­­­

学堂在线开放接口

版本修订记

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 负责人 | 日期 | 功能 |
| v0.1 | liyusheng@xuetangx.com | 2017.11.16 | 观看课程、获取用户成绩 |

1. API接口功能

### 1.1 请求学堂在线视频章节播放界面接口描述

|  |  |
| --- | --- |
| 功能描述 | 跳转到学堂在线课程章节播放界面 |
| 参数说明 | appId: 学堂在线分配给各合作平台的产品ID，String类型（必填）  value：将重要的请求参数对称加密后的数据，String类型（必填） |
| 请求方法 | GET |
| 请求头 | Get  api/connect/kyhz?appId=5ef52de7bbbaa0180de8&value=aes(json(appId+ user\_id + nickname + head\_url + app\_secret +next+course\_id))  Host: 127.0.0.1:8080  Content-Type: text/html;charset=UTF-8  注意：  这里的json结构如下：  {  “appId”:” 5ef52de7bbbaa0180de8”,  “app\_secret”:” 0366c00c1b2c12fcabf91d345da61b62d9bba060”,  “nickname”:” 张三”,  “head\_url”:” http://pic4.nipic.com/20091217/3885730\_124701000519\_2.jpg”,  “next”:” course\_detail”,  “course\_id”:” course-v1:TsinghuaX+80512073X+2017\_T2”,  “user\_id”:” 153341”,  } |
| 请求体 |  |
| 响应 | 1. 如果用户第一次通过此api访问学堂在线，则相当于注册新用户并且登录，如果不是第一次，就会直接登录到学堂在线，通过用户的user\_id判断。  2. appId和对称加密的key从学堂在线运营人员获取并安全保存，避免泄露带来不必要的安全问题。 |

### 1.2 value参数组成

##### 1）参数组成

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 是否必须 | 参数类型 | 字段长度 | 参数描述 |
| appId | 必须 | String | 32个字符 | 由学堂在线运营分配给接入的企业或者学校，表示应用的id |
| user\_id | 必须 | String | 64个字符 | 唯一标识，合作方的用户名(同一个appID对应的用户名不能重复) |
| nick\_name | 非必须 | String | 64个字符，UTF-8编码 | 用户的昵称，用于展示(出现重复随机加后缀) |
| head\_url | 非必须 | String |  | 用户的头像url |
| app\_secret | 必须 | String | 64个字符 | 由学堂在线提供给企事业或学校的唯一标识 |
| course\_id | 必须 | String | 64个字符，UTF-8编码 | 学堂在线course\_id |

##### 2）加密方法

①先将所有的参数变成json形式。例如：

{

“appId”:” 5ef52de7bbbaa0180de8”,

“app\_secret”:” 0366c00c1b2c12fcabf91d345da61b62d9bba060”,

“nickname”:” 张三”,

“head\_url”:” http://pic4.nipic.com/20091217/3885730\_124701000519\_2.jpg”,

“next”:” course\_detail”,

“course\_id”:” course-v1:TsinghuaX+80512073X+2017\_T2”,

“user\_id”:” 153341”,

}

②调用对称加密方法进行加密，加密算法如下(实现如附件Java或php)：

String value= encrypt(json)

参数说明：json:如①所示的json串

调用示例：

http://www.xuetangx.com/api/connect/kyhz?appId=5ef52de7bbbaa0180de8&value=9c8b6ab2089ccfb0c45f80a751c243ffa022170a86a65353886c231ddf7fcac728d89f74cbbf3c75454e04fc1cf29000698993b9e3dafa2332ad1fe03f2ef38e1b00a7b3e2f548df2e53

### 2. 获取用户在学堂在线成绩与课程结课状态

|  |  |
| --- | --- |
| 功能描述 | 获取单个用户在学堂在线成绩与课程结课状态 |
| 参数说明 |  |
| 请求方法 | Get |
| 请求头 | Get  /api/web/ kyhz /exam?appId=5ef52de7bbbaa0180de8&value=aes(json(appId+ user\_id + app\_secret +course\_id))  Host: 127.0.0.1:8080  Content-Type: application/json |
| 请求体 |  |
| 响应 | 响应码：200 (HTTP的响应码统一为200，接口调用结果在响应体中体现)  响应体：  code：200—接口调用成功。其他参考响应码  一个完整的响应体示例如下：  {  “code”: 200,  result:{  "user\_id":"123",//合作方用户id  "video\_progress":”90%”,//用户观看课程视频总进度  " homework\_score ":”80%”,//用户作业分数  " total\_score ":”79”,//总分数  "class\_ending":”true”//用户是否结课  }  } |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段长度 | 字段描述 |
| user\_id | String | 32个字符 | 唯一标识，用户名(相同产品ID对应的用户名不能重复) |
| course\_id | String |  | 学堂在线课程id |
| app\_id | String |  | 由学堂在线运营分配给接入的企业或者学校，表示应用的id |
| app\_secret | String |  | 由学堂在线提供给企事业或学校的唯一标识 |

