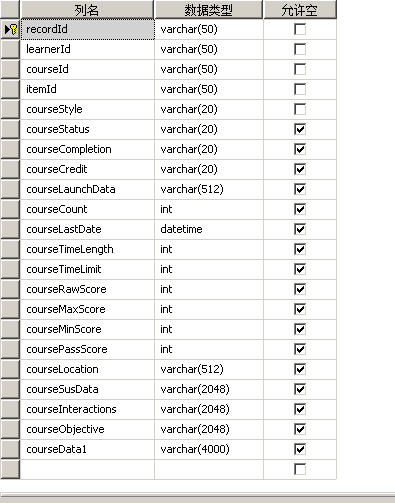
## SCORM2004 Player安装说明

2013-10-04

### 一、数据库

数据库字段的设置，参考如下：（下图来自MS SQL Server）



其中，每个字段的解释与第一次打开时，是否需要LMS进行相应初始化，如下表：

|  |  |  |
| --- | --- | --- |
| **字段** | **字段解释** | **是否初始化** |
| recordId | 每个SCO的学习记录ID，主键 | 第一次打开时自动初始化 |
| learnerId | 学员ID | 第一次打开时自动初始化 |
| courseId | 由一个或多个SCO组成的课件ID | 第一次打开时自动初始化，自由设定 |
| itemId | SCO的ID | 第一次打开时自动初始化，来自imsmanifest.xml文件中item节点里的identifier属性 |
| courseStyle | 课件的类型，如SCORM1.2、SCORM2004等 | 第一次打开时自动初始化 |
| courseStatus | SCO是否通过了考试 | SCO指定 |
| courseCompletion | SCO是否完成 | SCO指定 |
| courseCredit | SCO是否需要学分 | 第一次打开时自动初始化 |
| courseLaunchData | SCO启动时附带的数据 | 第一次打开时自动初始化，常为空 |
| courseCount | SCO学习次数 | 每次提交学习记录时累加一次 |
| courseLastDate | SCO最后一次学习时间 | 每次提交学习记录时LMS记录最后一次时刻 |
| courseTimeLength | SCO学习时长（秒数） | 每次提交学习记录时，将读取的时间累加过去时间 |
| courseTimeLimit | SCO最长学习时长（秒数） | 第一次打开时自动初始化，常为空 |
| courseRawScore | SCO的原始成绩 | 由课件自己设定 |
| courseMaxScore | SCO的最高分数 | 由课件自己设定 |
| courseMinScore | SCO的最低分数 | 由课件自己设定 |
| coursePassScore | SCO的及格成绩 | 由课件自己设定 |
| courseLocation | SCO的内部路径 | 由课件自己设定 |
| courseSusData | SCO的内部冗余字段 | 由课件自己设定 |
| courseInteractions | SCO的交互记录 | 由课件自己设定 |
| courseObjective | SCO的内部知识点 | 由课件自己设定 |
| courseData1 | SCO的冗余字段 | 由课件自己设定 |

### 二、安装步骤

1、在打开课件的课件列表html页面里，添加如下javascript代码：

//--SCORM2004 父窗口代码开始-------------

var adapterStr="";

var postStr="";

//预备未来提交地址自定义

var adapterURL="";

var divid=window.location.href.split("/");

adapterURL="http://"+divid[2]+"/";

//打开课件

function openCourse(thisURL){

//thisURL="url=courses/course1/imsmanifest.xml&cid=21321123&uid=test01";

//url为课件根目录下的imsmanifest.xml；cid为整个课件ID；uid为学员ID；

//如果没有url，不能够打开课件；如果没有cid或uid，则为浏览模式，非正常学习；

adapterStr="";

postStr="";

window.open("scorm2004Player/player.html?"+thisURL+"&rm="+Math.random());

}

//读取课件学习记录

function readCourseInfo(opStr){

if(opStr!=""){

adapterStr="";

//window.open(adapterURL+"adapterInfo.aspx?"+opStr);

