

Contents

1. Introduction to API.....	2
1.1. Sign-up for an API Key.....	2
1.2. Forming a Request	8
2. Using Java to do data scraping.....	9
2.1. The ApiExample	9
2.2. Coding a java file	13
2.2.1. Replacing the apiKey.....	14
2.2.2. Change the xml page you want to retrieve.....	15
2.2.3. Change the data you want to retrieve	15
2.2.4. Print out the data you want.....	17
2.3. Compile and execute your java file.....	17
3. Load data into MySQL database	18

1. Introduction to API

Web API is typically defined as a set of Hypertext Transfer Protocol (HTTP) request messages, along with a definition of the structure of response messages, which is usually in an Extensible Markup Language (XML) or JavaScript Object Notation (JSON) format. The Ohloh (Openhub) API is a free, REST-based programming interface to the Ohloh (Openhub) open source directory. You can use the Ohloh API to create your own applications and web services based on Ohloh data.

You can get the detail documents of ohloh API from

https://github.com/blackducksw/ohloh_api#ohloh-api-documentation

1.1. Sign-up for an API Key

Before you can access the Ohloh (Openhub) API you must register your application and obtain an API key. Bandwidth will initially be limited to 1,000 requests per API key per day.

<https://www.openhub.net/>

click "Join Now"

Welcome to Open Hub!

You can use Open Hub without logging in, or having an account. But with an account, you can do so much more:

- Claim all of your commits to projects across all of Open Hub, building a complete picture of your contributions to FOSS.
- Manage projects.
- Create project "stacks" that detail your usage of FOSS, and rate and review the projects you know.

[Login now](#) and start building your FOSS profile — it's free!

New Account

Login (required)

Your login is your public name on Open Hub and will be associated with all of your contributions.

Email Address (required)

You must provide a valid email to activate your account. You can opt-in to additional mail from Open Hub on the "Notifications" settings page.

If you have a [gravatar](#) for this email address, Open Hub will use it.

Verify Email Address (required)

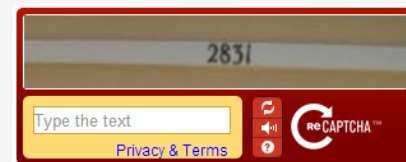
Password (required)

Password must be at least 5 characters long

Verify Password (required)

Verify Password must be at least 5 characters long

Captcha (required)



2831

Type the text

Privacy & Terms

reCAPTCHA

[Sign up](#)

Activate your account from your email account!

Click [Settings](#) in the webpage



yanjiaqi

Settings



Settings

Account Basics

ABC Login ID, Full Name, Email, Bio, Homepage, and Location for the account.

Password

Change password for the account.

API Keys

API Key settings and deletion, request new API keys.

Privacy

Opt-in / opt-out for emails from Open Hub, and OAuth access settings for external applications.

Account Summary

Settings

My Stacks

Widgets

Contributions

Contributions

Languages

Recognition

Kudos

Usage

Website Edits

Posts

Reviews



Click [API Keys](#)



yanjiaqi

Settings



Settings : API Keys

Request New API Key

About API Keys

- API keys are associated with your account, and a particular application or usage. Please refer to the [Open Hub Terms of Use and API Agreement](#) for more information
- On this page, you can request new API keys, or you can edit or remove existing API keys. You can also see your usage statistics for the API keys you have been granted.
- For more information on the Open Hub API, please see the [documentation](#).

Account Summary

Settings

My Stacks

Widgets

Contributions

Contributions

Languages

Recognition

Kudos

Usage

Website Edits

Posts

Reviews

About

Forums

Terms



Copyright © 2014 Black Duck Software, Inc. All Rights Reserved. and its contributors, Some Rights Reserved. Unless otherwise marked, this work is licensed under a Creative Commons Attribution 3.0 Unported License . Open Hub and the Open Hub logo are trademarks of Black Duck All Rights Reserved. in the United States and/or other jurisdictions. All other trademarks are the property of



© 2014 Black Duck Software, Inc.

Click Request New API Keys

Settings : API Keys : New

Application Name (required)
networktutorial

Home Page URL

An optional URL to a web page with information about this application.

Description (required)
for the tutorial

Enter a brief introductory paragraph about your application

Daily Request Limit: 1000
The default daily request limit is 1000.

I agree to use API keys associated with my account according to the Open Hub site [Terms of use](#) and [API Use Agreement](#)

Save Changes

After signing up, you can get your api key from your account information as follows:



yanjiaqi

Settings



Welcome to the Open Hub API program! Your new API key is shown below.

Settings : API Keys

networktutorial Created less than a minute ago, Hasn't been used yet
 for the tutorial

API Key	FRF42Zrmmfdq1ZnHM8XoMIA	Active ✓	can be used
OAuth Secret	t9HeT0waTCejikAdyQoHmBxECC9D8rsDwpbHDK94i0	0	requests all-time
		0	requests today
		limited to 1,000	requests a day

[Edit API Key](#)

[Request New API Key](#)

About API Keys

- API keys are associated with your account, and a particular application or usage. Please refer to the Open Hub Terms of Use and API Agreement for more information
- On this page, you can request new API keys, or you can edit or remove existing API keys. You can also see your usage statistics for the API keys you have been granted.
- For more information on the Open Hub API, please see the documentation.