调用示例：

http://www.xuetangx.com/api/web/ kyhz /exam/?appId=5ef52de7bbbaa0180de8&value=9c8b6ab2089ccfb0c45f80a751c243ffa022170a86a65353886c231ddf7fcac728d89f74cbbf3c75454e04fc1cf29000698993b9e3dafa2332ad1fe03f2ef38e1b00a7b3e2f548df2e53

# 附录

## 附录：对称加密算法(php)

**<?php**$privateKey = **"1234567812345678"**;  
$iv = **"1234567812345678"**;  
$data = **'{“user\_id”:”123”,”pwd”:”456”}'**;  
  
*//加密*$encrypted = mcrypt\_encrypt(***MCRYPT\_RIJNDAEL\_128***, $privateKey, $data, ***MCRYPT\_MODE\_CBC***, $iv);  
*#echo($encrypted);  
#echo '<br/>';***echo**(base64\_encode($encrypted));  
**echo '<br/>'**;  
  
*//解密*$encryptedData = base64\_decode(**"q4+tQ2+FXBvyPPkrNfb6WfdzJBDZX2pIiEkHckgcXpjiPweCBuj5sIJWQbJsDery"**);  
$decrypted = mcrypt\_decrypt(***MCRYPT\_RIJNDAEL\_128***, $privateKey, $encryptedData, ***MCRYPT\_MODE\_CBC***, $iv);  
**echo**($decrypted);  
**?>**

## 附录：对称加密算法(java)

**import** javax.crypto.Cipher;

**import** javax.crypto.spec.IvParameterSpec;

**import** javax.crypto.spec.SecretKeySpec;

**public** **static** String encrypt(String json) **throws** Exception {

**if**(**null**==json || "".equals(json)){

**return** **null**;

}

**try** {

String key = "1234567812345678";

String iv = "1234567812345678";

Cipher cipher = Cipher.*getInstance*("AES/CBC/NoPadding");

**int** blockSize = cipher.getBlockSize();

**byte**[] dataBytes = json.getBytes();

**int** plaintextLength = dataBytes.length;

**if** (plaintextLength % blockSize != 0) {

plaintextLength = plaintextLength + (blockSize - (plaintextLength % blockSize));

}

**byte**[] plaintext = **new** **byte**[plaintextLength];

System.*arraycopy*(dataBytes, 0, plaintext, 0, dataBytes.length);

SecretKeySpec keyspec = **new** SecretKeySpec(key.getBytes(), "AES");

IvParameterSpec ivspec = **new** IvParameterSpec(iv.getBytes());

cipher.init(Cipher.***ENCRYPT\_MODE***, keyspec, ivspec);

**byte**[] encrypted = cipher.doFinal(plaintext);

**return** **new** sun.misc.BASE64Encoder().encode(encrypted);

} **catch** (Exception e) {

e.printStackTrace();

**return** **null**;

}

}

**public** **static** String desEncrypt(String cryptData) **throws** Exception {

**if**(**null**==cryptData || "".equals(cryptData)){

**return** **null**;

}

**try**

{

String key = "1234567812345678";

String iv = "1234567812345678";

**byte**[] encrypted1 = **new** sun.misc.BASE64Decoder().decodeBuffer(cryptData);

Cipher cipher = Cipher.*getInstance*("AES/CBC/NoPadding");

SecretKeySpec keyspec = **new** SecretKeySpec(key.getBytes(), "AES");

IvParameterSpec ivspec = **new** IvParameterSpec(iv.getBytes());

cipher.init(Cipher.***DECRYPT\_MODE***, keyspec, ivspec);

**byte**[] original = cipher.doFinal(encrypted1);

String originalString = **new** String(original);

**return** originalString;

}

**catch** (Exception e) {

e.printStackTrace();

**return** **null**;

}

}