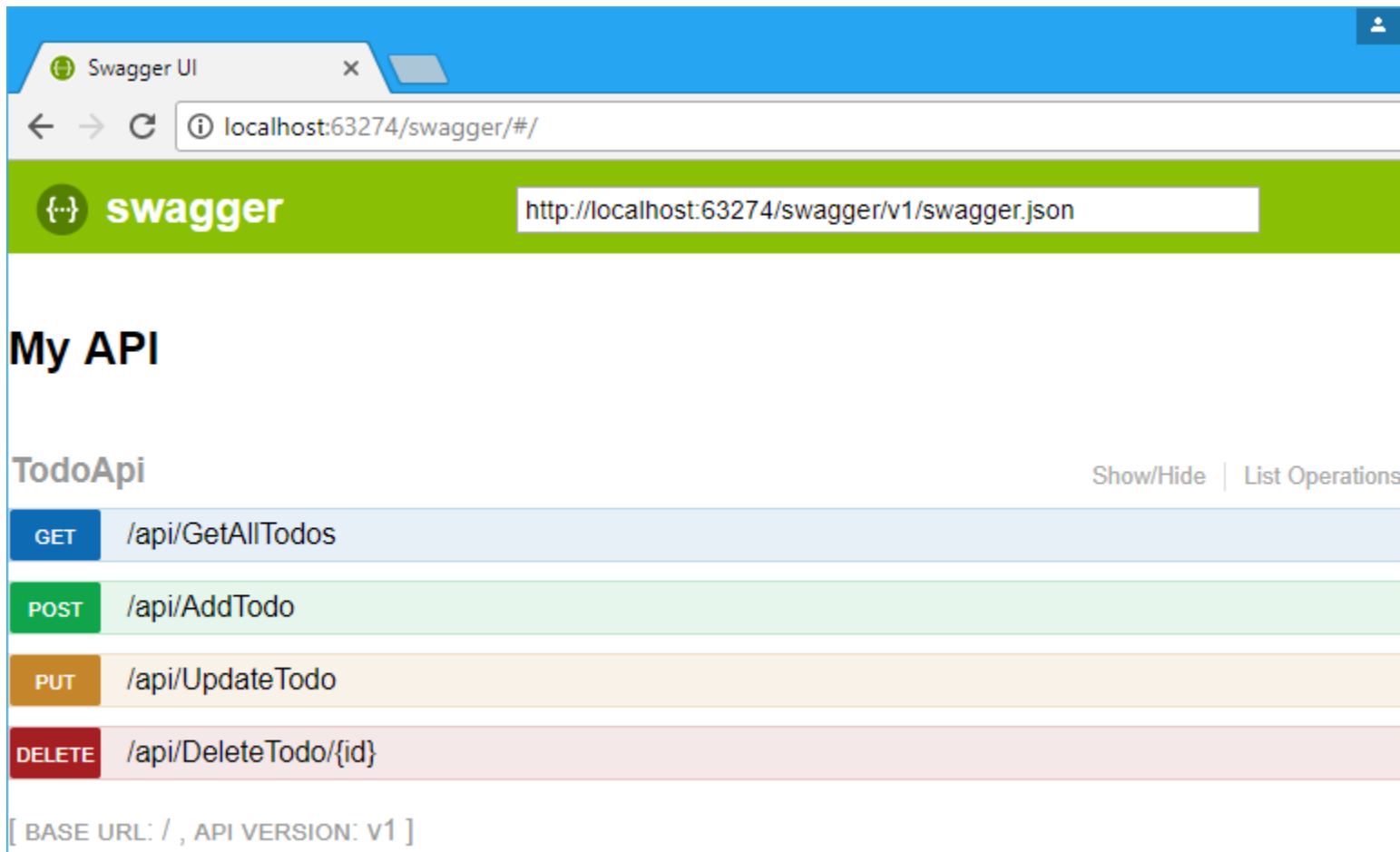


How to Use Swagger with ASP.NET Core Web APIs.

