Douban网络项目JSON解析过程

千锋iOS 欧阳坚

通过本课程，你将掌握

JSON协议解析	
JSON	
URL	

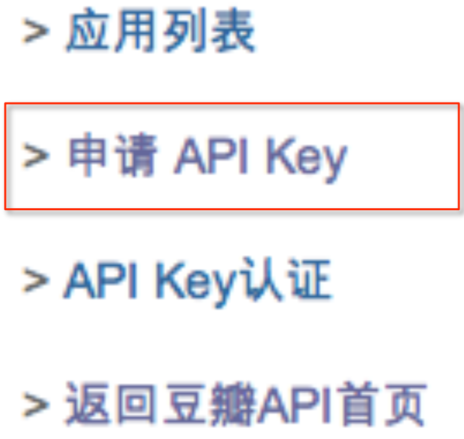
0 准备工作

申请douban API Key

豆瓣API申请

- http://www.douban.com/service/apikey/
- 申请的apikey
- c4579586f41a90372f762cb65c78be5d

- 点击网站右边的 申请API Key



> 应用列表

> 申请 API Key

> API Key认证

> 返回豆瓣API首页

在浏览器中输入一下网址

http://api.douban.com/book/subjects?q=maozedong&start-index=1&max-results=10&apikey=04f1ae6738f2fc450ed50b35aad8f4cf&alt=json

注意：上述是一行，请把apikey后面的数字换成自己的认证码

q=maozedong 表示要查询的字符串为maozedong，这里使用英语，汉字后面介绍

start-index=1 表示返回服务器数据库第一项

max-results=10 表示返回做多10项记录

alt=json 表示使用json数据格式返回，如果不写就返回xml

URL格式补充知识

http://api.douban.com/book/subjects?q=maozedong&start-index=1&max-results=10&apikey=04f1ae6738f2fc450ed50b35aad8f4cf&alt=json

