ساده ترین و سریعترین راه برای تست API های برنامه کاربردی چیست؟

ابزارهای زیادی وجود دارد که می توانید برای آزمایش یک API Controller برنامه کاربردی استفاده کنید.

برخی از آنهایی که خوب هستند:

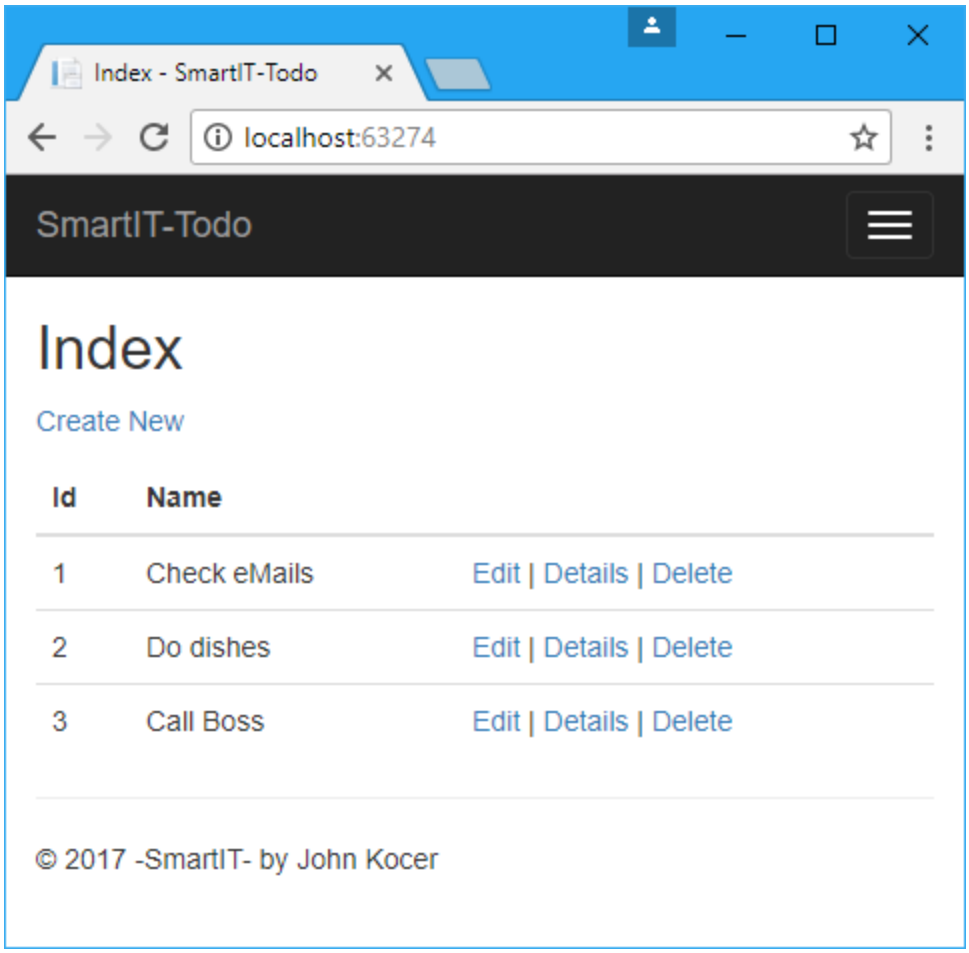
- Postman از فروشگاه Google Chrome
- Swagger که یک NuGet package است که یک صفحه خلاصه را به یک برنامه کاربردی تعریف می کند که applications APIs را توصیف می کند.
- Fiddler که یک محصول رایگان Telerik است که یک ابزار خطای HTTP مستقل و آزمایشی است.

اما، ساده ترین راه برای آزمایش یک برنامه API ، استفاده از Windows PowerShell یا Visual Studio Package Manager Console است. من به شما گام به گام نحوه استفاده از PowerShell یا PM ((Package Manager Console))را برای تست کاربرد application Web APIs نشان می دهم.

