SEARCHMONKEY APPLICATION

How to make a CUSTOM DATA SERVICE for use with an Infobar (using XPather)

OVERVIEW

SearchMonkey apps can display content scraped from almost any page on the internet.

Using a programming language called XSLT, this tutorial will teach you how to create a “custom data service” which will feed content to the SearchMonkey apps you will learn to create in the next tutorial.

Don’t worry, this sounds more technically complex than it actually is! **You don’t need to know XSLT code** to get your Custom Data Service up and running, thanks to the good people who made the wonderful XPather Firefox Add-on.

Custom Data Service – To Start:

1. Download **XPather**, the Firefox Add-on
   https://addons.mozilla.org/en-US/firefox/addon/1192

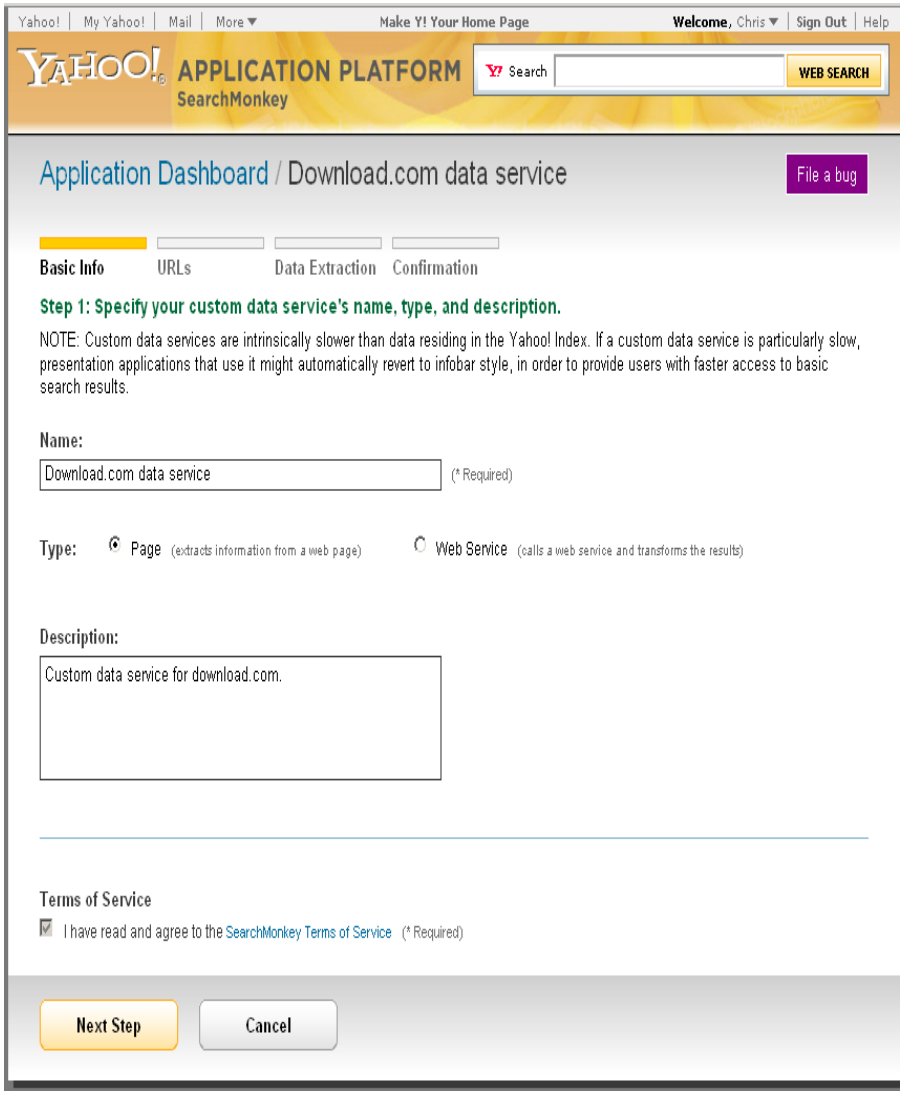
   This extension will help you to quickly and easily grab the XSLT code from a webpage and paste that code into your Custom Data Service

2. Go to http://developer.yahoo.com/searchmonkey/

   and Click on “Build an App”.

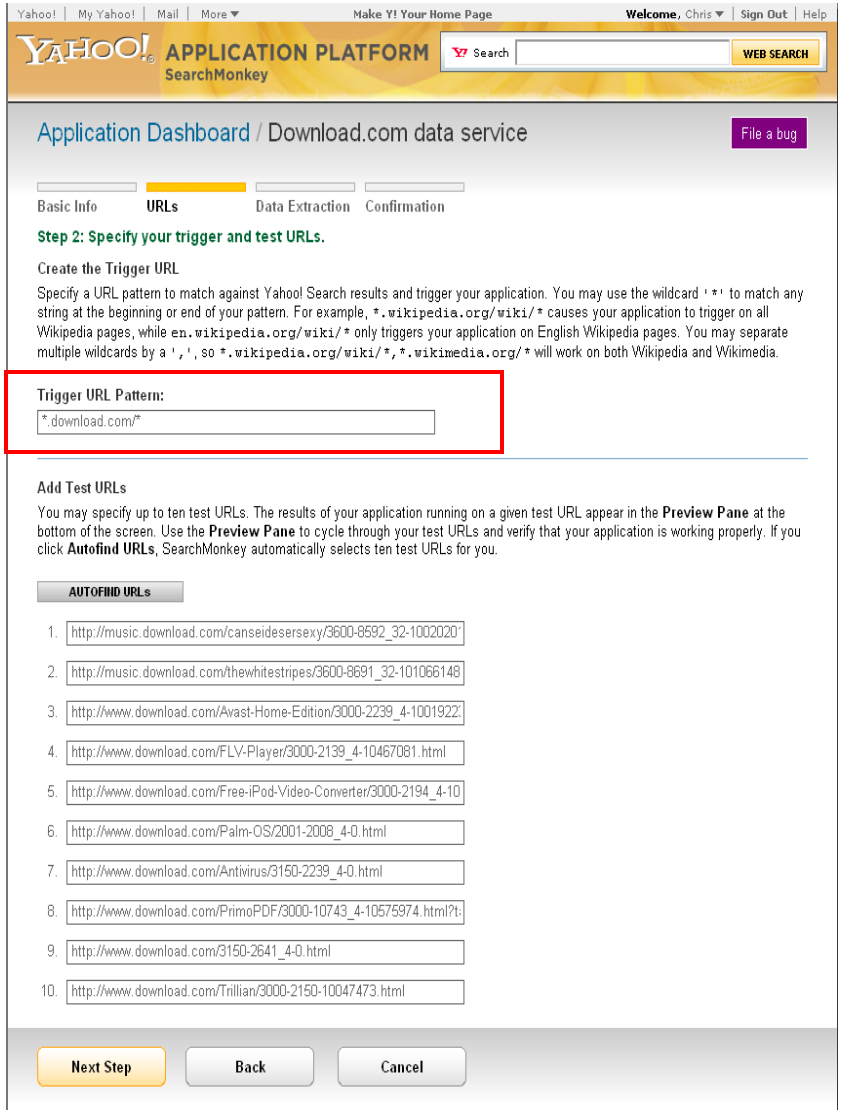
3. SCROLL DOWN to the *Custom Data Services* section and click on “Create a new data service”.

