ISSN 2250-3153

Client Side Binding of Dynamic Drop Downs

Tanuj Joshi

R&D Department, Syscom Corporation Limited

Abstract- This paper talks about

- **How to bind dynamic content in a webpage?**
- **What are the challenges to fetch dynamic content from the server?**
- **What can be the other approach to bind dynamic content?**

Index Terms- Client Side Binding of Dynamic Drop Downs, jQuery, How to write jQuery Code to bind Dynamic Content

Performance!!! This is what everyone expects from others and himself too in every aspect of life. Whether you are playing, studying or doing any activity performance always matters. When it comes to websites, performance is the major factor which decides the efficiency of that website. In this paper, I demonstrate a technique which increases the page performance of any website. A section in performance enhancement has been implemented using jQuery which demonstrates that how the page performance can be increased by reducing the response time for the page.

I. INTRODUCTION

It is very often that we have to bind dynamic dropdowns in our website or application. For implementing the same, various approaches are used. There are two basic approaches for doing this:

1: By Re-Loading Complete Page from the Server (Full Postback).

2: By Re-Loading Only Required Element of a Page (Partial Postback using AJAX).

I would like to introduce a different approach for binding the dynamic dropdowns in any web application. By using this approach, the load time of dependent dropdowns (or it may be any other control as per your need) is reduced and as a result the total response time of the application also decreases.

II. PURPOSE

The purpose for introducing this technique is mainly to reduce the response time of web pages that contains the dynamic dropdowns. Also, it reduces the round trips to the server for fetching the data of dynamic dropdowns. Usually in a web application, when the dynamic dropdown data needs to be populated, it is fetched from the server, this takes a lot of time thus response time of a web page increases. This problem is solved using my approach and ultimately the performance of a page is increased.

III. CLARIFY THE PROBLEM

For binding the dynamic content in page either people uses full postback approach or partial postback by using AJAX. Let's talk about the first one, in full postback when the dynamic content is required, the whole page gets loaded again. Whereas in partial postback, instead of loading the whole page, only the dropdown content is fetched from the server. Although in partial postback response time is much decreased but in both the cases, a call has to be made to the server for just fetching the values of the dropdowns or it may be some other small dynamic value. This reduces the performance of the page as round trips to the server are made each time the data is required for dynamic dropdowns.

IV. HOW THE SITUATION OCCURRED

Suppose there is an application that is hosted in a different region and it is being accessed at various other regions. In such cases, it takes a long time in binding dynamic dropdowns for the pages which results in a bad user experience. Reason being, the latency is very high because of the distance between the client and the server.

Moving further, suppose we have three dropdowns named as Region, Country and State. Data is dynamically binding to these dropdowns. Now, if we choose a region from the region dropdown, the data dynamically gets fetched for the country dropdown and same for the State dropdown. This whole process takes a lot of time as round trips are made to the server. Even if I am using partial postback approach then it takes 4-5 seconds to load the country dropdown. This is a major problem faced and I don't think using postback approach is suitable for such a small problem.

