

Basic Authentication

It takes a username and password, Base64 encodes them, and writes the resulting value to a variable. The resulting value is in the form of Base64Encoded String.

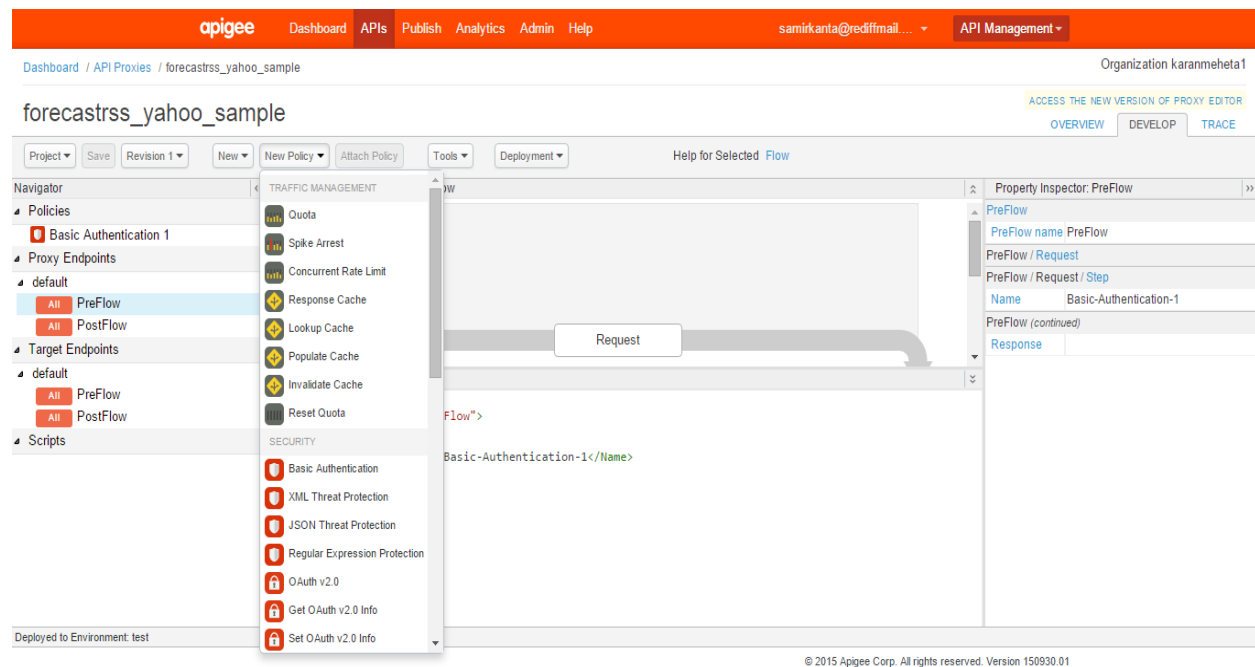
- Encode: Base64 encodes a username and password stored in variables
- Decode: Decodes the username and password from a Base64 encoded string

Here we used the sample yahoo weather URL which we used for Basic Authentication-


<http://weather.yahooapis.com/forecastrss?w=12896>

Step-1

First we need to select Basic Authentication From the Apigee Edge Console-



Step-2

New Policy:  Basic Authentication

Policy Display Name

Basic Authentication 2

Policy Name

Basic-Authentication-2

Attach Policy

☒

Flow

Flow PreFlow, Proxy Endpoint default ▼

Segment

☒ Request
☐ Response

Cancel


Add

- Give any name in the place of Policy Display Name.
- Same for Policy Name.
- Select the flow as per your requirement.
- Select the Segment as well.

Step-3

This is the default code we will gate after selecting Basic Authentication Policy.

Map: Endpoint default, Flow PreFlow



Basic Auth...

