Jonathan Roland

WEB 420 RESTful APIs

Instructor: Prof. Krasso

11/13/2020

Discussion 5.1

* **How does a client application find and parse an entity-body?**

For a client application to parse the entity-body of a server response, the Content-Type header in an HTTP request is used. The client must tell the server how it expects to parse any response. Examples:

* Content-type: text/html
* Content-type: application/json
* Content-type: application/atom+xml

The client can discover what HTTP requests can be made next if the parsed response body defines hypermedia controls such as HTML.

* **Explain what profiles are and how they are related to application semantics.**

Just because the client can parse a hypermedia document response does not mean it can understand the semantics of that application or protocol. Profiles describe additional meaning which can be applied on top of the document for greater context and functionality.

For example, lets say HTML describes a restaurant: the basic concepts like it being a structure with entrances and exits, tables, chairs, a kitchen, waitstaff and a menu you can order from. But if you read a sign outside or a description of the restaurant on Yelp that told you it was a high end French restaurant, you can apply greater context: expectations of menu items, pricing, the need to place reservations well in advance, and very likely the need to know some French to place your order.

This additional information is a profile. It adds context on top of the basic format, adding an understanding of the additional layer of rules and expectations so you can make decisions accordingly. In a web context, this additional understanding is typically served to the client via an extension in the response’s Link attribute. It describes a ‘menu’ of application-specific options for HTTP requests.

* **What are application semantics?**

Application semantics are (typically) shortened human readable strings which represent significant associations to the application. They can be divided into two categories: link relations and semantic descriptors.

Link relations are strings that represent hypermedia controls. These controls cause state changes in the application. These can be words like ‘next’,’prev’,’top’ and so on. It is imperative for developers to document these strings rather than let them be a mystery, relying on the client’s judgement that it means what they think it means. It is also important to realize that these are semantic links to hypermedia controls, not links in the traditional ‘href’ sense.

Semantic descriptors are strings that reflect similarly named keys in data within the document body. In a web store example, using ‘brand’ in some context of an element in your (extended) hypermedia indicates that this element is representing the brand attribute of this item. This type of usage is a bit loose and may be best to avoid.

* **Using your own words, what does a machine-readable profile mean?**

Profiles and application semantics represent human readable extended API definitions beyond basic hypermedia. However, these definitions are not necessarily readily available for review, and would require quite a lot of digging on the part of the client to discover and contextualize each one. Using machine-readable profile formats like XMDP, ALPS, or JSON-LD, the server can provide the client automatic links to the correct definition of semantic usage in a profile. This allows the client to more quickly contextualize the profile options the API is presenting them and write their client to match.

Reference:

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