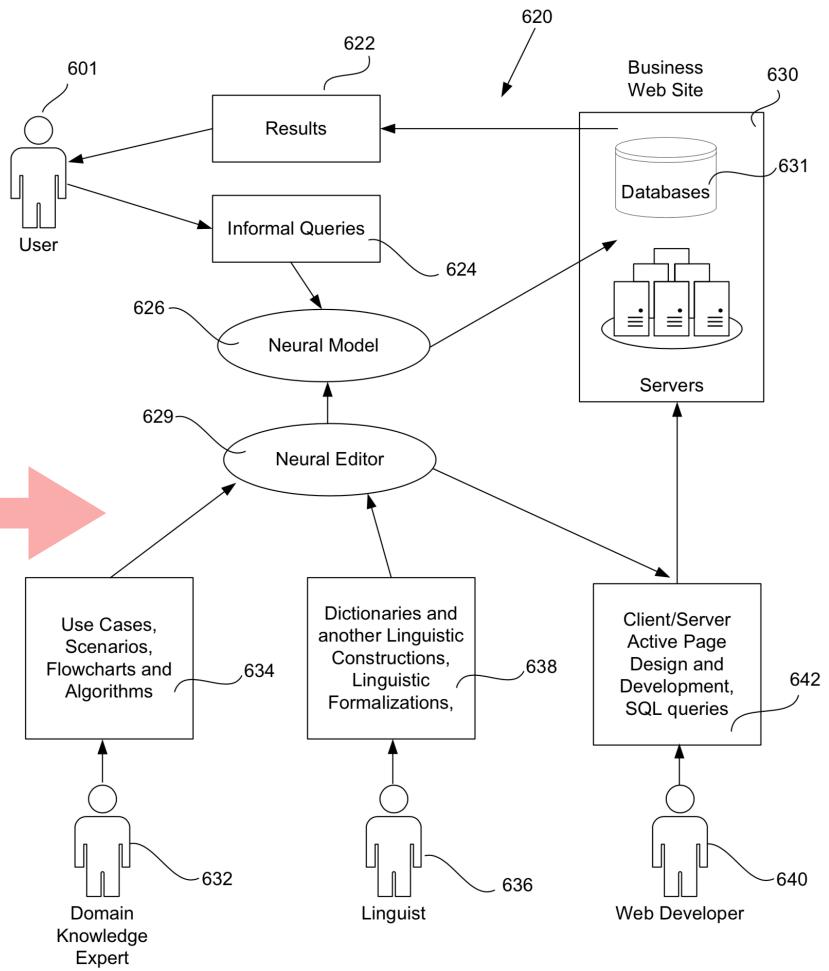
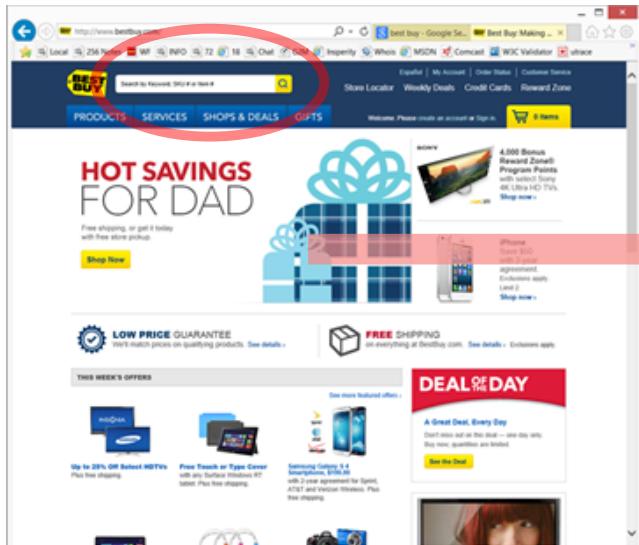