上述是一个典型的URL格式，使用了HTTP GET方式来获取数据。参数是通过 & 隔开，第一个参数q=maozedong之前必须用?隔开

http://api.douban.com 表示主机

/book/subjects 表示在服务器上的一个程序或者一个链接

在线JSON格式化工具

- http://www.bejson.com/go.html?u=http://www.bejson.com/jsonview2

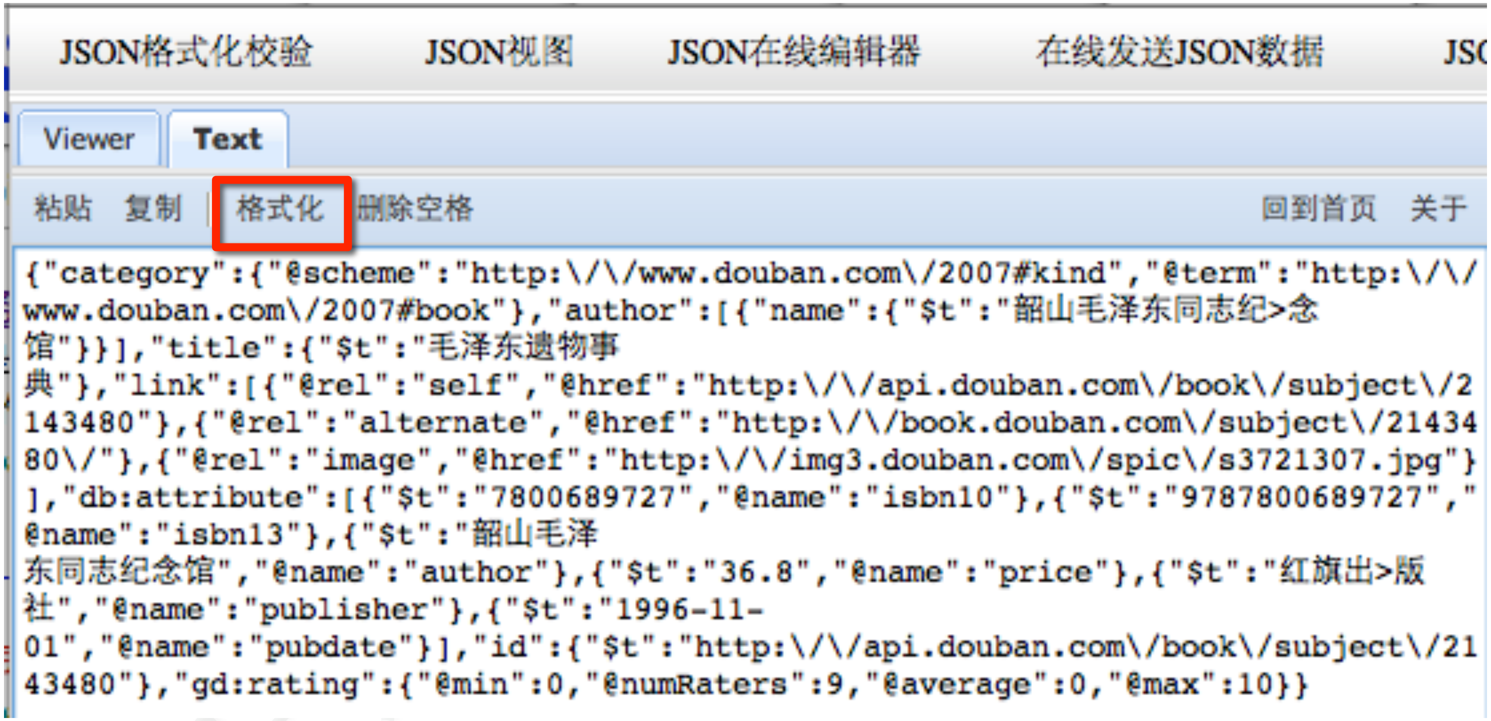
- 可以把JSON文本贴在上述网址上，然后格式化出可读的格式即可

- 步骤参考下面

JSON例子 可读性差

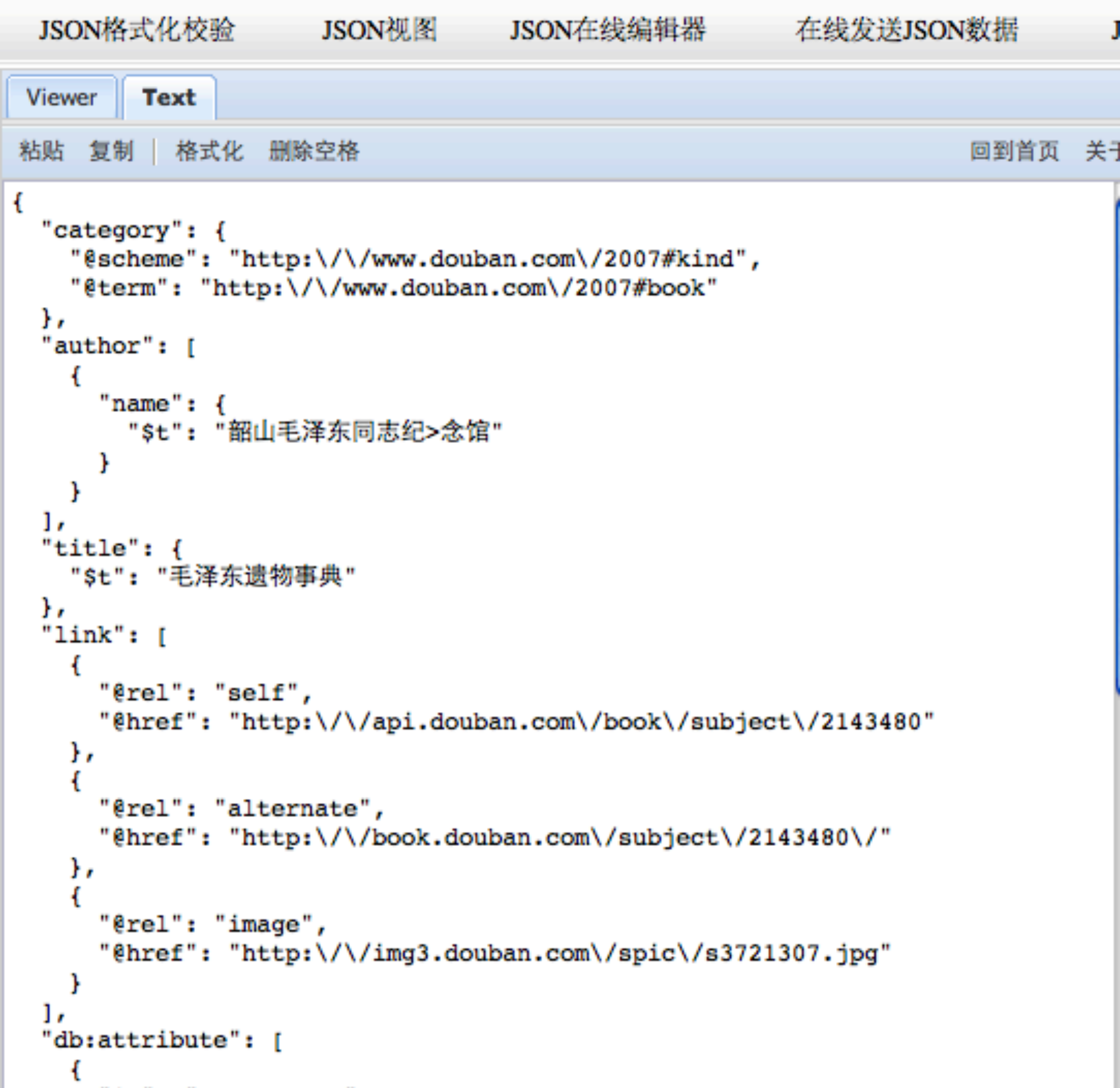
{"category":{"@scheme":"http:\/\/www.douban.com\/2007#kind","@term":"http:\/\/www.douban.com\/2007#book"},"author":[{"name":{"$t":"韶山毛泽东同志纪>念馆"}}],"title":{"$t":"毛泽东遗物事典"},"link":[{"@rel":"self","@href":"http:\/\/api.douban.com\/book\/subject\/2143480"},{"@rel":"alternate","@href":"http:\/\/book.douban.com\/subject\/2143480\/"},{"@rel":"image","@href":"http:\/\/img3.douban.com\/spic\/s3721307.jpg"}],"db:attribute":[{"$t":"7800689727","@name":"isbn10"},{"$t":"9787800689727","@name":"isbn13"},{"$t":"韶山毛泽东同志纪念馆","@name":"author"},{"$t":"36.8","@name":"price"},{"$t":"红旗出>版社","@name":"publisher"},{"$t":"1996-11-01","@name":"pubdate"}],"id":{"$t":"http:\/\/api.douban.com\/book\/subject\/2143480"},"gd:rating":{"@min":0,"@numRaters":9,"@average":0,"@max":10}}

