** |
| **3** | **三星** |

## 招生抓手所需算法接口

Name: KS

API地址： http://[IP地址](#_环境说明(URL)):8080/AlgoService/algo/ksx

使用语法算法时，所需的参数包含六个方面

1. 知识图谱编号，<**kmap\_code**> String 必要
2. 学生做题答案，**< answer\_status >**int(1,0) 必要
3. 知识图包含的知识点列表，< **element\_list**> String 必要
4. 知识图谱包含的每个知识点对应题目的数量，**<askedquestion\_num>S**tring 必要
5. 知识点概率状态,**< state\_probs>** String 必要
6. 前一个知识点编号，**<element\_id>** String 必要

**返回值：**

返回的JSON字符串中包含 6个方面的值：

**weak\_elements：薄弱知识点列表**

**next\_element：下一个知识点编号**

**askedquestion\_nums：答题编号**

**state\_probs：知识点概率状态**

**error： 错误信息**

**参数demo:**

**{"kmap\_code":"eg\_zs\_h\_9","answer\_status":"1","element\_list":"[\"eg\_zs\_h\_9.1\",\"eg\_zs\_h\_9.2\",\"eg\_zs\_h\_9.3\",\"eg\_zs\_h\_9.4\",\"eg\_zs\_h\_9.5\",\"eg\_zs\_h\_9.6\",\"eg\_zs\_h\_9.7\",\"eg\_zs\_h\_9.8\",\"eg\_zs\_h\_9.9\",\"eg\_zs\_h\_9.10\",\"eg\_zs\_h\_9.11\",\"eg\_zs\_h\_9.12\",\"eg\_zs\_h\_9.13\",\"eg\_zs\_h\_9.14\",\"eg\_zs\_h\_9.15\",\"eg\_zs\_h\_9.16\",\"eg\_zs\_h\_9.17\",\"eg\_zs\_h\_9.18\",\"eg\_zs\_h\_9.19\",\"eg\_zs\_h\_9.20\"]","askedquestion\_num":"[-2,-1,-2,-1,-2,-2,-2,-1,-2,-3,-1,-1,-1,-2,-2,-1,-2,-1,-2,-2]","state\_probs":"[]","element\_id":"eg\_zs\_h\_9.10"}**

**返回值demo:**

**{"weak\_elements":"[\"eg\_zs\_h\_9.1\",\"eg\_zs\_h\_9.3\",\"eg\_zs\_h\_9.5\",\"eg\_zs\_h\_9.6\",\"eg\_zs\_h\_9.7\",\"eg\_zs\_h\_9.9\",\"eg\_zs\_h\_9.14\",\"eg\_zs\_h\_9.15\",\"eg\_zs\_h\_9.17\",\"eg\_zs\_h\_9.19\",\"eg\_zs\_h\_9.20\"]","next\_element":"-1","askedquestion\_nums":"[-2,-1,-2,-1,-2,-2,-2,-1,-2,-1,-1,-1,-1,-2,-2,-1,-2,-1,-2,-2]","state\_probs":"","error":""}**

## 确认阅读能力级别(分级阅读)

Name: KSTRead

API地址： http://[IP地址](#_环境说明(URL)):8080/AlgoService/algo/kstread

使用语法算法时，所需的参数包含六个方面

1. 知识图谱编号，<**kmap\_code**> String 必要
2. 学生编号，**<** **usr\_id>**int 必要
3. 前一个知识点编号，**<** **pre\_knode>** String 必要
4. 所属年级编号，[年级编号表](#_附2：年级编号表)**，<cur\_grade>** String 必要
5. 题目是否答对集合，**<usr\_ans>**String [] 必要 (1-对，0-错)
6. 题目类型集合, **<ques\_types>** String [] 必要 仅有三个值 (1-词汇,2-语法,3-阅读)
7. 题目难度集合, **<ques\_diffs>** String 必要
8. 做题时长, **<** **duration >** String 必要 套卷做题时长(单位为分钟)

注意：**usr\_ans，ques\_types，ques\_diffs 题目对应的索引顺序 及 数量必须保持一致**

**返回值：**

返回的JSON字符串中包含 6个值：

**kmap\_code：知识图谱编号**

**usr\_id：用户编号**

**pre\_ability：前一个level级别对应的能力值**

**knode\_toaskq：下一个阅读级别，当该值为-1时，结束测试，ownlevel则为学生最终的阅读能力级别**

**ownlevel：当前用户的阅读能力级别**

**error： 错误信息**

**参数demo:**

**{"ques\_types":"[\"1\",\"2\",\"3\",\"1\",\"1\",\"1\",\"3\",\"1\",\"3\",\"2\",\"1\",\"2\",\"2\",\"1\",\"1\",\"3\",\"1\",\"1\",\"1\",\"1\",\"1\",\"2\",\"2\",\"3\",\"2\",\"3\",\"1\",\"2\",\"3\",\"3\",\"3\",\"3\",\"2\",\"2\",\"2\",\"1\",\"1\",\"2\",\"1\",\"2\",\"1\",\"2\",\"2\",\"1\",\"1\"]","duration":"19","ques\_diffs":"[\"1\",\"1\",\"1\",\"1\",\"3\",\"2\",\"3\",\"3\",\"3\",\"3\",\"1\",\"1\",\"3\",\"1\",\"3\",\"2\",\"1\",\"3\",\"2\",\"2\",\"2\",\"2\",\"1\",\"3\",\"3\",\"1\",\"3\",\"1\",\"3\",\"3\",\"2\",\"1\",\"3\",\"3\",\"2\",\"1\",\"3\",\"1\",\"1\",\"3\",\"3\",\"2\",\"2\",\"3\",\"3\"]","usr\_ans":"[\"1\",\"1\",\"0\",\"1\",\"1\",\"1\",\"1\",\"1\",\"0\",\"0\",\"0\",\"0\",\"0\",\"0\",\"0\",\"0\",\"1\",\"0\",\"0\",\"0\",\"0\",\"1\",\"0\",\"0\",\"1\",\"0\",\"1\",\"0\",\"1\",\"1\",\"1\",\"0\",\"0\",\"0\",\"0\",\"1\",\"1\",\"1\",\"1\",\"1\",\"0\",\"1\",\"0\",\"1\",\"1\"]","kmap\_code":"read\_lvl\_en","usr\_id":"110112","cur\_grade":"6","pre\_knode":"read\_lvl\_en.6"}**