Building your Custom Data Service
Step 1: **Basic Info**



1. Give your Data Service a nice name.
2. Select “Page” for Type.
3. Give your Data Service a nice description.
4. Agree to the Terms of Service. Thanks for being agreeable.
5. Click NEXT and you should be magically transported to the URLs page.

URLs - Trigger URL Pattern Overview



A note on making a Trigger URL Pattern:

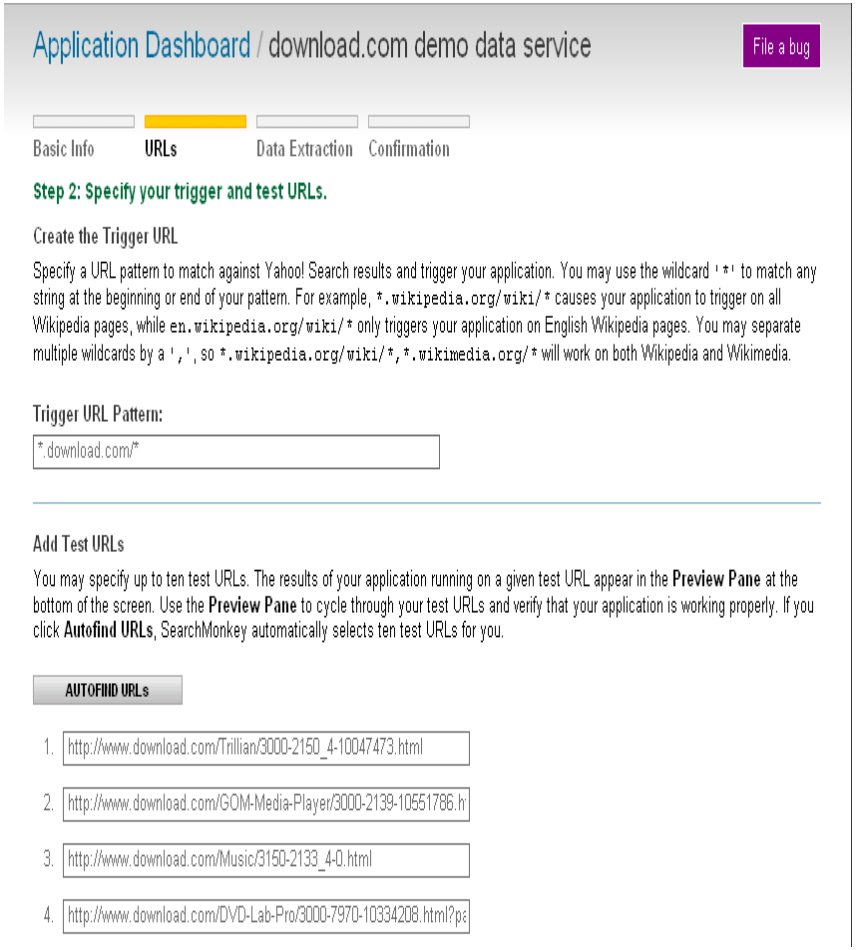
Trigger URL patterns are simply the root url of any page, minus the beginning portion.

For example, if you wanted to extract information from the Pitchforkmedia.com Album Review pages, you would go to: http://www.pitchforkmedia.com/article/record .

From this URL you would remove http://www and the page name, so that you are left with the root: .pitchforkmedia.com/article/record_review/*

If, on the other hand, you wanted to pull content from Pitchfork's feature stories, your "Trigger URL Pattern" would look more like this: pitchforkmedia.com/article/feature/*. Because Pitchfork keeps features in that folder, as you can see from this feature: http://www.pitchforkmedia.com/article/featur

Building your Custom Data Service
Step 2: **URLs**



1. For the tutorial, we will be extracting data from http://www.download.com/

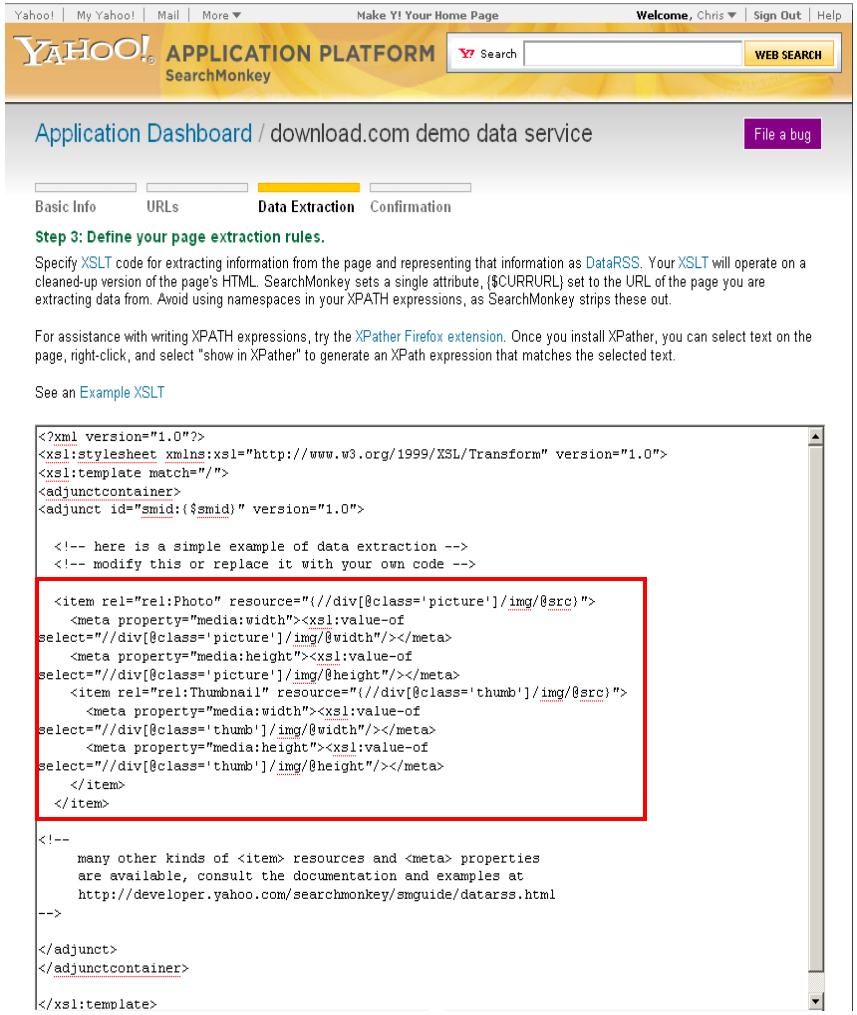
- Enter ***.download.com/*** as the *Trigger URL Pattern*.

