.NET GENES

ASP.NET MVC Cascade Dropdown Using jQuery And Entity Framework

- March 02, 2021

Team,

Here's an example of how to perform a cascading dropdown in ASP.NET MVC using jQuery and Entity Framework as the ORM. This example will use Microsoft's sample database called **WideWorldImporters**. So to start with, we need to create an ASP.NET MVC Project and add bootstrap and jQuery files through NuGet. Then add an ADO.NET Entity Data Model (Entity Framework) which maps to the WideWorldImporters database. Once done, add a new class called **CountryStatesCitiesViewModel** with an enumerable property called CountryList which contains country values that will bind to the country dropdownlist control in the UI.

```
public class CountryStatesCitiesViewModel
{
        public IEnumerable<SelectListItem> CountryList { get; set; }
}
```

Add a controller called Home with a default Index ActionResult method that retrieves country list from the database and store them to the model used for data binding of the dropdownlist control. I checked the WideWorldImporters db and found out that only the United States has corresponding state and cities. Next is add two methods that return a Json object. These methods query the database for the state provinces and it's corresponding cities. The parameters on both methods are the unique ID's and not their names.

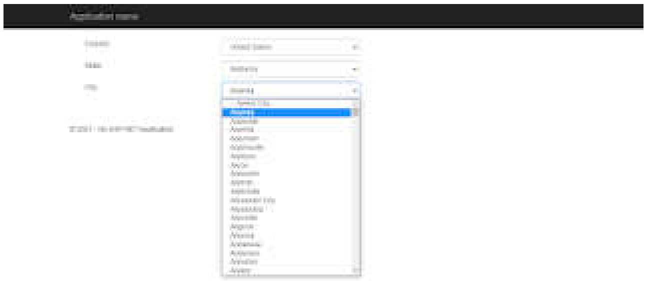
```
public class HomeController : Controller
{
  private CountryStatesCitiesViewModel model;
  private WideWorldImportersEntities _context;

  // GET: Home
  public ActionResult Index()
  {
         List<SelectListItem> itemCountries = new List<SelectListItem>
         _context = new WideWorldImportersEntities();
         model = new CountryStatesCitiesViewModel();

         var countries = (from ctr in _context.Countries select ctr).A
         {
                Value = x.CountryID.ToString(),
                Text = x.CountryName
         }).Where(z => z.Text == "United States");

         itemCountries.AddRange(countries);

         model.CountryList = itemCountries;
         return View(model);
  }

  public JsonResult GetStatesByID(int countryID)
  {
         _context = new WideWorldImportersEntities();
         return Json(_context.StateProvinces.Where(data => data.Countr
                                 , JsonRequestBehavior.AllowGet);
  }

  public JsonResult GetCitiesByID(int stateID)
  {
         _context = new WideWorldImportersEntities();
         return Json(_context.Cities.Where(data => data.StateProvinceI
                                 , JsonRequestBehavior.AllowGet);
  }
}
```

