Under the Covers: A Beginner's Guide to Learning about XML using Popfly

XML (eXtensible Markup Language) is a notation system designed to describe data. XML has become a standard for describing and transferring data over the Internet.

In this lesson you will learn to recognize XML by seeing how Popfly blocks use XML to describe their inputs, outputs, and operations. Next, you will examine data from an RSS feed to view its underlying XML, and build a Popfly mashup to display it. Finally, you will learn a little bit about web services, and examine the XML that is generated when a Popfly block calls a web service.

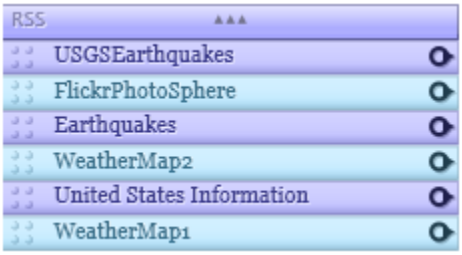
Prepared by Mark Frydenberg
Computer Information Systems Department
Bentley College, Waltham, MA
mfrydenberg@bentley.edu

Last Modified: 7/10/2008.

Under the Covers: A Beginner's Guide to Learning about XML using Popfly

Professor Popfly Mashups Referenced in this Lesson:

- ProfPopflyRSSFeed
  http://www.popfly.com/users/professorpopfly/ProfPopflyRSSFeed
- YahooTraffic XML (Block)
  http://www.popfly.com/users/professorpopfly/YahooTrafficXML
- YahooTrafficXMLMashup
  http://www.popfly.com/users/professorpopfly/YahooTrafficXMLMashup



Learning Outcomes

After completing this lesson, you should be able to:

- Describe differences between XML and HTML
- Identify different elements of an XML file
- Explain how Popfly uses XML to describe blocks
- Explain why XML is suited for data exchange
- Explain why comma separated files fall short in describing information
- Describe the relationship between XML and RSS
- Understand that web services provide information to other software applications

Overview

XML (eXtensible Markup Language) is a notation system designed to describe data. Because XML focuses on the meaning of data, many Web 2.0 companies that provide software services over the Internet use XML to represent the data that they make available. XML has become a standard language for describing and transferring data over the Internet.

In this lesson you will use Popfly to interact with XML in three different ways. First, you will learn to recognize XML by seeing how Popfly blocks use XML to describe their inputs, outputs, and operations. Next, you will examine data from an RSS feed to view its underlying XML, and make a Popfly mashup to display it. Finally, you will learn a little bit about web services, and examine the XML that is generated when a Popfly block calls a web service.

What is XML?

A simple way to represent information that a software application might process is by creating a text file containing comma separated values. For example, consider this sample data about different colleges:

```
Bentley College, 623, 576
Boston College, 667, 650
Northeastern University, 615, 596
Cornell University,,
```

While the college name is obvious, without markup it is not clear what the two numeric values represent. They could be the number of male and female students admitted this year, the average Math and Critical Reading SAT scores for the incoming freshman class, or the cost of student activity and technology fees on campus. One possible representation of this data in XML makes its meaning obvious:

```
<?xml version="1.0" encoding="utf-8"?>

<collegeList>
   <college name="Bentley College">
      <satMath>623</satMath>
      <satReading>576</satReading>
  </college>
  <college name="Boston College">
     <name>Boston College</name>
      <satMath>667</satMath>
      <satReading>650</satReading>
  </college>
  <college name="Northeastern University">
      <satMath>615</satMath>
      <satReading>596</satReading>
  </college>
  <college name="Cornell University" />
</collegeList>
```

Each item in the list is specified by an opening tag and a closing tag. Tags mark the beginning and end of a unit of information in an XML document. The entire document is contained within a single "root" element (in this case, <collegeList>). The closing tag begins with / in front of the tag name, and

XML requires that every opening tag have a corresponding closing tag. Each tag is specified in angled brackets < > and describes an element.