3. Click **Autofind URLs**

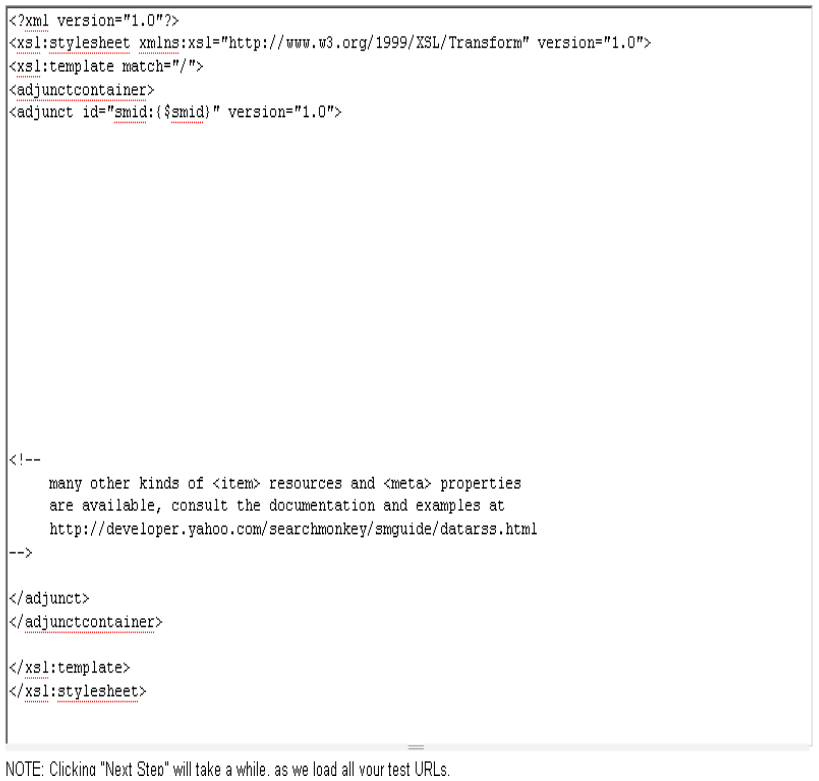
NOTE: If **Autofind URLs** isn't working for you, you can just enter five or six URLs by hand. For the sake of this tutorial, just make sure you have http://www.download.com/Trillian/3000-2 (Download.com's Trillian page) in the first slot so that your screen shots match mine.

4. Click NEXT and you should be magically transported to the Data Extraction page…

Building your Custom Data Service
Step 3: **Data Extraction**



1. On the Data Extraction page, delete the text in the middle starting with “<item” and ending with </item> so your page looks like this:

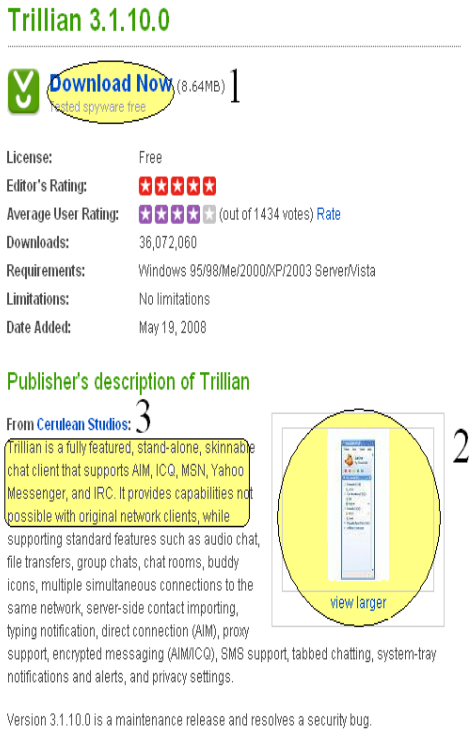


Building your Custom Data Service

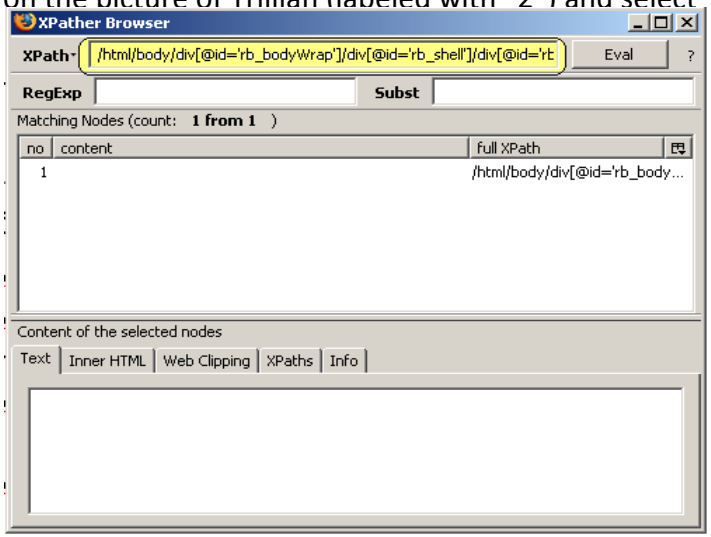
Step 3: **Data Extraction - Extracting a Picture with XPather**

- In a different browser window, open the page you want to extract data from. For the sake of this tutorial, please open: **http://www.download.com/Trillian/3000-2150-1004**

  Below is a screenshot with the elements that you will be extracting circled, numbered, and highlighted in a lovely yellow. (1. LINK, 2. PICTURE, 3. TEXT)