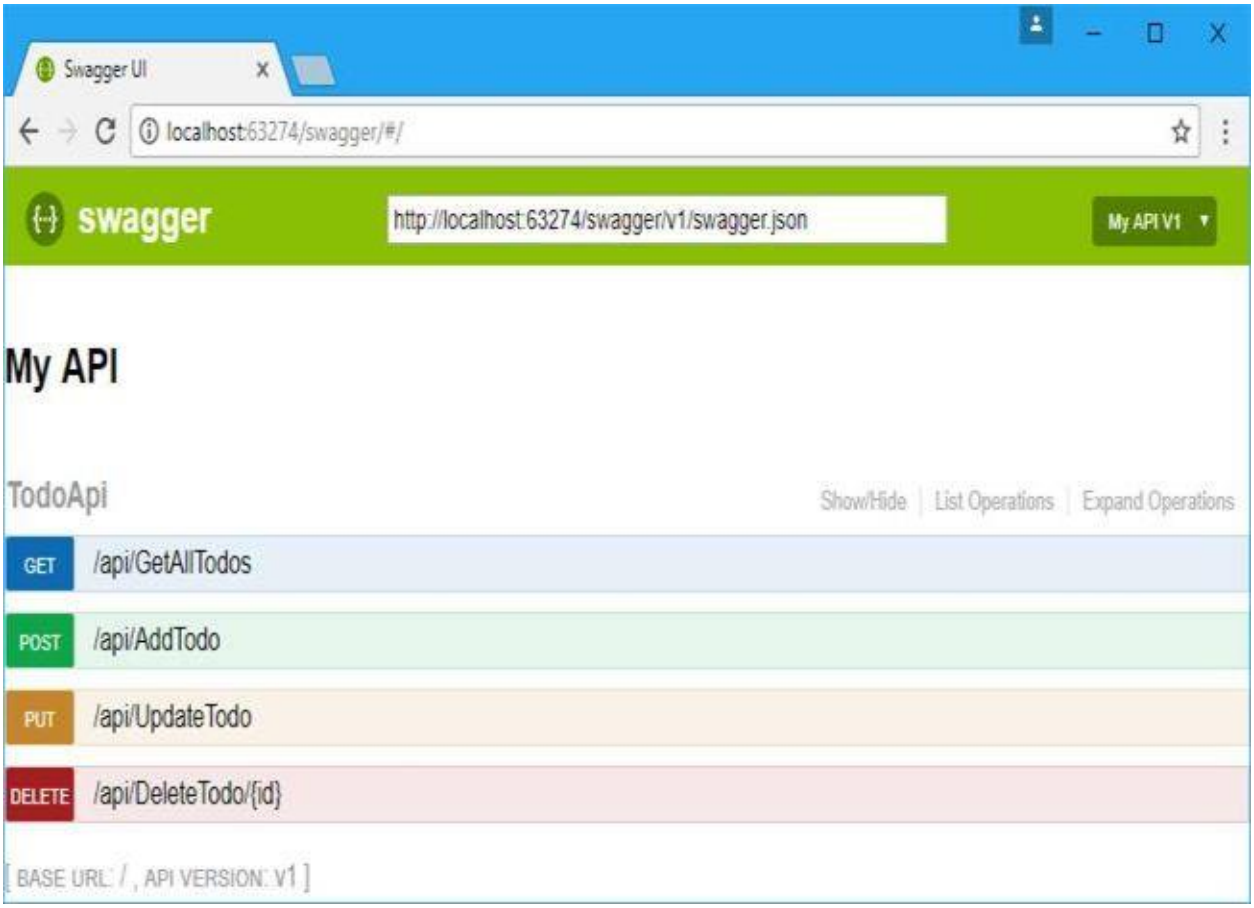
ما نیاز به یک پروژه وب API ساده داریم تا بتوانیم برخی از API های برنامه نمونه را آزمایش و اشکال زدایی کنیم.