One of the rules of XML is that every starting tag needs to have a corresponding closing tag. In some cases, there may not be any information specified between a starting and ending tag, so a shorthand notation is used: for example, there are not SAT scores available for Cornell University, so the tag `<college name="Cornell University"></college>` may be abbreviated `<college name="Cornell University" />`. Documents that follow these and other rules are said to be well-formed.

Some elements, such as `<collegeList>` and `<college>` contain other elements, making XML a structured markup language. The `<college>` element also has a `name` attribute that gives additional information about the college itself.

Note that there is no web page associated with information represented in XML. HTML (Hypertext Markup Language) defines the tags used for formatting and displaying data, without information as to what the data means. A new standard for HTML, called XHTML (eXtensible Hypertext Markup Language) applies XML-style rules to HTML tags with the intent of requiring well-formed documents.

Example 1: Look inside a Popfly Block

This example describes how to look inside of a Popfly block at its underlying XML. Popfly uses XML to specify the inputs, operations, and objects associated with a Popfly block.

Open Popfly, and create a new block. Select any of the available blocks from the area on the left. Clicking on a block will load its description and code into the Popfly code area. This example uses the PhotoStack block. The PhotoStack block has one operation called addImage, which has two inputs: imageUrl and the title of the image.

Here is the XML to describe the specifications for the PhotoStack block:

Block Description | Block Code | Block Files | Generate from WSDL

```
<?xml version="1.0" encoding="utf-8"?>
<block xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://www.popfly.com/schemas/blockschema.xsd"
  class="PhotoStackClass" hasInitialize="true" type="display">
  <blockIconUrl>/content/components/icons/block.png</blockIconUrl>
  <suggest input="imageoutput"/>
  <operations>
    <operation name="addImage">
      <description>
        Adds an image to the PhotoStack.
      </description>
      <inputs>
        <input name="imageUrl" required="true" type="imageUrl">
          <description>A URL pointing to an image</description>
          <defaultValue>http://www.popfly.com/Images/community_globe.jpg</defa
          <constraints />
        </input>
        <input name="title" required="false" type="title">
          <description>Title of the image</description>
          <defaultValue>The Popfly Community!</defaultValue>
          <constraints />
        </input>
      </inputs>
    </operation>
  </operations>
  <objects />
</block>
```

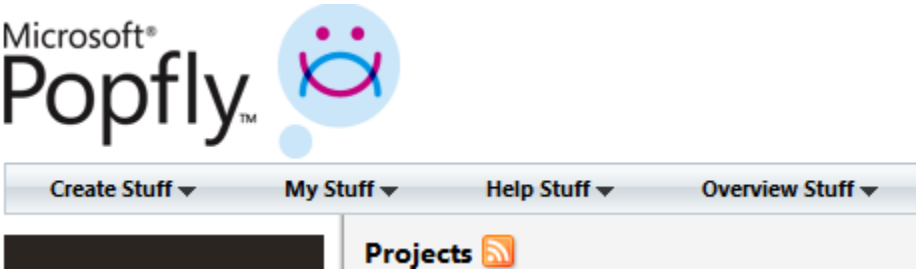
This is what you see when you open the PhotoStack block in the mashup editor:

Note the correspondence between the XML code in the Block Description and how the Popfly Mashup editor interprets and displays the XML data as the specification for the Popfly block. Each operation, input, default value, and description that is given in the Block Description appears as part of the block's specification.

Example 2: Look Inside an RSS Feed

RSS (Really Simple Syndication) feeds make use of XML to publish web content that is updated frequently. By providing the URL of an RSS feed to an aggregator program, the aggregator will periodically check the feed to see if there is anything new, and display the new items. This process gathers together data from many different web sites.