[Account Summary](#)
Settings

[Contributions](#)
Contributions

[Recognition](#)
Kudos

[Usage](#)
Website Edits

1.2. Forming a Request

The Ohloh API returns XML-formatted data in response to HTTP GET requests.

Each web page on Ohloh, there may be an equivalent XML-formatted version of the page.

You must do three things to receive an XML-formatted response:

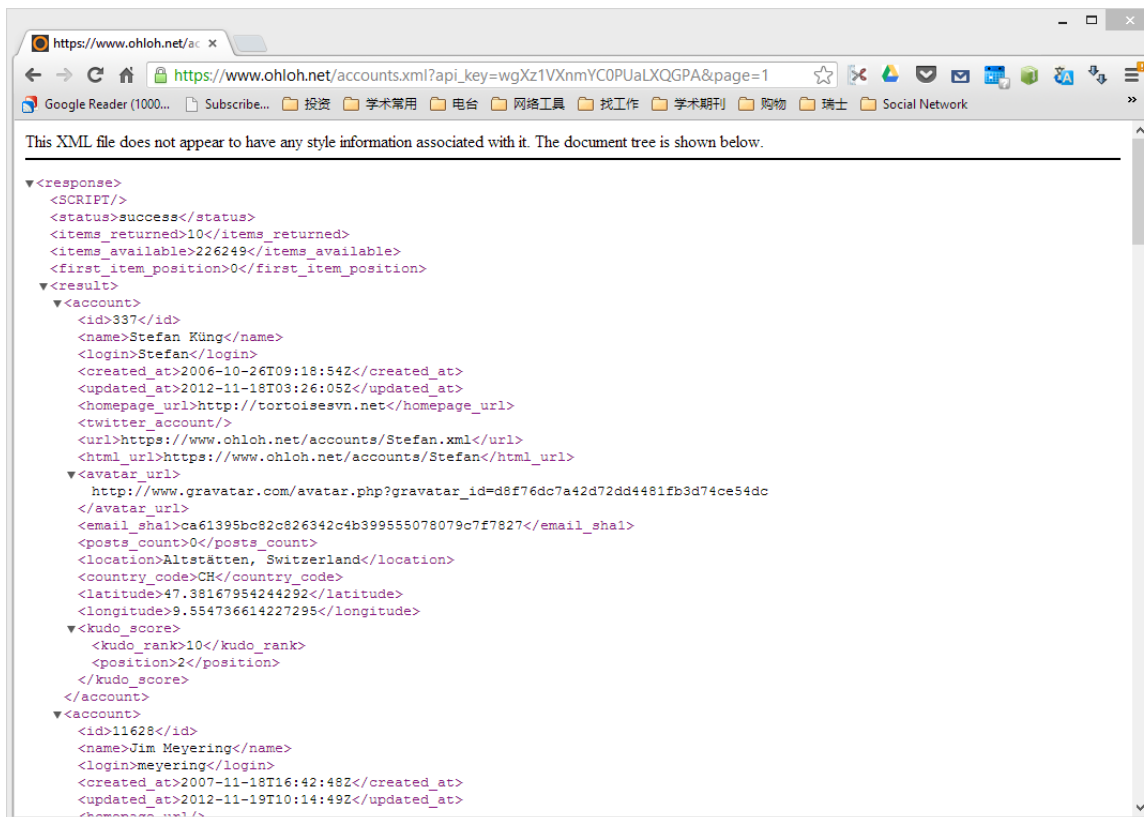
Append a .xml extension to the basic URL. For example, instead of `http://www.ohloh.net/projects/1`, which returns an HTML page, you would request `http://www.ohloh.net/projects/1.xml`.

Provide your API Key as an HTTP parameter. Your request will be forbidden without a valid `api_key`.

For example, to view the accounts in page 1 as XML, using an example API key, the complete URL would be:

https://www.ohloh.net/accounts.xml?api_key=wgXz1VXnmYCOPUaLXQGPA&page=1

you can get a page like this:



The screenshot shows a web browser window with the address bar containing the URL: `https://www.ohloh.net/accounts.xml?api_key=wgXz1VXnmYCOPUaLXQGPA&page=1`. The browser's developer tools are open, displaying the XML document tree. The XML content is as follows:

```
<?xml version="1.0" encoding="UTF-8" ?>
<response>
  <SCRIPT/>
  <status>success</status>
  <items_returned>10</items_returned>
  <items_available>226249</items_available>
  <first_item_position>0</first_item_position>
  <result>
    <account>
      <id>337</id>
      <name>Stefan KÜng</name>
      <login>Stefan</login>
      <created_at>2006-10-26T09:18:54Z</created_at>
      <updated_at>2012-11-18T03:26:05Z</updated_at>
      <homepage_url>http://tortoisesvn.net</homepage_url>
      <twitter_account/>
      <url>https://www.ohloh.net/accounts/Stefan.xml</url>
      <html_url>https://www.ohloh.net/accounts/Stefan</html_url>
    </account>
    <account>
      <id>11628</id>
      <name>Jim Meyering</name>
      <login>meyering</login>
      <created_at>2007-11-18T16:42:48Z</created_at>
      <updated_at>2012-11-19T10:14:49Z</updated_at>
      <homepage_url/>
```


2. Using Java to do data scraping

2.1. The ApiExample

Ohloh offers a tutorial on how to use api via an example. In this section, we will use this example to introduce some important codes.