یک پروژه ساده به نام TodoMvcSolution از لینک زیر دانلود کرده و اجرا کنید.

https://github.com/SmartITAz/TodoMvcSwaggerSolution



در اینجا API هایی هستند که می توانیم آنها را آزمایش کنیم. برای این برنامه نمونه Get, Post, Put, و Delete وجود دارد.



در اینجا Web APIs وجود دارد که ما می خواهیم تست کنیم:

```
//Copyright 2017 (c) SmartIT. All rights reserved.
//By John Kocer
// This file is for Swagger test, this application does not use this file
using System.Collections.Generic;
using Microsoft.AspNetCore.Mvc;
using SmartIT.Employee.MockDB;
namespace TodoAngular.Ui.Controllers
{
 [Produces("application/json")]
 [Route("api/Todo")]
 public class TodoApiController : Controller
 {
  TodoRepository _todoRepository = new TodoRepository();
```

```
    [Route("~/api/GetAllTodos")]
    [HttpGet]
    public IEnumerable<SmartIT.Employee.MockDB.Todo> GetAllTodos()
    {
      return _todoRepository.GetAll();
    }

    [Route("~/api/AddTodo")]
    [HttpPost]
    public SmartIT.Employee.MockDB.Todo AddTodo([FromBody]SmartIT.Employee.MockDB.Todo item)
    {
      return _todoRepository.Add(item);
    }

    [Route("~/api/UpdateTodo")]
    [HttpPut]
    public SmartIT.Employee.MockDB.Todo UpdateTodo([FromBody]SmartIT.Employee.MockDB.Todo item)
    {
      return  _todoRepository.Update(item);
    }

    [Route("~/api/DeleteTodo/{id}")]
    [HttpDelete]
    public void Delete(int id)
    {
      var findTodo = _todoRepository.FindById(id);
      if (findTodo != null)
        _todoRepository.Delete(findTodo);
    }
  }
}
```

نکته : local port number شما ممکن است متفاوت از ما باشد. از local port number خود استفاده کنید.

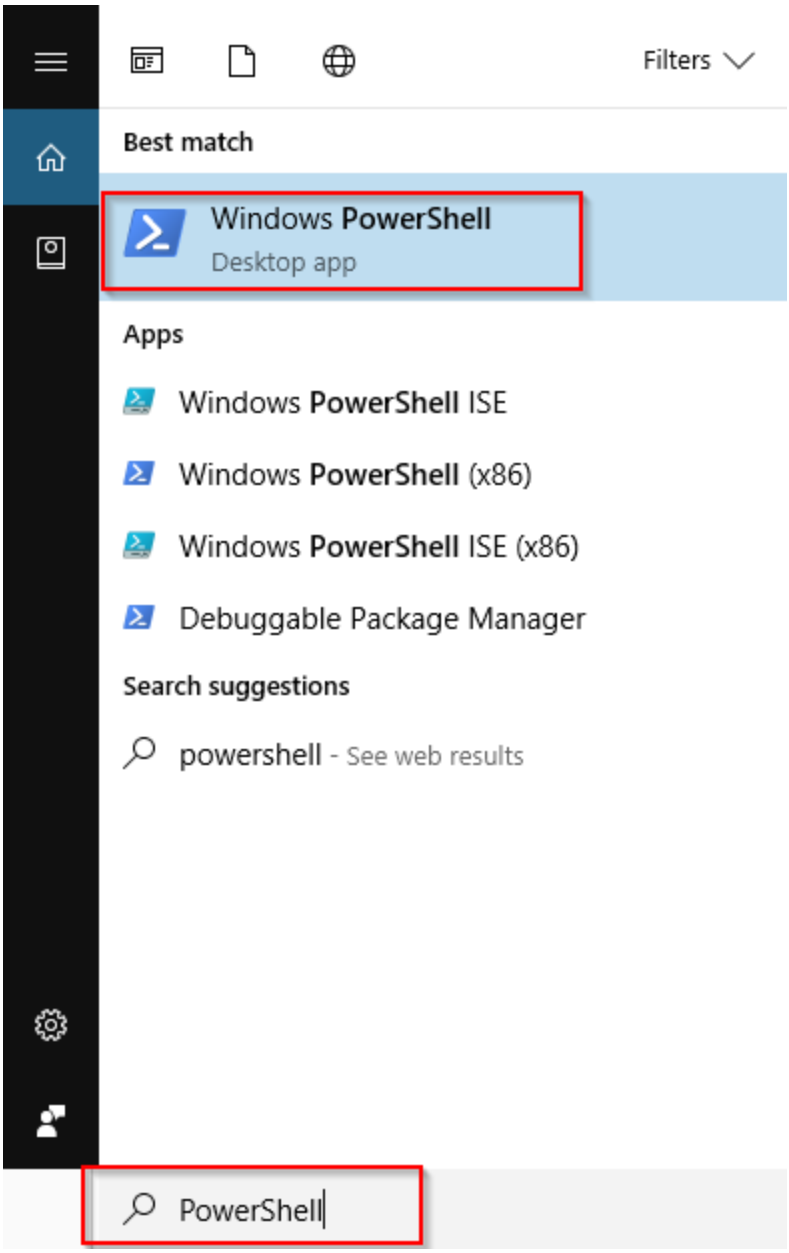
- http://localhost:63274/api/GetAllTodos // GET
- http://localhost:63274/api/AddTodo //POST
- http://localhost:63274/api/UpdateTodo //PUT
- http://localhost:63274/api/DeleteTodo/5 // DELETE