3. We are going to extract the picture first, so right click on the picture of Trillian (labeled with "2") and select

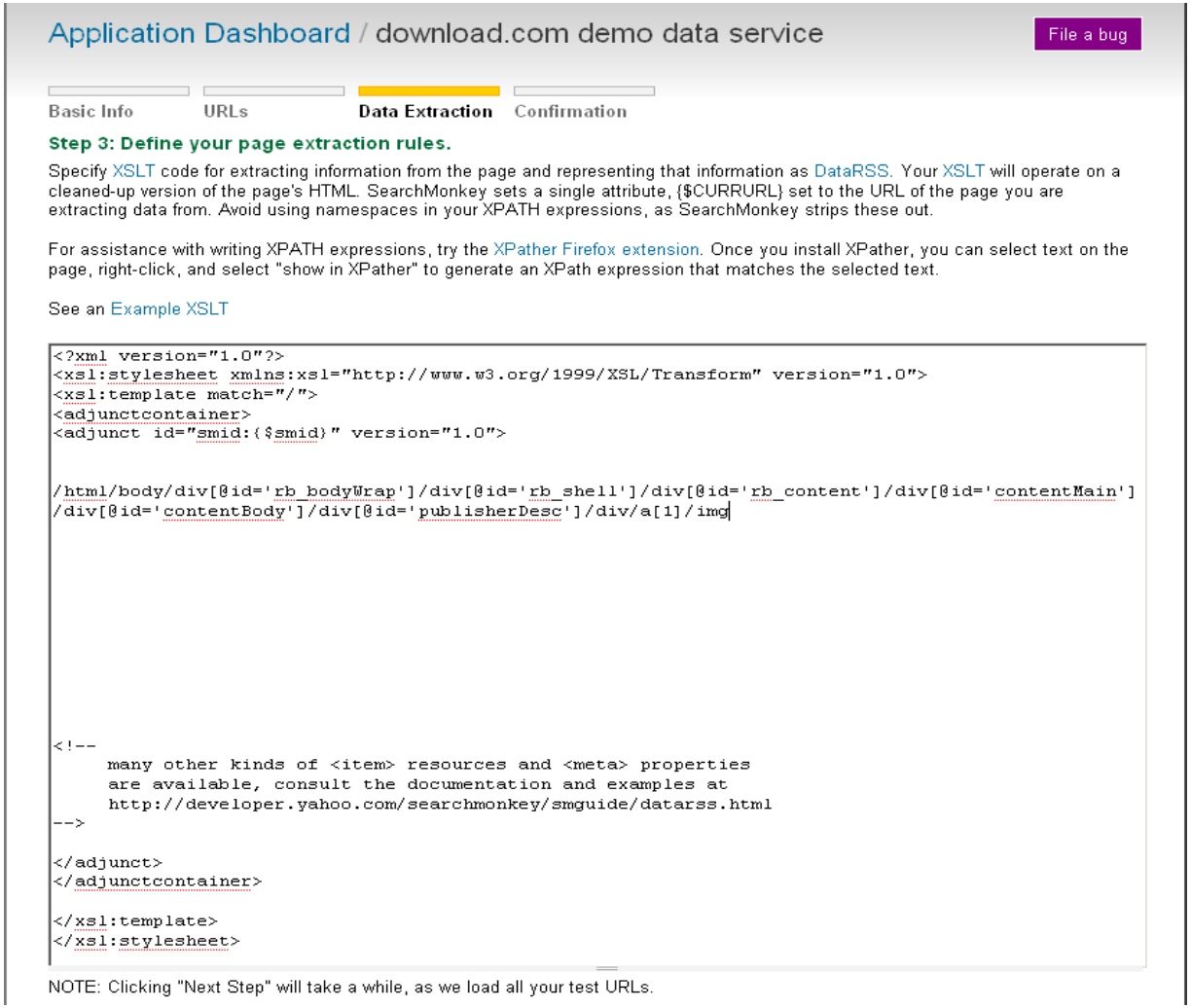


4. Copy the "XPath" at the top (highlighted in a lovely yellow)

5. Paste that XPath into the Data Extraction page in the Dashboard so that you end up with this…

Building your Custom Data Service

Step 3: **Data Extraction - Extracting a Picture with XPather**



good job! now, you have to tweak the code a bit…

Building your Custom Data Service

Step 3: **Data Extraction** - **Tweaking the picture's XPath**

Right now, your picture code looks like this

```
/
    html/body/div[@id='rb_bodyWrap']/div[@id='rb_shell']/
    div[@id='rb_content']/div[@id='contentMain']/div[@id='
    contentBody']/div[@id='publisherDesc']/div/a[1]/img
```

You need it to look like this

```
<meta rel="rel:photo"
    resource="{//*[@id='rb_bodyWrap']/div[@id='rb_shell']/
    div[@id='rb_content']/div[@id='contentMain']/div[@id='
    contentBody']/div[@id='publisherDesc']/div/a[1]/img/@s
    rc}" />
```

But how??? Well…

Building your Custom Data Service

Step 3: **Data Extraction** - **Tweaking the picture's XPath**

- First, trim off everything at the beginning of your line of code until you get to the first **div**. In this case, you're going to trim off **/html/body/** from the beginning. So you're left with:

div[@id='rb_bodyWrap']/div[@id='rb_shell']/div[@id='rb_content']/div[@id='contentMain']/div[@id='contentBody']/div[@id='publisherDesc']/div/a[1]/img

- Next, add **/@src** to the end of the line. NOTE: If you were extracting a link, you'd add **/@href**. And for text, add nothing.

div[@id='rb_bodyWrap']/div[@id='rb_shell']/div[@id='rb_content']/div[@id='contentMain']/div[@id='contentBody']/div[@id='publisherDesc']/div/a[1]/img/@src

- You're extracting a picture, so replace that **div** near the beginning of the line with an asterisk (*****). NOTE: For text and links, leave it as **div**.

***[@id='rb_bodyWrap']/div[@id='rb_shell']/div[@id='rb_content']/div[@id='contentMain']/div[@id='contentBody']/div[@id='publisherDesc']/div/a[1]/img/@src}" />**

Building your Custom Data Service

Step 3: **Data Extraction** - **Tweaking the picture's XPath**

4. Next, enclose your line of code with **resource="{//** at the beginning, and **}" /** **>** at the end (Remember to put a space between the end quote and back slash). Now, your code should look like this:

   resource="{//*[@id='rb_bodyWrap']/div[@id='rb_shell']/div[@id='rb_content']/div[@id='contentMain']/div[@id='contentBody']/div[@id='publisherDesc']/div/a[1]/img/@src}" />

5. Then, give it a name tag at the beginning of the line. The name is for your use only to describe what you have extracted. I'm going to call this data resource "photo" by adding **<meta rel="rel:photo"** and then a space. So your finished product should look like this:

   <meta rel="rel:photo"
   resource="{//*[@id='rb_bodyWrap']/div[@id='rb_shell']/div[@id='rb_content']/div[@id='contentMain']/div[@id='contentBody']/div[@id='publisherDesc']/div/a[1]/img/@src}" />

NOTE: If you were extracting a link, you'd add **<item rel** at the beginning.