You can download the example from

<http://meta.ohloh.net/wp-content/uploads/2012/02/ApiExample.java>

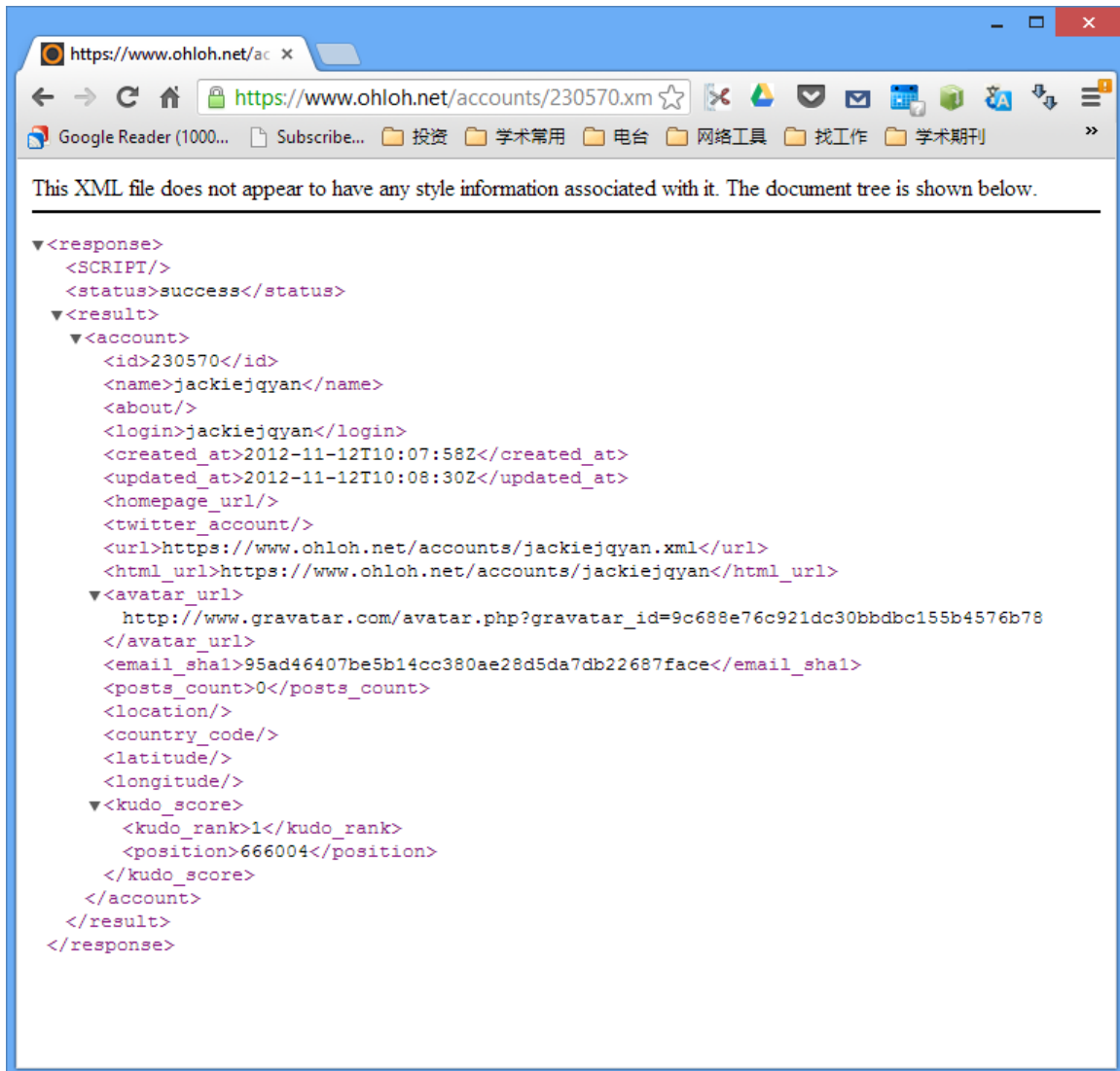
This example is used to retrieve data from the url

```
URL url = new URL("http://www.ohloh.net/accounts/" + emailDigest + ".xml?api_key=" + apiKey + "&v=1");
```

