

Module 3.1 Handout – Demystifying the Web

Feeds Examples – Flickr

Before starting the examples, go to: <https://www.flickr.com/services/feeds/>

- This page has documentation about how to use the Flickr public feed service.
- The page shows that the base URL to use the public feed service is:
`https://api.flickr.com/services/feeds/photos_public.gne`
- Think of this service as a way to issue queries to Flickr and get back a list of results.
- The service supports an array of optional parameters
- The general form is:
`baseURL?param1=value¶m2=value...¶mN=value`
(note that the first parameter is prefaced by `?`, but later parameters are separated by `&`)

Example #1: Flickr public feed

- Try this URL to see an unfiltered feed:
`https://api.flickr.com/services/feeds/photos_public.gne`
- Now add `?tags=uncsils` to filter to see only photos tagged with 'uncsils':
`https://api.flickr.com/services/feeds/photos_public.gne?tags=uncsils`
- Now add `&format=json` to change the output format to be JSON:
`https://api.flickr.com/services/feeds/photos_public.gne?tags=uncsils&format=json`
(note that the first parameter is prefaced by `?`, but later parameters are separated by `&`)
(also note that `49938518@N04` is the `author_id` for uncsils)
- Look at the JSON feed data – fields for `title`, `link`, `date_taken`, `description`, `author_id`, `tags`
- Copy and paste one of the URLs in a link field into a new browser window:
e.g.: `https://www.flickr.com/photos/uncsils/4818686971/`
- Consider the difference between these two queries to the service:
`https://api.flickr.com/services/feeds/photos_public.gne?tags=uncsils&format=json`
`https://api.flickr.com/services/feeds/photos_public.gne?tags=uncsils,epa&format=json`
- Now try to use the `tagmode` parameter to find pictures tagged with either `uncsils` or `unc`

Example #2: Flickr favorite photos feed

- Try this URL to see a feed of the public favorite photos for a user:
`https://api.flickr.com/services/feeds/photos_faves.gne?id=49938518@N04`
(uncsils has no favorites)

REST API Examples – Flickr

Before starting the examples, go to: <https://www.flickr.com/services/api/>

- This page has documentation about a wide range of Flickr API methods.
- Many of these requires a developer API key and use OAuth to allow an application to interact with Flickr on behalf of a user. For simplicity, we will use methods that do not require this.
- However, it is useful to understand an overview of OAuth (many online services use it):
<https://www.flickr.com/services/api/auth.oauth.html>

Example #1: Camera Brand models

- a) Review this API documentation:
<https://www.flickr.com/services/api/flickr.cameras.getBrandModels.html>
- b) Click the link at the bottom of the page for the “API Explorer”:
<https://www.flickr.com/services/api/explore/flickr.cameras.getBrandModels>
- c) Enter a camera brand in the value box (e.g., apple, samsung, canon)
- d) Select output JSON
- e) Click “Call method...” button
- f) Review the output listing of different camera models for the specified brand.
- g) Look at the URL at the bottom of the page – this is the REST call that was made
- h) Copy the URL and paste it into a new browser window

Example #2: Get Places

- a) Review this API documentation:
<https://www.flickr.com/services/api/flickr.places.find.html>
- b) Click the link at the bottom of the page for the “API Explorer”:
<https://www.flickr.com/services/api/explore/flickr.places.find>
- c) For the value, enter a place such as:
Chapel Hill, NC
Washington Monument
IAD
- d) Click “Call method...” button
- e) Some queries will return multiple “places”.

Example #3: Get Places for Tags

- a) Review this API documentation:
<https://www.flickr.com/services/api/flickr.places.placesForTags.html>
- b) Click the link at the bottom of the page for the “API Explorer”:
<https://www.flickr.com/services/api/explore/flickr.places.placesForTags>
- c) Enter 22 for place_type_id (22 is neighborhood, 7 is locality, 8 is region)
- d) Enter unc for tags
- e) Click “Call method...” button
- f) List will show different places that photos have been taken with that tag.
- g) Try other tags (e.g., mountain, snow)