



Embarcadero Conference

Documente sua API REST em Delphi utilizando SwagDoc

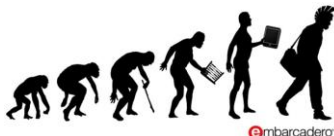
Marcelo Jaloto



Embarcadero

Conference

Marcelo Jaloto – Resumo



- + de 18 anos de experiência na área de desenvolvimento de softwares;
- Pós-graduado em Engenharia de Softwares;
- Certificação Delphi Developer pela Embarcadero Technologies;
- Certificação Scrum Master pela Scrum Alliance;
- Especialista na linguagem Delphi;
- Arquiteto e Desenvolvedor de software sênior;
- Professor Delphi, orientação a objetos e lógica de programação;
- Palestrante e escritor de artigos técnicos...



Para mais informações acesse o meu linkedin:
<https://www.linkedin.com/in/marcelojaloto>



Motivação para criação do SwagDoc



A demanda de integrações exigiu a necessidade de uma boa documentação da API;



Não encontramos nenhuma biblioteca em Delphi com a finalidade exclusiva para gerar documentação Swagger;



A mais popular ferramenta para gerar documentação de uma API REST;

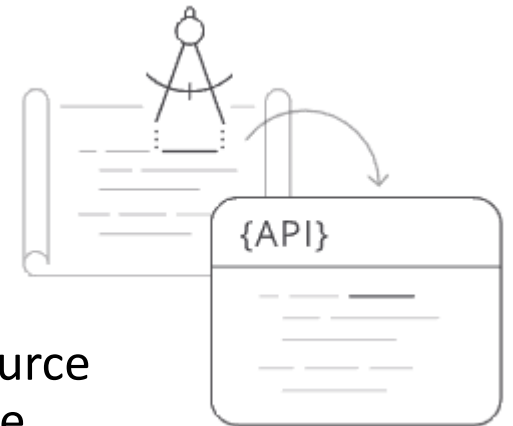


Nasceu então o SwagDoc, a primeira biblioteca open source para gerar documentação de uma API REST em Delphi.

<https://github.com/marcelojaloto/SwagDoc>



O time da Seventh então teve uma ideia de solução para ser desenvolvida de forma open source;



Porque documentar uma API REST?



1 Cria um contrato de uso bem definido e detalhado;

2 Facilita a comunicação;

3 Facilita o trabalho em paralelo;

4 Ganha produtividade de desenvolvimento;

5 Facilita a equipe de testes garantir qualidade;

6 Gera uma manual de consulta rápido;

7 Promove o sucesso de uma API.

Agenda



- ~~Introdução~~
- API REST
- Swagger
- Especificação OpenAPI
- Especificação Json Schema
- SwagDoc
- O Código Fonte
- Demonstração de Exemplos
- Deploy de arquivos Swagger
- Extras...

API REST

Transferência de Estado

API Employee

GET /employees

POST /employees

GET /employees/{id}

PUT /employees/{id}

DELETE /employees/{id}

API REST - SERVER
(fornecedor de serviços)

Response:

HTTP/1.1 200 OK
Date: Sun, 14 Oct 2018 23:06:58 GMT
Content-Type: application/json

```
{  
  "employees": [...]  
}
```