Where emailDigest is the message digest of the account's email address

Take my account as an example, the url would be

https://www.ohloh.net/accounts/230570.xml?api_key=wgXz1VXnmYC0PUaLXQGPA&v=1



```
URLConnection con = url.openConnection();
DocumentBuilder builder = DocumentBuilderFactory.newInstance().newDocumentBuilder();
Document doc = builder.parse(con.getInputStream());
```

Is to create a document from this URL's input stream and parse.

```
NodeList responseNodes = doc.getElementsByTagName("response");
```

Is to get the elements between the tags of "response", as shown in the following picture.

https://www.ohloh.net/ac x

https://www.ohloh.net/accounts/230570.xml

Google Reader (1000... Subscribe... 投资 学术常用 电台 网络工具 找工作 学术期刊

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<response>
  <SCRIPT/>
  <status>success</status>
  <result>
    <account>
      <id>230570</id>
      <name>jackiejqyan</name>
      <about/>
      <login>jackiejqyan</login>
      <created_at>2012-11-12T10:07:58Z</created_at>
      <updated_at>2012-11-12T10:08:30Z</updated_at>
      <homepage_url/>
      <twitter_account/>
      <url>https://www.ohloh.net/accou
      <html_url>https://www.ohloh.net/accounts/jackiejqyan</html_url>
    </account>
    <avatar_url>
      http://www.gravatar.com/avatar.php?gravatar_id=9c688e76c921dc30bbdbc155b4576b78
    </avatar_url>
    <email_shai>95ad46407be5b14cc390ae28d5da7db22687face</email_shai>
    <posts_count>0</posts_count>
    <location/>
    <country_code/>
    <latitude/>
    <longitude/>
    <kudo_score>
      <kudo_rank>1</kudo_rank>
      <position>666004</position>
    </kudo_score>
  </result>
</response>
```

doc.getElementsByTagName("response")

Elements between the tags of "response" and "</response"

Element resultElement = (Element)element.getElementsByTagName("result").item(0);
Is to get the elements between the tags of "result"

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<response>
  <SCRIPT/>
  <status>success</status>
  <result>
    <account>
      <id>230570</id>
      <name>jackiejqyan</name>
      <about/>
      <login>jackiejqyan</login>
      <created_at>2012-11-12T10:07</created_at>
      <updated_at>2012-11-12T10:08</updated_at>
      <homepage_url/>
      <twitter_account/>
      <url>https://www.ohloh.net/a
      <html_url>https://www.ohloh.net/accounts/jackiejqyan</html_url>
      <avatar_url>
        http://www.gravatar.com/avatar.php?gravatar_id=9c688e76c921dc30bbdbc155b4576b78
      </avatar_url>
      <email_shai>95ad46407be5b14cc380ae28d5da7db22687face</email_shai>
      <posts_count>0</posts_count>
      <location/>
      <country_code/>
      <latitude/>
      <longitude/>
      <kudo_score>
        <kudo_rank>1</kudo_rank>
        <position>666004</position>
      </kudo_score>
    </account>
  </result>
</response>

```

element.getElementsByTagName("result").item(0);
Elements between the tags of "<result>" and "</result>"

```
String realName =
accountElement.getElementsByTagName("name").item(0).getTextContent();
System.out.println("Located the real name: " + realName);
```

Is to get elements between the tags of "name" and print out the result.

https://www.ohloh.net/ac x

https://www.ohloh.net/accounts/230570.xml

Google Reader (1000... Subscribe... 投资 学术常用 电台 网络工具 找工作 学术期刊

This XML file does not appear to have any style information associated with it. The document tree is shown below.

