Name: Shant Ahuja Date: 5-9-20

Lab 4 – Web Services and JSON

Objectives

Part 1 – Web Services Introduction

Part 2 – Functional Requirements

Part 3 – Example User Interface

Reflection

Background/Scenario

With offline computers quickly becoming the minority, being able to write programs that effectively communicate with the Internet is a very important skill to have. In the past, most programs acquire their data from manual input. Now, most applications receive their data from web services, whether from internet third party sources or intranet business servers.

In this lab, you will be consuming, caching, and displaying data from a couple of different web services. The web services you will be accessing data from will be Reddit and Bing.

Required Resources

- Visual Studio
- Sample Photos

Part 1: Web Services Introduction

There is an unfathomable amount of content on the Internet. While most content can be freely viewed via your web browser, most web service APIs require authentication in order to access the data. I chose these services for you to implement in this lab because they work without requiring you to sign up for login credentials and are straightforward to implement.

I highly encourage you to explore the internet and see if you can find more APIs that you are interested in accessing. https://github.com/toddmotto/public-apis is an extensive list of some public APIs that have a wide range of data to access. From cat photos to weather, I'm sure you will find many interesting APIs to poke around with there.

Any API URLs mentioned in the lab can be viewed directly from your browser. Just simply paste the URL and you can immediately see the data returned from the API. Note, not all APIs return formatted and indented JSON so it may not always look nice and pretty in the browser but your code will have no problem parsing it either way.

Bing Photo of the Day – http://www.bing.com/

When viewing the Bing homepage, there is a new photo that is featured as the background that changes every day. If you are using Windows 10, you may also notice the same thing happening with your lock screen.

The URL to access the API is:

```
http://www.bing.com/HPImageArchive.aspx?format=js&idx=0&n=8&mkt=en-US
```

There are a few customizable parameters you can change, most notably the idx and the n parameters. The idx parameter is the start index of the photo, 0 being the most recent. The n parameter is how many to return in the query, in this case 8 photos will be returned.

Reddit – http://www.reddit.com/

Reddit is a website where users collectively post links and content from the Internet. All content is categorized into collections known as subreddits. If you view the homepage, you will see a curated collection of some of the more popular subreddits' contents.

The Reddit API is very easy to use. If you are at any page, you can just simply add ".json" to the end of the URL before the parameters, if any. For example, to get the json object representing the homepage http://www.reddit.com/, change it to http://www.reddit.com/.json and you will see all the data from the homepage.

For this lab, you will be accessing the /r/EarthPorn subreddit. Although the name of the subreddit hints at inappropriate content, there is nothing inappropriate about this subreddit's content. But be warned, Reddit is a user generated content site and there is an equal amount of inappropriate content to match the normal content. Be aware of what you browse.

The /r/EarthPorn subreddit contains posts of scenic landscape shots from around the world. Here is the URL to the API:

```
http://www.reddit.com/r/EarthPorn/top/.json
```

The "top" part of the URL denotes that it will return the highest upvoted posts from the past 24 hours. There are a lot of other parameters you can specify but for simplicity, this is what we will use for the lab. If you are interested in more of what the API has to offer, check out Reddit's API documentation here: https://www.reddit.com/dev/api/

Note: Most web service APIs have rate limits imposed on them. Rate limits are limits on how many times you can access the API before it locks you out. If you debug often, you may want to save the API response to a file and simulate querying the API by reading back the saved data.

Part 2: Functional Requirements

Summary

The goal of this application is to acquire photo information from various sources and display the photos and various metadata in a meaningful way to the user. The application will also be versatile in loading from sources via a plugin architecture.

Technical Requirements

- The user interface must be programmed in WPF.
- All non-static content displayed in the user interface must be initialized with data binding.
- The application must implement a plugin interface for photo acquisition.
  - All photo content retrieval code must be isolated to individual plugins. For example, a separate dll for returning Reddit content and a separate dll for Bing content.
  - Plugins must be stored in a Plugins subdirectory at the application root.
  - The application must dynamically find and load the plugins.
  - The application and plugins must share a common type. (Refer to Lab 1)

User Interface Requirements

- The user interface must provide:
  - A listing of all plugins detected.
  - A means of displaying the plugin's photos.
  - A thumbnail gallery-like view of the photos.

- A means to display the full resolution of the photo.
- Metadata about the photo.
  - Different web services will provide different metadata. The user interface must be able to display different metadata provided by the API. For example, the Reddit API has upvote count whereas the Bing API has copyright.