1

Request

App

Code: Basic-Authentication-1

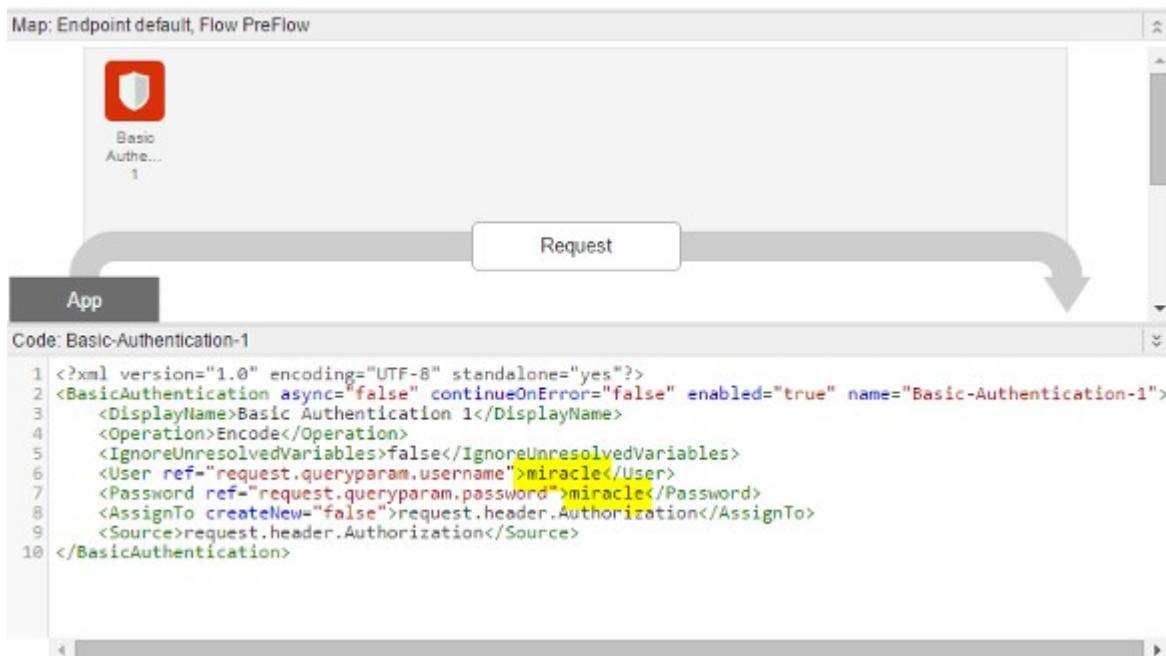
```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <BasicAuthentication async="false" continueOnError="false" enabled="true" name="Basic-Authentication-1">
3   <DisplayName>Basic Authentication 1</DisplayName>
4   <Operation>Encode</Operation>
5   <IgnoreUnresolvedVariables>false</IgnoreUnresolvedVariables>
6   <User ref="request.queryparam.username">miracle</User>
7   <Password ref="request.queryparam.password">miracle</Password>
8   <AssignTo createNew="false">request.header.Authorization</AssignTo>
9   <Source>request.header.Authorization</Source>
10 </BasicAuthentication>

```

Step-4

Give your desired Username and Password in the code.