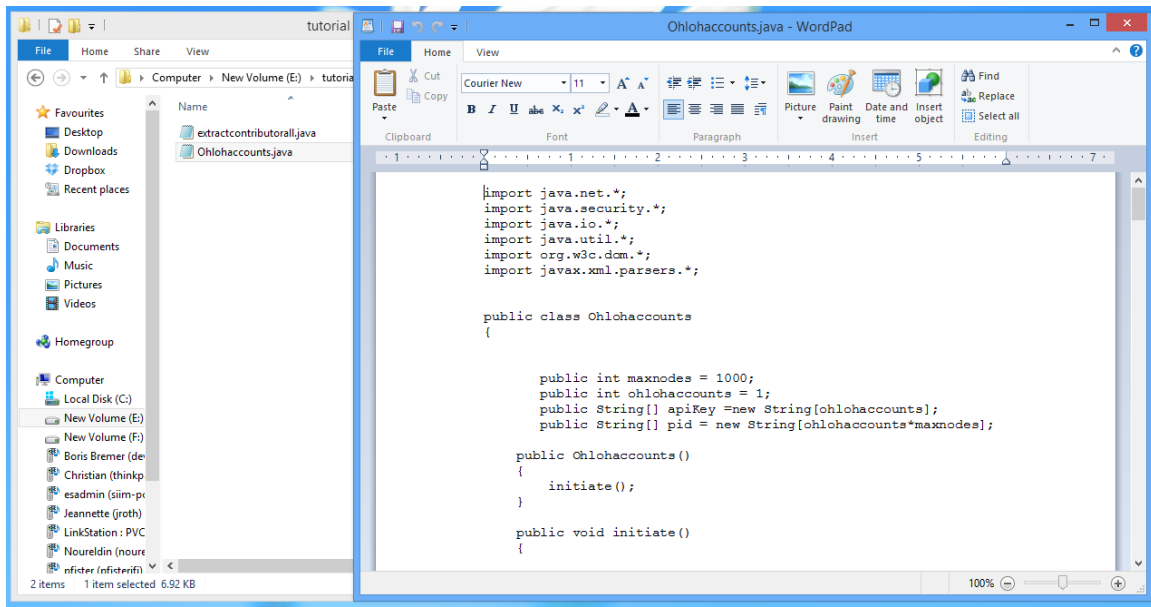
```
<?xml version="1.0" encoding="UTF-8" ?>
<response>
  <SCRIPT/>
  <status>success</status>
  <result>
    <account>
      <id>230570</id>
      <name>jackiejqyan</name>
      <about/>
      <login>jackiejqyan</login>
      <created_at>2012-11-12T10:07:58Z</created_at>
      <updated_at>2012-11-12T10:08:30Z</updated_at>
      <homepage_url/>
      <twitter account/>
      <url/>
      <html/>
      <html/>
    </account>
    <avatar>
      http://www.ohloh.net/avatars/230570/230570.jpg
    </avatar>
    <email/>
    <posts/>
    <location/>
    <count/>
    <latitude/>
    <longitude/>
    <kudo_score>
      <kudo_rank>1</kudo_rank>
      <position>666004</position>
    </kudo_score>
  </result>
</response>
```

accountElement.getElementsByTagName("name").item(0).getTextContent();

Elements between the tags of "<name>" and "</name>"

2.2. Coding a java file

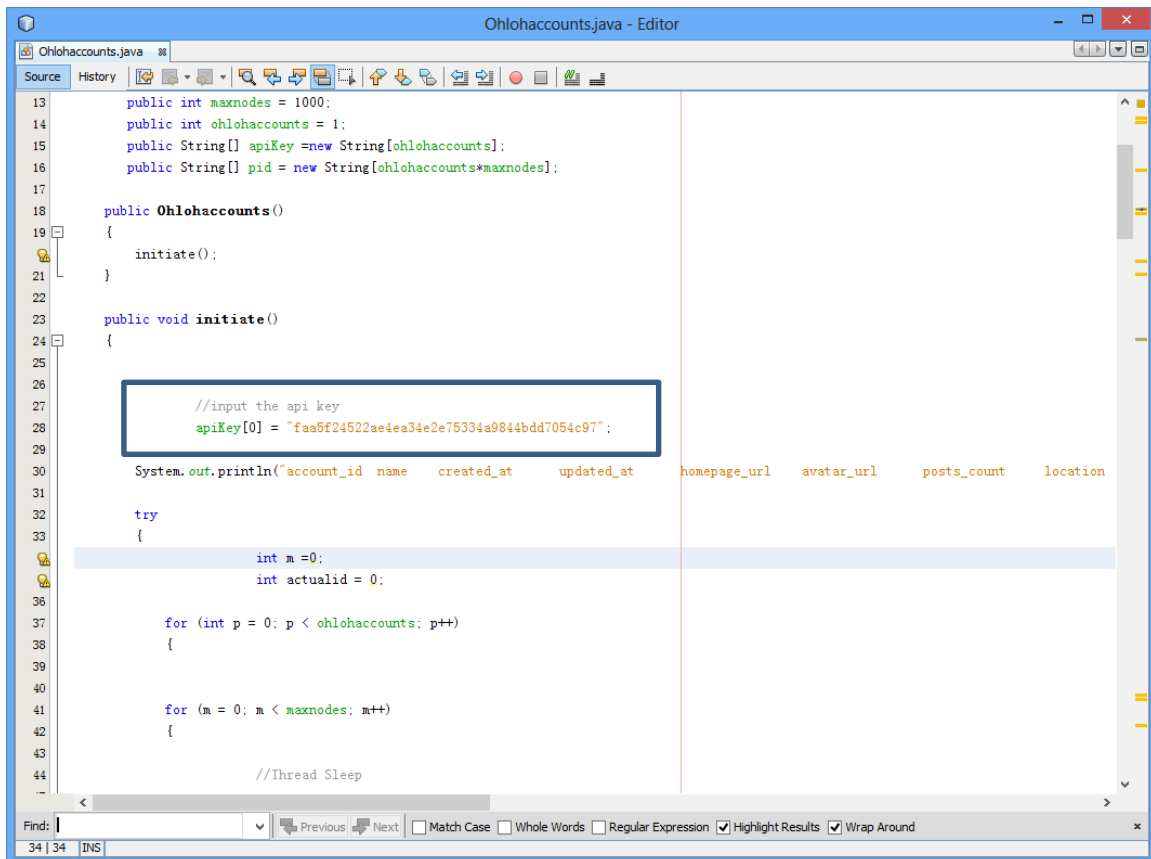
You can use the java file we provided and have some revisions depending on your scraping needs.