Building your Custom Data Service

Step 3: **Data Extraction - Tweaking the picture's XPath**

6. Click Save & Refresh, and now the lower window panel should look just like this:

YOU DID IT! I'm proud of you. Next: Extracting Text…

*** If it doesn't look like the above example, see our troubleshooting page.**

Building your Custom Data Service

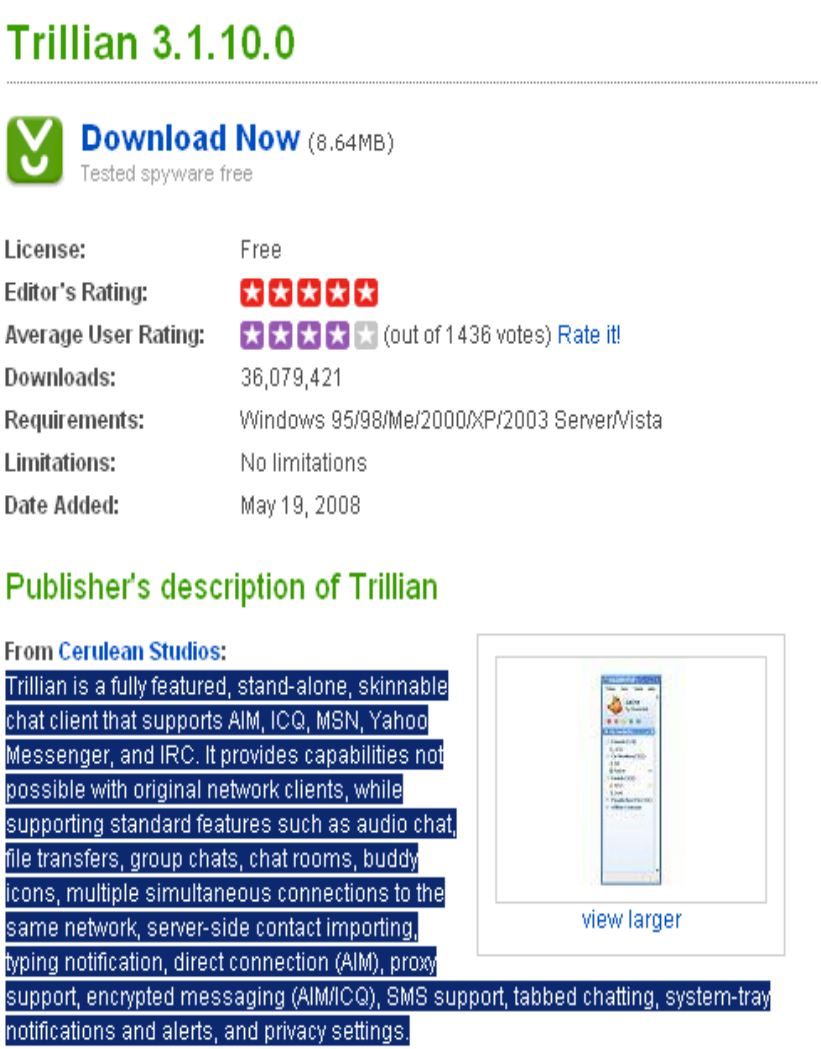
Step 3: **Data Extraction** - **Extracting Text with XPather**

- Go back to the Trillion download.com page: http://www.download.com/Trillia

3. Highlight the text on that page that you want to extract.

   For now, you're going to highlight the Publisher's Description of Trillian. Like in this screenshot →

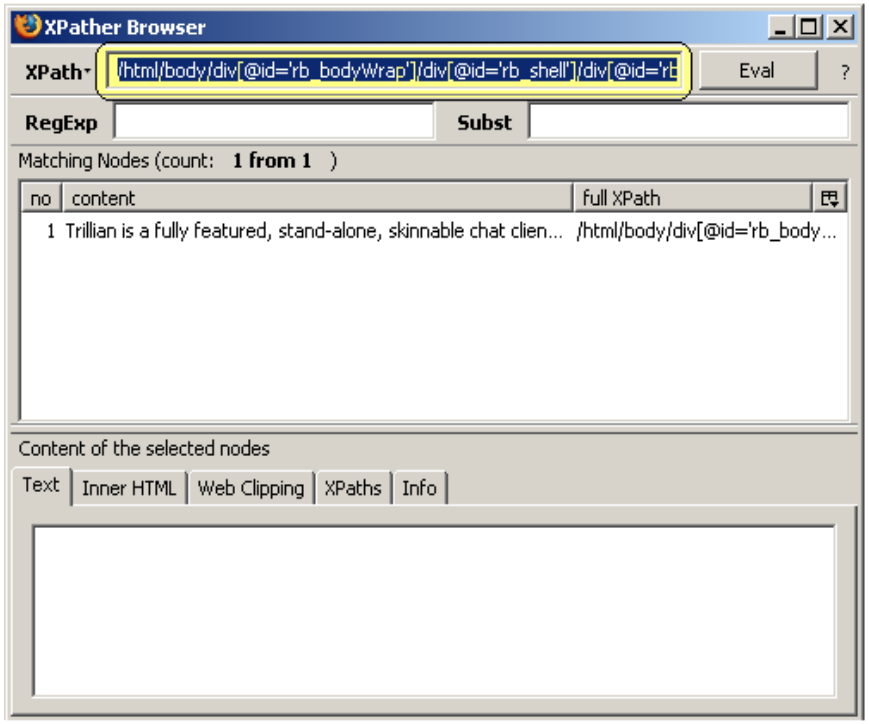
3. Right click on the highlighted text and select "Show in XPather" from the drop-down menu.