The  icon is commonly used on web sites to represent an RSS feed. You can click on the RSS icon on your Projects page in Popfly Mashup Creator to see the RSS feed and then view the source of that page to see the underlying XML for the feed. This URL is for Professor Popfly's Popfly Projects RSS feed: http://www.popfly.com/users/professorpopfly/projects.rss The URL for your project feed is similar; simply replace *professorpopfly* with your Popfly user name.



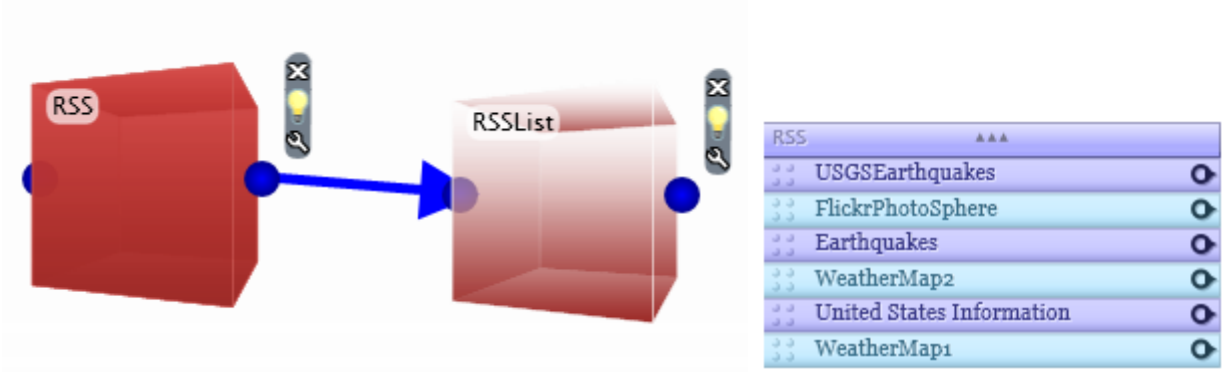
Every time you save a Popfly mashup, Popfly stores the mashup along with information about it in a database. Whenever you request your Popfly mash up RSS feed, this request runs a program on the Popfly server to query its database and generate an RSS feed file "on the fly." It is common for XML to be generated as the output of computer programs that query databases for specific information and then return that information in an XML marked-up format.

A portion of the resulting XML for a Popfly Projects RSS feed is shown below.

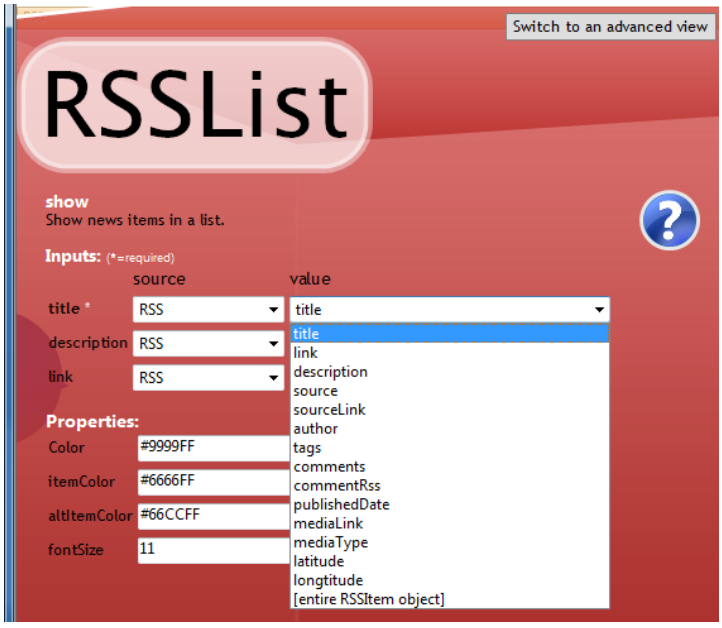
```
<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0">
  <channel>
    <title>professorpopfly's Popfly Feed</title>
    <link>http://www.popfly.com/users/professorpopfly</link>
    <description>Popfly projects shared by professorpopfly.
</description>
    <lastBuildDate>Tue, 27 May 2008 03:16:26 GMT</lastBuildDate>
    <item>
      <title>USGSEarthquakes</title>
      <link>
http://www.popfly.com/users/professorpopfly/USGSEarthquakes.details
</link>
      <description>Mashup of earthquake data from US Geological
Survey. &amp;#40;earthquake.usgs.gov&amp;#41;.  This one includes
little maps in the description, so it&amp;#39;s prettier than my other
Earthquake mashup.</description>
      <pubDate>Tue, 27 May 2008 03:16:26 GMT</pubDate>
    </item>
    <item>
      <title>FlickrPhotoSphere</title>
<link>http://www.popfly.com/users/professorpopfly/FlickrPhotoSphere.de
tails</link>
      <description>Display photos on a sphere.</description>
      <pubDate>Mon, 26 May 2008 05:43:30 GMT</pubDate>
    </item>
…
  </channel>
</rss>
```