In the following, we will take Ohlohaccounts.java as an example.

2.2.1. Replacing the apiKey

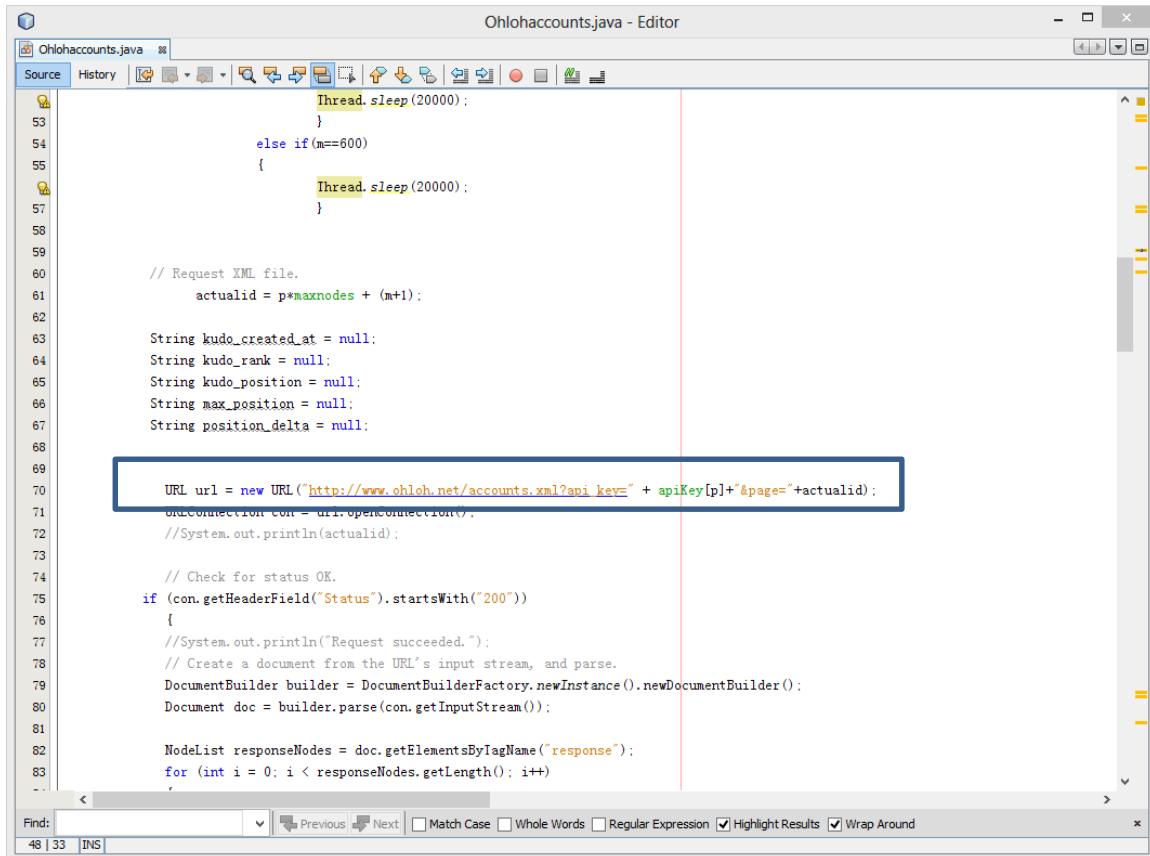
You need to replace the apikey in the file.



Because an Ohloh api has a limitation of 1000 requests per day, you may need to apply several apikeys according to your needs.