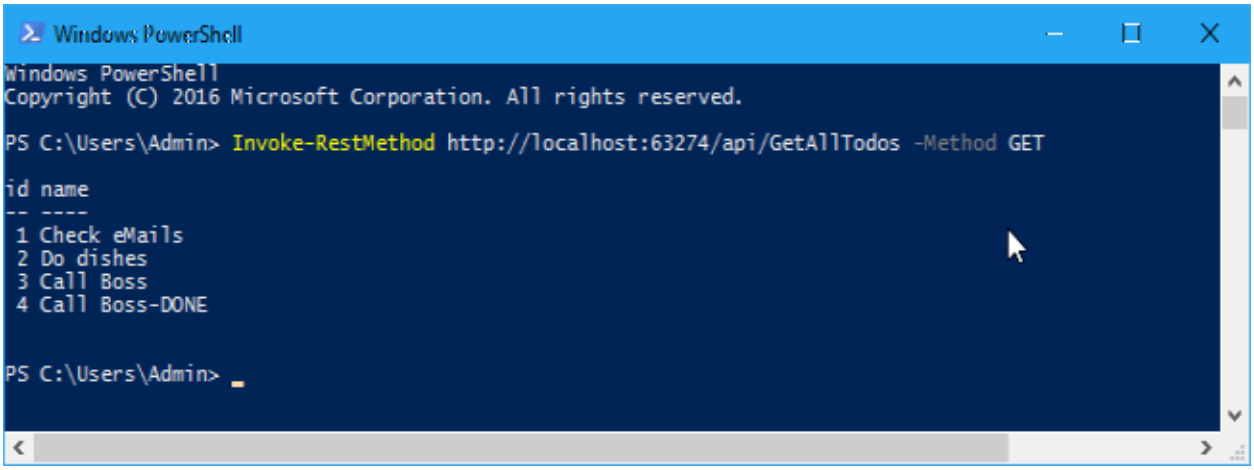
PowerShell commands for above HTTP APIs

- تست عملیاتGET
  - Invoke-RestMethod http://localhost:63274/api/GetAllTodos -Method GET
- تست عملیات POST
  - Invoke-RestMethod http://localhost:63274/api/AddTodo -Method POST -Body(@{id="0"; name="Call Boss-OK" } | ConvertTo-Json) -ContentType ""application/json
- تست عملیات PUT
  - Invoke-RestMethod http://localhost:63274/api/UpdateTodo -Method PUT -Body(@{id="4"; name="Call Boss-DONE" } | ConvertTo-Json) -ContentType "application/json"
- تست عملیات DELETE
  - Invoke-RestMethod http://localhost:63274/api/DeleteTodo/5 -Method DELETE

Start a PowerShell window

از جستجوی ویندوز، PowerShell را تایپ کرده و برنامه Windows PowerShell را انتخاب کنید.


