Jonathan Roland

WEB 420 RESTful APIs

Instructor: Prof. Krasso

11/06/2020

Discussion 4.1

A collection is a type of resource that represents a grouping of other resources through linking. The linked individual resources can be referred to as items, entries, or members of the collection. These resources exist independently with their own URLs but are linked by reference in the collection. One way to think of a collection would be a grouping of attributes which describe a class of thing, or an individual thing. This does not have to be a total description of the thing’s attributes, but it may describe some facet of it by linking cumulative information.

The collection pattern describes a common pattern that emerges across domains. One of those patterns is Collection + JSON. Per the name, this pattern uses a JSON document as the linking format. This collection stores the protocols for HTTP GET requests to various attributes of an object. It follows a specific pattern which includes the following properties:

* **href**: the link to the collection itself.
* **items**: the members of the collection and their partial representations.
* **links**: resources that are external to, but related to, the collection.
* **queries**: methods for searching the collection.
* **template**: the method of adding a new item to the collection.

And earlier implementation of collections uses XML documents rather than JSON for representation. Specifically, the Atom file format used by AtomPub presents a competing collection pattern using XML documents. The structure differs with a wrapping XML element called a feed which contains multiple entries. The feed is analogous to a collection, and entries are analogous to items.

Atom documents are heavily used by Google, who demonstrate one of their core differences with Collection+JSON: extensibility. Atom-based APIs can be extended to define nearly anything, and Google uses this for custom APIs for spreadsheets, media, and map data.

Though collection patterns differ in implementation, they all achieve the same generic goals. They provide a way to coordinate API requests and representations into one cumulative resource. Every collection pattern must succeed in presenting options for GET, POST-to-Append, PUT, PATCH, and DELETE, as well as make provisions for paginating large collections.

Reference:

Richardson, L., Amundsen, M., & Ruby, S. (2013). *RESTful Web APIs: Services for a Changing World* (1st ed.). O’Reilly Media.