ChatBot in 7 Days

How to use Neural Networks to create Chatbot in a week

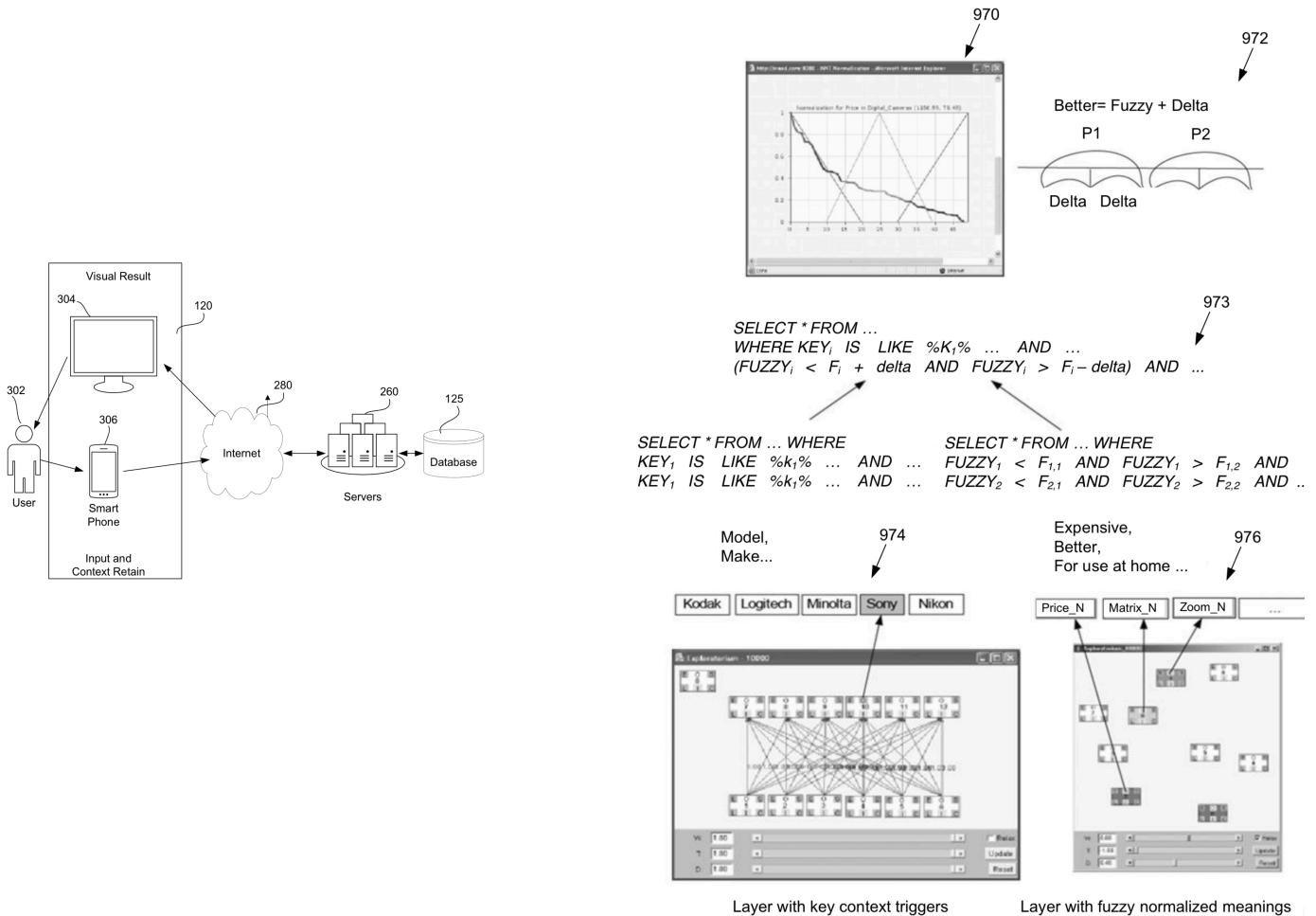
Development Diary: February 22, 2018 - February 28, 2018

Best Buy



Introduction

We will build a ChatBot from scratch, using patented SDK and in very generic development environment. The goal is to show how easy is the process of building interactive contextual search with Neural Networks toolkit. As an example we will use [BestBuy website](#) and its [Development API](#). The process will takes approximately 7 days, and we will keep the diary with daily report.



Day 1

Project preparation.

The very first step - registration on BestBuy Development site and obtaining API key. To do this we registered on <https://developer.bestbuy.com> and fill application form to obtain API Key. This Key required to send requests to BestBuys database.

Learning Best Buy API.

The next step is to learn and experiment with BestBuy API (BB API). It takes some time to understand the structure, component, formats of API. The most important for us is Query Builder - <http://bestbuyapis.github.io/bby-query-builder/#/productSearch> Most of the time we spent to understand the structure of JSON queries and responses.

