A proper way
to develop a REST service

Sergey Ogorodnikov
Senior C# developer
One Inc
Omsk, 4/18/2019

Use Swagger, Luke!

Swagger is an open-source software framework backed by a large ecosystem of tools that helps developers design, build, document, and consume RESTful Web services.

While most users identify Swagger by the Swagger UI tool, the Swagger toolset includes support for automated documentation, code generation, and test-case generation.

https://swagger.io/

The OpenAPI Specification

The **OpenAPI Specification** defines a standard, programming language-agnostic interface description for REST APIs, which allows both humans and computers to discover and understand the capabilities of a service

https://www.openapis.org

Swashbuckle

Swashbuckle **by:** domaindrivendev

Swashbuckle - Swagger for WebApi

↓ 8,180,841 total downloads | 🕐 last updated 7/8/2017 | ⚑ Latest version: 5.6.0

Seamlessly adds a Swagger to WebApi projects!

```csharp
public static void Register()
{
    GlobalConfiguration.Configuration
        .EnableSwagger(c =>
            {
                c.SingleApiVersion("v1", "OneInc.ProcessOne.Web.API");

                c.SchemaFilter<ReusableEnumSchemaFilter>();

                c.SchemaId(t => GetSchemaId(t));
            })
        .EnableSwaggerUi(c => { });
}
```

ProcessOne API Specification

```
swagger:                                    "2.0"
info:
    version:                                "v1"
    title:                                  "OneInc.ProcessOne.Web.API"
host:                                       "localhost"
basePath:                                   "/ProcessOne"
schemes:                                    [-]
paths:
    /api/CreditCard/Save:
        post:
            tags:
                0:                          "CreditCard"
            operationId:                    "CreditCard_Save"
            consumes:                       [-]
            produces:                       [-]
            parameters:
                0:
                    name:                   "request"
                    in:                     "body"
                    required:               true
                    schema:
                        $ref:               "#/definitions/SaveCreditCardRequest"
            responses:
                200:
                    description:            "OK"
                    schema:
                        $ref:               "#/definitions/SaveCreditCardResponse"
    /api/CreditCard/Update:                 {-}
    /api/CreditCard/Charge:                 {-}
```

ProcessOne API Specification

```yaml
definitions:
    SaveCreditCardRequest:
        required:
            0:                                      "authenticationKey"
        type:                                       "object"
        properties:
            creditCard:
                $ref:                               "#/definitions/CreditCardDetails"
            callbackId:
                type:                               "string"
            clientReferenceData:
                $ref:                               "#/definitions/ClientReferenceData"
            authenticationKey:
                type:                               "string"
    CreditCardDetails:
        type:                                       "object"
        properties:
            expirationYear:
                format:                             "int32"
                type:                               "integer"
            expirationMonth:
```

Swagger-UI

swagger `http://localhost/ProcessOne/swagger/docs/v1` `api_key` **Explore**

OneInc.ProcessOne.Web.API

CreditCard   Show/Hide | List Operations | Expand Operations

POST /api/CreditCard/Save

POST /api/CreditCard/Update

POST /api/CreditCard/Charge

Response Class (Status 200)
OK

Model | Example Value

```
{
  "token": "string",
  "transactionId": "string",
  "transactionDate": "2019-01-14T20:04:38.395Z",
  "timeZone": "string",
  "authorizationCode": "string",
  "batchNumber": 0,
  "creditCardType": 0,
  "holderName": "string",
  "holderZip": "string",
```

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
request	(required)		body	Model | Example Value

```
{
  "amount": 0,
  "chargeFee": true,
  "token": "string",
```

7

Swagger-codegen

swagger-api/swagger-codegen   ● HTML   ★ 9.1k

swagger-codegen contains a template-driven
engine to generate documentation, API clients and
server stubs in differen…

`swagger-codegen` `openapi-specification` `rest`

Updated 14 hours ago   368 issues need help

Supported languages (with code and mustache templates):
- Java (client, service, JMeter, different http libraries, frameworks), Scala, Kotlin
- CSharp (dotnet2, dotnet4, .Net Core, client, controller)
- Typescript (angualr, angularjs, fetch, aurelia, jquery, node...)
- python, php, ruby/groovy, rust, r, perl, go…
- ada, eiffel…
- confluence/wiki

https://docs.microsoft.com/en-us/aspnet/core/tutorials/getting-started-with-swashbuckle

8

Swagger-codegen + crutch

Our custom swagger-codegen with a little crutch:

PROC-12940 add nuspec file to generate nuget package with custom swagger-codegen

PROC-12940 Support x-ms-enum

PROC-12940 A swagger-codegen repository is created from commit ddc4154de984a4d219fabb27ed5a3d65aac86bdb, tag v2.4.0

```
c.SchemaFilter<ReusableEnumSchemaFilter>();
```

A script to generate client libraries with our custom swagger-

```
C:\Projects\processone\SDK\SwaggerGenerator\generate.bat
"../../.nuget/nuget.exe" install -outputDirectory "./packages/" -excludeVersion -configFile "../..

#generate OneInc.ProcessOne.RestApi.Client.Java
java -jar ./packages/OneInc.ProcessOne.Libs.SwaggerCodegen/lib/swagger-codegen-cli.jar generate ^
    -i http://localhost/processone/swagger/docs/v1 ^
    -l java ^
    -o ./../OneInc.ProcessOne.RestApi.Client.Java/ ^
    -c ./Configuration/java.json ^
    --http-user-agent processone-client
```

ProcessOne Java SDK

Autorest

Azure/autorest   ● TypeScript   ★ 2.3k

OpenAPI (f.k.a Swagger) Specification code
generator. Supports C#, PowerShell, Go, Java,
Node.js, TypeScript, Python,...