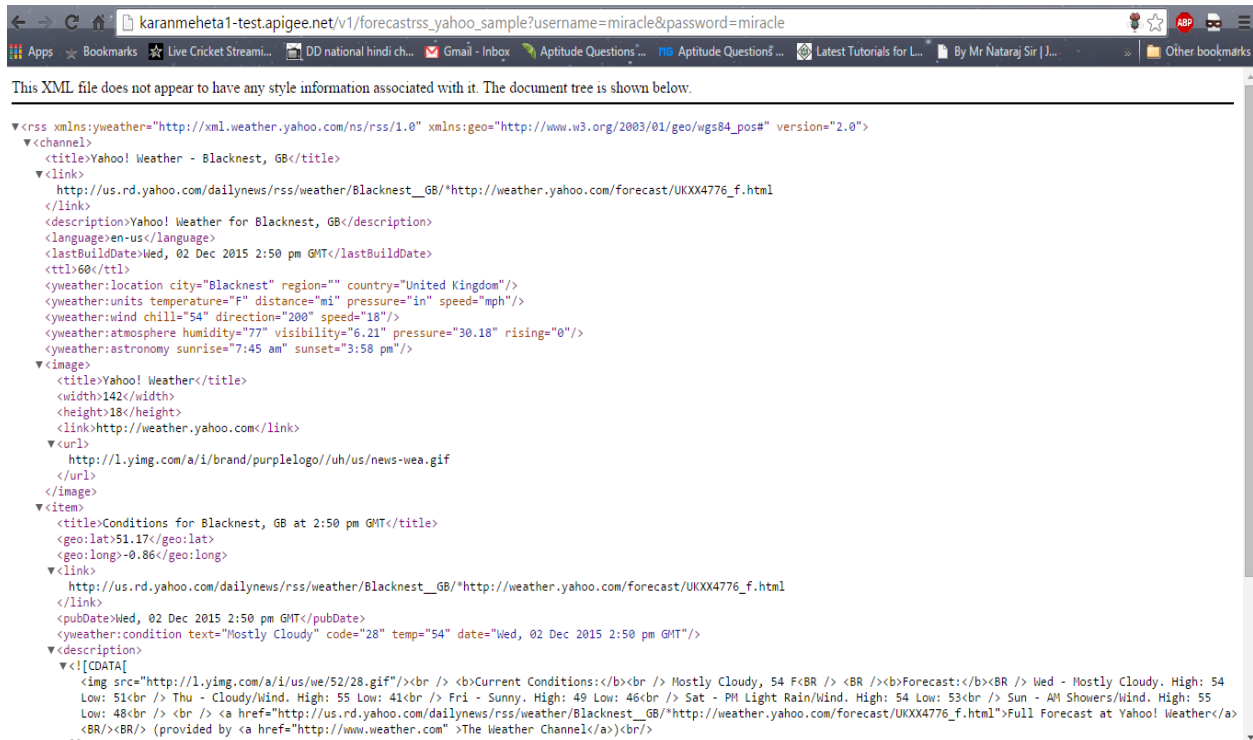


- Here we gave Username as-miracle and Password as-miracle

Step-5

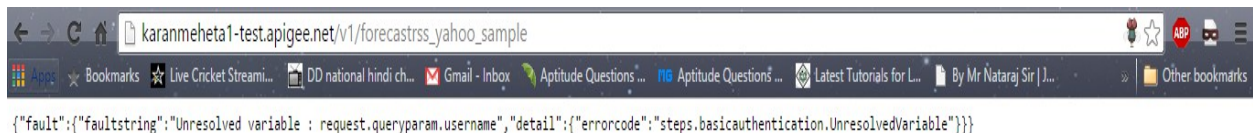
We need to pass username and password in the browser/Api Console-

http://karanmeheta1-test.apigee.net/v1/forecastrss_yahoo_sample?username=miracle&password=miracle



```
<?xml version="1.0" encoding="UTF-8" ?>
<rss xmlns:yweather="http://xml.weather.yahoo.com/ns/rss/1.0" xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#" version="2.0">
  <channel>
    <title>Yahoo! Weather - Blacknest, GB</title>
    <link>http://us.rd.yahoo.com/dailynews/rss/weather/Blacknest_GB/*http://weather.yahoo.com/forecast/UKXX4776_f.html</link>
    <description>Yahoo! Weather for Blacknest, GB</description>
    <language>en-us</language>
    <lastBuildDate>Wed, 02 Dec 2015 2:50 pm GMT</lastBuildDate>
    <ttl>60</ttl>
    <yweather:location city="Blacknest" region="" country="United Kingdom"/>
    <yweather:units temperature="F" distance="mi" pressure="in" speed="mph"/>
    <yweather:wind chill="54" direction="200" speed="18"/>
    <yweather:atmosphere humidity="77" visibility="6.21" pressure="30.18" rising="0"/>
    <yweather:astronomy sunrise="7:45 am" sunset="3:58 pm"/>
    <image>
      <title>Yahoo! Weather</title>
      <width>142</width>
      <height>18</height>
      <link>http://weather.yahoo.com</link>
    </image>
    <item>
      <title>Conditions for Blacknest, GB at 2:50 pm GMT</title>
      <geo:lat>51.17</geo:lat>
      <geo:long>-0.86</geo:long>
      <link>http://us.rd.yahoo.com/dailynews/rss/weather/Blacknest_GB/*http://weather.yahoo.com/forecast/UKXX4776_f.html</link>
      <pubDate>Wed, 02 Dec 2015 2:50 pm GMT</pubDate>
      <yweather:condition text="Mostly Cloudy" code="28" temp="54" date="Wed, 02 Dec 2015 2:50 pm GMT"/>
      <description>
        <![CDATA[
          <br />
          <b>Current Conditions:</b><br />
          Mostly Cloudy, 54 F<br />
          <b>Forecast:</b><br />
          Wed - Mostly Cloudy. High: 54 Low: 51<br />
          Thu - Cloudy/Wind. High: 55 Low: 41<br />
          Fri - Sunny. High: 49 Low: 46<br />
          Sat - PM Light Rain/Wind. High: 54 Low: 53<br />
          Sun - AM Showers/Wind. High: 55 Low: 48<br />
          <br />
          <a href="http://us.rd.yahoo.com/dailynews/rss/weather/Blacknest_GB/*http://weather.yahoo.com/forecast/UKXX4776_f.html">Full Forecast at Yahoo! Weather</a>
          <br />
          (provided by <a href="http://www.weather.com">The Weather Channel</a>)<br />
        ]]>
      </description>
    </item>
  </channel>
</rss>
```