For example, informal request - "Show me cameras around \$200 with the best reviews" can be reformulated in formal request:

The screenshot shows the Best Buy Query Builder interface. On the left, there's a sidebar with links for Products, Stores, Categories, Open Box, Recommendations, and Smart Lists. The main area has a title 'Products API' and a sub-section '1 Search for Products'. It includes a dropdown for 'Digital Cameras', a search bar 'Enter Keyword(s)', and three filter fields: 'Regular Price' (100), 'Regular Price' (300), and 'Customer Review Average' (4). Below this is a section '2 Build Your Response' with 'Product Attributes (optional)' checkboxes for Name, Regular Price, Image, URL, and Customer Review Average. There are also 'Facets' and 'Sort By' sections. At the bottom is a '3 Pagination' section with 'Results Per Page' set to 100 and 'Page' set to 1. On the right, there's a 'URL Breakdown' section with the URL query parameters listed, a 'Complete URL' field with the full API endpoint, and a 'response' field showing a JSON object representing the search results. The JSON includes fields like 'from', 'to', 'currentPage', 'total', 'totalPages', 'queryTime', 'totalTime', 'partial', 'canonicalUrl', and a list of products with their names, regular prices, and URLs.

```
baseURL : https://api.bestbuy.com/v1/products
categoryPath.id : (categoryPath.id=abcat0401000)
attribute :
(regularPrice>100&regularPrice<300&customerReviewAverage>=4)
apiKey : ?apiKey=7ksBhjryZShxfJSEk2VB07u
sortOptions : &sort=customerReviewAverage.dsc
showOptions :
&show=name,regularPrice,image,url,customerReviewAverage
pagination : pageSize=100
facets : &facet=regularPrice,50
responseFormat : &format=json
```

```
#request: [ ]  
https://api.bestbuy.com/v1/products(regularPrice>100&regularPrice<300&customerReviewAverage>=4&(categoryPath.id=abcat0401000))?  
apiKey=7ksBhjryZShxfJSEk2VB07u&sort=customerReviewAverage.dsc&show=name,regularPrice,image,url,customerReviewAverage&facet=regularPrice,50&pageSize=100&format=json
```

```
{  
  "from": 1,  
  "to": 40,  
  "currentPage": 1,  
  "total": 40,  
  "totalPages": 1,  
  "queryTime": "0.123",  
  "totalTime": "0.243",  
  "partial": false,  
  "canonicalUrl":  
    "/v1/products(regularPrice>100&regularPrice<300&customerReviewAverage>=4&(categoryPath.id=abcat0401000))?  
show=name,regularPrice,image,url,customerReviewAverage&sort=customerReviewAverage.dsc&pageSize=100&format=json&facet=regularPrice,50&apiKey=7ksBhjryZShxfJSEk2VB07u",  
  "products": [  
    {  
      "name": "Panasonic - LUMIX DMC-TS30 16.1-Megapixel Waterproof  
Digital Camera - Blue",  
      "regularPrice": 179.99,  
      "image":  
        "https://img.bbstatic.com/BestBuy_US/images/products/4032/4032159_ra.jpg" }]
```

At the same time, on the background, we start architectural design and preparation for presentation of responses in more dynamic visual form.

To make this solution generic as possible, we will use basic Web Server and will build interactive chat page using only standard JavaScript.

Our Neural Network API published as Open Source under GNU license in [GitHub](#), and we will add this solution into the same GitHub depository.

Local Development Environment.