格式化 步骤1



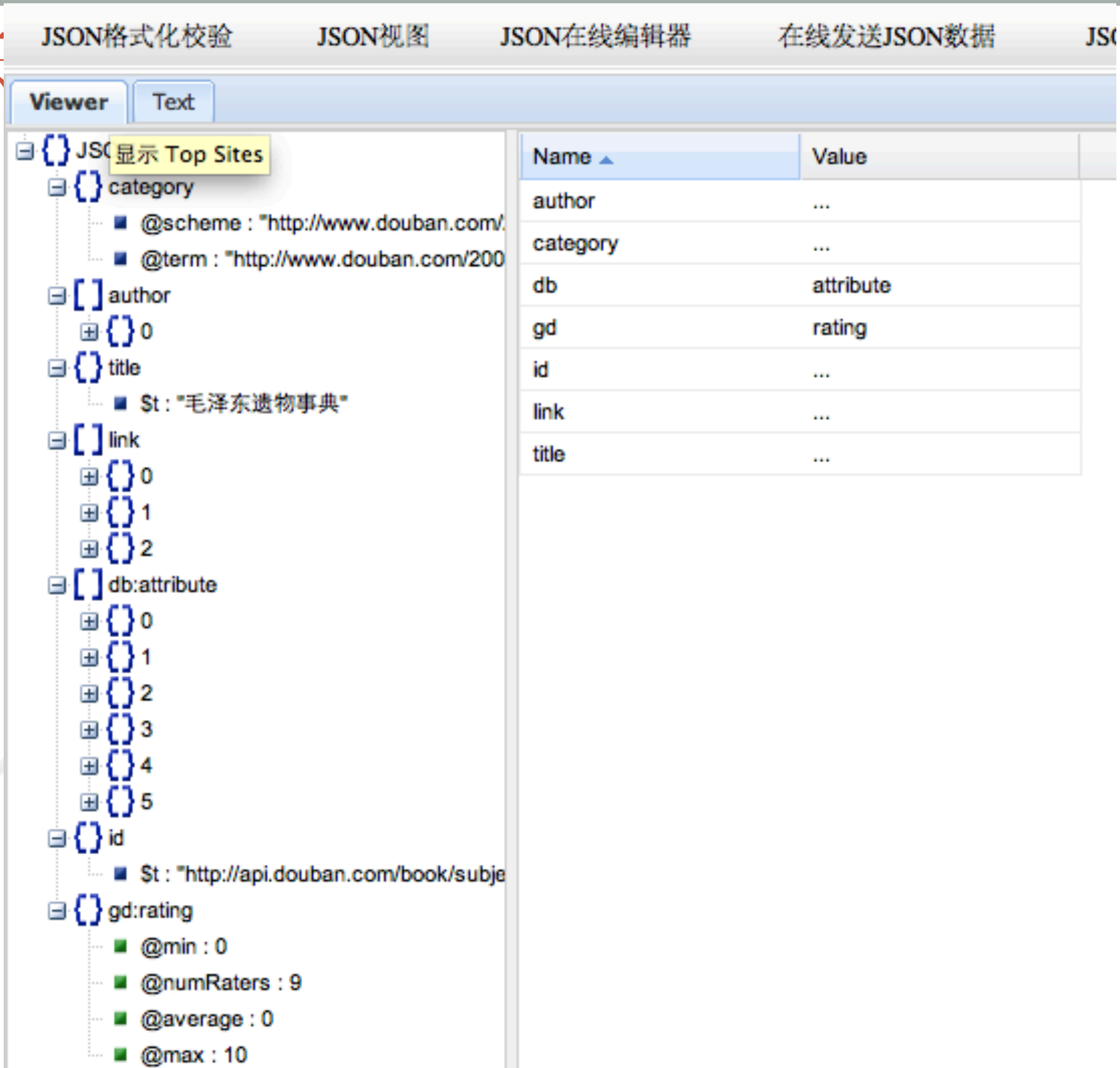
```
{"category":{"@scheme":"http:\/\/www.douban.com\/2007#kind","@term":"http:\/\/
www.douban.com\/2007#book"},"author":[{"name":{"$t":"韶山毛泽东同志纪>念
馆"}}],"title":{"$t":"毛泽东遗物事
典"},"link":[{"@rel":"self","@href":"http:\/\/api.douban.com\/book\/subject\/2
143480"},{"@rel":"alternate","@href":"http:\/\/book.douban.com\/subject\/21434
80\/"},{"@rel":"image","@href":"http:\/\/img3.douban.com\/spic\/s3721307.jpg"}
],"db:attribute":[{"$t":"7800689727","@name":"isbn10"},{"$t":"9787800689727","
@name":"isbn13"},{"$t":"韶山毛泽
东同志纪念馆","@name":"author"},{"$t":"36.8","@name":"price"},{"$t":"红旗出>版
社","@name":"publisher"},{"$t":"1996-11-
01","@name":"pubdate"}],"id":{"$t":"http:\/\/api.douban.com\/book\/subject\/21
43480"},"gd:rating":{"@min":0,"@numRaters":9,"@average":0,"@max":10}}
```