Note that every RSS feed, whether it is describing Popfly mashups, blog posts, or the most recent photos you posted on flickr, all contain (at least) the <rss>, <channel>, <item>, <title>, <link>, <description>, and <pubDate> tags. These are the most common; the RSS specification also allows for some other tags to mark up content appropriate information. Because of this, it is easy to combine and process RSS feeds, as we will see in the RSS lesson. Popfly's RSS block returns these and other values from an RSS feed. To view these values, connect the RSS block to another block in Popfly, and specify values from the RSS block as inputs.

In this example, we create a simple mashup to display RSS feeds. The mashup uses the RSS block to obtain the items from a specified RSS feed, and RSSList to display the items using an attractive, dynamic interface. Use the getItems operation from the RSS block, and enter the URL for your Popfly mashups (or another) RSS feed. Connect that to an RSSList block to create a fancy RSS List display. Select customized colors for the RSSList if you wish.



Before previewing the mashup, look inside the RSSList block by clicking on its wrench. Notice that the items in the drop down list are exactly those items output from the RSS feed. Popfly matches up outputs of one block with inputs to another as best as it can, based on common field names and data types.



Example 3: Look inside Yahoo Traffic and other Web Services

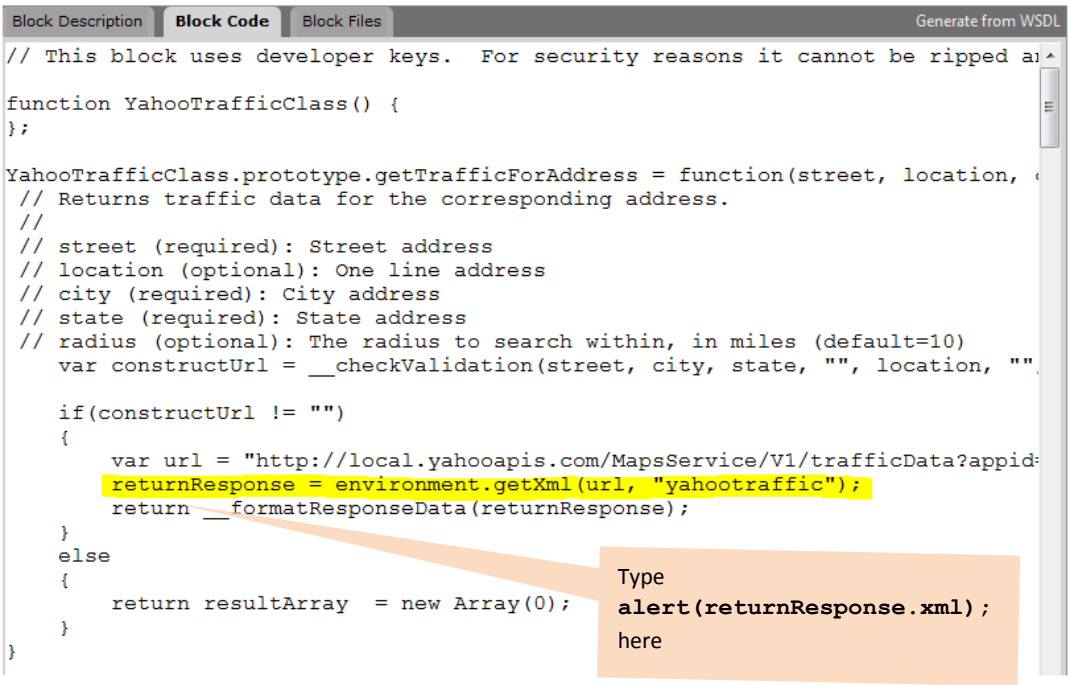
In their white paper entitled *Mashups, Interoperability and eInnovation* (http://cyber.law.harvard.edu/interop/pdfs/interop-mashups.pdf), Professors John Palfrey and Urs Gasser, of Harvard University's Berkman Center for Internet and Society, in Cambridge, MA, describe how web services are changing the way people use the World Wide Web:

> "More and more, the Web is being used not only as a portal for information but also for application-to-application communication. *Web services* is the general term for the interfaces available that connect different applications to each other, or any technology that supports machine-to-machine interaction over a network."

Many Popfly blocks access web services in order to obtain their information. A web service is a method or operation that one application (such as your Popfly mashup) can invoke on another remote computer located somewhere else on the Internet. The web service provides its output in XML format back to the application that called it. Web services provide a framework for applications and organizations to share data with each other over the Internet.

In this example you will see how to examine the XML data returned from a web service. This example makes use of the Yahoo Traffic block in Popfly. Yahoo Traffic is a web service provided by Yahoo (http://developer.yahoo.com/traffic/). The Popfly mashup that we are about to build (running on a Microsoft web server) will call upon the Yahoo service (running on servers at yahoo.com) to provide it with the traffic data.

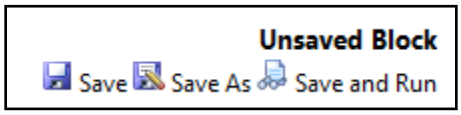
Open Popfly and create a new block. Drag the Yahoo Traffic block into the design area, and notice that its Block Description will appear. Click on the *Block Code* tab.



Look for a line of JavaScript that contains the words `environment.getXml(…)`. This is the statement that Popfly uses to get the actual XML data from the web service. To display the XML that is returned, you need to add one line of JavaScript below the line containing the call to `environment.getXml(…)` to display the raw XML in a pop-up alert box.