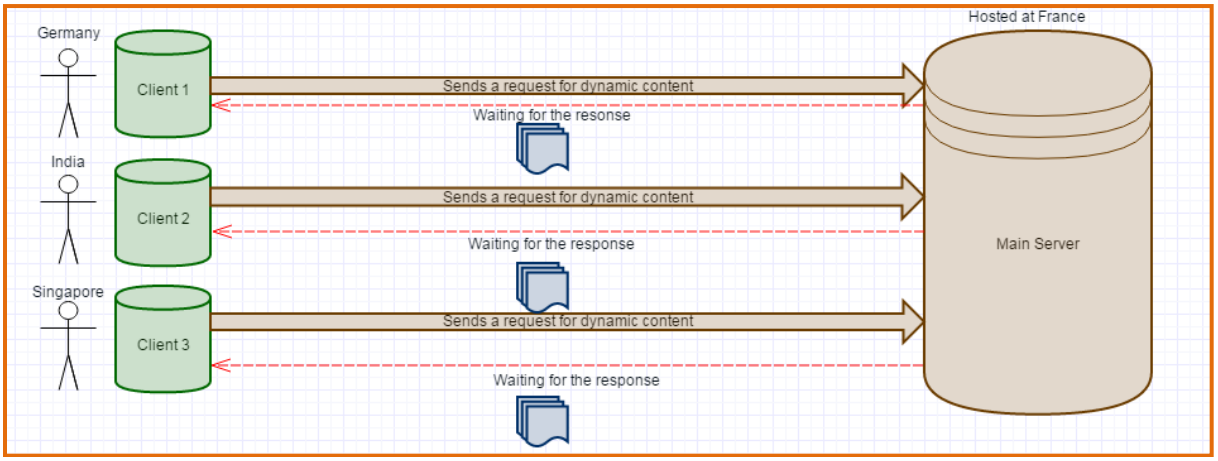

Figure 1: Clients are waiting for the Dynamic Content

V. SOLUTION

Here I have come up with a solution by using the jQuery. This solution loads all the data from the server to the local machine of the client at the time of page loading. As all the content is present in the local machine and server round trip needs not to be done for fetching the dynamic content this increases the response time of a page. Also, the approach is language independent (means on can use Java, C# or any other language at server side) making it flexible enough to be implemented using any language. Through this, let's see how I am achieving the same:

PREREQUISITES:
As I have already mentioned that we have 3 dependent dropdowns in my page Region, Country and State. In which Region dropdown is a parent dropdown, Country dropdown is a parent-child dropdown and State is a child dropdown. So the value of child and parent-child dropdowns must contain some logical data that will link it with its parent dropdown. Before using this approach we have to ensure that at client end the data of the drop down should be bind in following format:

```
<select id="Region">
                    <option>--Select--</option>
                    <option value="1">Africa</option>
                    <option value="2">Asia</option>
                    <option value="3">Europe</option>
                    <option value="4">Oceania</option>
                    <option value="5">North America</option>
                    <option value="6">South America</option>
</select>
```

Figure 2: HTML of Region Dropdown List Rendered at Client Machine

```
<select id="Country">
                    <option>--Select--</option>
                    <option value="5,2">Afghanistan</option>
                    <option value="6,2">Bangladesh</option>
                    <option value="7,2">Bhutan</option>
                    <option value="8,2">China</option>
                    <option value="9,2">Egypt</option>
                    <option value="10,2">Georgia</option>
                    <option value="11,2">India</option>
                     …….
</select>
```

Figure 3: HTML of Country Dropdown List Rendered at Client Machine

Purpose to show this html code is to notice the highlighted value field. Here value attribute contains comma separated string in which first part indicates the ID of the Country and the second part indicates ID of the Region.

```
<select id="State">
                    <option>--Select--</option>
                    <option value="5,11">Andhra Pradesh</option>
                    <option value="6,11">Arunachal Pradesh</option>
                    <option value="7,11">Assam</option>
                    <option value="8,11">Bihar</option>
                    <option value="9,11">Chhattisgarh</option>
                    <option value="10,11">Goa</option>
                    <option value="11,11">Gujarat</option>
                    <option value="12,11">Haryana</option>
                     ……….
</select>
```

Figure 4: HTML of State Dropdown List Rendered at Client Machine

In the above code first part of value field indicates the ID of the state and the other part indicates ID of the corresponding country.

PROCEDURE

As I have mentioned above that Region drop down is parent drop down and Country drop down is a child drop down of Region as its value depends on selected Region. A child drop down must contain all its values with corresponding Parent value.
Now I will explain by quoting an example so that it will be clearer. What I am doing, loading the complete data of all dynamic dropdown when the page loaded first time and assigning the whole data of Country drop down into an array at client side by using jQuery. This array will contain Country Name, Country ID and Region ID. When a Region is selected then I am clearing all data from Country dropdown and re-assigning the data to Country dropdown from the array based on selected Region ID. Now you can see data to Country dropdown is populated without any postback.

