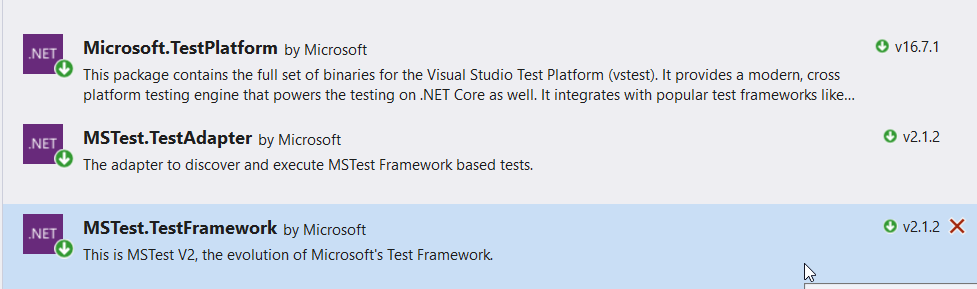
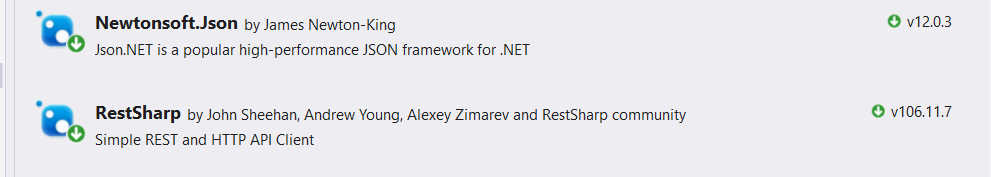
# How to Automate REST API with RESTSHARP .NET C#

1. Add Class libray .Net Framework
2. Add NuGet Packages.





1. Add REST API Helper Class. Capture Response Content. Verify response Status Code & Content, Deserialized Content

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Newtonsoft.Json;

using RestSharp;

using System.IO;

using System.Collections;

namespace RESTAPIDemo

{

public class RESTAPIHelper<T>

{

public RestClient restClient;

public RestRequest restRequest;

public string baseUrl = "";

public RestClient SetUrl(string URL, string endpoint)

{

this.baseUrl = URL;

var url = Path.Combine(baseUrl, endpoint);

var restClient = new RestClient(url);

return restClient;

}

public RestRequest CreatePostRequest(string payload)

{

var restRequest = new RestRequest(Method.POST);

restRequest.AddHeader("Accept", "application/json");

restRequest.AddParameter("application/json", payload, ParameterType.RequestBody);

return restRequest;

}

public RestRequest CreatePutRequest(string payload)

{

var restRequest = new RestRequest(Method.PUT);

restRequest.AddHeader("Accept", "application/json");

restRequest.AddParameter("application/json", payload, ParameterType.RequestBody);

return restRequest;

}

public RestRequest CreateGetRequest(Hashtable parametersList)

{

var restRequest = new RestRequest(Method.GET);

//Add header parameters

restRequest.AddHeader("Accept", "\*/\*");

//Add URL parameters

if (parametersList != null)

{

foreach (DictionaryEntry de in parametersList)

{

restRequest.AddQueryParameter(de.Key.ToString(), de.Value.ToString());

}

}

return restRequest;

}

public RestRequest CreateDeleteRequest()

{

var restRequest = new RestRequest(Method.DELETE);

restRequest.AddHeader("Accept", "application/json");

return restRequest;

}

public IRestResponse GetResponse(RestClient client, RestRequest request)

{

return client.Execute(request);

}

public DTO GetContent<DTO>(IRestResponse response)

{

var content = response.Content;

DTO dtoObject = JsonConvert.DeserializeObject<DTO>(content);

return dtoObject;

}

}

}

1. Create DTO folder, Convert JSON to C#, Add Class ‘Employee’ in it.

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using RESTAPIDemo;

namespace RESTDEMO1.DTO

{

class Employee

{

public class DataEmployee

{

public int id { get; set; }

public string employee\_name { get; set; }

public int employee\_salary { get; set; }

public int employee\_age { get; set; }

public string profile\_image { get; set; }

}

public class RootEmployee

{

public string status { get; set; }

public DataEmployee data { get; set; }

public string message { get; set; }

}

public static RootEmployee CreateEmployee(string baseUrl, string endpoint)

{

var user = new RESTAPIHelper<RootEmployee>();

var url = user.SetUrl(baseUrl, endpoint);

var jsonReq = @"{

""name"": ""farah"",

""salary"": ""2000"",

""age"": ""30"",

}";

var request = user.CreatePostRequest(jsonReq);

var response = user.GetResponse(url, request);

RootEmployee content = user.GetContent<RootEmployee>(response);

return content;

}

internal static RootEmployee GetEmployeeDetails(string v1, string v2)

{

throw new NotImplementedException();

}

public static RootEmployee GetEmployeeDetails(string baseUrl, string endpoint, Hashtable parameterList)

{

var user = new RESTAPIHelper<RootEmployee>();

var url = user.SetUrl(baseUrl, endpoint);

var request = user.CreateGetRequest(parameterList);

var response = user.GetResponse(url, request);

RootEmployee content = user.GetContent<RootEmployee>(response);

return content;

}

}

}

1. Add REST API Execution Test Method

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using RESTDEMO1.DTO;

using static RESTDEMO1.DTO.Employee;

namespace TestProject.Execution

{

[TestClass]

public class RESTAPIExecution

{

[TestMethod, TestCategory("Employee"), Description("Get Employee ")]

public void GetEmployee()

{

/\*Hashtable parametersList = new Hashtable

{

{ "id", "1"}

};\*/

Hashtable parametersList = null;

RootEmployee employeeData = Employee.GetEmployeeDetails("http://dummy.restapiexample.com/", "api/v1/employee/1", parametersList);

Assert.AreEqual(employeeData.data.id,1);

}

[TestMethod, TestCategory("Employee"), Description("Create Employee ")]

public void CreateEmployee()

{

RootEmployee employee = Employee.CreateEmployee("http://dummy.restapiexample.com/", "api/v1/create");

Assert.AreEqual(employee.status, "success");

Assert.AreEqual(employee.message, "Successfully! Record has been added.");

}

}

}