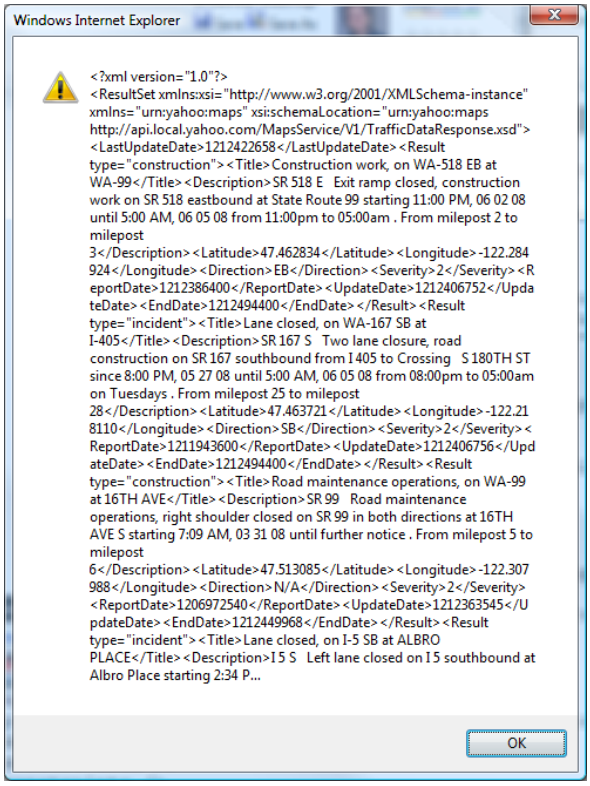
```
alert(returnResponse.xml);
```

In this block, `returnResponse` is the name of the variable containing the XML response. Note that other blocks may use differently named variables before the = sign. Be sure to spell the variable name correctly, using upper and lower case letters in exactly the same way as it appears in the source code line from which you are copying.

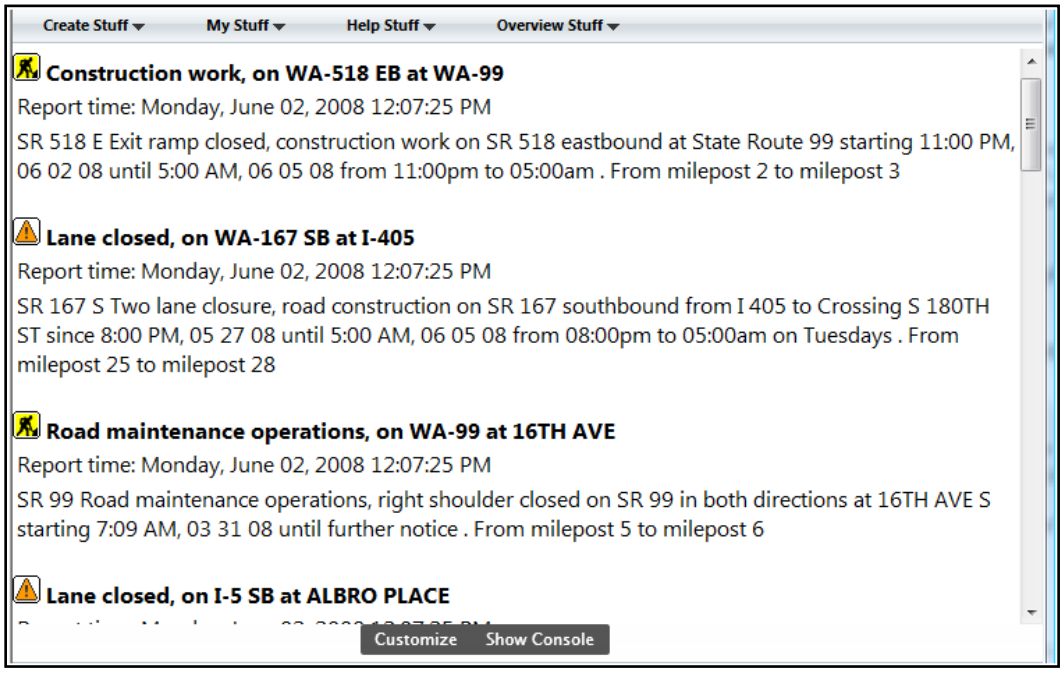


Click Save and Run. Give your block a name. (ProfessorPopfly named this block YahooTrafficXML). A new browser window running the Popfly Mashup Creator will appear containing your new block. Click Preview to run the block as a mashup. An alert box will appear in which you can see the XML for the data from this web service. If you don't see the XML code in an alert box, it means probably you didn't type in the alert statement correctly.

> Note: Internet Explorer 7 only displays a portion of the XML in the alert box. Firefox will open a larger alert box to display all of the XML if necessary.



Click OK on the alert box after you've examined the XML. When the mashup completes, look at its output in your browser. What is different between the XML data shown in the alert box and the output of the mashup as shown in the browser?