显示结果



```
{
  "category": {
    "@scheme": "http:\/\/www.douban.com\/2007#kind",
    "@term": "http:\/\/www.douban.com\/2007#book"
  },
  "author": [
    {
      "name": {
        "$t": "韶山毛泽东同志纪>念馆"
      }
    }
  ],
  "title": {
    "$t": "毛泽东遗物事典"
  },
  "link": [
    {
      "@rel": "self",
      "@href": "http:\/\/api.douban.com\/book\/subject\/2143480"
    },
    {
      "@rel": "alternate",
      "@href": "http:\/\/book.douban.com\/subject\/2143480\/"
    },
    {
      "@rel": "image",
      "@href": "http:\/\/img3.douban.com\/spic\/s3721307.jpg"
    }
  ],
  "db:attribute": [
    {
```

层次显示



```
⊟ { } JSO
  ⊟ { } category
      ■ @scheme : "http://www.douban.com/
      ■ @term : "http://www.douban.com/200
  ⊟ [ ] author
    ⊞ { } 0
  ⊟ { } title
      ■ $t : "毛泽东遗物事典"
  ⊟ [ ] link
    ⊞ { } 0
    ⊞ { } 1
    ⊞ { } 2
  ⊟ [ ] db:attribute
    ⊞ { } 0
    ⊞ { } 1
    ⊞ { } 2
    ⊞ { } 3
    ⊞ { } 4
    ⊞ { } 5
  ⊟ { } id
      ■ $t : "http://api.douban.com/book/subje
  ⊟ { } gd:rating
      ■ @min : 0
      ■ @numRaters : 9
      ■ @average : 0
      ■ @max : 10
```

Name ▲	Value
author	...
category	...
db	attribute
gd	rating
id	...
link	...
title	...

1 创建工程

创建一个iOS工程

创建一个RootViewController

创建DoubanBook工程

- 创建Empty Application，命名为DoubanBook
- 然后创建一个RootViewController类(基UIViewController，确保选中with XIB for user interface)

AppDelegate.m 增加RootViewController

增加头文件包含

```
#import "RootViewController.h"
```

添加调用代码

```
- (BOOL)application:(UIApplication *)application
            didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
…
   RootViewController *rvc = [[[RootViewController alloc]  initWithNibName:
           @"RootViewController" bundle:nil] autorelease];
   UINavigationController *nav = [[[UINavigationController alloc]
            initWithRootViewController:rvc] autorelease];
   self.window.rootViewController = nav;
   [self.window makeKeyAndVisible];
   return YES;
}
```

2 同步下载

直接下载，等待下载

RootViewController.h 定义

```
@interface RootViewController : UIViewController
{
    int startIndex;
    int maxResults;
}

@end
```

这里startIndex从什么位置开始下载

maxResults最多下载记录条数

RootViewController.m 定义NSURL

```
- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view from its nib.
    NSString *searchWord = @"maozedong";
    startIndex = 1;
    maxResults = 10;

    NSString *urlString = [NSString stringWithFormat:@"http://api.douban.com/book/
subjects?q=%@&start-index=%d&max-results=
%d&apikey=04f1ae6738f2fc450ed50b35aad8f4cf&alt=json",
                              searchWord, startIndex, maxResults];
    NSLog(@"url string is %@", urlString);
    NSURL *url = [NSURL URLWithString:urlString];

    [self downloadSync:url];
}
```

RootViewController.m 定义downloadSync:函数

```
- (void) downloadSync:(NSURL *)url {
    /* 1. 把字符串转化为NSURL对象  */
    NSError *error = nil;
    NSString *jsonStr =
    [NSString stringWithContentsOfURL:url
            encoding:NSUTF8StringEncoding
            error:&error];
    /* 2. 同步下载url里面的内容存放在jsonStr中  */
    NSLog(@"json:%@", jsonStr);
}
```

运行一下

- 在命令行中可以看出输出的JSON结果

3 异步下载

直接下载，等待下载

RootViewController.h 定义网络

```
#import <UIKit/UIKit.h>
@interface RootViewController : UIViewController
 <NSURLConnectionDataDelegate>
{
    int startIndex;
    int maxResults;
    NSMutableData *responseData;
    /* 用来存放从服务器传过来的一切数据 */
    NSURLConnection *myConnection;
    /* 存放和服务器连接的对象 */
}
```

RootViewController.m 析构函数

```
- (void) dealloc
{
    [responseData release];
    [myConnection release];

    [super dealloc];
}
```

RootViewController.m 定义异步下载函数

```
- (void) downloadAsync:(NSURL *)url {
    /* 0. 创建responseData数据区 ∗/
    /* 1. 构造网络请求 30s超时时间 ∗/
    responseData = [[NSMutableData alloc] initWithLength:0];
    NSURLRequest *myRequest =
    [NSURLRequest requestWithURL:url
                cachePolicy:NSURLRequestReloadIgnoringLocalCacheData
                timeoutInterval:30];    /* 2. 建立网络连接 ∗/
    myConnection = [[NSURLConnection alloc] initWithRequest:
              myRequest delegate:self];
    /* 以请求myRequest来建立一个网络连接，并且使用代理self来
        接收该连接传过来的数据 ∗/
    /* 3. 让网络指示器显示进度条 ∗/
    [UIApplication sharedApplication].networkActivityIndicatorVisible = YES;
}
```