Filters
Best match
Windows PowerShell
Desktop app
Apps
Windows PowerShell ISE
Windows PowerShell (x86)
Windows PowerShell ISE (x86)
Debuggable Package Manager
Search suggestions
powershell - See web results
PowerShell



Testing GET with PS

یک دستور GET برای PowerShell را کامپایل کنید و روی خط فرمان قرار دهید. (این یک خط پیوسته است)

Invoke-RestMethod http://localhost:63274/api/GetAllTodos -Method GET

Response

Windows PowerShell

Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Admin> Invoke-RestMethod http://localhost:63274/api/GetAllTodos -Method GET

id name

-- ----

1 Check eMails

2 Do dishes

3 Call Boss

4 Call Boss-DONE

PS C:\Users\Admin>

Testing POST with PS

یک دستور POST برای PowerShell را کامپایل کنید و روی خط فرمان قرار دهید. (این یک خط پیوسته است).

ما یک Body با آیتم our Todo تشکیل می دهیم ({ "id="0"; name="Call Boss-OK}).

```
Invoke-RestMethod http://localhost:63274/api/AddTodo -Method POST -Body(@{id="0";
"name="Call Boss-OK

"ConvertTo-Json) -ContentType "application/json | {
```

Response

ما یک Id= 4 را با یک مورد Todo درج شده مانند زیر دریافت کردیم.

```
PS C:\Users\Admin> Invoke-RestMethod http://localhost:63274/api/AddTodo -Method POST -
Body(@{id="0"; name="Call Boss-OK"

  } | ConvertTo-Json) -ContentType "application/json"

id name

-- ----

 4 Call Boss-OK

PS C:\Users\Admin>
```

Testing PUT with PS: Compile a PUT command for PowerShell and paste on to command line. (It is one continuous line)
We are composing a Body with our Todo item {id="0"; name="Call Boss-OK" }

```
Invoke-RestMethod http://localhost:63274/api/UpdateTodo -Method PUT -Body(@{id="4";
name="Call Boss-DONE"  } | ConvertTo-Json) -ContentType "application/json"
```

Response

ما یک Call =Boss-DONE name با یک Todo به روز شده دریافت کردیم.

```
PS C:\Users\Admin> Invoke-RestMethod http://localhost:63274/api/UpdateTodo -Method PUT -
Body(@{id="4"; name="Call Boss-D

ONE"  } | ConvertTo-Json) -ContentType "application/json"

id name
```

-- ----

4 Call Boss-DONE

PS C:\Users\Admin>

Testing DELETE with PS

یک دستور Delete برای PowerShell را کامپایل کنید و روی خط فرمان قرار دهید. (این یک خط پیوسته است).

Invoke-RestMethod http://localhost:63274/api/DeleteTodo/4 -Method DELETE

Response

ما هیچ اطلاعاتی دریافت نکرده ایم مانند زیر:

PS C:\Users\Admin> Invoke-RestMethod http://localhost:63274/api/DeleteTodo/4 -Method DELETE

PS C:\Users\Admin>

نکته : ما می توانیم همه اینها را در کنسول Visual Studio Package Manager به جای PowerShell انجام دهیم.

برای استفاده از Package Manager Console، کنسول PM را باز کنید و متد GET کامپایل شده را وارد کنید.

How to debug an application API?

ما می توانیم breakpoint را در ویژوال استودیو API قرار دهیم و سپس آن API را از PowerShell یا Package Manager Console اشکال زدایی کنیم.

این کار تست و اشکال زدایی Web API را تکمیل خواهد کرد.

Summary

در این مقاله ما یاد گرفتیم که چگونه Web APIs را آزمایش و اشکالزدایی کنیم. شما می توانید کد منبع را از لینک زیر دانلود کنید.

https://github.com/SmartITAz/TodoMvcSwaggerSolution