Building your Custom Data Service

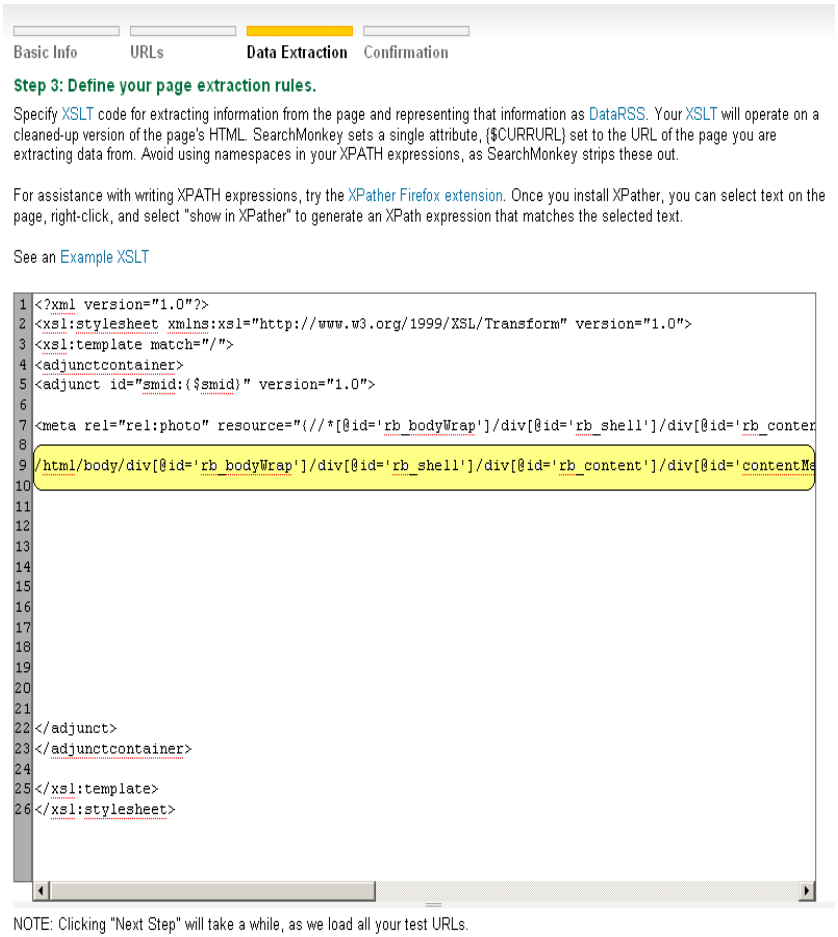
Step 3: **Data Extraction - Extracting Text with XPather**

Your Xpather Browser should look like this (see below):



4. **Select and COPY the text in the "XPath" box** (highlighted in a lovely yellow above).

5. Paste that text into the Search Monkey application dashboard, a line or two below your "rel:photo" code, so that your dashboard looks like this (see below):



Building your Custom Data Service

Step 3: **Data Extraction** – **Tweaking the Text's XPath**

Freshly copied from XPather, the text XPath looks like this:

/html/body/div[@id='rb_bodyWrap']/div[@id='rb_shell']/div[@id='rb_content']/div[@id='contentMain']/div[@id='contentBody']/div[@id='publisherDesc']/p[1]

But, in order for the Search Monkey tool to make sense of it, you need it to look like this:

<meta rel="rel:text" resource="{//div[@id='rb_bodyWrap']/div[@id='rb_shell']/div[@id='rb_content']/div[@id='contentMain']/div[@id='contentBody']/div[@id='publisherDesc']/p[1]}" />

Building your Custom Data Service

Step 3: **Data Extraction - Tweaking the Text's XPath**

1. First trim off everything at the beginning of your line of code until you get to the first **div**. In this case, you're going to trim off **/html/body/** from the beginning. So you're left with:

   div[@id='rb_bodyWrap']/div[@id='rb_shell']/div[@id='rb_content']/div[@id='contentMain']/div[@id='contentBody']/div[@id='publisherDesc']/p[1]

- Then, enclose your line of code with **resource="{//** at the beginning, and **}" />** at the end (remember to include a space between the end quote and the back slash). Now, your code should look like this:

  resource="{//div[@id='rb_bodyWrap']/div[@id='rb_shell']/div[@id='rb_content']/div[@id='contentMain']/div[@id='contentBody']/div[@id='publisherDesc']/p[1]}" />

  NOTE: We are extracting text, so we're going to leave that **div** near the beginning of the line. If we were extracting a picture, we would replace that **div** with an asterisk *****.

3. Then, give it a name. I'm going to call this data resource "text" by adding **<meta rel="rel:text" and a SPACE** to the beginning of the line so that it looks like this:

   <meta rel="rel:text"
   resource="{//div[@id='rb_bodyWrap']/div[@id='rb_shell']/div[@id='rb_content']/div[@id='contentMain']/div[@id='contentBody']/div[@id='publisherDesc']/p[1]}" />

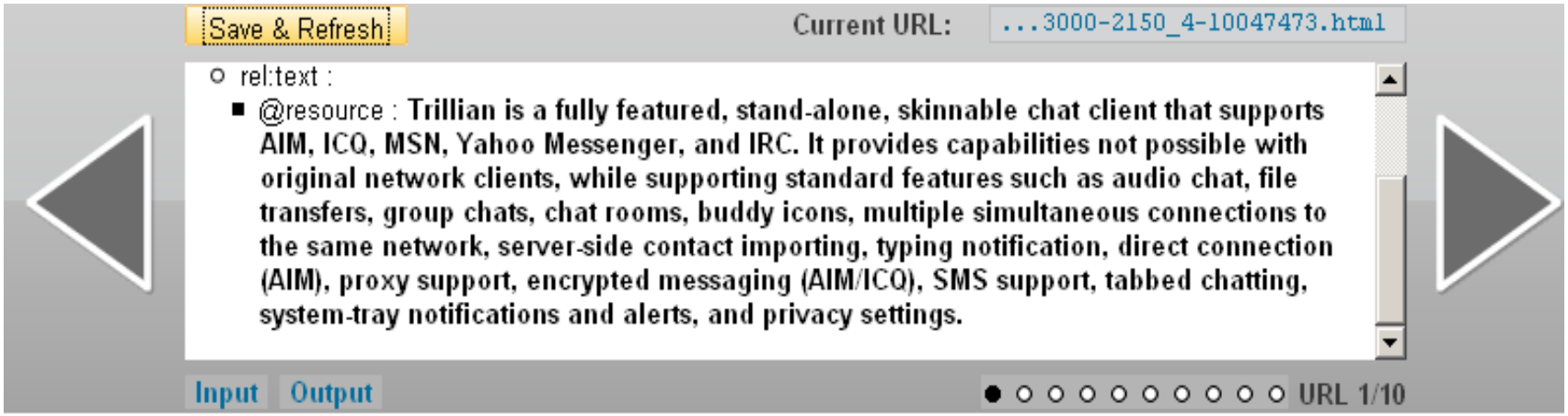
Building your Custom Data Service

Step 3: **Data Extraction - Tweaking the Text's XPath**

4. Your finished product should look like this:
   <meta rel="rel:text" resource="{//div[@id='rb_bodyWrap']/

   div[@id='rb_shell']/div[@id='rb_content']/div[@id='contentMain']/div[@id='contentBody']/div[@id='publisherDesc']/p[1]}" />