RootViewController.m 接受响应头

```
/* 1. 网络开始接收数据  */
/* 网络响应头接收完成，真正数据开始马上要来到了  */-
(void) connection:(NSURLConnection *)connection
        didReceiveResponse:(NSURLResponse *)response
{
   NSLog(@"begin to receive real data");
   [responseData setLength:0];
}
```

RootViewController.m 接受数据

```
/* 2. 网络正在接收数据  */
/* data参数就是系统每次接收一大段数据就会调用该方法  */
- (void) connection:(NSURLConnection *)connection
            didReceiveData:(NSData *)data{
    [responseData appendData:data];
    /* 把网络传给我们的数据data附加到
           responseData之后即可  */
}
```

RootViewController.m 数据接收完成

```
/* 2. 网络数据接收完成 */
- (void) connectionDidFinishLoading:(NSURLConnection *)connection
{
    NSLog(@"receive finish");
    NSString *str = [[NSString alloc] initWithData:responseData
              encoding:NSUTF8StringEncoding];
    NSLog(@"receive string is %@", str);
    [str release];

    [UIApplication sharedApplication].networkActivityIndicatorVisible
                        = NO;
}
```

RootViewController.m 调用异步函数

```
- (void)viewDidLoad{
    [super viewDidLoad];
    NSString *searchWord = @"maozedong";
    startIndex = 1;
    maxResults = 10;
    NSString *urlString = [NSString stringWithFormat:@"http://api.douban.com/boo
k/subjects?q=%@&start-index=%d&max-results=
%d&apikey=04f1ae6738f2fc450ed50b35aad
8f4cf&alt=json",
                          searchWord, startIndex, maxResults];
    NSLog(@"url string is %@", urlString);
    NSURL *url = [NSURL URLWithString:urlString];

    //[self downloadSync:url];
    [self downloadAsync:url];
}
```

运行一下

- 可以看出在connectionDidFinishLoading:中也打印了网络输出结果

思考NSData和NSString之间关系

- NSData和NSString之间如何转换

加入汉字支持

- 汉字在UTF-8中占用3个字节，在网络编码中，使用了%然后加上字节16进制的标准进行

NSString+URLEncoding.h 创建文件

```
#import <Foundation/Foundation.h>

@interface NSString(URLEncoding)
- (NSString *)urlEncodeString;
@end
```

NSString+URLEncoding.m 实现

```
#import "NSString+URLEncoding.h"

@implementation NSString(URLEncoding)
- (NSString *)urlEncodeString {
    NSString *result = (NSString *)
            CFURLCreateStringByAddingPercentEscapes(NULL,
              (CFStringRef)self,
              NULL,
              (CFStringRef)@";/?:@&=$+{}<>,",
              kCFStringEncodingUTF8);
    return [result autorelease];
}
@end
```

RootViewController.m 修改

```
#import "NSString+URLEncoding.h"
…
- (void)viewDidLoad
{
    [super viewDidLoad];
     NSString *searchWord = @"毛泽东";
    searchWord = [searchWord urlEncodeString];
    startIndex = 1;
    maxResults = 20;
…
}
```

运行一下

- 可以看出可以搜素出汉字结果了

额外技术Unicode转码

```
NSString  ∗name = @"北京";
char converted[1000];
memset(converted, '\0', 1000);
[name getCString:converted maxLength:(1000)
           encoding:NSUnicodeStringEncoding];
int len = strlen(converted);
for (int i =0; i < len/2; i++) {
     NSLog(@"%04X", (unsigned short)((short *)converted)[i]);
}
```

4 JSON解析

JSON解析

JSON介绍(SBJson)

- https://github.com/stig/json-framework
- SBJson (aka json-framework)
- JSON (JavaScript Object Notation) is a light-weight data interchange format that's easy to read and write for humans and computers alike. This library implements strict JSON parsing and generation in Objective-C.

将SBJson目录导入到工程中

- JSON是著名开源框架，老版的名字叫JSON，新版3.x改名了SBJson，增加了iOS5.x支持，这里使用新版本的代码

创建BookModel类

- 创建BookModel类，该类用来表示书的所有对象属性

BookModel.h 定义书的字段

```
#import <Foundation/Foundation.h>

@interface BookModel : NSObject
{
    NSString *_title;
    NSString *_author;
}

@property (nonatomic, retain) NSString *title;
@property (nonatomic, retain) NSString *author;

@end
```

BookModel.m 实现

```
#import "BookModel.h"

@implementation BookModel

@synthesize title = _title;
@synthesize author = _author;

- (void) dealloc
{
    self.title = nil;
    [_author release], _author = nil;
    [super dealloc];
}
@end
```

RootViewController.h 增加数组

```
#import <UIKit/UIKit.h>

@interface RootViewController : UIViewController
    <NSURLConnectionDataDelegate>
{
    int startIndex;
    int maxResults;
    NSMutableData *responseData;
    /* 用来存放从服务器传过来的一切数据 */
    NSURLConnection *myConnection;
    /* 存放和服务器连接的对象 */
    NSMutableArray *bookArray;
}
```