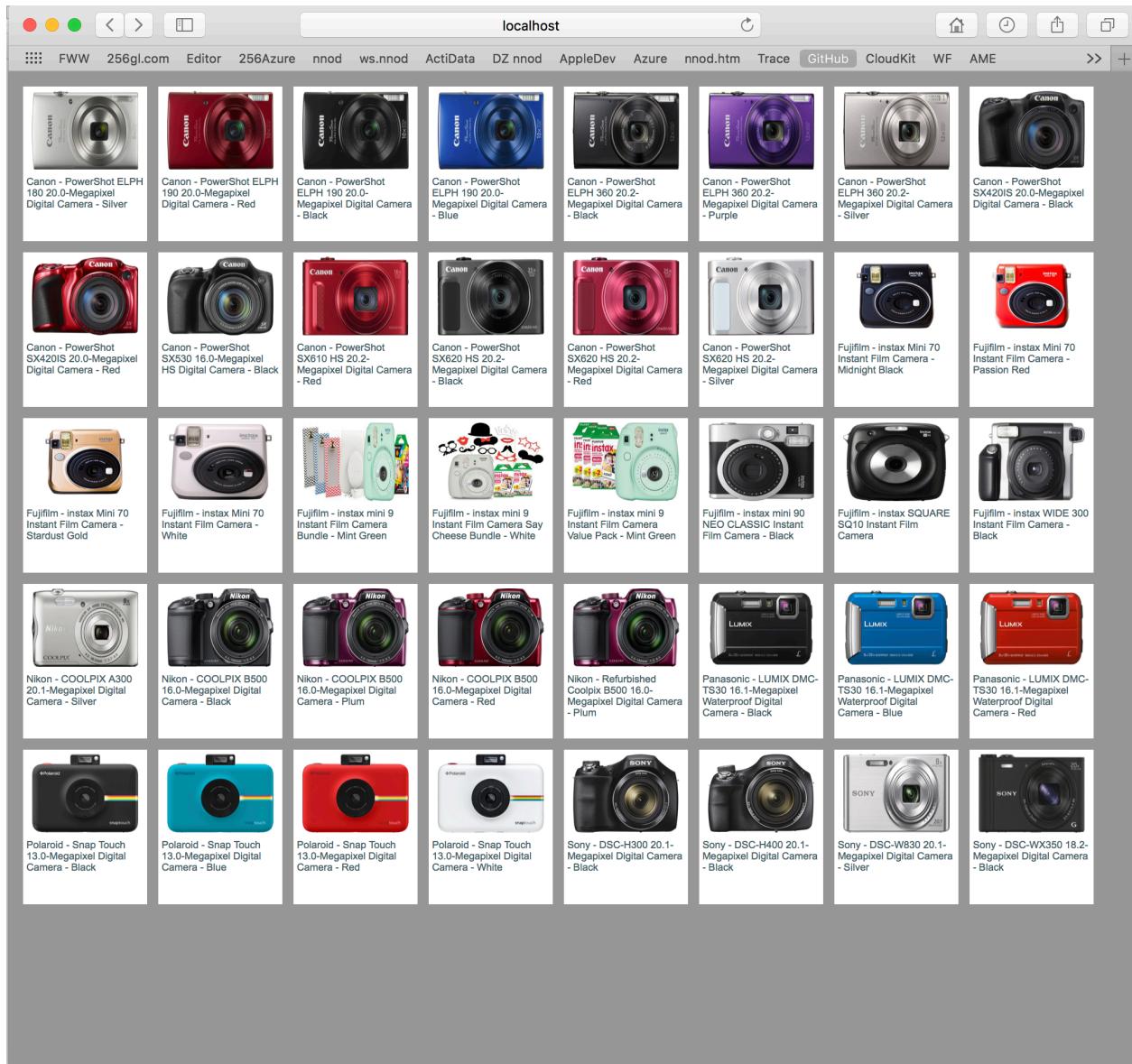
The whole process will be made in Mac OS but there are no any differences in the implementation for Windows. We will use Apache Web Server on Mac and IIS Web Server on Windows.

First PHP Script.

To make the very first call to [BB API](#) we can use PHP:

```
<body style="background-color:#999999;">
<?php
$url = "https://api.bestbuy.com/v1/
products(customerReviewAverage%3E=4&regularPrice%3E150&regularPrice%3C250&(ca
tegoryPath.id=abcat0401000))?
apiKey=xxxxxx&sort=name.asc&show=manufacturer,name,image,url,regularPrice&fac
et=regularPrice,50&pageSize=100&format=json";
$data = file_get_contents($url);
$json = json_decode($data, true);
$array = $json[ "products" ];
$db_counter = 1;
foreach ($array as $obj) {
    $db_counter = $db_counter + 1;
    echo "<div class=\"thumbnail_wrapper\"><table cellspacing=\"0\" "
cellpadding="0"><tr>\n" .
    "<td id=\"td_thumbnail_\" . $db_counter . \""
class=\"td_thumbnail_image\">\n" .
    "<a href=\"#\"><img src=\"\" . $obj[ 'image' ] . "\" height='70' width='95'
" .
    "\" onclick=\"javascript:openThumbnailLink('" . $obj[ 'url' ] . "'); return
false;\"></a></td>\n" .
    "</tr><tr><td id=\"td_thumtitle_\" . db_counter . \""
class=\"td_thumbnail_title\">\n" . " " . $obj[ 'name' ] . "\n" .
    "</td></tr></table></div> \n\n";
}
?>
</body>
```

The result:



Day 1 conclusion.