Popfly PopQuiz

1. Think of some things that you interact with every day (for example, favorite TV shows, a recipe, or your friends). On paper, or using a text editor such as Notepad, create an XML file that shows how you might represent this data in XML.
2. How does the XML in the block description tab of a Popfly block describe the block?
3. What XML tags describe the entries in an RSS feed?
4. In the Yahoo Traffic web services mashup, what is different between the XML data shown in the alert box and the output of the mashup as shown in the browser? Why?
5. Mashups are sometimes called "distributed applications" because the data and the processing of that data originate from, and occur on, different computers connected over the Internet. How does XML facilitate sharing of data across these different systems?

Try It in Popfly

Modify or build these Popfly mashups to demonstrate your understanding of this lesson. The more ducks, the bigger the challenge.

Open Popfly, select Create a Block, and select any Popfly block of interest. Look at the XML that specifies its Block Description.

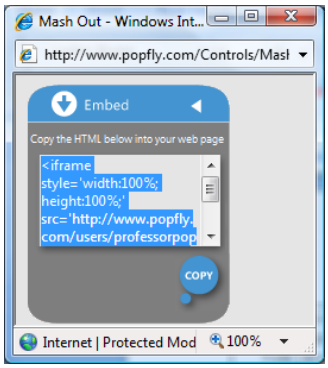
Identify the root element, an element that has an attribute, an element that contains another element, and possibly an empty element (with no content.)

When you are signed in to Popfly, click on the RSS icon on the Projects page (http://www.popfly.com/Profile/MyProjects.aspx) to display the RSS feed for your Popfly projects in your browser. Then right-click on View Source to examine the XML that represents the RSS Feed for your Popfly Projects.

Copy the URL between one of the <link> tags and paste it into your browser. To which Popfly page does it take you?

Build the RSS List Display mashup described in this lesson using your Popfly Mashups feed or another favorite RSS feed. Add custom HTML to give an appropriate title to your list.

To embed this mashup on your blog or web site, click *MashOut* for this project on your Popfly projects page. Choose *Embed* and copy the HTML code, then paste it in the appropriate place on your blog or web site.




```
<iframe style='width:100%; height:100%;'
src='http://www.popfly.com/users/profes
sorpopfly/ProfPopflyRSSFeed.small'
frameborder='no'
allowtransparency='true'></iframe>
```

Tip: Change the 100% for width and height to 150px and 200px (the px means pixels) or another appropriate size so it fits nicely in the side bar of your blog or web site.

Select a Popfly block that you think uses a web service to obtain its data. News and information blocks, stocks, and GeoCoding (such as YahooGeoCoding or GeoNames) are safe bets to try.

Repeat the Web Services exercise to add an alert box within the Block Code to display an alert box in which you can view the XML that the web service returns.

Add the statement `alert(____.xml);` filling the blank with the name of the variable before the = sign on the `environment.getXml` line of code in the block you have chosen.

Be sure that the block's default inputs provide enough information for the block to run as a mashup (or specify suitable defaults so that it runs). Run the mashup containing your new block to examine the raw XML obtained from a third-party web service. Compare the XML with the output of the mashup.

Learn More about XML

- XML Tutorial (http://www.w3schools.com/xml/default.asp)

  This tutorial describes the basics of XML, some ways XML can be used, and how to create, view, validate, and process XML documents.

- RSS in Plain English (http://blip.tv/file/205570/)

  An entertaining and informative video. What else can we say?

- RSS http://en.wikipedia.org/wiki/RSS_(file_format)

  This Wikipedia article describes the origin and formats for RSS feeds, and some applications that make use of them.

- Mashups, Interoperability and eInnovation (http://cyber.law.harvard.edu/interop/pdfs/interop-mashups.pdf)

  This whitepaper gives an overview of mashups and web services, and describes why they have become popular constructs in recent years. It also addresses some of the technological and social issues related to creating mashups.

- Web Services (http://www.ignyte.com/services-web-services.html)

  This web page describes web services and the web service used in Popfly's LocalMovies block.