RootViewController.m

```
#import "RootViewController.h"
#import "NSString+URLEncoding.h”
#import "BookModel.h”
#import "SBJson.h”
…
- (void) dealloc
{
    [bookArray release];
    [responseData release];
    [myConnection release];

    [super dealloc];
}
```

RootViewController.m 增加数组创建

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view from its
nib.
    bookArray = [[NSMutableArray alloc] init];
    NSString  *searchWord = @"毛泽东";
    searchWord = [searchWord urlEncodeString];
    startIndex = 1;
    maxResults = 10;

…
}
```

RootViewController.m 实现JSON解析

```
- (void) parseCommon:(NSDictionary *)jsonParser
{
    NSDictionary *t1 = [jsonParser
              objectForKey:@"opensearch:totalResults"];
    NSString *t2 = [t1 objectForKey:@"$t"];
    NSLog(@"$t is %@", t2);
    startIndex = [[[jsonParser objectForKey:@"opensearch:startIndex"]
              objectForKey:@"$t"] intValue];
    int itemsPerPage = [[[jsonParser objectForKey:
             @"opensearch:itemsPerPage"] objectForKey:@"$t"] intValue];
    NSLog(@"startIndex %d itemsPerPage %d",
          startIndex, itemsPerPage);
```

RootViewController.m 实现JSON解析(续)

```
    NSArray *books = [jsonParser objectForKey:@"entry"];
    for (NSDictionary *perBook in books) {
        NSString *name = [[perBook objectForKey:@"title"] objectForKey:@"$t"];
        NSString *author = [[[[perBook objectForKey:@"author"]
               objectAtIndex:0] objectForKey:@"name"] objectForKey:@"$t"];
        NSLog(@"book name %@ author %@", name, author);
        BookModel *oneBook = [[BookModel alloc] init];
        oneBook.title = name;
        oneBook.author = author;
        [bookArray addObject:oneBook];
        [oneBook release];
        startIndex++;
    }
    [myTableView reloadData];
}
```

RootViewController.m 2中解析

```
- (void) parseStringJson:(NSString *)str
{
    /* 使用json解析 */
    NSDictionary *jsonParser = [str JSONValue];
    [self parseCommon:jsonParser];
}

- (void) parseDataJson:(NSData *)data
{
    /* 使用json解析 */
    NSDictionary *jsonParser = [data JSONValue];
    [self parseCommon:jsonParser];
}
```

RootViewController.m 调用解析函数

```
/* 2. 网络数据接收完成 */
- (void) connectionDidFinishLoading:(NSURLConnection *)connection
{
#define USE_STRING_PARSE 0
#if USE_STRING_PARSE
    NSString *str = [[NSString alloc] initWithData:responseData
encoding:NSUTF8StringEncoding];
    [self parseStringJson:str];
    [str release];
#else
    [self parseDataJson:responseData];
#endif

    [UIApplication sharedApplication].networkActivityIndicatorVisible = NO;
}
```

运行一下

- 可以看出在bookArray数组中填充了图书数据

5 UITableView展示

把JSON解析的结果刷新到界面上

RootViewController.h 增加UITableView

```
@interface RootViewController : UIViewController
    <NSURLConnectionDataDelegate, UITableViewDelegate,
UITableViewDataSource>
{
    int startIndex;
    int maxResults;
    NSMutableData *responseData;
    /* 用来存放从服务器传过来的一切数据 */
    NSURLConnection *myConnection;
    /* 存放和服务器连接的对象 */

    NSMutableArray *bookArray;

    UITableView *myTableView;
}
```

RootViewController.m 创建Table

```
- (void)viewDidLoad
{    [super viewDidLoad];
    // Do any additional setup after loading the view from its nib.
    bookArray = [[NSMutableArray alloc] init];

    myTableView = [[UITableView alloc] initWithFrame:self.view.bounds 
style:UITableViewStylePlain];
    myTableView.autoresizingMask = UIViewAutoresizingFlexibleWidth | 
UIViewAutoresizingFlexibleHeight;
    myTableView.delegate = self;
    myTableView.dataSource = self;
    [self.view addSubview:myTableView];
…
}
```

RootViewController.m

```
- (NSInteger) tableView:(UITableView *)tableView
            numberOfRowsInSection:(NSInteger)section
{
    return [bookArray count];
}
```

RootViewController.m

```
- (UITableViewCell *) tableView:(UITableView *)tableView
               cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    NSString *CellID = @"Cell";
    UITableViewCell *cell = [tableView
dequeueReusableCellWithIdentifier:CellID];
    if (cell == nil) {
        cell = [[[UITableViewCell alloc] initWithStyle:
              UITableViewCellStyleDefault reuseIdentifier:CellID] autorelease];
    }
    BookModel *book = [bookArray objectAtIndex:indexPath.row];
    cell.textLabel.text = book.title;
    return cell;
}
```

RootViewController.m 刷新表

