using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace \_03\_JsonResultSample.Controllers

{

public class HomeController : Controller

{

//

// GET: /Home/

public ActionResult Index()

{

return View();

}

}

}

@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>Index</title>

</head>

<body>

<div>

@Html.ActionLink("Action #1", "Action1", "Data") <br />

<a id="sendRequest" href="#">Action #2 (JavaScript)</a>

<div id="result"></div>

<a id="sendRequest2" href="#">Action3</a>

<div id="result2">

</div>

<script src="~/Scripts/jquery-2.0.2.min.js"></script>

<script>

$(document).ready(function () {

$("#sendRequest").click(function () {

// Получение данных с помощью GET запроса

$.ajax({

url: "@Url.Action("Action2", "Data")",

type : "POST",

success: function (data) {

$("#result")

.append("Name: " + data.Name + "<br />")

.append("Price: " + data.Price + "<br />")

.append("Description: " + data.Description + "<br />");

}

});

});

$("#sendRequest2").click(function () {

$.ajax({

url: "@Url.Action("Action3", "Data")",

type: "POST",

success: function (data) {

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

//{

// $("#result2")

// .append("Name: " + data[i].Name + "<br />")

// .append("Price: " + data[i].Price + "<br />")

// .append("Description: " + data[i].Description + "<br />");

//}

for (var i in data) {

$("#result2")

.append("Name: " + data[i].Name + "<br />")

.append("Price: " + data[i].Price + "<br />")

.append("Description: " + data[i].Description + "<br />");

}

}

});

});

});

</script>

</div>

</body>

</html>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace \_03\_JsonResultSample.Controllers

{

public class DataController : Controller

{

//

// GET: /Data/

[HttpGet] // атрибут устанавливает ограничение на метод Action1 и позволяет выполнять его только при GET запросе

public ActionResult Action1()

{

var product = new

{

Name = "Item 1",

Price = 10,

Description = "Dummy Text ....",

};

// По умолчанию MVC Framework не будет возвращать JSON в ответ на GET запрос по причнинам безопасности.

// Возвращать в ответ на GET запрос JSON безопасно если в ответе не содержатся чувствительные данные.

// http://haacked.com/archive/2009/06/25/json-hijacking.aspx/

return Json(product, JsonRequestBehavior.AllowGet);

}

[HttpPost] // атрибут устанавливает ограничение на метод Action2 и позволяет выполнять его только при POST запросе

public ActionResult Action2()

{

var product = new

{

Name = "Item 1",

Price = 10,

Description = "Dummy Text ....",

};

return Json(product);

}

[HttpPost]

public ActionResult Action3()

{

List<Product> products = new List<Product>(){

new Product(){Name="Item 1",Price = 10, Description="Dummy text1"},

new Product(){Name="Item 2",Price = 20, Description="Dummy text2"},

new Product(){Name="Item 3",Price = 30, Description="Dummy text3"},

new Product(){Name="Item 4",Price = 40, Description="Dummy text4"},

new Product(){Name="Item 5",Price = 50, Description="Dummy text5"},

new Product(){Name="Item 6",Price = 60, Description="Dummy text6"},

};

return Json(products);

}

}

class Product

{

public string Name { get; set; }

public int Price { get; set; }

public string Description { get; set; }

}

}

Action1

{"Name":"Item 1","Price":10,"Description":"Dummy Text ...."}

Action2

[Action #1](http://localhost:13379/Data/Action1)   
[Action #2 (JavaScript)](http://localhost:13379/)

Name: Item 1  
Price: 10  
Description: Dummy Text ....

[Action3](http://localhost:13379/)

Action3

[Action #1](http://localhost:13379/Data/Action1)   
[Action #2 (JavaScript)](http://localhost:13379/)

[Action3](http://localhost:13379/)

Name: Item 1  
Price: 10  
Description: Dummy text1  
Name: Item 2  
Price: 20  
Description: Dummy text2  
Name: Item 3  
Price: 30  
Description: Dummy text3  
Name: Item 4  
Price: 40  
Description: Dummy text4  
Name: Item 5  
Price: 50  
Description: Dummy text5  
Name: Item 6  
Price: 60  
Description: Dummy text6