

厦门大学图书馆馆藏书目信息 API 开发实例及其应用

肖铮 陈晓亮

(厦门大学图书馆, 厦门 361005)

文摘: Web2.0 环境中 OpenAPI 成为了越来越多网站开放数据服务的选择。本文以厦门大学图书馆馆藏书目信息 API 的开发为实例, 探索了图书馆开放数据服务的方式, 并介绍了以此 API 为基础的豆瓣网 Mashup 应用, 说明了其的实用价值。

关键词: Web2.0; Library2.0; API; Mashup; OPAC

0. 引言

Web2.0, 从最初的概念, 到今天众多蓬勃发展的 Web2.0 网站, 在理念和技术上都为业界带来了许多革命性的变化。Web2.0 理念中的开放精神推动了众多网站数据和服务的开放, 而开放带给这些网站的好处则是更多的应用和更旺的人气。从 Google 到 Amazon, 再到图书馆界的大佬 OCLC 无不通过 OpenAPI 的形式向外开放其数据和服务, 而基于这些开放接口的第三方应用则时刻围绕着用户需求, 如雨后春笋般大量涌出, 改变着用户的使用习惯。

图书馆的服务近年来也深受 Web2.0 之风影响, 并逐步形成了图书馆 2.0 理念和众多相关服务。在 Talis 关于图书馆 2.0 的白皮书中^[1]提到, 图书馆 2.0 的四项原则之一就是图书馆无处不在, 图书馆将相关内容复制到用户在任何时候需要的任何地方。而要实现这一原则, 图书馆必须首先拥有一个积极的开放心态, 再要有一种可行的开放方式。

图书馆 2.0 理念^[2]为图书馆开放数据服务提供了理论基础, 而众多 Web 公司的 OpenAPI 服务则为之提供了技术实现的可能。

1. 开发背景

1.1 需求分析

图书馆拥有大量的数据资源, 虽然馆藏书目信息只是其中的一小部分, 却与图书馆服务紧密相关。在没有 OPAC 的时代, 读者需要从书目柜中查找图书卡片, 再从卡片中读取索书号, 并最终在具体架位寻找要借的图书, 如果图书已被借走, 则读者只能在一番忙碌之后空手而归。而在如今普遍使用 OPAC 的时代, 读者则首先通过 OPAC 系统查找图书, 然后确定典藏情况、是否有可借复本、图书索书号、馆藏地等信息。换言之, 读者到图书馆借书, 首先接触到的就是馆藏信息。可见, 获取馆藏信息是读者实实在在的需求, 也是图书馆开放数据的一个很好的切入点。

2.2 功能分析

馆藏书目信息 API 接收来自第三方的请求, 并发送响应给第三方。根据以上的需求分析, 可以对请求和响应作出明确的定义。在请求部分, 以图书 ISBN 或者图书题名为参数。在响应部分, 返回是否典藏、图书题名、图书索书号、馆藏地、图书状态。具体功能包括:

(1) 根据图书 ISBN 号, 返回该 ISBN 号图书在图书馆中的馆藏情况, 包括: 是否典藏、图书索书号、馆藏地、图书状态。

(2) 若在 ISBN 匹配中, 未在馆藏记录数据库中找到该图书, 则根据图书题名查找本馆是否有同名图书, 如果存在同名图书, 则返回该图书题名。

2.3 技术可行性分析

厦门大学图书馆采用汇文 Libsys 系统, 其底层数据库为 Oracle, 其 OPAC 系统使用 PHP 开发, 通过 Apache 提供 Web 服务。因此, 在技术上可选择:

(1) PHP: 作为 API 接口实现的编程语言。PHP 作为目前流行的 Web 编程语言, 拥有丰富的函数库,

可快速实现各种 Web 应用。

(2) REST: Representational State Transfer 是一种轻量级的接口服务规范。REST 是基于 HTTP、URL、HTML、XML 这些广泛使用的协议和标准。使用 URI 表示资源,对资源的操作通过 HTTP 协议提供的 GET、POST、PUT、DELETE 方法,以 HTML、XML 作为资源的传递格式^[3]。