Photo Source Plugin Requirements

- Bing Photo of the Day
  - Display the most recent 8 photos.
  - Display the following metadata:
    - Image filename
    - Copyright
    - URL
- Reddit /r/EarthPorn
  - Display the top 25 images from the past 24 hours.
  - Display the following metadata:
    - Post title
    - Username of submitter
    - Post time and date
    - Number of upvotes
    - URL of the post permalink
- File System
  - Display all the photos from a directory. (Can be hard coded. See extra credit)
  - Display the following metadata:
    - Filename
    - Date taken
    - Path of the file
- All Photos
  - Height and Width

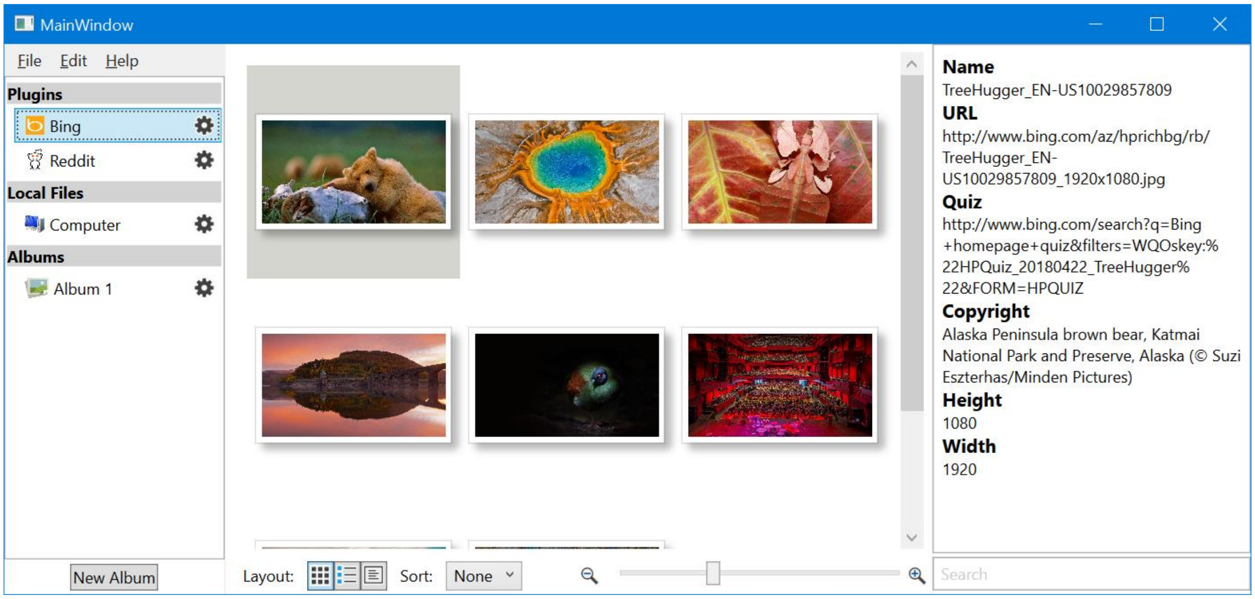
Extra Credit (Optional)

- Provide a way to customize plugin data
  - Bing
    - Number of photos
    - Start index
  - Reddit
    - Number of photos
    - Start index

  - File System
    - Directory path to load the photos from
- Lazy load plugin data
  - Only load the plugin's photo data when the plugin data is requested by the user interface. The application should not make any web requests until the plugin data is requested for the first time.
- Photo EXIF data
  - If the photo contains EXIF metadata, parse and load the metadata. (E.g. exposure, GPS coordinates, camera model, etc…)
- Create a new plugin
  - Find another web api and create a plugin that consumes the web API. (E.g. Twitter, Instagram, etc…)