```
- (void) parseCommon:(NSDictionary *)jsonParser
{    /* 使用json解析 */
…
    NSLog(@"array is %@", bookArray);
    [myTableView reloadData];
}
```

运行一下

- 可以看出

5 作业

全部解析JSON，显示详细的内容

6 ASIHttpRequest框架

ASIHttpRequest请求

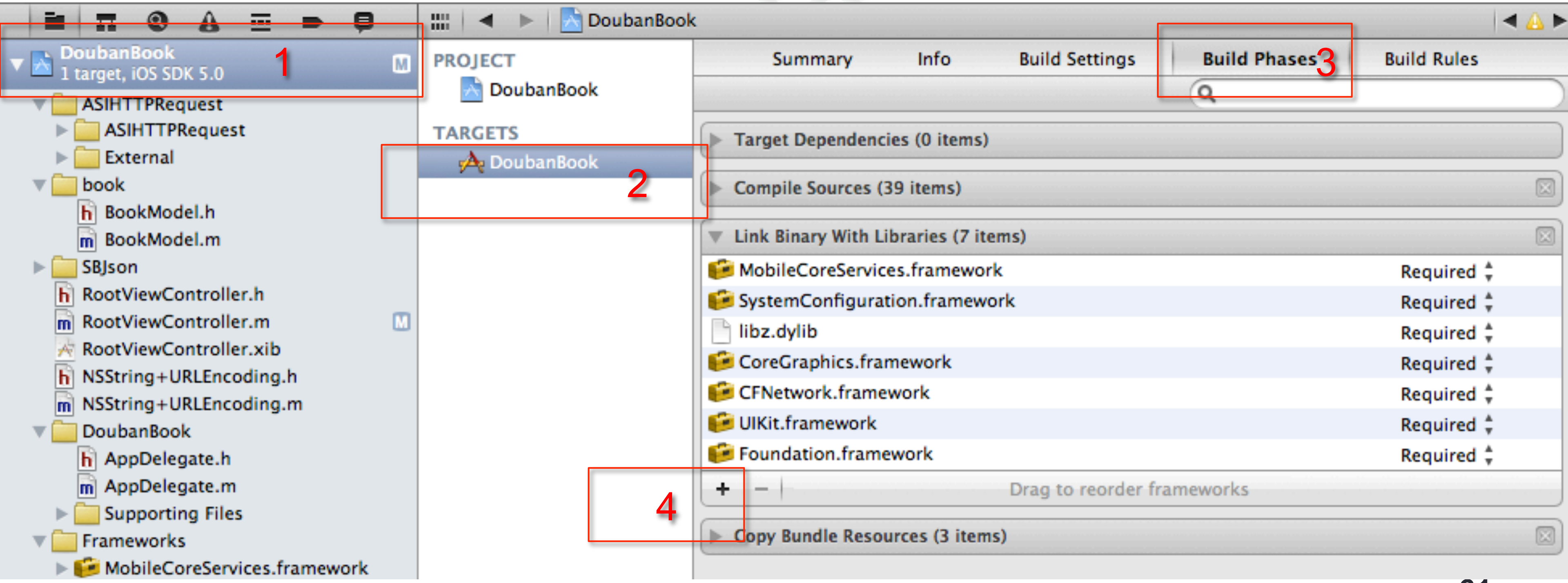
准备工作

- 把ASIHTTPRequest文件夹导入到项目中
- 增加Frameworks

- CFNetwork.framework
- CoreGraphics.framework
- libz.dylib
- SystemConfiguration.framework
- MobileCoreServices.framework

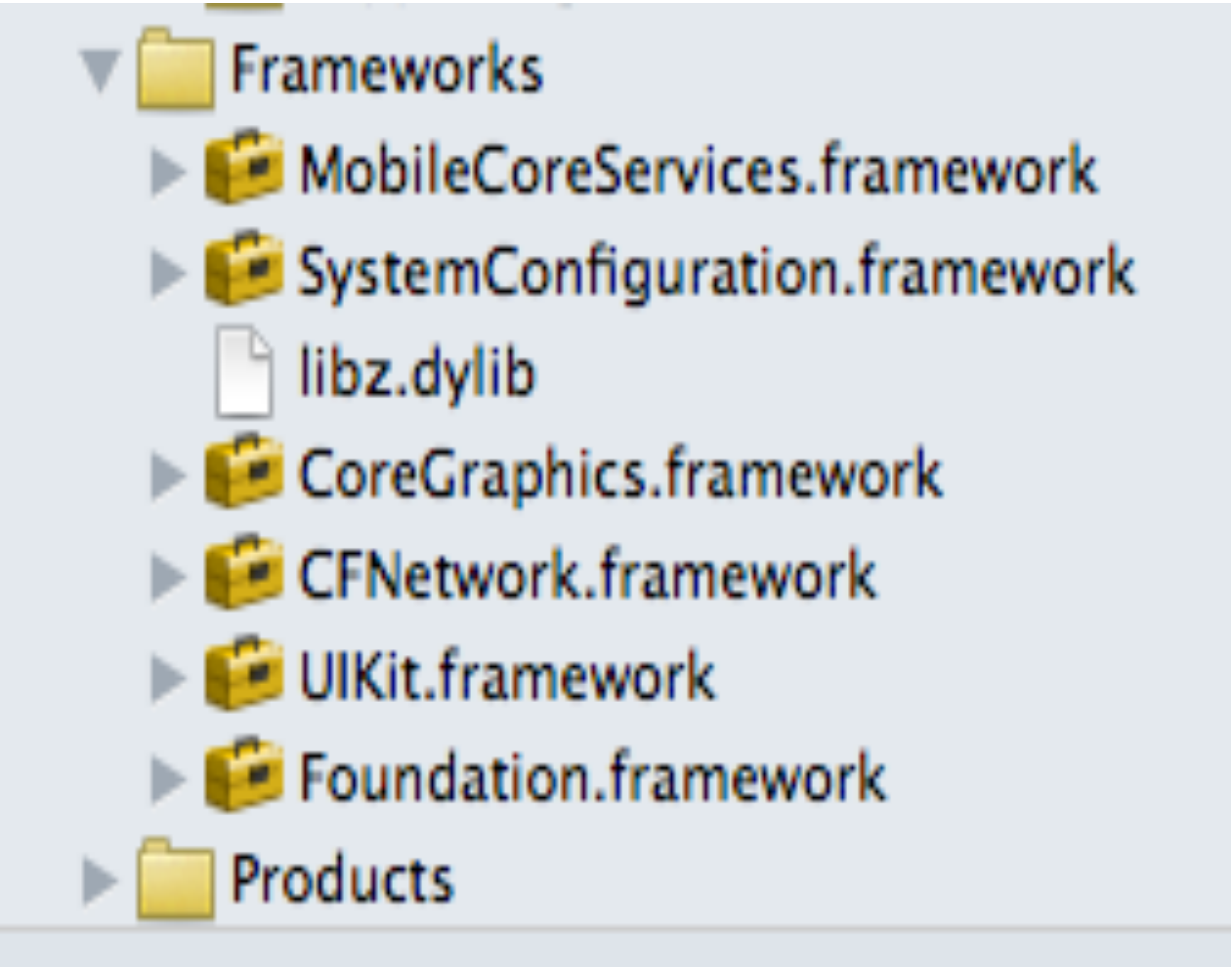
如何增加库

- 按照顺序点击增加库
- 按照1， 2， 3， 4顺序



添加库后效果图

- 添加入右图所有的库


Frameworks
MobileCoreServices.framework
SystemConfiguration.framework
libz.dylib
CoreGraphics.framework
CFNetwork.framework
UIKit.framework
Foundation.framework
Products

ASIHttpRequest同步异步

ASIHTTPRequest同步下载

ASIHttpRequest同步下载字符串

