Discussion 4.1

Domain-specific design is a way to represent a problem so a computer what kind of problem it has encountered while at the same time remaining understandable to a human from a semantic point of view. For example, say, through your computer (Google Street View or something), you are at Caramel Street on 95th, and you have a choice to go north to Huntington Street or south to Berkshire Street. Domain-specific design means that when you choose to go north and send off your request, the server will know to put you into the resource that corresponds with your choice of north, that is, Huntington Street on 95th. This is done through link relations, which are simply strings that represent the type of relationship one resource has to another[[1]](#footnote-1). For example, on the computer over the Internet, the resource that represents Caramel Street at 95th has its own link, and the resource representing Huntington at 95th has a different link. The link relation from Caramel to Huntington would probably be labeled “north.” A human reading the API response for Caramel would see the word “north” and assume that meant that Huntington Street is north of Caramel Street, while the computer would see the Huntington Street link associated with “north,” so even though the computer may not know what the word north means to a human, it would know that if the human selected the link related to the word north it is to retrieve the Huntington Street resource.

A collection is a grouping of resources that focuses on links to those resources. A JSON Collection means that the collection is simply a grouping of JSON objects. And a generic JSON collection means that the the collection does not actually care what the group of JSON objects is[[2]](#footnote-2), whether it’s a list of winter coats on Amazon or an advanced cloud-based task system through a mobile application. Instead, you can add your own application semantics to the collection. Application semantics are simply the context of the resource provided. For example, one semantic Amazon might use in a collection is “size,” because a customer might want to get a list of coats in a specific size.

API pagination is where, when faced with a massive number of resources as results, the computer limits the results to a certain number and includes a link to get more of the resources in the results. This makes a response easier to handle. The usual example for this is when you have a 295 results from searching for a library book. The web page shows the first ten, and below that list is the the following: “1 2 3 4 5 Next”. Clicking on 2 will take you to the next ten results. Clicking a different number will take you to ten different resource results, while clicking Next will take you to the ten results that come after the ten you are viewing.

1. Kin Lane. (2017, Aug 11). Link Relation Types for APIs (article). Retrieved from <https://apievangelist.com/2017/08/11/link-relation-types-for-apis/> [↑](#footnote-ref-1)
2. Darrel Miller, et al. Designing Evolvable Web APIs with ASP.NET (book). Retrieved from <https://www.oreilly.com/library/view/designing-evolvable-web/9781449337919/ch06.html> [↑](#footnote-ref-2)