`openapi`   `javascript`   `ruby`   `python`   `java`
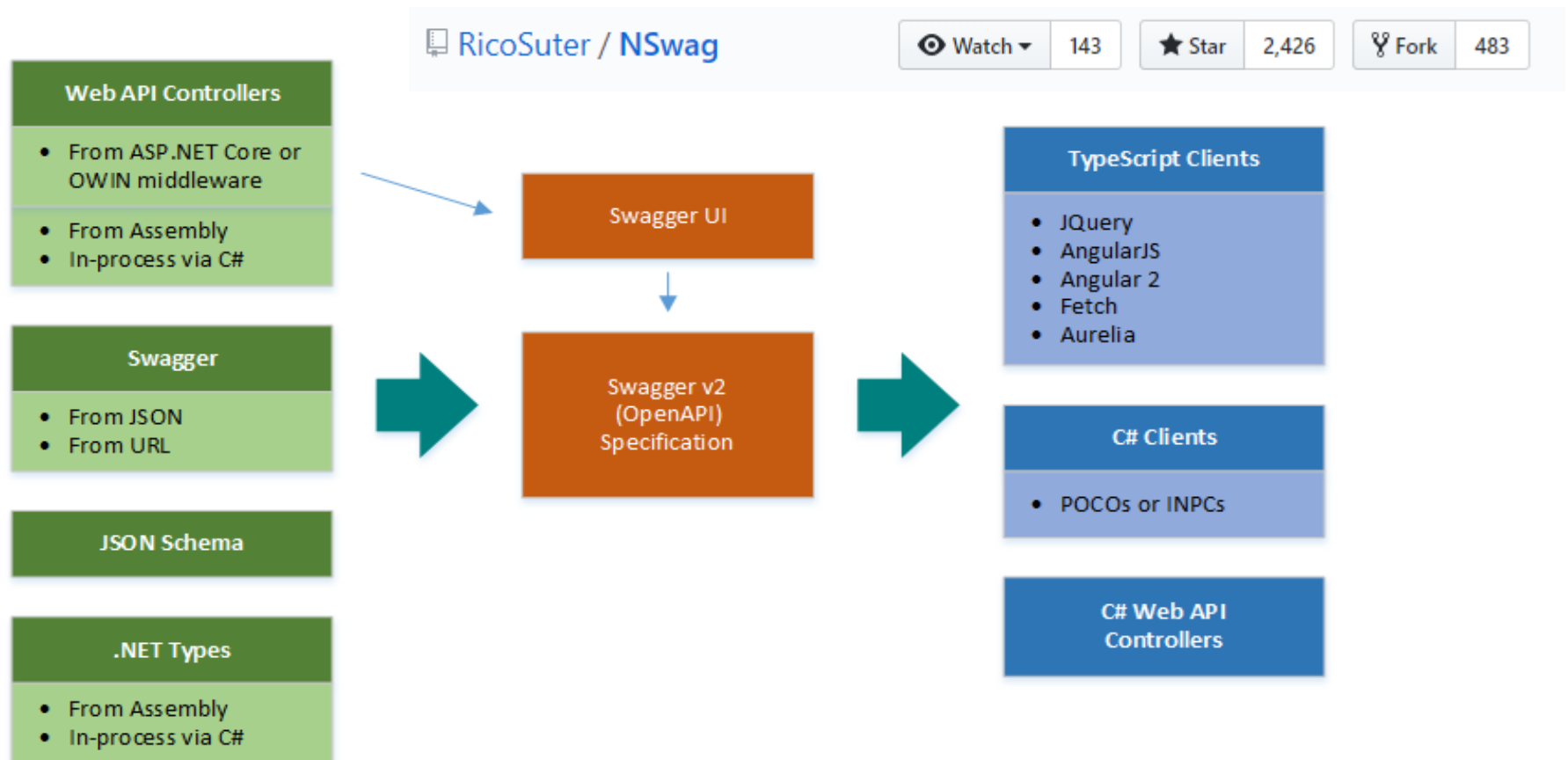
MIT license   Updated 19 hours ago

Made by Microsoft, distributed as NPM package, runs on Node.js

Supports languages (with extension packages):
- CSharp both client and service (generated code uses Microsoft.Rest.ClientRuntime package; supports .Net standart)
- Typescript
- Java
- Python

NSwag

The project combines the functionality of Swashbuckle (Swagger generation) and AutoRest (client generation) in one toolchain. This way a lot of incompatibilites can be avoided and features which are not well described by the Swagger specification or JSON Schema are better supported (e.g. inheritance, enum and reference handling).

12

Generate client for frontend

*.tst template file should be added to solution:

```
$Classes(*Model)[ // Find all classes with a name ending with Model
    class $Name {
        constructor($Properties[public $name: $Type][, ]) {
        }
    }
]
```

13

Generate client for frontend

Our custom utility

- creates typescript client using just compiled assembly metadata
- may work as standalone tool
- is a part of build process

Go futher

Each microservice may use its own client

+ will implement only needed methods
+ decoupling
+ no need to write client foreach service

- should write more code

Use HATEOAS/HAL - let REST be self-descriptive in runtime
 + decoupling
 + independent from paths

 - difficult to implement

Questions?