(3) JSON: JavaScript Object Notation 是一种轻量级的数据交换格式。易于人阅读和编写。同时也易于机器解析和生成。JSON 采用完全独立于语言的文本格式,使 JSON 成为理想的数据交换语言^[4]。使用 PHP 的 json_encode 函数¹,可以方便地将 PHP 数组转化为 JSON 数据。

3. 开发步骤

3.1 系统流程图

以 ISBN 或者图书题名作为输入参数,经过对底层数据库查询,最终输出馆藏信息。系统流程图如图 1 所示:



图 1 馆藏书目信息 API 系统流程图

3.2 核心代码片段

(1) 接收参数

```
$isbn=$_REQUEST["isbn"];
```

```
$title=$_REQUEST["title"];
```

(2) 通过 ISBN 对数据库记录进行查找

```
$strSQL="select marc_rec_no from marc_idx where marc_idx_cont = '$isbn' and marc_idx_grp='05' ";
```

```
$stmt = $dbh->prepare($strSQL);
```

```
$stmt->execute();
```

```
$mode = OCI_FETCHSTATEMENT_BY_ROW;
```

```
$result = $stmt->fetchall($mode);
```

```
$num=count($result);
```

(2) 通过题名对数据库进行查找

```
$strSQL = "select MARC_REC_NO from MARC_IDX where MARC_IDX_CONT='$title' and MARC_DATA_CODE='200a'";
```

```
$stmt = $dbh->prepare($strSQL);
```

```
$stmt ->execute();
```

```
$mode=OCI_FETCHSTATEMENT_BY_ROW;
```

¹ 只有 PHP5 以上版本支持该函数

```
$aryItem=$stmt->fetchall($mode);
```

```
$num=count($aryItem);
```

(3) 生成结果数组, 并转化为 JSON 对象。因存在复本的原因, 图书馆藏地和图书状态数据需要使用数组。

```
$strMarcNo=$result[0]['MARC_REC_NO'];
```

```
$strCallNoOne = $aryItem[0]['CALL_NO'];
```

```
for ($i=0; $i<count($aryItem); $i++)
```

```
{
```

```
    $aryBook_Loc_NAME[] = $aryItem[$i]['DIS_LOCATION'];
```

```
    $aryBook_Loc_DEPT[] = $aryItem[$i]['LOCATION_DEPT'];
```

```
    $aryBook_stat_code[] = $aryItem[$i]['BOOK_STAT'];
```

```
}
```

```
$json_data=array('marc_no'=>$strMarcNo,'call_no'=>$strCallNoOne,'Loc_NAME'=>$aryBook_Loc_NAME,
```

```
'loc_DEPT'=>$aryBook_Loc_DEPT,'book_stat'=>$aryBook_stat_code);
```

3.3 API 的部署与发布

将程序代码命名为 `getBookInfo.php`, 上传至 OPAC Web 服务器根目录下的 API 目录中, 就完成了部署。

3.4 API 使用说明

此 API 程序支持 `isbn` 和 `title` 两个参数, 分别代表图书 ISBN 号和图书题名。可同时以 ISBN 和图书题名两个参数作为查询条件, 也可以任选其一作为查询条件。使用时, 通过 HTTP 协议向服务器发送请求, 请求经服务器处理, 最终返回 JSON 数据。以 ISBN 为 9787115168917 的图书为例:

(1) 向服务器端发送请求: `http://服务器地址/api/getBookInfo.php?isbn=9787115168917`

(2) 服务器响应结果为:

```
{
```

```
"marc_no":658173,
```

```
"call_no":"TP393.09/161.5",
```

```
"Loc_NAME":["嘉庚馆藏一(漳州校区)","东部分馆(曾厝垵)","理工与人文借阅区"],
```

```
"loc_DEPT":["漳州","海韵","本部"],
```

```
"book_stat":["可借","可借","借出-应还日期: 2009-09-24"]
```

```
}
```