5. Now click Save & Refresh, and now the lower window panel should look just like this:



YAY! Next: Extracting Links…

Building your Custom Data Service

Step 3: **Data Extraction – Extracting a Link with XPather**

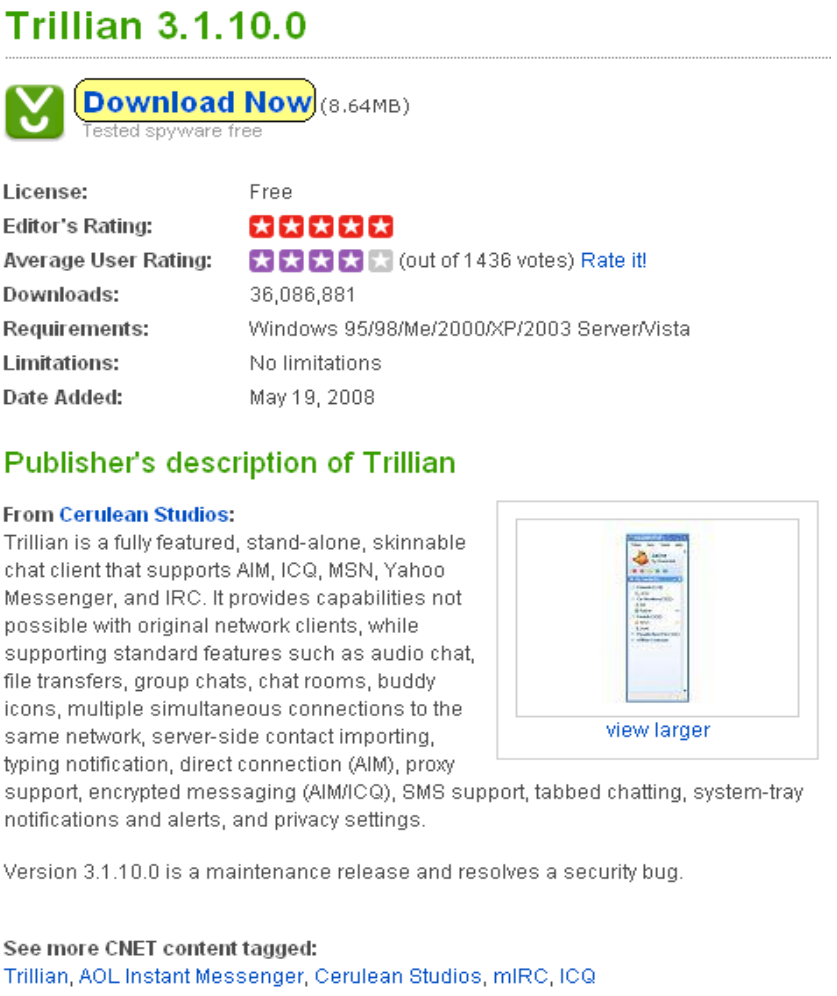
- Go back to your open download.com page: http://www.download.com/Trillian/

3. Right Click on the link you want to extract.

   For now, grab the big "Download Now" link at the top (highlighted here in a lovely yellow) →

6. Select "Show in XPather" from the menu that appears.

   Now, your XPather Browser should look like this (next page)…



Trillian 3.1.10.0

Download Now (8.64MB)
Tested spyware free

License:	Free
Editor's Rating:	★★★★★
Average User Rating:	★★★★☆ (out of 1436 votes) Rate it!
Downloads:	36,086,881
Requirements:	Windows 95/98/Me/2000/XP/2003 Server/Vista
Limitations:	No limitations
Date Added:	May 19, 2008

Publisher's description of Trillian

From Cerulean Studios:
Trillian is a fully featured, stand-alone, skinnable chat client that supports AIM, ICQ, MSN, Yahoo Messenger, and IRC. It provides capabilities not possible with original network clients, while supporting standard features such as audio chat, file transfers, group chats, chat rooms, buddy icons, multiple simultaneous connections to the same network, server-side contact importing, typing notification, direct connection (AIM), proxy support, encrypted messaging (AIM/ICQ), SMS support, tabbed chatting, system-tray notifications and alerts, and privacy settings.

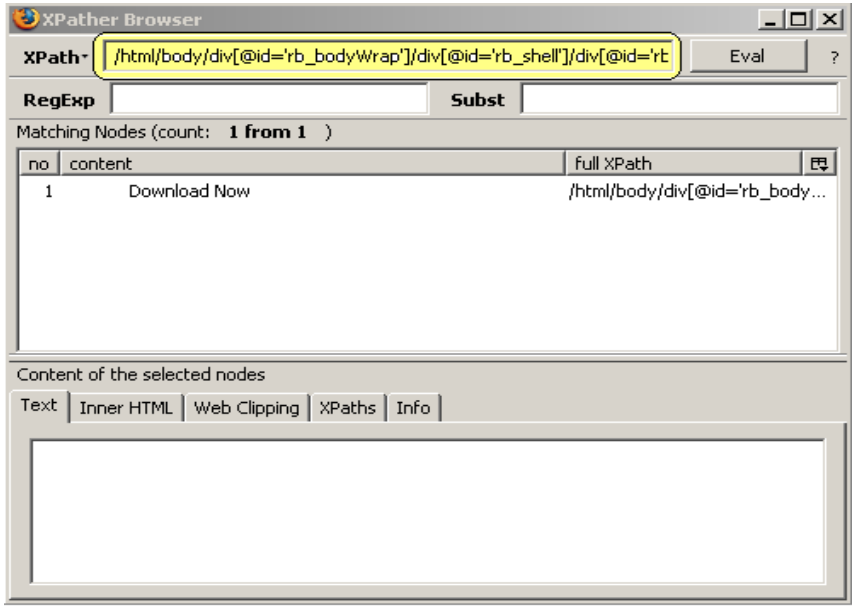
view larger

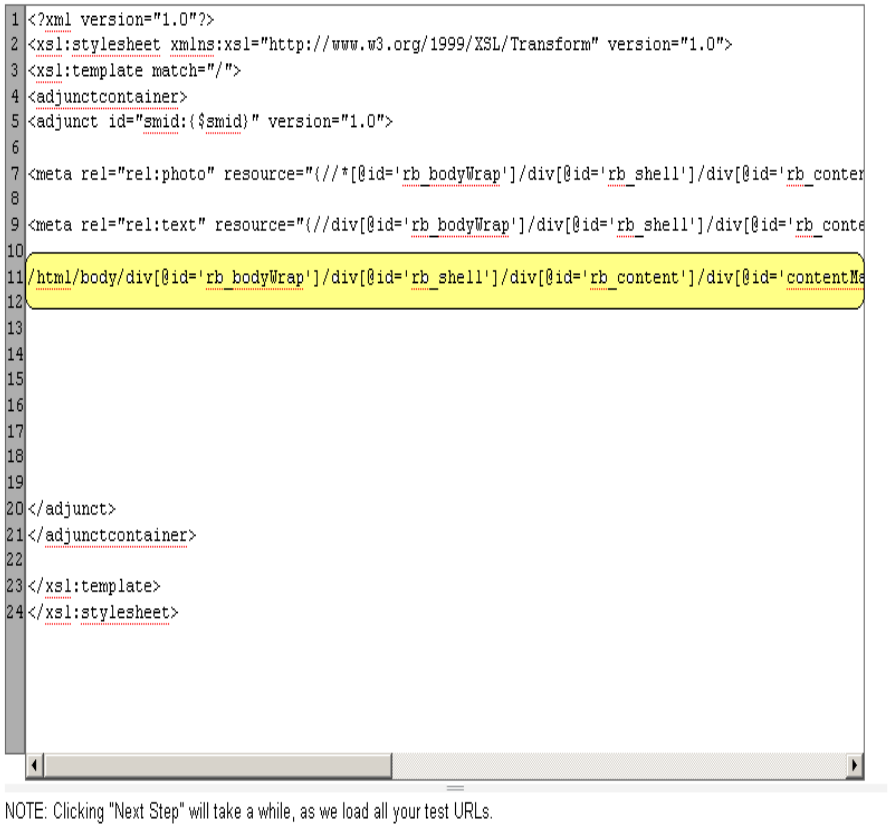
Version 3.1.10.0 is a maintenance release and resolves a security bug.