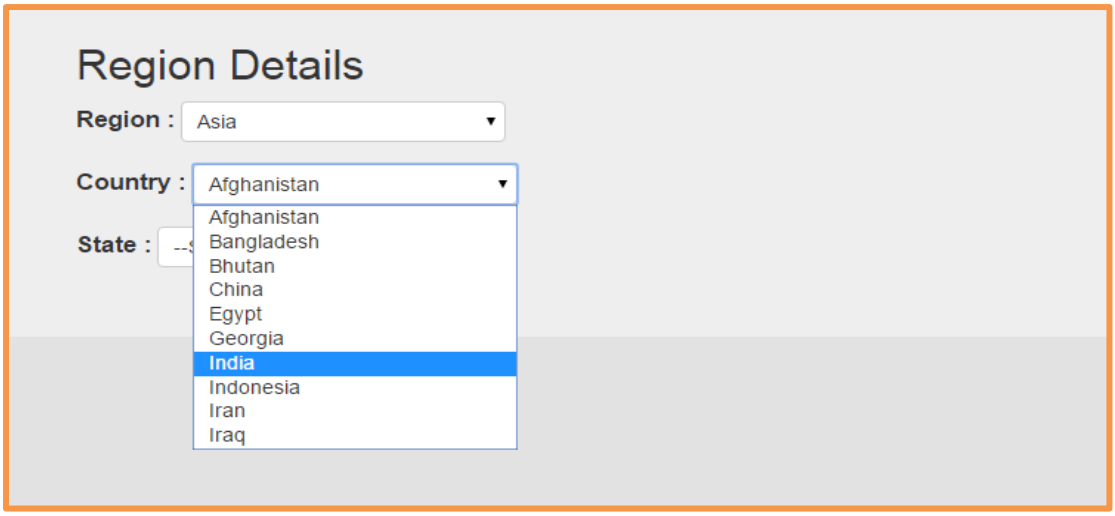

Figure 5: Dropdown at Client Machine

Following are the steps to implement the client side binding of dynamic dropdowns by using jQuery:

NOTE: Here I am displaying the code to bind Country on the basis of selected Region as my purpose is to give an idea only.

STEP 1:
Declare an array to hold all the items of the Country dropdown.

```
var arrCountry[];
```

STEP 2:
Iterate over Country DropDown and insert items on array arrCountry[].

```
$("#Country option").each(function (index) {
                var val = $(this).val();
                var CombinedID = val.split(",");

                var item = {}
                item["RegionID"] = CombinedID[1];
                item["CountryID"] = CombinedID[0];

                item["CountryName"] = $(this).text();

                arrCountry.push(item);
 });
```

STEP 3:
Declare a variable myJsonString.

```
var myJsonString;
```

STEP 4:
Declare a variable myJsonString.

```
var myJsonString;
```

STEP 5:
Convert arrCountry[] into JSON text and store it in a string.

```
myJsonString = JSON.stringify(arrCountry);
```

STEP 6:
Store JSON string into a hidden field or in hidden text area.

```
$("#data").val(myJsonString);
```

STEP 7:
Create a function to Handle the Change Event of Primary Dropdown.

a. Declare a variable RegionID that will contain the selected RegionID from dropdown.

```
var selectedRegionID = $(this).val();
```

b. Parse the data into JSON contained in hidden field and assign this data into a new variable CountryData.

```
var CountryData = jQuery.parseJSON($("#data").val());
```

c. Remove all the options from Country Dropdown.

```
$("#Country").find("option").remove();
```

d. Iterate over CountryData and assign data to country dropdown.

```
jQuery.each(CountryData, function (i, jArray) {

                    if (selectedRegionID == jArray["RegionID"]) {
                        $('#Country').append($('<option>', {
                            value: jArray["CountryID"],
                            text: jArray["CountryName"]
                        }));

                    }
            });
```

VI. CONCLUSION

Although and is widely used but in my opinion there are some scenarios where latency is very high, it is not preferable. For such scenarios I prefer the jQuery approach for binding the dynamic content at client side.

ACKNOWLEDGEMENT

This work was supported by Syscom Corporation Ltd.

REFERENCES

http://www.w3schools.com/jquery/
https://jquery.com/

AUTHORS

First Author – Tanuj Joshi , M.C.A, Software Engineer, tanuj.joshi@live.com