2.2.2. Change the xml page you want to retrieve.

As we mentioned, we retrieve data from the xml page, so you should change the xml page according to your requirements.



```
Ohlohaccounts.java - Editor
Ohlohaccounts.java
Source History
Thread.sleep(20000);
}
else if(m==600)
{
Thread.sleep(20000);
}

// Request XML file.
actualid = p*maxnodes + (m+1);

String kudo_created_at = null;
String kudo_rank = null;
String kudo_position = null;
String max_position = null;
String position_delta = null;

URL url = new URL("http://www.ohloh.net/accounts.xml?api_key=" + apiKey[p]+"&page="+actualid);
URLConnection con = url.openConnection();
//System.out.println(actualid);

// Check for status OK.
if (con.getHeaderField("Status").startsWith("200"))
{
//System.out.println("Request succeeded.");
// Create a document from the URL's input stream, and parse.
DocumentBuilder builder = DocumentBuilderFactory.newInstance().newDocumentBuilder();
Document doc = builder.parse(con.getInputStream());

NodeList responseNodes = doc.getElementsByTagName("response");
for (int i = 0; i < responseNodes.getLength(); i++)
{
}
}

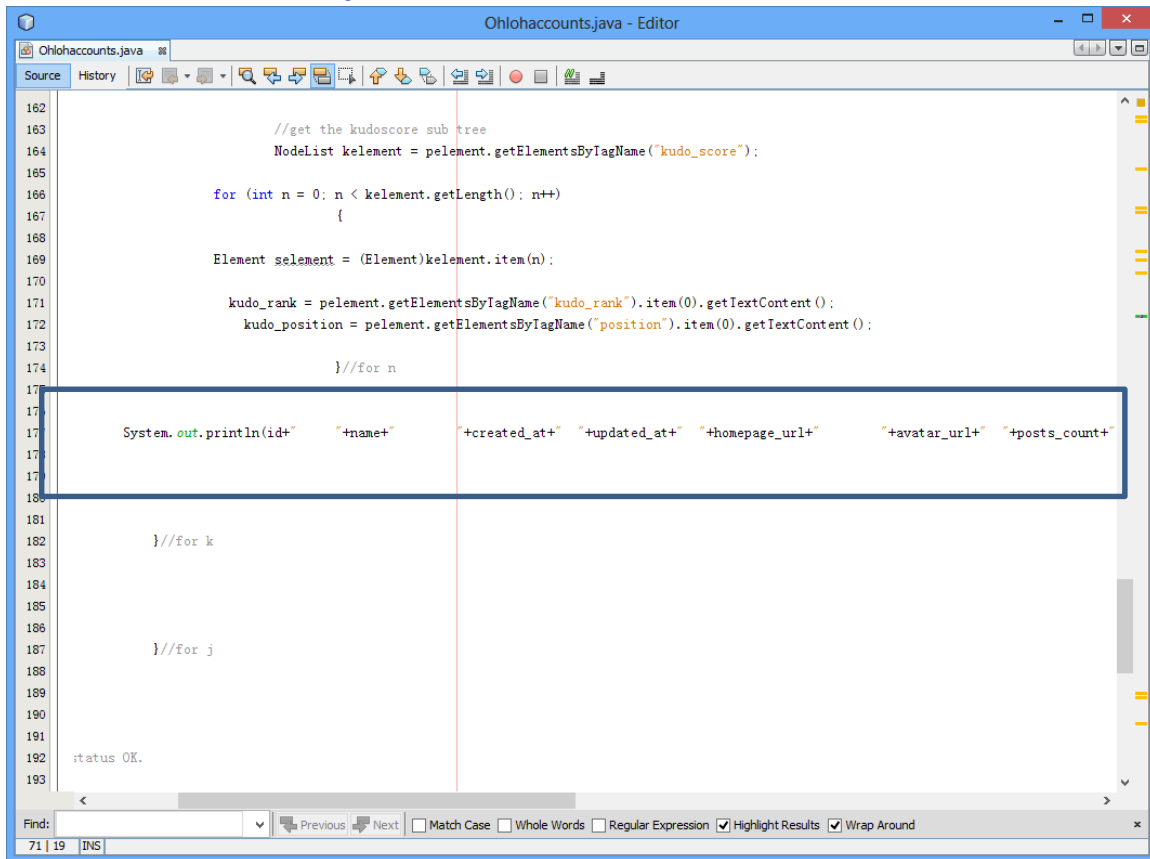
Find:
Previous Next Match Case Whole Words Regular Expression Highlight Results Wrap Around
48 | 33 | INS
```

2.2.3. Change the data you want to retrieve

You should change the definition of the data to be retrieved according to your needs. For example, in the following figure, we retrieve the data of "id", "name", "created_at" ..., you should change the data according to your requirements.

```
Ohlohaccounts.java - Editor
Ohlohaccounts.java
Source History
107
108
109     for (int j = 0; j < resultElement.getLength(); j++)
110     {
111         Element relement = (Element)resultElement.item(j);
112         NodeList projectElement = relement.getElementsByTagName("account");
113
114
115         for (int k = 0; k < projectElement.getLength(); k++)
116         {
117             Element pelement = (Element)projectElement.item(k);
118             // Lookup name.
119             String id = pelement.getElementsByTagName("id").item(0).getTextContent();
120             String name = pelement.getElementsByTagName("name").item(0).getTextContent();
121             name = name.replaceAll(",","");
122             name = new String(name.getBytes(),"UTF-8");
123
124
125
126
127             String created_at = pelement.getElementsByTagName("created_at").item(0).getTextContent();
128             created_at = created_at.replaceAll("T"," ");
129             created_at = created_at.replaceAll("Z","");
130
131
132
133             String updated_at = pelement.getElementsByTagName("updated_at").item(0).getTextContent();
134             // String description = pelement.getElementsByTagName("description").item(0).getTextContent();
135             updated_at = updated_at.replaceAll("T"," ");
136             updated_at = updated_at.replaceAll("Z","");
137
138
Find:
Previous Next Match Case Whole Words Regular Expression Highlight Results Wrap Around
71 | 19 | INS
```


2.2.4. Print out the data you want.



```
162
163 //get the kudoscree sub tree
164 NodeList kelement = pelement.getElementsByTagName("kudo_score");
165
166 for (int n = 0; n < kelement.getLength(); n++)
167 {
168
169     Element selement = (Element)kelement.item(n);
170
171     kudo_rank = pelement.getElementsByTagName("kudo_rank").item(0).getTextContent();
172     kudo_position = pelement.getElementsByTagName("position").item(0).getTextContent();
173
174     }//for n
175
176
177     System.out.println(id+" "+name+" "+created_at+" "+updated_at+" "+homepage_url+" "+avatar_url+" "+posts_count+"
178
179
180
181
182     }//for k
183
184
185
186
187     }//for j
188
189
190
191     status OK.
192
193
```

2.3. Compile and execute your java file

Run cmd and make the place where you locate your java file as the current directory.