$.ajax({

type: "post",

url: adapterURL+"adapterInfo.aspx",

cache: false,

data: opStr,

success: function(msg){

adapterStr=msg;

},

dataType: "text",

error : function() {

adapterStr="";

}

});

}

}

//写入课件学习记录

function writeCourseInfo(){

if(postStr!=""){

//window.open(adapterURL+"adapterInfo.aspx?"+postStr);

$.ajax({

type: "post",

url: adapterURL+"adapterInfo.aspx",

cache: false,

data: postStr,

success: function(msg){

postStr="";

//如果上传成功后，需要刷新页面，请执行reload，否则可以注释

window.location.reload();

},

dataType: "text",

error : function() {

//

}

});

}

}

//--SCORM2004 父窗口代码结束-----------

其中，课件使用openCourse函数打开，如：

<a href="javascript:openCourse('url=courses/course1/imsmanifest.xml&cid=21321123&uid=

test01')">SCORM 2004 course</a>

adapterInfo.aspx文件为LMS服务器端文件，用于读写学习记录到数据库中；可以根据需要修改其路径。

2、在LMS根目录下，建立js文件夹，拷贝jquery-1.7.1.min.js到该文件夹中。

推荐使用jquery-1.7.1.min.js，以保证对IE6、7的更好兼容。

3、将scorm2004Player文件夹拷贝到LMS的根目录下；

### 三、LMS服务器端文件的制作

由于解析XML文件的解释器，在各个浏览器中兼容性难以保证，故采用字符串的方式。通过在服务器端文件URL后添加参数，采用post访问相应数据。

前台和服务器端统一约定区分字符串的区隔符，有三个级别：分别为dot1="\_XvYvX\_"；dot2="\_XyXyX\_"； dot3="\_XyOyX\_"。

#### 1、读取课程各SCO状态

读取各SCO状态的格式，类似于：http：//xxxxx.xxx? op=l&uid=111&cid=222

其中，op=l，代表读取记录；uid代表当前用户id；cid代表该SCO所在课件的id。

服务器端文件以字符串形式打印如下格式到前台：

最后完成的小节id + dot1 + 小节id=completed + dot1+ 小节id= incomplete + dot1+ 小节id= incomplete + dot1+ 小节id=completed + dot1…

服务器判断最后完成的小节id，以最晚时刻SCO的id为最后学习的SCO。

如S100002\_XvYvX\_S100001=completed\_XvYvX\_S110001=incomplete\_XvYvX\_S110002=completed\_XvYvX\_S110003=completed\_XvYvX\_S100002=incomplete\_XvYvX\_S100003=completed\_XvYvX\_S100004=completed

#### 2、读取学习记录

读取学习记录的格式，类似于：http：//xxxxx.xxx? op=r&uid=111&cid=222&iid=333

其中，op=r，代表读取记录，op=w，代表写入记录；uid代表当前用户id；cid代表该SCO所在课件的id，iid代表SCO的id。

服务器端文件以字符串形式打印如下格式到前台：

doStr=用户名 ;

doStr+=dot1+是否通过考试 ;

doStr+=dot1+是否完成;

doStr+=dot1+是否Credit ;

doStr+=dot1+启动数据;

doStr+=dot1+学习次数;

doStr+=dot1+最后时刻;

doStr+=dot1+累积时长;

doStr+=dot1+时长限制;

doStr+=dot1+原始成绩;

doStr+=dot1+最高成绩;

doStr+=dot1+最低成绩;

doStr+=dot1+及格成绩;

doStr+=dot1+课件路径;

doStr+=dot1+Sus数据;

doStr+=dot1+ Obj数据;

doStr+=dot1+交互数据;

doStr+=dot1+冗余字段;

Response.Write(doStr);