Visual Studio 2017, ASP.NET Core 2.0, By John Kocer- SmartIT

Tags: Swagger, Visual Studio 2017, ASP.NET Core, Web API, API Test, Do-to List, SmartIT



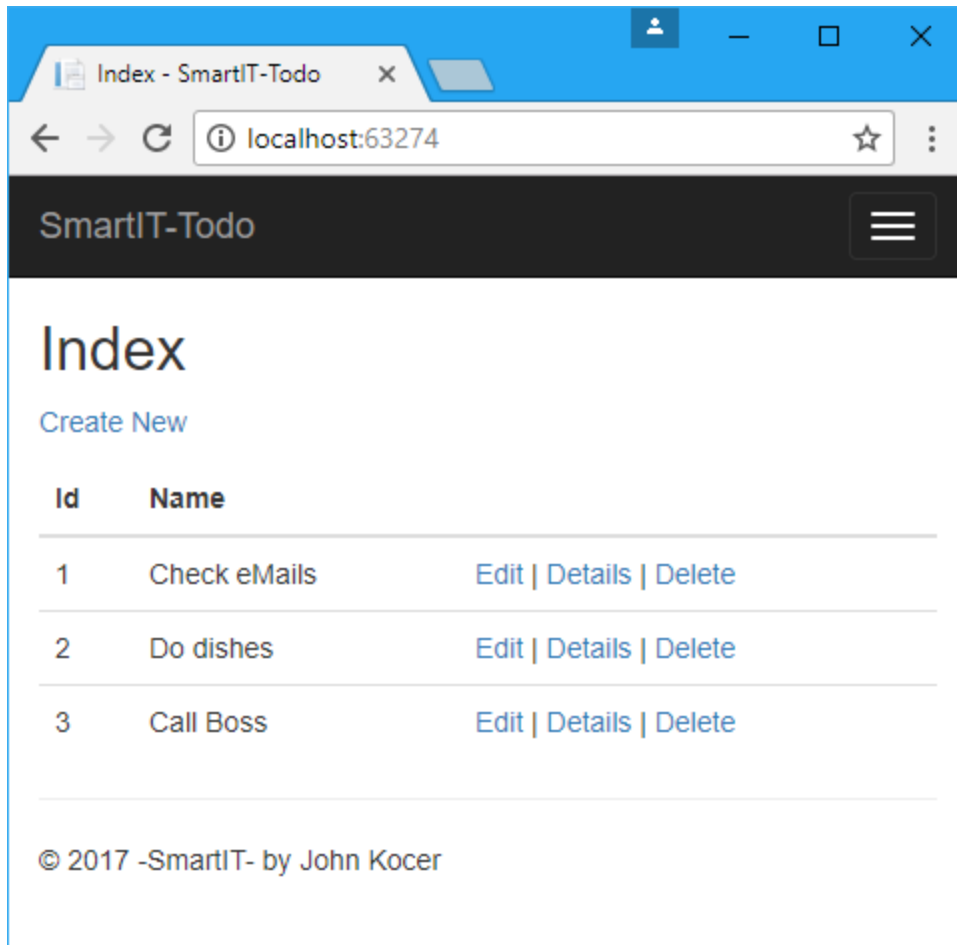
Visual Studio 2017, ASP.NET Core 2.0, By John Kocer- SmartIT

If you are developer, tester or a manager, sometimes understanding the various methods of API can be a challenge when building and consuming the application.

Generating good documentation and help pages for your Web API, using Swagger with .NET Core is as easy as adding a couple of NuGet and modifying the Startup.cs

Let's start a downloading simple To-do projects from Github.

1. Download and run below TodoMvcSolution from below link.
<https://github.com/SmartITAz/TodoMvcSolution>



2. NuGet Packages

Install the below NuGet package

```
Install-Package Swashbuckle.AspNetCore
```

Configure Swagger in the `Startup.cs`.