Use `javac Ohlohaccounts.java` to compile the java file,

and use `java Ohlohaccounts>E:\tutorial\accounts.txt` to execute the file and save the results to `accnts.txt`.

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\Jiaqi Yan>E:
E:\>cd tutorial
E:\tutorial>javac Ohlohaccounts.java
E:\tutorial>java Ohlohaccounts>E:\tutorial\accounts.txt
E:\tutorial>

```

After the execution is done, you will get an txt file name “accounts.txt” storing the data you have retrieved.

account_id	name	created_at	updated_at	homepage_url	avatar_url	posts_count	location	color
337	Stefan K?ng	2006-10-26 09:18:54	2012-11-18 03:26:05	http://tortoisesvn.net	http://www.gravatar.com/av			
11628	Jim Meyering	2007-11-18 16:42:48	2012-11-19 11:18:05		http://www.gravatar.com/avatar.php?gravatar			
75847	odvarko	2010-03-03 13:16:49	2012-11-19 01:30:03		http://www.gravatar.com/avatar.php?gravatar_id=34b			
5439	Junio C Hamano	2007-06-29 19:01:21	2012-11-17 14:17:36	http://git-blame.blogspot.com	http://www.gravata			
28796	Damon Kohler	2008-12-19 20:46:32	2012-11-19 09:40:48	http://www.damonkohler.com/	http://www.gravata			
29081	Roland McGrath	2008-12-24 06:08:06	2012-11-19 11:15:03	http://www.frob.com/	http://www.gravatar.com/av			
15765	Eion Robb	2008-03-06 08:32:07	2012-11-19 06:47:55	http://eion.robbmob.com/blog/	http://www.gravata			
19797	Tim Kosse	2008-06-17 13:30:05	2012-11-16 14:21:50	http://filezilla-project.org	http://www.gravata			
18327	Mark Story	2008-05-11 17:01:21	2012-11-19 09:49:08	http://mark-story.com	http://www.gravatar.com/av			
3597	Karl Fogel	2007-05-10 23:31:57	2012-11-18 18:42:31	http://www.red-bean.com/kfogel/	http://www.gravata			
15138	William A. Rowe Jr.	2008-02-22 04:40:39	2012-11-19 01:53:35		http://www.gravatar.com/avatar.php			
214053	misha	2012-06-25 18:37:08	2012-11-17 13:51:11	http://www.gravatar.com/avatar.php?gravatar_id=369				
11203	Michael Natterer	2007-11-05 12:52:54	2012-11-19 08:50:35		http://www.gravatar.com/avatar.php			
3472	Sven Neumann	2007-05-07 21:02:24	2012-11-19 09:06:13	http://sven.gimp.org/	http://www.gravatar.com/av			
2232	C. Michael Pilato	2007-03-09 16:57:48	2012-11-18 02:29:01	http://www.cmichaelpilato.com/	http://www			
9897	Linus.Torvalds	2007-09-29 02:36:23	2012-11-17 14:20:37	http://www.gravatar.com/avatar.php?gravatar				
3592	Hyrum Wright	2007-05-10 20:19:09	2012-11-16 14:32:39	http://www.hyumwright.org/	http://www.gravata			
21262	Sergey Poznyakoff	2008-07-19 21:21:43	2012-11-19 11:05:11	http://gray.gnu.org.ua	http://www.gravata			
2217	Mark Phippard	2007-03-09 00:46:25	2012-11-18 22:06:49	http://markhipp.blogspot.com/	http://www.gravata			
3739	Paolo Bonzini	2007-05-14 14:48:17	2012-11-19 11:11:45	http://www.inf.unisi.ch/phd/bonzini	http://www			
17455	Bert Huijben	2008-04-17 15:46:49	2012-11-17 14:14:11	http://b.qqn.nl/	http://www.gravatar.com/av			
50801	trawick	2009-07-09 16:10:31	2012-11-19 05:16:28	http://www.gravatar.com/avatar.php?gravatar_id=2ba				
8977	jacob	2007-09-03 15:51:26	2012-11-19 06:37:02	http://www.gravatar.com/avatar.php?gravatar_id=304				
96540	WU Jun	2010-07-06 09:59:39	2012-11-01 18:54:11	http://lihdd.net	http://www.gravatar.com/avatar.php			

3. Load data into MySQL database

Please refer to the tutorial 1: MySQL Database and Relational Data Processing