

به نام خدا

آموزش ساخت وب سرویس در دات نت

سرویس دهنده Restful

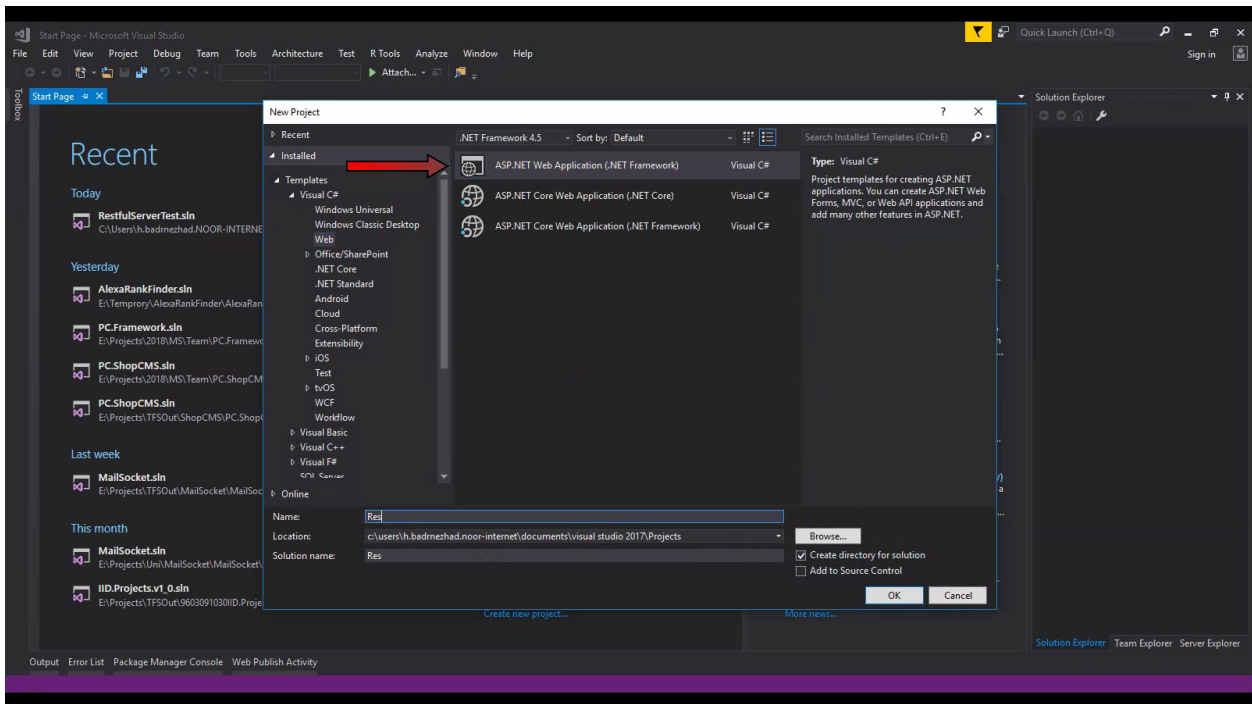
و سرویس گیرنده MVC

حسین بدرنژاد

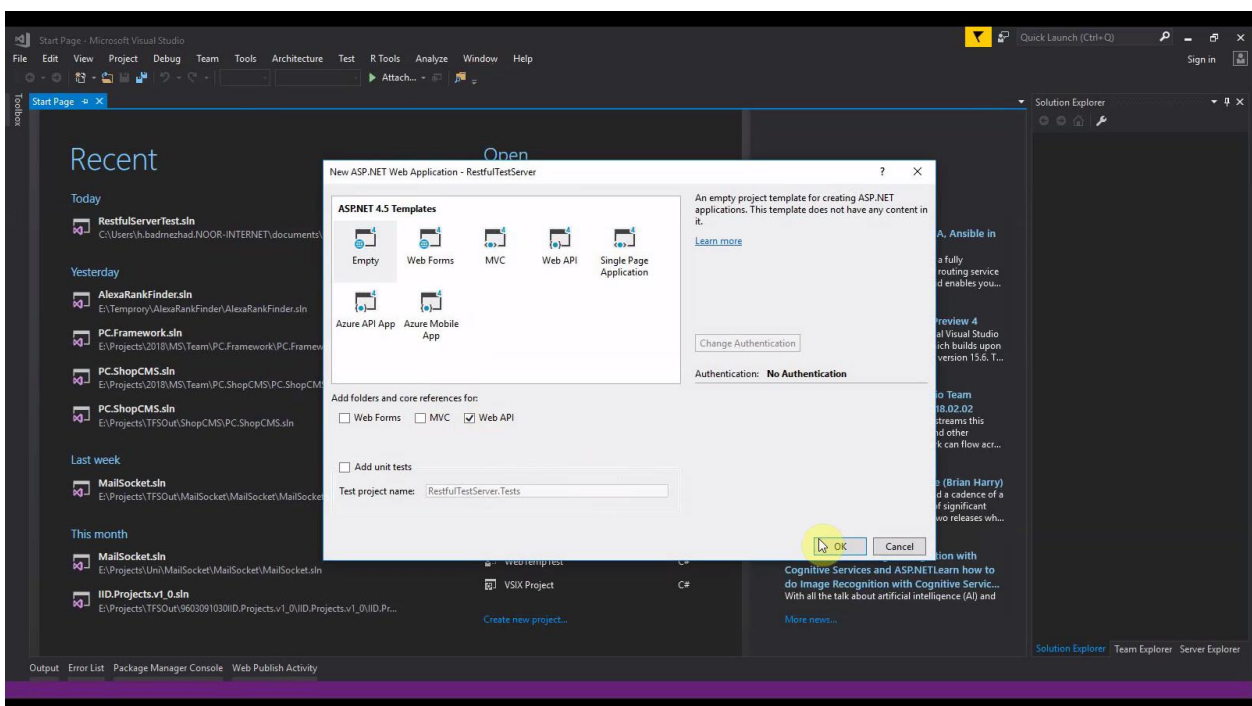
زمستان ۹۶

در این آموزش قصد داریم یک وب سرویس Restful توسط WebApi تحت فریم ورک .Net MS پیاده سازی کنیم، سپس توسط ASP.NET MVC سمت سرویس گیرنده را پیاده سازی میکنیم.

جهت انجام این کار ابتدا یک پروژه جدید از نوع ASP.NET Web Application میسازیم



سپس آن را از نوع Empty و هسته آن را از نوع Web Api انتخاب میکنیم



پس از ساخت این پروژه یک کنترلر از نوع WebApi2 به آن اضافه مینماییم.

سپس مدل های خود را به پروژه اضافه مینماییم. در این مثال از مدل مخاطبین (Contacts) استفاده شده است.

```
01 public class ContactModel
02 {
03     public int Id { get; set; }
04     public string FirstName { get; set; }
05     public string LastName { get; set; }
06     public string Mobile { get; set; }
07 }
```

حالا اطلاعاتی که توسط وب سرویس قرار است ارائه داده شود را، از دیتابیس، فایل یا ... توسط لایه های دیگر پیاده سازی کنید و در کنترل به صورت زیر ارسال مینماییم. در این مثال یک کلاس سرویس پیاده سازی شده که دو ردیف اطلاعات جهت تست ارائه شده.

```
01 public class ContactService
02 {
03     private List<ContactModel> lstTest;
04
05     public ContactService()
06     {
07         lstTest = new List<ContactModel>();
08         lstTest.Add(new ContactModel() { Id = 1, FirstName = "Hossein", LastName
09 = "Badrnezhad", Mobile = "09388283893" });
10         lstTest.Add(new ContactModel() { Id = 2, FirstName = "Test", LastName =
11 "User", Mobile = "09121234567" });
12     }
13
14     public List<ContactModel> GetAllContacts()
15     {
16         return lstTest;
17     }
18
19     public List<ContactModel> GetContactByName(string name)
20     {
21         return lstTest.Where(x => x.FirstName.ToLower().Contains(name.ToLower())
22 ||
23 x.LastName.ToLower().Contains(name.ToLower())).ToList();
24     }
25
26     public ContactModel GetContactById(int id)
27     {
28         return lstTest.FirstOrDefault(x => x.Id == id);
29     }
30 }
```