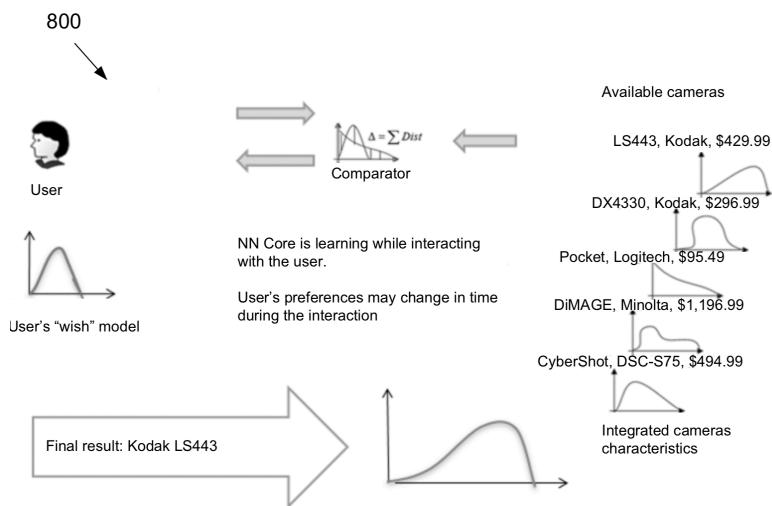
BB API Key obtained and we have full access to BB DataBase. The very first Web Page with implementation of requests to BB API built. Clear understanding of BB API now allow to start building Linguistic model for interactive search.

Day 2

Architectural/Marketing.

Value - one of the most critical questions needed to be answered before and during any development. What is the criteria we should put to measure the project advantages from similar? In our case we will use Cost of service and User's Information Demand satisfaction.

When User get an idea about buying something, he/she need to gather some information about the product. And before the very last moment when sale completed, the information for decision making will be gathered from many sources. Web, personal notes, social networks and finally expert are the network of participants in interactive process of finding the best possible item from many available in variety of sources.



Basically the process includes collecting data from different sources, compare result and repeat the process until the threshold level for the decision reached.

The cost of this process has two components: User's cost, which includes the time and potential difference if the item is wrong, or the same item can be purchased cheaper. And sale's coat, which includes the cost of consulting, and potential cost of loosing the client if he/she decided to switch to different store because of whatever reason.

We will use Best Buy Web site as one of the sources for measuring success. Criteria #1 will be based on the time required for the user to find enough information about the product and make decision to move forward and visit Best Buy or to switch to another retailer.

PHP -> JavaScript.

Even PHP is very popular and widely supported language, wherever it is possible we will try to use JavaScript for the reason of minimization of support and scalability. If our solution will be fully JavaScript based and will not require any external server support, but only BB API, that's mean we automatically solve the problem of scalability.

Changing PHP to JavaScript is relatively easy:

```
<!DOCTYPE HTML>
<html>
<head>
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
    <meta name="apple-mobile-web-app-capable" content="yes" />
    <meta name="apple-mobile-web-app-status-bar-style" content="black" />
    <meta name="viewport" content="minimum-scale=1.0, maximum-scale=5.0, user-scalable=yes, initial-scale=1.0" />
    <meta name="viewport" content="user-scalable=yes">

    <title>256gl NNOD</title>
    <link rel="stylesheet" type="text/css" href="style.css" />
    <script type="text/javascript">
        function openThumbnailLink(url) {
            window.open(url, "_blank", "toolbar=no, scrollbars=yes, resizable=yes,
top=10, left=10, width=800, height=800");
            return true;
        };
    </script>
</head>
<body style="background-color:#999999;">
    <div id="demo"></div>
    <script>
        var xmlhttp = new XMLHttpRequest();
        xmlhttp.onreadystatechange = function() {
            if (this.readyState == 4 && this.status == 200) {
                var myObj = JSON.parse(this.responseText);
                var result = myObj.products;
                var x = "";
                for (i in result) {
                    x = x + "<div class=\"thumbnail_wrapper\"><table
cellspacing=\"0\" cellpadding=\"0\"><tr>\n" +
                        "<td id=\"td_thumbnail_" + i + "\""
class="td_thumbnail_image\">\n" +
                        "<a href=\"#\"><img src=\"" + result[i].image + "\"
height='70' width='95' " +
                        "\" onclick=\"javascript:openThumbnailLink('" +
result[i].url + "'); return false;\"></a></td>\n" +
                        "</tr><tr><td id=\"td_thumbtile_" + i + "\""
class="td_thumbnail_title\">\n" + " " + result[i].name + "<br>$" +
result[i].regularPrice + "\n" +
```