If we won't pass the username and password then we will get the error message like this-



```
{ "fault": { "faultstring": "Unresolved variable : request.queryparam.username", "detail": { "errorCode": "steps.basicauthentication.UnresolvedVariable" } } }
```

OAuth 2.0

OAuth is an Open Authentication Framework, enables a third-party application to obtain limited access to an HTTP service.

OAuth allows notifying a resource provider (e.g. Twitter) that the resource owner (e.g. you) grants permission to a third-party (e.g. a Twitter Application) access to their information (e.g. the list of your tweets).

Step-1

For providing security to the Applications in Apigee Edge, we are using OAuth 2.0 policy.

To achieve this we will use Postman (It is an Application where we can give the I/P and view the results).

As discussed earlier we need to generate **Consumer Key** and **Consumer Secret** by going into Developer Apps also by registering our Products and API Proxy.

- There is one default API Proxy called “**oauth**” is there in Apigee Edge Console by default soon after you login for the first time into Apigee Edge Console.

You have to add one Assign Message Policy into it, then the complete code looks like this-

The screenshot shows the Apigee Edge console interface. The top navigation bar includes 'Dashboard / API Proxies / oauth' and the organization 'samirkantanka'. The main area is titled 'oauth' and shows the 'Add_Post' policy selected in the 'Policies' section. The 'Code: Add_Post' section displays the following XML code:

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <AssignMessage async="false" continueOnError="false" enabled="true" name="Add_Post">
3   <DisplayName>Add_Post</DisplayName>
4
5   <Set>
6
7     <Verb>POST</Verb>
8     <Payload contentType="text/xml">
9
10      <client_id>HjmJReFnWWzh4AoKPDw8cjZnbF4GY4iW</client_id>
11      <client_secret>X0jVA0hoya0ZYXcr</client_secret>
12
13    </Payload>
14
15    </Set>
16
17    <AssignTo createNew="false" transport="http" type="request"/>
18  </AssignMessage>
```

The right sidebar shows the 'Property Inspector' for the 'Add_Post' policy, with the following configuration:

AssignMessage	
AssignMessage async	false
AssignMessage continue...	false
AssignMessage enabled	true
AssignMessage name	Add_Post
DisplayName	Add_Post
AssignMessage / Set	
Verb	POST
Payload contentType	text/xml
Payload	<client_id>HjmJReFn...
AssignMessage (continued)	
AssignTo createNew	false
AssignTo transport	http
AssignTo type	request
AssignTo	

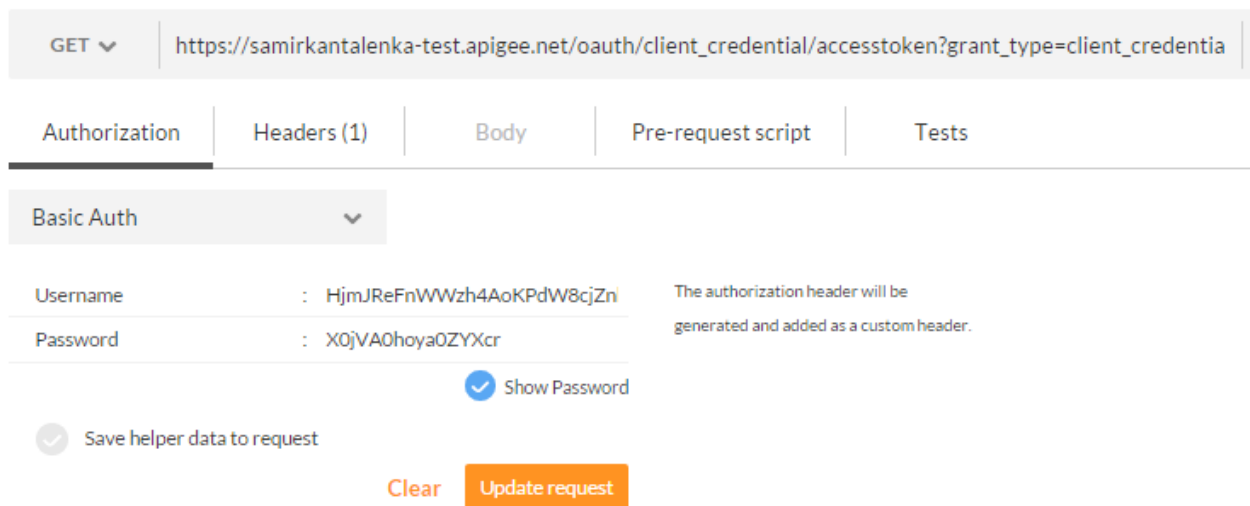
Complete Code-

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<AssignMessage async="false" continueOnError="false" enabled="true" name="Add_Post">
<DisplayName>Add_Post</DisplayName>
<Set>
<Verb>POST</Verb>
<Payload contentType="text/xml">
<client_id>HjmJReFnWWzh4AoKPDw8cjZnbF4GY4iW</client_id>
<client_secret>X0jVA0hoya0ZYXcr</client_secret>
```

```
</Payload>
</Set>
<AssignTo createNew="false" transport="http" type="request"/>
</AssignMessage>
```