حال کنترلر را به این صورت پیاده سازی میکنیم.

```
01 public class ContactsController : ApiController
02 {
03     [Route("api/contacts")]
04     public HttpResponseMessage Get()
05     {
06         ContactService service = new ContactService();
07         var contacts = service.GetAllContacts();
08         var response = Request.CreateResponse(HttpStatusCode.OK, contacts);
09         return response;
10     }
11     [Route("api/contacts/{id?}")]
12     public HttpResponseMessage Get(int id)
13     {
14         ContactService service = new ContactService();
15         var contacts = service.GetContactById(id);
16         var response = Request.CreateResponse(HttpStatusCode.OK, contacts);
17         return response;
18     }
19     [Route("api/contacts/{name:alpha}")]
20     public HttpResponseMessage Get(string name)
21     {
22         ContactService service = new ContactService();
23         var contacts = service.GetContactByName(name);
24         var response = Request.CreateResponse(HttpStatusCode.OK, contacts);
25         return response;
26     }
27 }
```

حالا برنامه را اجرا میکنیم. با استفاده از آدرس <http://localhost:59981/api/contacts> متد دریافت همه مخاطبین صدا زده میشود و اطلاعات دریافتی بصورت xml قابل نمایش میباشد.

```
<ArrayOfContactModel xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://schemas.datacontract.org/2004/07/RestfulTestServer.Models">
  <ContactModel>
    <FirstName>Hossein</FirstName>
    <Id>1</Id>
    <LastName>Badrnezhad</LastName>
    <Mobile>09388283893</Mobile>
  </ContactModel>
  <ContactModel>
    <FirstName>Test</FirstName>
    <Id>1</Id>
    <LastName>User</LastName>
    <Mobile>09121234567</Mobile>
  </ContactModel>
</ArrayOfContactModel>
```

حال برای تغییر قالب از xml به json یک کلاس به نام BrowserJsonFormatter میسازیم و آن را از JsonMediaTypeFormatter ارث میبریم.

```

01 public class BrowserJsonFormatter : JsonMediaTypeFormatter
02 {
03     public BrowserJsonFormatter()
04     {
05         this.SupportedMediaTypes.Add(new MediaTypeHeaderValue("text/html"));
06         this.SerializerSettings.Formatting = Formatting.Indented;
07     }
08
09     public override void SetDefaultContentHeaders(Type type, HttpContentHeaders
headers, MediaTypeHeaderValue mediaType)
10     {
11         base.SetDefaultContentHeaders(type, headers, mediaType);
12         headers.ContentType = new MediaTypeHeaderValue("application/json");
13     }
14 }

```

حال فایل WebApiConfig را باز کرده و کد زیر را به آن در متد Register اضافه کنید.

```

01 config.Formatters.Add(new BrowserJsonFormatter());