**返回值demo:**

**{"ownlevel":"read\_lvl\_en.1","kmap\_code":"read\_lvl\_en","usr\_id":"110112","error":"","pre\_ability":"0.6587999999999999","knode\_toaskq":"-1"}**

## DL测评新算法

Name: KstDL

API地址： http://[IP地址](#_环境说明(URL)):8080/AlgoService/algo/kstdl

使用语法算法时，所需的参数包含六个方面

1. 知识图谱编号，<**kmap\_code**> String 必要
2. 学生编号，**<** **user\_id>**String 必要
3. 前一个知识点编号，**<** **pre\_knode>** String 必要
4. 学生所选自评水平，< **init\_kstatus** > Integer (1,2,3,4,5) 必要
5. 题目是否答对集合，**<** **score >**Integer 必要 (1-对，0-错)
6. 题目难度集合, **<** **difficulty >** double[] 必要
7. 学生的能力估计对应的最大似然函数概率值，< **likelihood**> [99] 必要
8. 能力评估类型(1-assessment,2-practice) < **type** > double [] 必要
9. 知识点难度级别，< **level\_mode** > int 测评标准难度级别
10. 难度级别数，<total\_level> int (只有3或9两种值，默认为3) 可选 默认是3
11. 注意：difficulty 、score 的每个元素一一对应

**返回值：**

返回的JSON字符串中包含 6个方面的值：

1. **user\_id：用户编号**
2. **kmap\_code：知识图谱编号**
3. **knode\_toaskq：所得知识点**
4. **weak\_elems：薄弱知识点列表**
5. **ability** ：**pre\_knode对应的能力值能力值**
6. **likelihood：** **该学生目前的能力估计对应的最大似然函数概率值(99个)**
7. **abilityprob：对pre\_knode整体评估能力可能性值**
8. **error ： 错误信息**

**参数demo:**

**{"ques\_types":"[\"1\",\"2\",\"3\",\"1\",\"1\",\"1\",\"3\",\"1\",\"3\",\"2\",\"1\",\"2\",\"2\",\"1\",\"1\",\"3\",\"1\",\"1\",\"1\",\"1\",\"1\",\"2\",\"2\",\"3\",\"2\",\"3\",\"1\",\"2\",\"3\",\"3\",\"3\",\"3\",\"2\",\"2\",\"2\",\"1\",\"1\",\"2\",\"1\",\"2\",\"1\",\"2\",\"2\",\"1\",\"1\"]","duration":"19","ques\_diffs":"[\"1\",\"1\",\"1\",\"1\",\"3\",\"2\",\"3\",\"3\",\"3\",\"3\",\"1\",\"1\",\"3\",\"1\",\"3\",\"2\",\"1\",\"3\",\"2\",\"2\",\"2\",\"2\",\"1\",\"3\",\"3\",\"1\",\"3\",\"1\",\"3\",\"3\",\"2\",\"1\",\"3\",\"3\",\"2\",\"1\",\"3\",\"1\",\"1\",\"3\",\"3\",\"2\",\"2\",\"3\",\"3\"]","usr\_ans":"[\"1\",\"1\",\"0\",\"1\",\"1\",\"1\",\"1\",\"1\",\"0\",\"0\",\"0\",\"0\",\"0\",\"0\",\"0\",\"0\",\"1\",\"0\",\"0\",\"0\",\"0\",\"1\",\"0\",\"0\",\"1\",\"0\",\"1\",\"0\",\"1\",\"1\",\"1\",\"0\",\"0\",\"0\",\"0\",\"1\",\"1\",\"1\",\"1\",\"1\",\"0\",\"1\",\"0\",\"1\",\"1\"]","kmap\_code":"read\_lvl\_en","usr\_id":"110112","cur\_grade":"6","pre\_knode":"read\_lvl\_en.6"}**

**返回值demo:**

**{"ownlevel":"read\_lvl\_en.1","kmap\_code":"read\_lvl\_en","usr\_id":"110112","error":"","pre\_ability":"0.6587999999999999","knode\_toaskq":"-1"}**

# 附录：

## 附1：系统编号表

|  |  |
| --- | --- |
| 年级描述 | 值 |
| 数学春季正式产品 | 1 |
|  |  |

## 附2：年级编号表

|  |  |
| --- | --- |
| 年级描述 | 值 |
| 七年级上学期 | 1 |
| 七年级下学期 | 2 |
| 八年级上学期 | 3 |
| 八年级下学期 | 4 |
| 九年级上学期 | 5 |
| 九年级下学期 | 6 |