注意：必须严格按照此格式返回数据；约定以dot1="\_XvYvX\_" 来区分字符串；

其字段说明如下：

|  |  |
| --- | --- |
| **字段** | **字符说明** |
| doStr=用户名 ; | 当前学员的姓名 |
| doStr+=dot1+是否通过考试 ; | 是否考试通过，Passed：通过；failed：未通过；unknown：未知； |
| doStr+=dot1+是否完成; | 是否学习完成，completed：完成；incomplete：未完成；not attempted：未开始；unknown：未知； |
| doStr+=dot1+是否Credit ; | 积分 |
| doStr+=dot1+启动数据; | 课件自定义 |
| doStr+=dot1+学习次数; | 学习次数，1代表1次，20代表20次 |
| doStr+=dot1+最后时刻; | 最后一次学习时刻，如2013-10-05 14:30:45 |
| doStr+=dot1+累积时长; | 累积学习的秒数，如1250，代表学习了1250秒 |
| doStr+=dot1+时长限制; | 学习超时后的动作。exit,message：退出，有提示；continue,message：继续学习，有提示；exit,no message：退出，无提示；continue,no message：继续学习，无提示 |
| doStr+=dot1+原始成绩; | 课件原始成绩，80代表80分 |
| doStr+=dot1+最高成绩; | 课件最高分，默认为100 |
| doStr+=dot1+最低成绩; | 课件最低分，默认为0 |
| doStr+=dot1+及格成绩; | 课件及格分数线 |
| doStr+=dot1+课件路径; | 课件自定义，默认为空 |
| doStr+=dot1+Sus数据; | 课件自定义，默认为空 |
| doStr+=dot1+ Obj数据; | 课件自定义，默认为空 |
| doStr+=dot1+交互数据; | 课件自定义，默认为空 |
| doStr+=dot1+冗余字段; | 课件自定义，默认为空 |

#### 3、写入学习记录

写入方式如同读取学习记录，post如下格式的数据到服务器端：

op=w&uid=111&cid=222&iid=333&args=学习记录

其中，学习记录字符串的格式如下：

\_tmpStr = 是否完成;

\_tmpStr = dot1 + 内部地址;

\_tmpStr = dot1 + 原始成绩;

\_tmpStr = dot1 + 最高分数;

\_tmpStr = dot1 + 最低分数;

\_tmpStr = dot1 + 当前SCO学习时长;

\_tmpStr = dot1 + 是否完成当前SCO;

\_tmpStr = dot1 + 内部冗余字段;

\_tmpStr = dot1 + 内部知识点数据;

\_tmpStr = dot1 + SCO交互数据;

\_tmpStr = dot1 + 其他冗余字段;

在写入学习记录时，需要累加学习时长，累加学习次数。此外，还需要记录最后学习的时刻。

其字段说明如下：

|  |  |
| --- | --- |
| **字段** | **字符说明** |
| \_tmpStr = 是否完成; | completed：完成；incomplete：未完成；not attempted：未开始；unknown：未知； |
| \_tmpStr = dot1 + 内部地址; | 课件自定义的字符串 |
| \_tmpStr = dot1 + 原始成绩; | 课件自定义，通常为百分制 |
| \_tmpStr = dot1 + 最高分数; | 课件自定义，默认为100 |
| \_tmpStr = dot1 + 最低分数; | 课件自定义，默认为0 |
| \_tmpStr = dot1 + 当前SCO学习时长; | 本次学习的秒数，单位为秒 |
| \_tmpStr = dot1 + 是否完成当前SCO; | Passed：通过；failed：未通过；unknown：未知； |
| \_tmpStr = dot1 + 内部冗余字段; | 课件自定义 |
| \_tmpStr = dot1 + 内部知识点数据; | 课件自定义 |
| \_tmpStr = dot1 + SCO交互数据; | 课件自定义 |
| \_tmpStr = dot1 + 其他冗余字段; | 课件自定义 |

### 四、调试

将Player.html文件的第55行，var debug=true 意为打印调试信息；若设置为var debug=false，意为不打印调试信息。