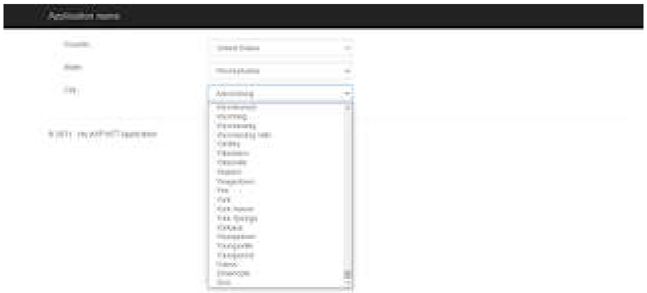
The last part is to add an Index page with three dropdown controls designated for the country, state province and city. The country dropdown values were populated with values during Index method execution. The state dropdown will be populated with values when a specific country was chosen by the user and likewise the city dropdown will be populated with values if a selection happens from the state dropdown. The cascade effect was done using jQuery by calling the change function for the country and state dropdowns that is shown in the script section of the page.

```
@{
    ViewBag.Title = "Index";
}
@model CascadingDropdown.Models.CountryStatesCitiesViewModel

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Edit</title>
    <script type="text/javascript" src="~/scripts/jquery-1.10.2.min.js
```

```
<script type="text/javascript">
    $(function () {
        $("#CountryID").change(function () {
            var selectedCountry = $(this).val();
            if (selectedCountry != "") {
                $.ajax({
                    url: "/Home/GetStatesByID",
                    data: { countryID: selectedCountry },
                    success: function (result) {
                        if (result != undefined) {
                            $("#StateID").empty();
                            $("#StateID").append($("<option></opti
                            $.each(result, function (index, elem)
                                if (index == 0) {
                                    $("#StateID").append($("<optio
                                        .attr("selected", "selecte
                                        .attr("value", elem.value)
                                        .text(elem.text));
                                }
                                else {
                                    $("#StateID").append($("<optio
                                }
                            });
                            $("#StateID").change();
                        }
                    }
                });
            }
        });

        $("#StateID").change(function () {
            var selectedState = $(this).val();
            if (selectedState != "") {
                $.ajax({
                    url: "/Home/GetCitiesByID",
                    data: { stateID: selectedState },
                    success: function (result) {
                        if (result != undefined) {
                            $("#CityID").empty();
                            $("#CityID").append($("<option></optio
                            $.each(result, function (index, elem)
                                if (index == 0) {
                                    $("#CityID").append($("<option
                                        .attr("selected", "selecte
                                        .attr("value", elem.value)
                                        .text(elem.text));
                                }
                                else {
                                    $("#CityID").append($("<option
                                }
                            });
                            }
                        }
                    });
                }
                else {
                    $("#CityID").empty();
                    $("#CityID").append($("<option></option>").attr("v
                }
            });

        });
    </script>
    <style type="text/css">
        .row{
            margin-bottom: 10px !important;
        }
    </style>
</head>
<body>
    <div class="container">
        <br />
        <div class="row">
            <div class="col-md-12">
                <div class="col-md-3">
                    Country:
                </div>
                <div class="col-md-9">
                    @Html.DropDownListFor(x => x.CountryList, new Sele
                                      "-- Select Country --", new { id
                </div>
            </div>
        </div>
        <div class="row">
            <div class="col-md-12">
                <div class="col-md-3">
                    State:
                </div>
                <div class="col-md-9">
                    @Html.DropDownList("StateID", new List<SelectListI
                </div>
            </div>
        </div>
        <div class="row">
            <div class="col-md-12">
                <div class="col-md-3">
                    City:
                </div>
                <div class="col-md-9">
                    @Html.DropDownList("CityID", new List<SelectListIt
                </div>
            </div>
            </div>
        </div>
</body>
</html>
```

Output With Changes Done On States Dropdown





ASP.NET MVC Entity Framework jQuery

Enter Comment

Popular Posts From This Blog

WPF CRUD Application Using DataGrid, MVVM Pattern, Entity Framework, And C#.NET

- *February 04, 2021*

Image

Good day to all! Here's an example of a WPF CRUD (Create,Update and Delete) project using the DataGrid control, ADO.NET Entity Framework 6.x, C#.NET and Model–View–Viewmodel(MVVM) architectural pattern. This post was based from this tutor …

READ MORE

How To Install Or Add Icons In ASP.NET Core MVC Pages Using Font Awesome

- April 19, 2021

Image This post illustrates on how to install or add icons to your ASP.NET Core MVC website pages using Font Awesome. If you added Bootstrap 4.x to your project, you might noticed that when you typed glyphicon in a class attribu …

READ MORE

How To Insert Or Add Emojis In Microsoft Teams Status Message

- July 09, 2021

Image Good day Gents! I was trying to update my Microsoft Teams status and was wondering if I can add or insert emoji. For editing or adding new teams status, only text fields are available and there's no emoji icon similar to wr …

READ MORE

Powered by Blogger

←

.NET GENES

Greg Esguerra is a .NET Applications

Developer and IT Instructor from Cebu City, Philippines with experience working on projects related to web, desktop, database and business intelligence using Microsoft Technology Stack.

VISIT PROFILE

FAVORITES

InfoVerze

PAGES

About Me
Disclaimer
Contact Us

ARCHIVE

LABELS

FOLLOWERS

POPULAR POSTS

Add Checkbox In WPF Datagrid DatagridTemplateColumnHeader (For Checkall)

- September 23, 2012

Image

Here's how to add checkbox in WPF Datagrid DatagridTem …

READ MORE

Change BIOS Boot Options For Acer Aspire V5-431P/471P

- September 18, 2013

Image

Hello and Good afternoon! Here's a useful Acer Aspire V5-43 …

READ MORE

Add Placeholder (Watermark) To Html.TextBoxFor() Helper In ASP.NET MVC 4

- March 12, 2014

Image

After working with simple form html elements, I decided to g …

READ MORE

Bootstrap DatePicker Control In ASP.NET Web Forms

- June 16, 2016

Image

Here's how to integrate Bootstrap DatePicker widget in yo …

READ MORE

Clean Invalid XML Characters In C#

- February 03, 2013

Here's a cool way to clean Large XML files with invalid xml characters. Note: Stream from is the origi …

READ MORE