- Add the Swagger generator to the service collection after `services.AddMvc();`
- Enable the middleware for serving the generated JSON document after `app.UseStaticFiles();`
- Add the below background `yellow lines` in your `Startup` file.

```

using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Swashbuckle.AspNetCore.Swagger;

namespace Todo.Mvc.Ui
{
    public class Startup
    {
        public Startup(IConfiguration configuration)
        {
            Configuration = configuration;
        }

        public IConfiguration Configuration { get; }

        // This method gets called by the runtime. Use this method to add services to the container.
        public void ConfigureServices(IServiceCollection services)
        {
            services.AddMvc();

            // Register the Swagger generator, defining one or more Swagger documents
            services.AddSwaggerGen(c =>
            {
                c.SwaggerDoc("v1", new Info { Title = "My API", Version = "v1" });
            });
        }

        // This method gets called by the runtime. Use this method to configure the HTTP request
        pipeline.
        public void Configure(IApplicationBuilder app, IHostingEnvironment env)
        {
            if (env.IsDevelopment())
            {
                app.UseDeveloperExceptionPage();
                app.UseBrowserLink();
            }
            else
            {
                app.UseExceptionHandler("/Home/Error");
            }

            app.UseStaticFiles();

            // Enable middleware to serve generated Swagger as a JSON endpoint.
            app.UseSwagger();

            // Enable middleware to serve swagger-ui (HTML, JS, CSS, etc.), specifying the Swagger JSON
            endpoint.
            app.UseSwaggerUI(c =>
            {
                c.SwaggerEndpoint("/swagger/v1/swagger.json", "My API V1");
            });

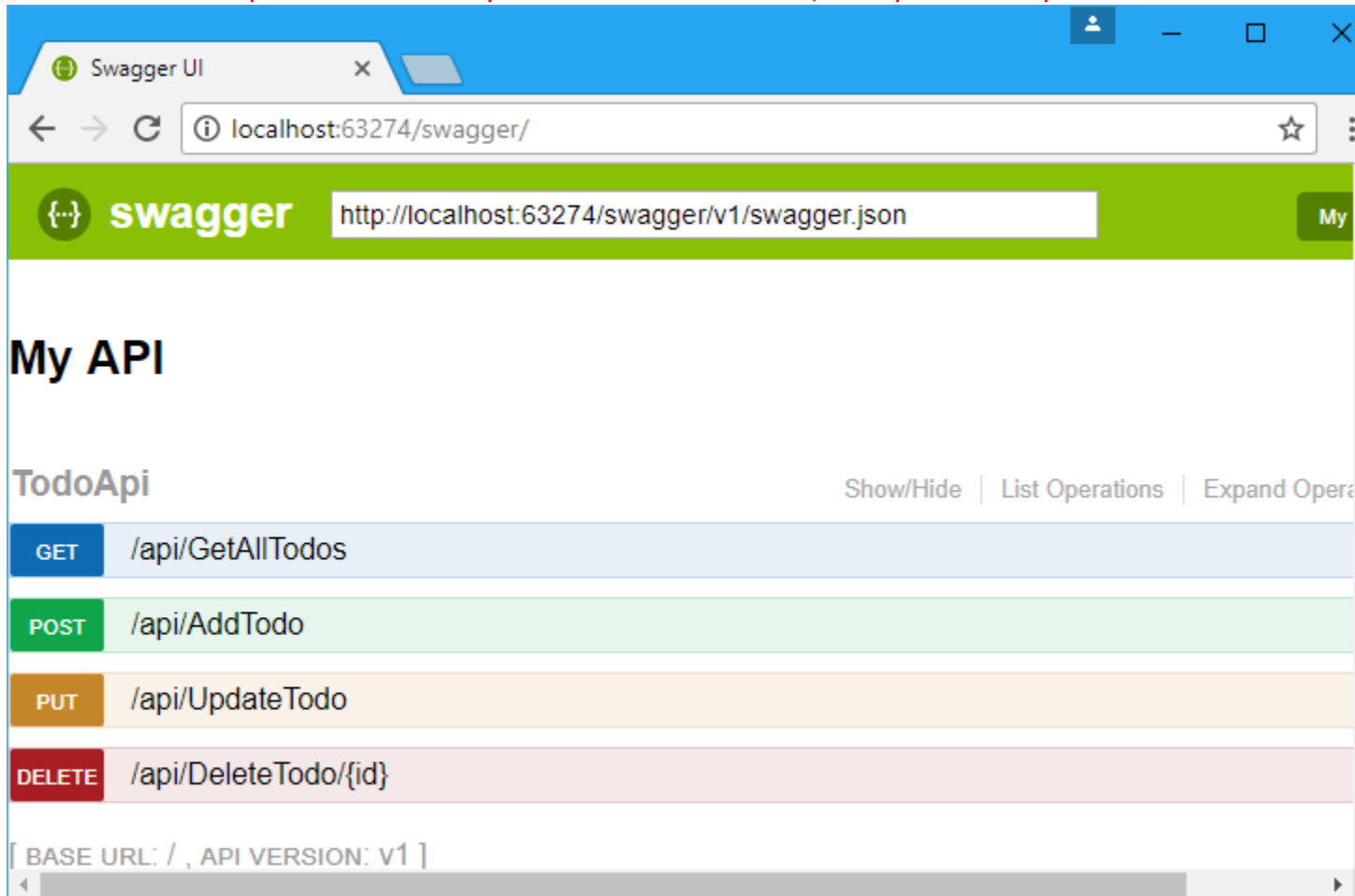
            app.UseMvc(routes =>
            {
                routes.MapRoute(
                    name: "default",
                    template: "{controller=Todo}/{action=Index}/{id?}");
            });
        }
    }
}