```
- (void) downloadASIHTTPSync:(NSURL *)url
{
    ASIHTTPRequest *request =
            [ASIHTTPRequest requestWithURL:url];
    [request startSynchronous];
    NSString *response = [request responseString];
    NSLog(@"download sync %@", response);
}
```

ASIHttpRequest同步下载二进制

```
- (void) downloadASIHTTPSync:(NSURL *)url
{
    ASIHTTPRequest *request =
            [ASIHTTPRequest requestWithURL:url];
    [request startSynchronous];
    NSData *responseData = [request responseData];
    [responseData  writeToFile:@"/testmusic.mp3”
                    atomically:YES]
}
```

ASIHTTPRequest异步下载

异步下载音乐

```
- (void) downloadMusic {
    NSString *musicURL = @"http://zhangmenshiting2.baidu.com/data2/music/13870770/13870770.mp3?xcode=69f706b1a5bcf5e531505161a949b71e&mid=0.41260424528396";
    NSURL *url = [NSURL URLWithString:musicURL];
    ASIHTTPRequest *request = [ASIHTTPRequest requestWithURL:url];
    [request setTag:100];
    [request setDelegate:self];
    [request startAsynchronous];
}
```

异步下载文本

```
- (void) downloadASIHTTPASync:(NSURL *)url
{
    ASIHTTPRequest *request = [ASIHTTPRequest
requestWithURL:url];
    [request setTag:101];
    [request setDelegate:self];
    [request startAsynchronous];
}
```

下载结束函数

```
- (void)requestFinished:(ASIHTTPRequest *)request
{
    if (request.tag == 101) {
        NSString *responseString = [request responseString];
        NSLog(@"response string is %@", responseString);
    } else if (request.tag == 100) {
        NSData *responseData = [request responseData];
        [responseData writeToFile:@"/testmusic.mp3"
atomically:YES];
    }
}
```

Douban网络项目 XML解析过程

千锋iOS 欧阳坚

添加头文件和库

- 选中工程
- 然后选择Target或者工程
- 然后选择 Build Settings
- 然后确保选择All
- 然后翻到Search Paths
- 然后选择Header Search Paths
- 然后输入 /usr/include/libxml2

- <?xml version="1.0" encoding="UTF-8"?>

- <feed xmlns="http://www.w3.org/2005/Atom" xmlns:db="http://www.douban.com/xmlns/" xmlns:gd="http://schemas.google.com/g/2005" xmlns:openSearch="http://a9.com/-/spec/opensearchrss/1.0/" xmlns:opensearch="http://a9.com/-/spec/opensearchrss/1.0/">

</feed>

<title>搜索 maozedong 的结果</title>

<opensearch:startIndex>1</opensearch:startIndex>

<opensearch:totalResults>3</opensearch:totalResults>

- <catogory> </catogory>
- <catogory/>

- http://www.douban.com/service/apikey/

课后作业1

实验心得

- 我叫_艾锋_，今天我学习了
  1. ________________
  2. ________________
  3. ________________
- 我觉得最难的部分是________，因为_______。
- 我觉得掌握的最熟练的是____。
- 我相信自己一定能够努力学好iPhone编程！

课后作业2

- 整理今天学到的新函数，记录到一个文本文件中

课后作业3

- 尽可能重复整个实验
- 次数不限

放飞梦想 大有希望！

千锋寄语