```

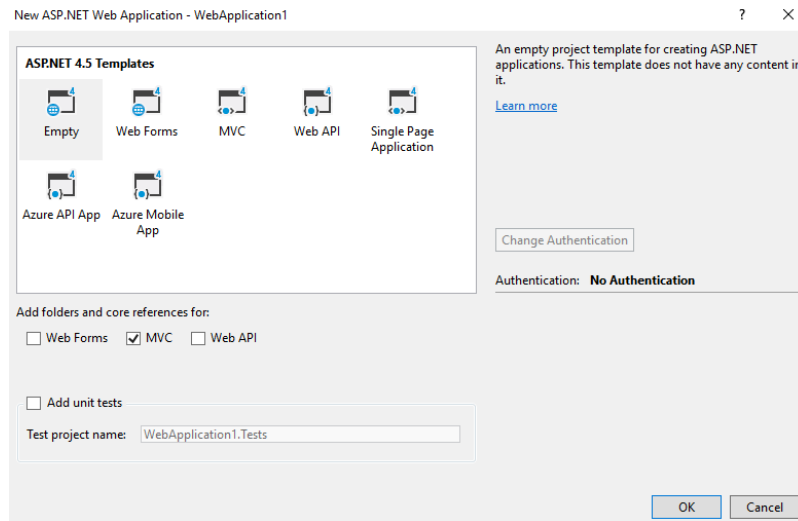
حالا دوباره آدرس <http://localhost:59981/api/contacts> را باز کنید. فرمت به json تغییر پیدا کرده است.

```

[
  {
    "Id": 1,
    "FirstName": "Hossein",
    "LastName": "Badrnezhad",
    "Mobile": "09388283893"
  },
  {
    "Id": 2,
    "FirstName": "Test",
    "LastName": "User",
    "Mobile": "09121234567"
  }
]

```

حال نوبت به سرویس گیرنده می‌باشد. ابتدا یک پروژه جدید از نوع ASP.NET Web Application بسازید و نوع هسته آن را MVC انتخاب کنید.



پس از ساخت پروژه یک کنترلر از نوع MVC5 به آن اضافه کنید. سپس همان مدلی که سمت سرور وجود دارد را سمت کلاینت نیز بیاورید.

جهت استفاده از برخی متدها و کلاس‌ها لازم است تا چند کتابخانه به پروژه اضافه شود. جهت انجام این کار ابتدا از پنجره Solution روی Reference ها کلیک راست نمایید سپس روی Add Reference کلیک نمایید. از پنجره باز شده به بخش Assemblies رفته و به دنبال System.Net.Http و System.Net.Http بگردید و آنها را اضافه نمایید.

حالا مجدداً روی Reference ها کلیک راست نمایید سپس روی Manage Nuget Packages کلیک نمایید. به تب Browse رفته و به دنبال Newtonsoft.Json بگردید و آن را نصب نمایید.

حالا یک کلاس به سرویس‌ها اضافه نمایید که در آن متدهای وب سرویس صدا زده می‌شود.

```
01 public class ContactService
02 {
03     public static List<ContactModel> GetAllContacts()
04     {
05         try
06         {
07             var url = ServiceUrls.getAllContacts;
08             var client = new HttpClient();
09             JsonSerializerSettings settings = new JsonSerializerSettings();
10             settings.ContractResolver = new
11             CamelCasePropertyNamesContractResolver();
12             var response_data = client.GetAsync(url);
13             var response =
14             response_data.Result.Content.ReadAsStringAsync().Result;
15             var result =
16             JsonConvert.DeserializeObject<List<ContactModel>>(response);
17             return result;
18         }
19         catch (Exception)
20         {
21             return new List<ContactModel>();
22         }
23     }
24 }
```

```
20     }
21   }
22 }
23
```

حالا کنترلر را بصورت زیر پیاده سازی نمایید.

```
01 public class HomeController : Controller
02 {
03     // GET: Home
04     public ActionResult Index()
05     {
06         var ContactsData = ContactService.GetAllContacts();
07         return View(ContactsData);
08     }
09 }
```

سپس روی View کلیک راست نموده و Add View را بزنید تا View آن ساخته شود.

کد View صفحه Index را بصورت زیر تغییر دهید و برنامه را اجرا نمایید.

```
01 @using System.Collections.Generic;
02 @using RestfulTestClient.Models;
03 @{
04     ViewBag.Title = "Index";
05
06     List<ContactModel> dataList = Model;
07 }
08
09 <h2>Index</h2>
10
11 @foreach (var item in dataList)
12 {
13     <div>
14         <h1>@item.FirstName @item.LastName</h1>
15         <h2>@item.Mobile</h2>
16     </div>
17     <hr />
18 }
```