4. 基于馆藏书目信息 API 的豆瓣网 Mashup 应用实例

Web2.0 的开放性促使 Mashup 应用的出现^[5]。Mashup 应用是指通过调用不同来源的数据或服务, 利用程序对请求的数据或服务进行重新组合, 而形成新服务的一种应用。Mashup 应用的基础则是 Web 上开放的 API 服务, API 为 Mashup 应用开发者提供了一条获取相关资源的捷径。

豆瓣网是目前国内最为成功的 Web2.0 网站之一, 其以“口味”聚合朋友的特殊社交模式, 基于大众分类标签的内容组织体系, 通过个人历史记录提供个性化推荐的特有功能, 以及无所不有的小组都吸引着越来越多的用户。

豆瓣网为用户提供了一个了解图书内容和大众评价的平台, 用户依据豆瓣上的图书简介, 图书评分和大众评论可以快速判断某本图书是否适合自己, 为自己选择阅读内容提供快速参考, 而这些都是图书馆 OPAC 系统所欠缺的。正是因为如此, 一部分读者在借书之前都到豆瓣去看看所借图书简介, 其他读者评价如何。另一部分读者, 直接以豆瓣网作为个人阅读管理中心, 在豆瓣上查找发现自己想要阅读的图书。

对于第一类读者, 我们已经在 OPAC 页面上, 通过对豆瓣 API 的 Mashup 应用, 将豆瓣提供的图书内容简介、分类标签融合到 OPAC 中, 并且提供了到该图书豆瓣页面的链接, 使读者可以直接在 OPAC 中对图书进行搜索和筛选。

对于第二类读者, 图书馆可以主动融入读者的网络环境, 在读者熟悉的网站中融入图书馆服务, 使用户能感受到“无所不在”的图书馆服务。

4.1 豆瓣网集成馆藏信息的实现基础

通过在客户端 Mashup，将馆藏信息置于豆瓣页面。在浏览器上使用 Ajax 技术，通过 JavaScript 脚本调用馆藏书目信息 API，并改写豆瓣 HTML 页面。对于客户端有以下要求^[6]：

(1) Firefox 浏览器。支持多种互联网新技术标准的浏览器，尤其是其所具有的高度可扩展性，已被越来越多的用户所青睐。

(2) GreaseMonkey 扩展。Firefox 浏览器的一个特殊扩展，支持浏览器运行客户端 js 代码，而使用户能够扩展自己对于某些网页或内容的处理功能。

(3) js 脚本文件。调用馆藏信息 API 并修改豆瓣页面的用户脚本。

(4) jQuery^[7]。轻量级的 JavaScript 库，实现了 DOM 操控、CSS 处理与导航、事件处理、良好用户界面效果和 Ajax 功能。使用 jQuery 可编写短小精悍的用户脚本。

4.2 系统架构和工作流程

客户端 Mashup 应用工作流程图如图 2 所示：

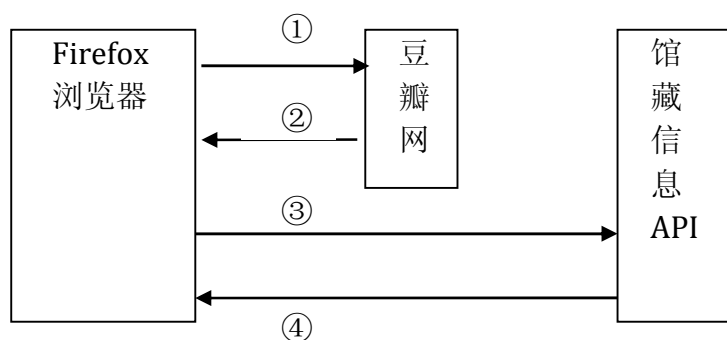


图 2 客户端 Mashup 应用工作流程图

(1) 用户浏览豆瓣网。

(2) Firefox 浏览器中的 GreaseMonkey 扩展触发用户 js 脚本。程序分析豆瓣网页面，获取图书 ISBN、图书题名。

(3) 向馆藏信息 API 发送查询请求。

(4) Firefox 浏览器接收 API 响应，处理页面。

4.3 用户 js 脚本代码片段

(1) 分析豆瓣页面，获取图书 ISBN 和图书题名

```
var title = $('h1').text();
$("#info .obmo .pl").each(function(i)
{
    if ($(this).text() == 'ISBN:'){
        var isbn = $(this)[0].nextSibling.nodeValue;
        isbn = isbn.substr(1,13);
    }
})
```