See more CNET content tagged:
Trillian, AOL Instant Messenger, Cerulean Studios, mIRC, ICQ

Building your Custom Data Service

Step 3: **Data Extraction - Extracting a Link with XPather**





NOTE: Clicking "Next Step" will take a while, as we load all your test URLs.

4. Copy the text from the XPath box (highlighted above in a LOVELY yellow).

5. Paste that below the picture and text code in your Data Extraction page in the Search Monkey Application Dashboard
   so that it looks like this

Done and done! Now you're ready for the next part…Tweaking the XPath

Building your Custom Data Service

Step 3: **Data Extraction** – Tweaking the **Link's** **XPath**

So, XPather gave you this:

```
/html/body/div[@id='rb_bodyWrap']/div[@id='rb_shell']
/div[@id='rb_content']/div[@id='contentMain']/div[@id=
'contentBody']/div[@id='dlNow']/a[2]
```

But you really want this:

```
<item rel="rel:link"
resource="{//div[@id='rb_bodyWrap']/div[@id='rb_shell'
]/div[@id='rb_content']/div[@id='contentMain']/div[@id
='contentBody']/div[@id='dlNow']/a[2]/@href}" />
```

What're ya gonna do?

Building your Custom Data Service

Step 3: **Data Extraction** – Tweaking the **Link's XPath**

- **Trim off the first part of your code until you get to the first *div*.**
  So, in this case, you're going to trim off **/html/body/** from the beginning of the line so that you're left with:

  div[@id='rb_bodyWrap']/div[@id='rb_shell']/div[@id='rb_content']/div[@id='contentMain']/div[@id='contentBody']/div[@id='dlNow']/a[2]

- **Enclose your code with *resource="{//* at the beginning, and */@href}" />* at the end.**

  resource="{//div[@id='rb_bodyWrap']/div[@id='rb_shell']/div[@id='rb_content']/div[@id='contentMain']/div[@id='contentBody']/div[@id='dlNow']/a[2] /@href}" />

NOTE: Because this is a link, you're adding an **@href** to this code. If this was a picture, you would add an **@src**, and you were grabbing text, you would add nothing.

- **Give your data resource a name by adding *<item rel="rel:link" and a SPACE*** **to the beginning of the line:**

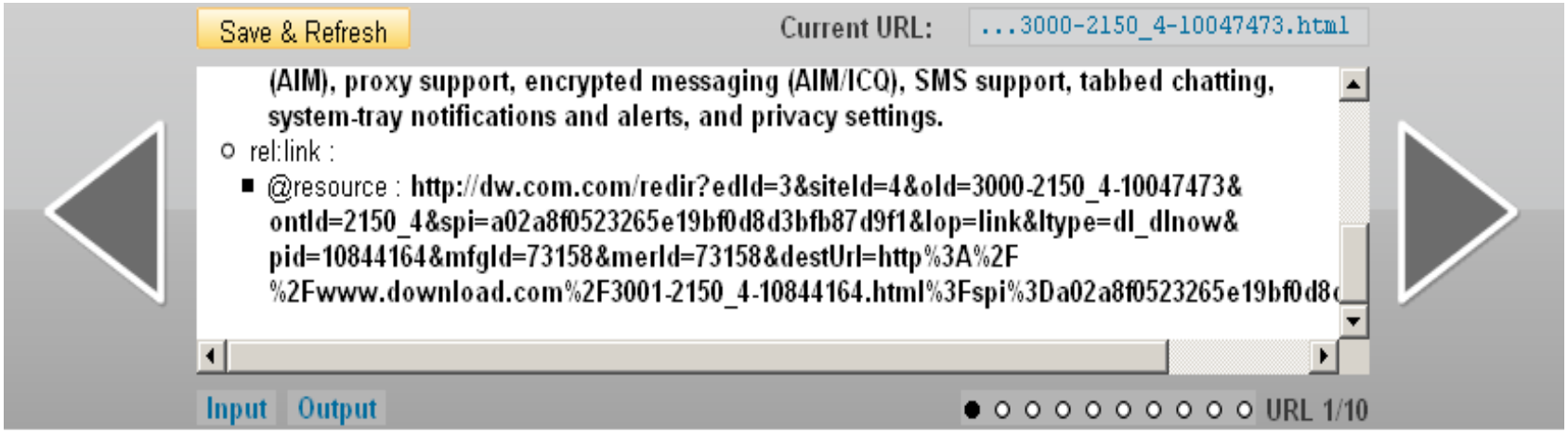
  <item rel="rel:link“ resource="{//div[@id='rb_bodyWrap']/div[@id='rb_shell']/div[@id='rb_content']/div[@id='contentMain']/div[@id='contentBody']/div[@id='dlNow']/a[2]/@href}" />

NOTE: Because this is a link, you're tagging it with **<item rel** as opposed to **<meta rel** which you would use if this was a picture or text.

Building your Custom Data Service

Step 3: **Data Extraction** – Tweaking the **Link's XPath**

4. **At last, click Save & Refresh, and now that lower panel should look something like this:**



You're done! Go ahead and click on "Next Step" at the bottom of the dashboard.

Building your Custom Data Service

Step 4: **Confirmation**

The hardest part is over. Now that you have extracted data you can arrange it into a SearchMonkey App by **Creating a Presentation Application**

To get started, click on the "Create a new Presentation Application" link from the confirmation page.

Note: You can also create the Presentation App at a later date by going to the Developer Tool Home page, clicking on “Create an App” and choosing “Create a new Application” under Presentation Applications.