```

```
}
```

Go to <http://localhost:63274/swagger/>

Note: Your local port number may be different than our, use your local port number.



-Click on Get Try it out and you will see the response in Response body section.

Swagger UI

localhost:63274/swagger/#!/TodoApi/ApiGetAllTodosGet

GET /api/GetAllTodos

Response Class (Status 200)

Success

Model | Example Value

```
[
  {
    "id": 0,
    "name": "string"
  }
]
```

Response Content Type

Try it out! [Hide Response](#)

Curl

```
curl -X GET --header 'Accept: application/json' 'http://localhost:63274/api/GetAllTodo
```

Request URL

```
http://localhost:63274/api/GetAllTodos
```

Response Body

```
[
  {
    "id": 1,
    "name": "Check eMails"
  },
  {
    "id": 2,
    "name": "Do dishes"
  },
  {
    "id": 3,
    "name": "Call Boss"
  }
]
```

Here is the Web APIs

```
//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);
        }
    }
}
```

NOTE:

We have two controller files TodoController and TodoApiController. TodoController is MVC Controller and Swagger did not show the API methods, because it does not know how to route and also controller methods return a View.

```

public class TodoController : Controller
{
    IRepository _todoRepository = new IRepository();
    // GET: Todo
    public ActionResult Index()
    {
        var todos = (List<SmartIT.Employee.MockDB.Todo>)_todoRepository.GetAll();
        return View(todos);
    }

    // GET: Todo/Details/5
    public ActionResult Details(int id)
    {
        var findTodo = _todoRepository.FindById(id);
        return View(findTodo);
    }

    // GET: Todo/Create
    public ActionResult Create()
    {
        return View();
    }

    // POST: Todo/Create
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Create(IFormCollection collection)
    {
        try
        {
            _todoRepository.Add(new SmartIT.Employee.MockDB.Todo() { Name = collection["Name"] });

            return RedirectToAction(nameof(Index));
        }
        catch
        {
            return View();
        }
    }
}

```

Summary

In this article, we will learned

How to Use Swagger with ASP.NET Core Web APIs

Download color pdf version of this article with pictures.

<https://github.com/SmartITAz/ToDoMvcSwaggerSolution/blob/master/SwaggerAspNetCoreVs2017.pdf>

NOTE: If you need to copy and paste the code download the pdf file locally.

Download source code from GitHub: <https://github.com/SmartITAz/ToDoMvcSwaggerSolution>

Visual Studio 2017, ASP.NET Core 2.0, By John Kocer- SmartIT

Tags: Swagger, Visual Studio 2017, ASP.NET Core, Web API, API Test, Do-to List, SmartIT