(2) 向馆藏信息 API 发送请求，接收响应 JSON 数据

```
$.getJSON("http://210.34.4.43/api/getLOC.php?callback=?", {isbn:isbn,title:title},function(json){ ... })
```

(3) 处理 JSON 数据，生成插入豆瓣页面的 HTML 代码

```
for (i=0;i<json.Loc_NAME.length;i++){
    htmlStr += "<li style='font-size:12px'>"+json.call_no+"&nbsp;&nbsp;&nbsp;"+json.Loc_NAME[i];
    if (json.book_stat[i] == "可借")
    {
        htmlStr += "&nbsp;&nbsp;&nbsp;<font color='#006600'>"+json.book_stat[i]+"</font></li>";
    }
}
```

```

else
    htmlStr += "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<font color='red'>"+json.book_stat[i]+"</font></li>";
}

```

(4) 向豆瓣页面插入新生成 HTML 代码

```
$(".aside div:eq(0)").after(htmlStr);
```

4.4 实现效果

用户在浏览器上看到的最终效果如图 3 所示，在豆瓣页面的右侧边栏中，新加入了一栏“在哪借这本书？”-- 该栏显示该书在厦门大学图书馆的馆藏情况。



图 3 豆瓣网集成馆藏信息的实际页面效果

5 总结与展望

通过图书馆馆藏书目数据 API 的开发和应用，一方面深度挖掘了图书馆现有的数据资源并释放出其潜在价值，另一方面用户的个性化需求得到了满足，实现了图书馆和用户的双赢局面。API-Mashup 的应用架构，为图书馆提供了一个根据用户需求，释放内在资源，增加数据价值，融入用户环境的良好服务模式。图书馆可以根据自身特点和用户需求逐步提供不同的 API，并在此基础上形成应用前景更为广泛的开放平台。

但需要注意的是，在 API 的建设中，需要面临对来自数据安全、系统性能、运营管理三个方面的挑战^[8]。这也是笔者在未来的实践和应用中要逐步解决的问题，使得 API 应用日趋完善。

参考文献

- [1] Card K, Miller P. Do libraires matter? The rise of Library 2.0[EB/OL]. [2009-8-25]. http://www.talis.com/applications/downloads/white_papers/DoLibrariesMatter.pdf
- [2] 范并思, 胡小菁. 图书馆 2.0: 构建新的图书馆服务[J]. 大学图书馆学报. 2006 (01): 2-7.
- [3] Wikipedia. Representational State Transfer[EB/OL]. [2009-8-25]. http://en.wikipedia.org/wiki/Representational_State_Transfer
- [4] JSON 中国. 介绍 JSON[EB/OL]. [2009-8-25]. <http://www.json.org/json-zh.html>
- [5] 刘炜, 葛秋妍. Web2.0 技术图书馆应用分析. [2009-8-25]. <http://www.libnet.sh.cn/sztsg/fulltext/reports/2006/libraryTech20.pdf>
- [6] 图书馆吱声. 豆瓣网的图书馆馆藏书目信息扩展脚本[EB/OL]. [2009-8-25]. http://blog.xmulib.org/cgi-bin/mt/mt-search.cgi?tag=OPAC&blog_id=4&IncludeBlogs=4
- [7] jQuery[EB/OL]. [2009-8-25]. <http://jquery.com/>
- [8] 陈定权, 莫秀娟. OpenAPI 及其在数字图书馆中的应用[J]. 图书馆论坛. 2009, 29 (1) 69-71.

作者简介

肖 铮 厦门大学图书馆技术部，工程师，联系电话 0592-2184973，电子邮件 zhx@xmilib.org

陈晓亮 厦门大学图书馆特藏部，馆员，联系电话 0592-2184818，电子邮件 soggchen@gmail.com