The **Consumer Key** and **Consumer Secret** has to be mentioned in the code as shown in above code.

- Open Postman go to Authorization and Select Basic Auth form the drop down list and there give the **Consumer Key** in the place of Username and **Consumer Secret** in the place of Password as shown bellow-



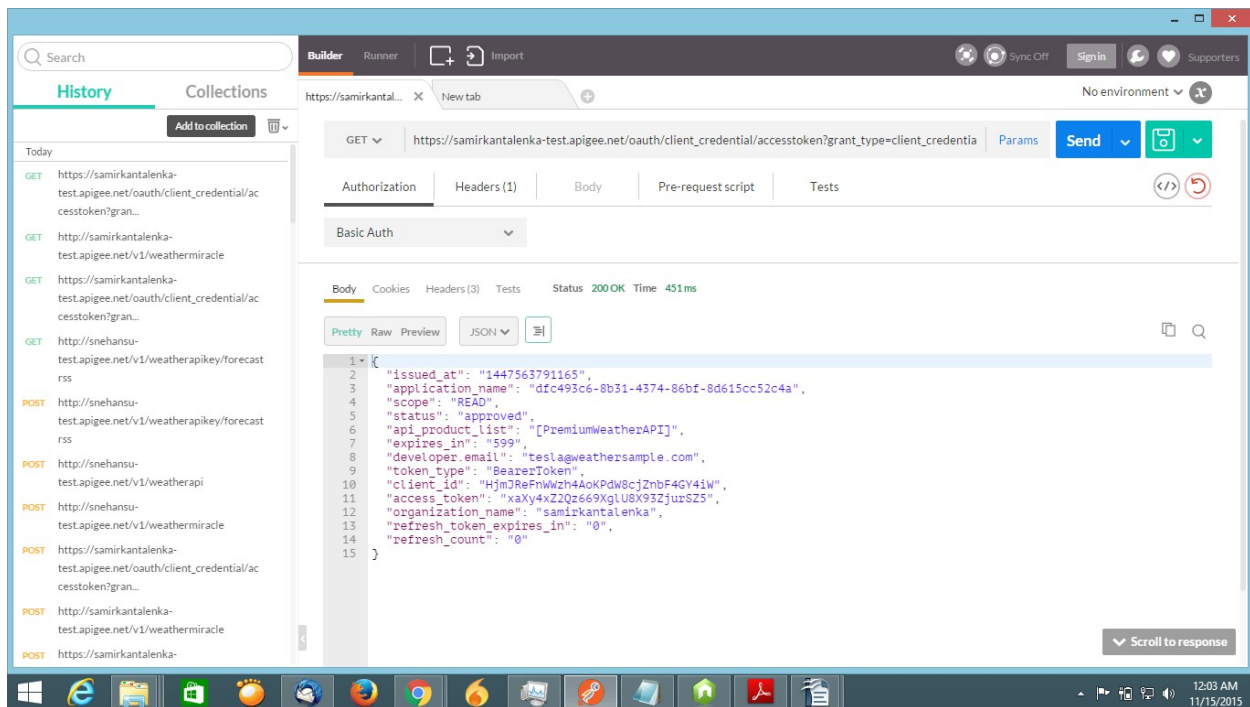
The screenshot shows the Postman interface with the Authorization tab selected. The URL bar at the top contains "GET" and "https://samirkantalenka-test.apigee.net/oauth/client_credential/accesstoken?grant_type=client_credentia". Below the URL bar, the Authorization tab is active, showing a dropdown menu set to "Basic Auth". The Username field is populated with "HjmJReFnWWzh4AoKPdW8cjZn" and the Password field is populated with "XQjVA0hoyaOZYXcr". A "Show Password" button is visible next to the password field. Below the fields, there is a "Save helper data to request" checkbox, a "Clear" button, and an "Update request" button. A note on the right states: "The authorization header will be generated and added as a custom header."

