**Cascading DropDownLists**



public class Employee

{

public int Id { get; set; }

public string Surname { get; set; }

public List<Employee> **FetchEmployees()**

{

return new List<Employee>

{

new Employee {Id = 1, Surname = "Hanks"},

new Employee {Id = 2, Surname = "Woods"},

};

}

public Employee **FetchEmployee**(int id)

{

var employees = **FetchEmployees();**

return (from p in employees where p.Id == id

select p).**First();**

}

}

public class EmployeeCar

{

public int Id { get; set; }

public string Car { get; set; }

private static List<EmployeeCar> **LoadData()**

{

return new List<EmployeeCar>

{

new EmployeeCar {Id = 1, Car = "Ford"},

new EmployeeCar {Id = 1, Car = "Holden"},

new EmployeeCar {Id = 1, Car = "Honda"},

new EmployeeCar {Id = 2, Car = "Toyota"},

new EmployeeCar {Id = 2, Car = "Volvo"},

};

}

public List<EmployeeCar> FetchEmployeeCars(int id)

{

return (from p in **LoadData()** where p.Id == id

select p).**ToList();**

}

}

Add code to the page load event to fill the drop down list with the employees and a method to be called through jQuery to update the employee’s cars when the user changes the selected employee.

[WebMethod]

public **static** List<EmployeeCar> FetchEmployeeCars(int id)

{

var emp = new EmployeeCar();

return emp.FetchEmployeeCars(id);

}

protected void Page\_Load(object sender, EventArgs e)

{

if (!IsPostBack)

{

var employees = new Employee();

ddlEmployee.DataSource = employees.FetchEmployees();

ddlEmployee.DataTextField = "Surname";

ddlEmployee.DataValueField = "Id";

ddlEmployee.DataBind();

}

}

The *FetchEmployeeCars* method has been decorated with the [WebMethod](http://msdn.microsoft.com/en-us/library/byxd99hx(VS.80).aspx) attribute and it’s static. This will allow the call from jQuery using the [Ajax](http://docs.jquery.com/Ajax) method:

<script language="javascript" type="text/javascript">

// when page loads

$(function () {

// When the page loads it looks for the values

var ddl = $("#ddlEmployee");

ddl.focus();

$("#ddlCars").hide();

// It BINDS CHANGE KEYUP to trigger when selecting

ddl.bind("change keyup", function () {

loadCars($(this).val());

});

});

// Perform AJAX call to function FetchEmployeeCars [Web Method]

function loadCars(selectedItem) {

if (selectedItem == "0") {

$("#ddlCars").fadeOut("slow"); // fadeOut

return;

}

$("#ddlCars").fadeIn("slow"); // fadeIn

$.ajax({

type: "POST",

url: "Default3.aspx/FetchEmployeeCars",

data: "{id: " + selectedItem + "}",

contentType: "application/json; charset=utf-8",

dataType: "json",

async: true,

success: function (data) {

LoadComboCars(data.d);

}

});

}

// Once it has the data then load the combo

function LoadComboCars(data) {

var carros2 = $("#ddlCars");

// Clean Cars Combo Box

var carros = document.getElementById("ddlCars");

carros.options.length = 0;

for (var i = 0; i < data.length; i++) {

var opt = document.createElement("option");

opt.text = data[i].Car;

opt.value = data[i].Id;

// Add an Option object to Drop Down/List Box

carros.options.add(opt);

}

}

</script>

ANOTHER OPTION TO ADD TO A COMBO BOX IS

for (var i = 0; i < data.length; i++) {

$("select[name$=ddlCars]").append(

$("<option></option>").val(data[i].Id).html(data[i].Car)

);

}

**Add the Body**

<body>

<form id="form1" runat="server">

<div>

<asp:DropDownList ID="ddlEmployee" runat="server" AppendDataBoundItems="true">

<asp:ListItem Text="(Please Select)" Value="0" Selected="True" />

</asp:DropDownList>

</br>

</br>

<asp:DropDownList ID="ddlCars" runat="server" ></asp:DropDownList>

</div>

</form>

</body>

</html>

Otra forma

<script type="text/javascript">

$(function() {

$('#ddlEmployee').change(loadCars);

});

function loadCars() {

var selectedItem = $('#ddlEmployee').val();

if (selectedItem == "0") {

$("#ddlCars").fadeOut("slow"); // fadeOut

return;

}

…. And on on

**The easiest way to add items to a combo box**

var opt = document.createElement("option");

opt.text = "-- Select --";

opt.value = 0;

carros.options.add(opt);