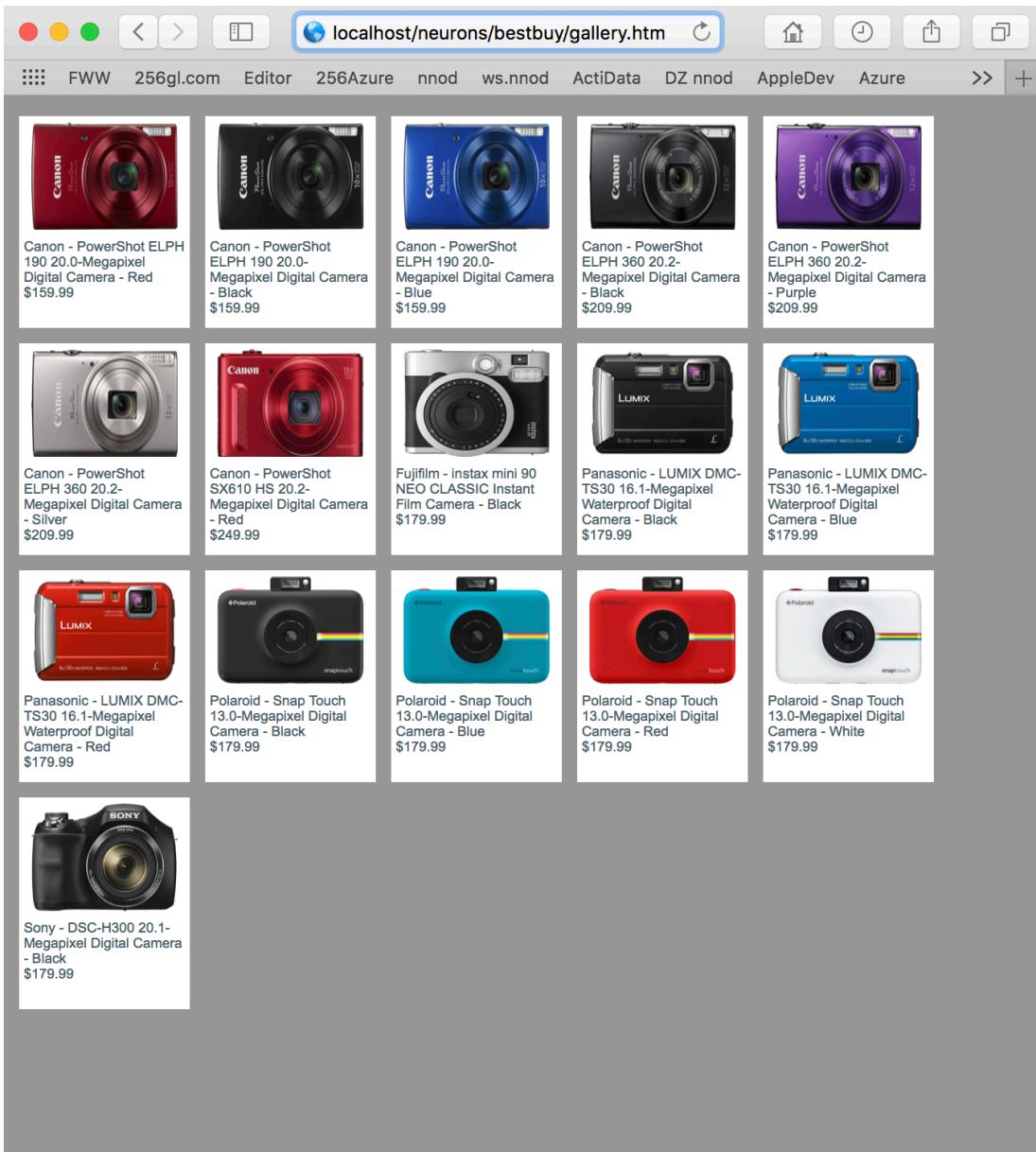
```

        "</td></tr></table></div> \n\n";
    }
    document.getElementById( "demo" ).innerHTML = x;
}
};

url = "https://api.bestbuy.com/v1/products\(customerReviewAverage%3E=4&regularPrice%3E150&regularPrice%3C250&\(categoryPath.id=abcat0401000\)\)?apiKey=xxxxxx&sort=name.asc&show=manufacturer,name,image,url,regularPrice&facet=regularPrice\_50&pageSize=100&format=json";

xmlhttp.open( "GET", url, true);
xmlhttp.send();
</script>
</body>
</html>

```



Parameters in HTTP Request