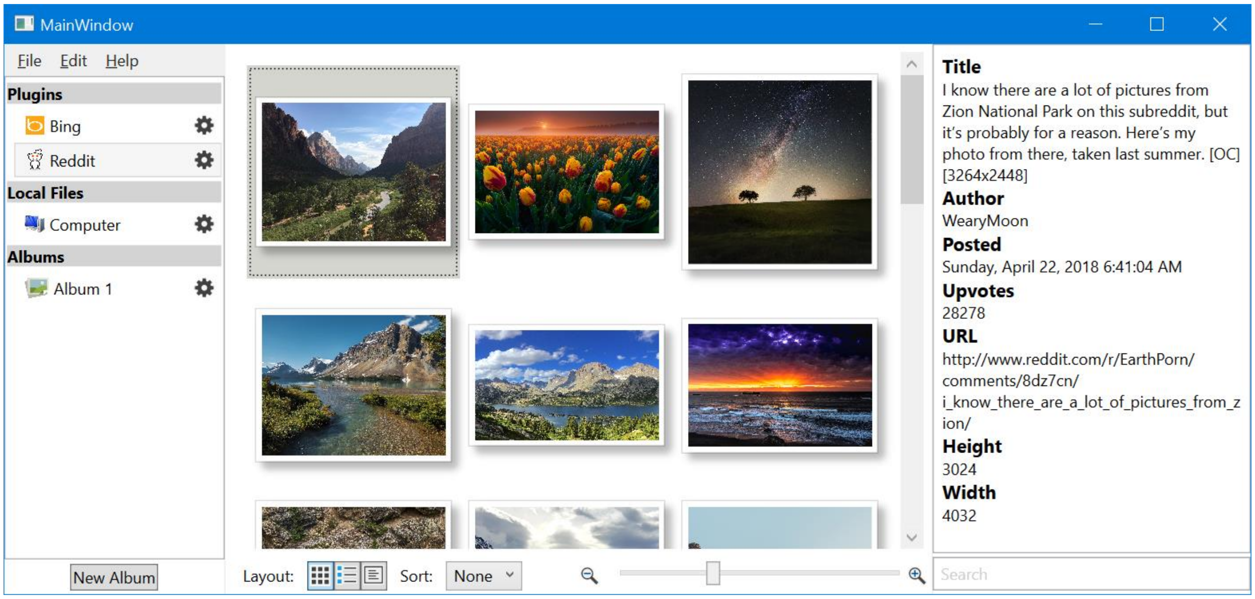
Part 3: User Interface Example

Again, here is an example of a user interface that meets all the requirements stated above. There are other features that the example user interface has that were not described by the requirements to give some example on experimentation with user interface design.

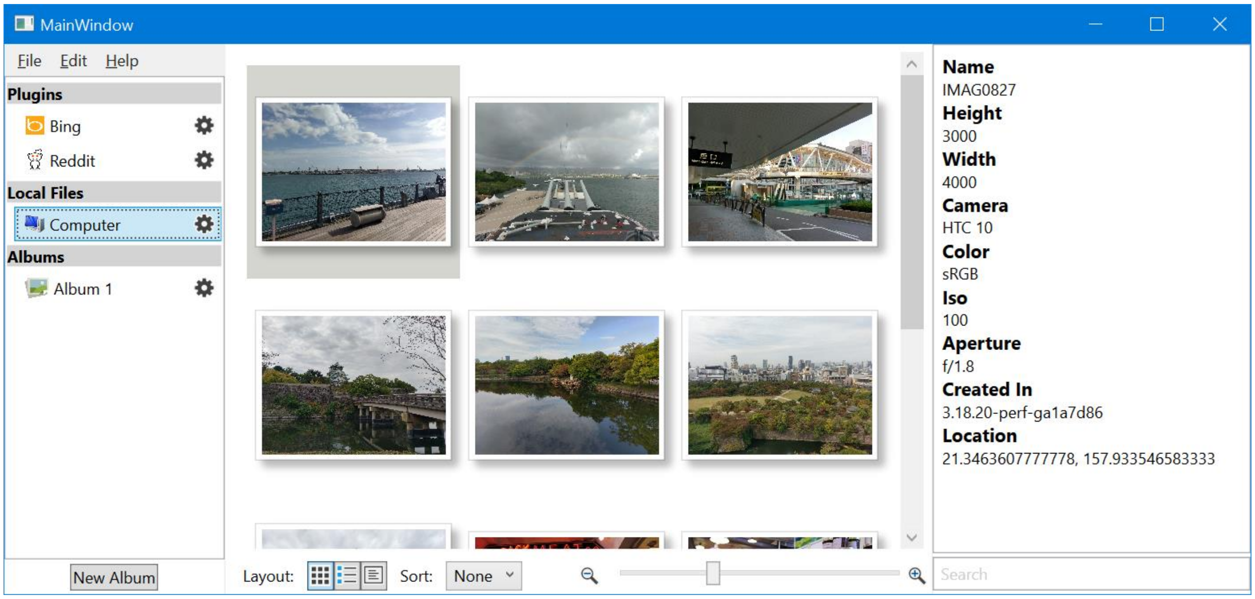


Sort: None

Name
TreeHugger_EN-US10029857809
URL
http://www.bing.com/az/hprichbg/rb/TreeHugger_EN-US10029857809_1920x1080.jpg
Quiz
http://www.bing.com/search?q=Bing+homepage+quiz&filters=WQOskey:%22HPQuiz_20180422_TreeHugger%22&FORM=HPQUIZ
Copyright
Alaska Peninsula brown bear, Katmai National Park and Preserve, Alaska (© Suzi Eszterhas/Minden Pictures)
Height
1080
Width
1920



Title
I know there are a lot of pictures from Zion National Park on this subreddit, but it's probably for a reason. Here's my photo from there, taken last summer. [OC] [3264x2448]
Author
WearyMoon
Posted
Sunday, April 22, 2018 6:41:04 AM
Upvotes
28278
URL
http://www.reddit.com/r/EarthPorn/comments/8dz7cn/i_know_there_are_a_lot_of_pictures_from_zion/
Height
3024
Width
4032



Name
IMAG0827
Height
3000
Width
4000
Camera
HTC 10
Color
sRGB
Iso
100
Aperture
f/1.8
Created In
3.18.20-perf-ga1a7d86
Location
21.3463607777778, 157.933546583333

Reflection

1. Both the Bing and Reddit web services have options to support both XML and JSON by simply changing a parameter. (Try it for yourself) After working with Binary and XML from Lab 3 and now JSON from this lab, which format do you find easiest to work with? Do you prefer any one over the others?

   I like json just because I know it more

2. How did you access the JSON data that you deserialized from the web API? Did you deserialize the data into .NET classes or did you deserialze it into an object and reference the properties based on keys? Or did you even use some other method (dynamic?)

   deserialized into object and reference properties :D