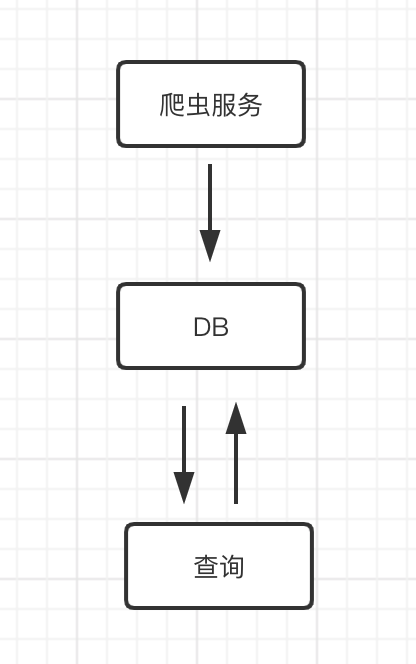
# 腾讯课堂数据抓取方案说明书

2020年4月11日

## 一、目的

抓取腾讯企鹅辅导官网的数据：<https://fudao.qq.com/>。统计每天的各个学科课程数量。可以查看历史数据。点击数量可以跳转页面查看每一门课的详情信息，包括课程id、课程标题、价格、老师等信息。

## 二、项目流程图



项目流程分三步：

1. 抓包分析官网接口，获取爬虫路径；
2. 定时去抓取官方网站数据，并入数据库；
3. 定义api接口，页面通过接口获取数据并展示。

## 三、数据抓取模块设计

通过官网分析可知，课程可以按年级划分，也可以按类型划分，对应关系如下，其中1代表有这门课，0代表没有：



抓包根据接口返回数据，可以获得两种课程信息，一类是基本课程，例如语文、数学、英语等，每门课有多个开课时间；另一类是特殊课程，例如讲座，这类课都上免费而且只有一个开课时间。接口如下：

a接口：

[https://fudao.qq.com/cgi-proxy/course/discover\_subject?client=4&platform=3&version=30&grade={grade}&subject={subject}&showid=0&page=1&size=10&t=0.24549794142780268](https://fudao.qq.com/cgi-proxy/course/discover_subject?client=4&platform=3&version=30&grade=%7bgrade%7d&subject=%7bsubject%7d&showid=0&page=1&size=10&t=0.24549794142780268)

b接口：

<https://fudao.qq.com/cgi-proxy/course/get_course_package_info?client=4&platform=3&version=30&subject_package_id=>{subject\_package\_id}&t=0.6655795271578293

（1）基本课程需要分两次访问，先访问a接口获取基本信息，再访问b接口获取详细信息;

（2）特殊课程只需要访问一次a接口。

## 四、数据库设计

数据可分成两个部分存储：

（1）基本课程信息 t\_sys\_course\_base\_info

CREATE TABLE `t\_sys\_course\_base\_info` (

`id` int(10) NOT NULL AUTO\_INCREMENT COMMENT 'id',

`grade` int(20) NOT NULL COMMENT '年级id',

`subject` int(20) NOT NULL COMMENT '科目id',

`subject\_package\_id` varchar(512) NOT NULL COMMENT '课程id',

`title` varchar(1024) NOT NULL COMMENT '课程标题',

`course\_bgtime` int(11) COMMENT '课程开始时间',

`course\_endtime` int(11) COMMENT '课程结束时间',

`sold\_count` int(10) NOT NULL DEFAULT 0 COMMENT '销量',

`course\_min\_price` int(10) NOT NULL DEFAULT 0 COMMENT '课程最低价',

`course\_max\_price` int(10) NOT NULL DEFAULT 0 COMMENT '课程最高价',

`created` date COMMENT '创建时间',

PRIMARY KEY (`id`),

UNIQUE KEY `idx\_pid` (`subject\_package\_id`,`created`) USING BTREE,

KEY `idx\_sid` (`subject`)

) ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8 COMMENT='课程基本信息';

（2）基本课程详细表t\_sys\_course\_detail\_info

CREATE TABLE `t\_sys\_course\_detail\_info` (

`id` int(10) NOT NULL AUTO\_INCREMENT COMMENT 'id',

`subject\_package\_id` varchar(1024) NOT NULL COMMENT '课程id',

`cid` int(20) NOT NULL COMMENT '子课程id',

`name` varchar(100) NOT NULL COMMENT '子课程标题',

`time\_plan` varchar(100) COMMENT '子课程计划',

`tu\_name` varchar(100) NOT NULL COMMENT '授课老师名',

`te\_name` varchar(100) NOT NULL COMMENT '辅导老师名',

`created` date COMMENT '创建时间',

PRIMARY KEY (`id`),

UNIQUE KEY `idx\_cid` (`cid`,`created`) USING BTREE,

KEY `idx\_pid` (`subject\_package\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8 COMMENT='课程详细信息';

（3）特殊课程信息表

CREATE TABLE `t\_spe\_course\_info` (

`id` int(10) NOT NULL AUTO\_INCREMENT COMMENT 'id',

`grade` int(20) NOT NULL COMMENT '年级id',

`subject` int(20) NOT NULL COMMENT '科目id',

`cid` int(20) NOT NULL COMMENT '子课程id',

`name` varchar(100) NOT NULL COMMENT '子课程标题',

`time\_plan` varchar(100) COMMENT '子课程计划',

`te\_name` varchar(100) NOT NULL COMMENT '授课老师名',

`created` date COMMENT '创建时间',

PRIMARY KEY (`id`),

UNIQUE KEY `idx\_cid` (`cid`,`created`) USING BTREE

) ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8 COMMENT='特殊课程详细信息';

## 五、web接口设计

#### 5.1获取课程基本信息

**简要描述：**

* 按时间和课程类型划分的课程列表

**请求URL：**

* http://xxxxx.com/txcourse/courselist

**请求方式：**

* GET

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| start\_time | date | 否 | 默认当天，格式YYYY-MM-DD |
| end\_time | date | 否 | 默认当天，格式YYYY-MM-DD |

**返回参数说明：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 示例 | 说明 |
| errno | int |  |  |
| error | string |  |  |
| data\_list | array |  | 课程json数组，见下方 |
| data\_list[0].subject | int | 6001 | 课程id |
| data\_list[0].num | int |  | 数量 |
| data\_list[0].created | date | 6001 | 时间 |

#### 5.2获取课程详细信息，

**简要描述：**

* 按课程集体开课时间、价格、老师等信息

**请求URL：**

* http://xxxxx.com/txcourse/coursedetail

**请求方式：**

* POST

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| start\_time | date | 是 | 默认当天，格式YYYY-MM-DD |
| subject | int | 是 | 课程类id |

**返回参数说明：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 示例 | 说明 |
| errno | int |  |  |
| error | string |  |  |
| data\_list | array |  | 课程json数组，见下方 |
| data\_list[0].grade | int | 6001 | 年级 |
| data\_list[0].name | string |  | 课程名 |
| data\_list[0].cid | date |  | 课程id |
| data\_list[0].time\_plan | string |  | 课程规划时间 |
| data\_list[0].te\_name | string |  | 授课老师 |
| data\_list[0].tu\_name | string |  | 辅导老师 |