Pass the request URL as "GET" method and the URL will be-

https://samirkantalenka-test.apigee.net/oauth/client_credential/accesstoken?grant_type=client_credentials

Step-2

- After Passing the Username and Password from the Postman, Postman will generate one **"Access Token"**.



```
{
  "issued_at": "1449858257903",
  "application_name": "dfc493c6-8b31-4374-86bf-8d615cc52c4a",
  "scope": "READ",
  "status": "approved",
  "api_product_list": "[PremiumWeatherAPI]",
  "expires_in": "599",
  "developer_email": "tesla@weathersample.com",
  "token_type": "BearerToken",
  "client_id": "HjmJReFnWzh4AoKpDw8cjZnbF4GY4iW",
  "access_token": "SIYRSq4CCcC0jNmD8zUmL0Tt5h4s",
  "organization_name": "samirkantalenka",
  "refresh_token_expires_in": "0",
  "refresh_count": "0"
}
```

Step-3

- We need to Pass that Access token from the Postman as Headers- Headers->In the Header field Pass Header as (Authorization) and Value as(Bearer:Pass the access token value)
i.e-"Bearer:SIYRSq4CCcC0jNmD8zUmL0Tt5h4s".

The URL we need Pass is-

<http://samirkantalenka-test.apigee.net/v1/weathermiracle>

The output we will get like this-

The screenshot displays the Swagger UI interface for the API endpoint `http://samirkantalenka-test.apigee.net/v1/weathermiracle`. The interface is divided into several sections:

- History:** A list of recent API calls on the left side, including GET and POST requests to various endpoints.
- Builder:** The main area for constructing the API call. It shows the method `GET` and the URL `http://samirkantalenka-test.apigee.net/v1/weathermiracle`.
- Authorization:** A section for setting authentication, currently showing a Bearer token: `Bearer 81JrLWWQWZ2CGeSImmmWMahWQyz`.
- Body:** A section for the request body, currently empty.
- Response:** A section showing the response body, which is an XML document. The status is `200 OK` and the time taken is `823 ms`.

The XML response is as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<rss version="2.0" xmlns:yweather="http://xml.weather.yahoo.com/ns/rss/1.0" xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#">
  <channel>
    <title>Yahoo! Weather - Blacknest, GB</title>
    <link>http://us.rd.yahoo.com/dailynews/rss/weather/Blacknest__GB/*http://weather.yahoo.com/forecast/UKXX4776_f.html</link>
    <description>Yahoo! Weather for Blacknest, GB</description>
    <language>en-us</language>
    <lastBuildDate>Tue, 10 Nov 2015 6:50 am GMT</lastBuildDate>
    <ttl>60</ttl>
    <yweather:location city="Blacknest" region="" country="United Kingdom"/>
    <yweather:units temperature="F" distance="mi" pressure="in" speed="mph"/>
    <yweather:wind chill="57" direction="220" speed="16" />
    <yweather:atmosphere humidity="94" visibility="6.21" pressure="30.15" rising="0" />
    <yweather:astronomy sunrise="7:11 am" sunset="4:22 pm"/>
    <image>
      <title>Yahoo! Weather</title>
      <width>142</width>
      <height>18</height>
      <Link>http://weather.yahoo.com</Link>
    </image>
  </channel>
</rss>
```