Request:
Get /employees/10 HTTP/1.1

HTTP

CLIENT

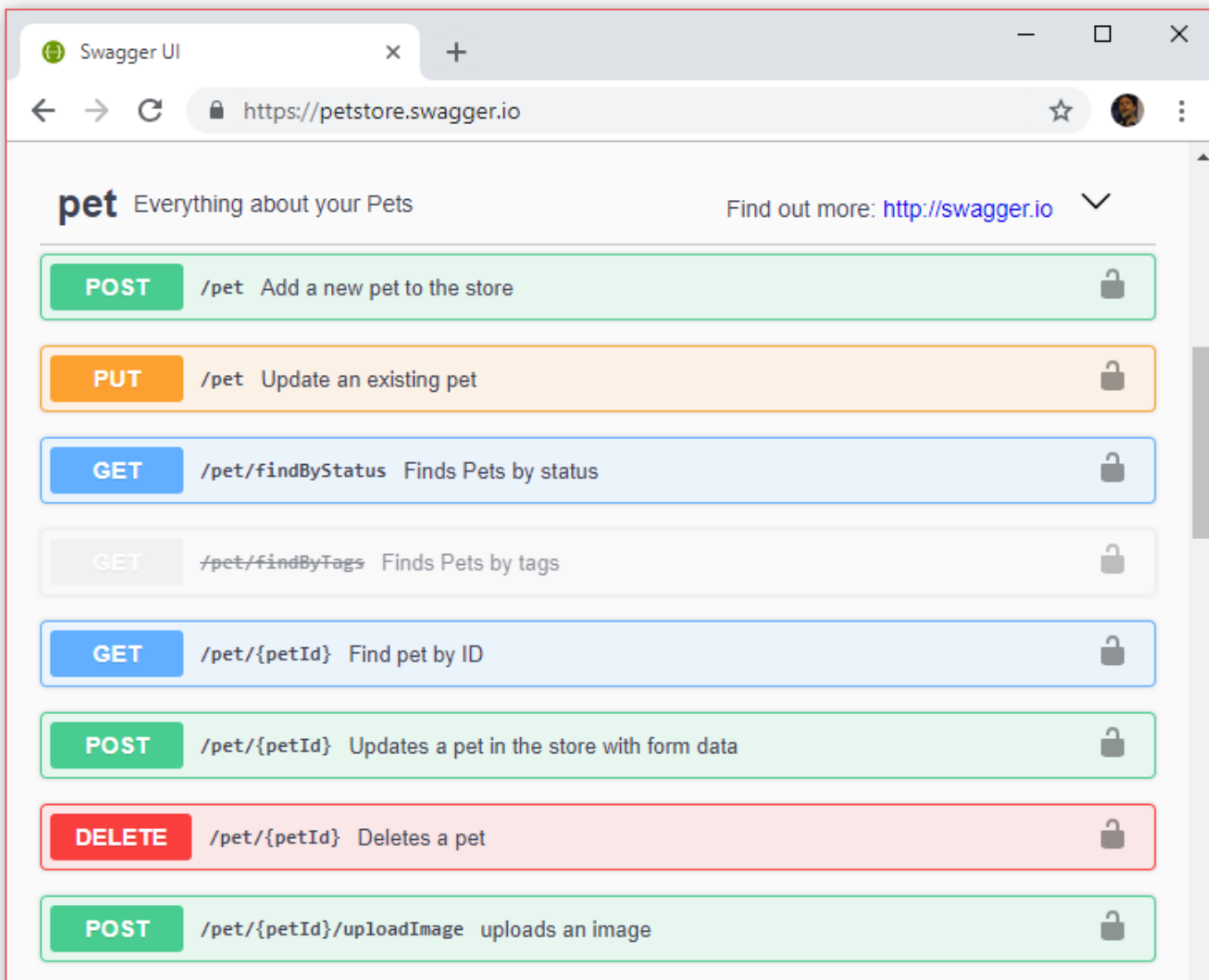
(consumidor do serviço)

```
{  
  "employees": [  
    {  
      "employee": {  
        "id": 0,  
        "name": "string",  
        "phone": "string",  
        "hireDate": "string",  
        "salary": 0  
      }  
    }  
  ]  
}
```

Representação do recurso
em formato JSON

- **API:** Application Programming Interface (Interface de Programação de Aplicações)
É de um conjunto de funcionalidades fornecidos para ser consumidos por uma aplicação.
- **REST:** REpresentational State Transfer (Transferência de Estado Representacional)
É um modelo de arquitetura criado por Roy Fielding, que define um conjunto de restrições e propriedades baseados em HTTP.

Swagger



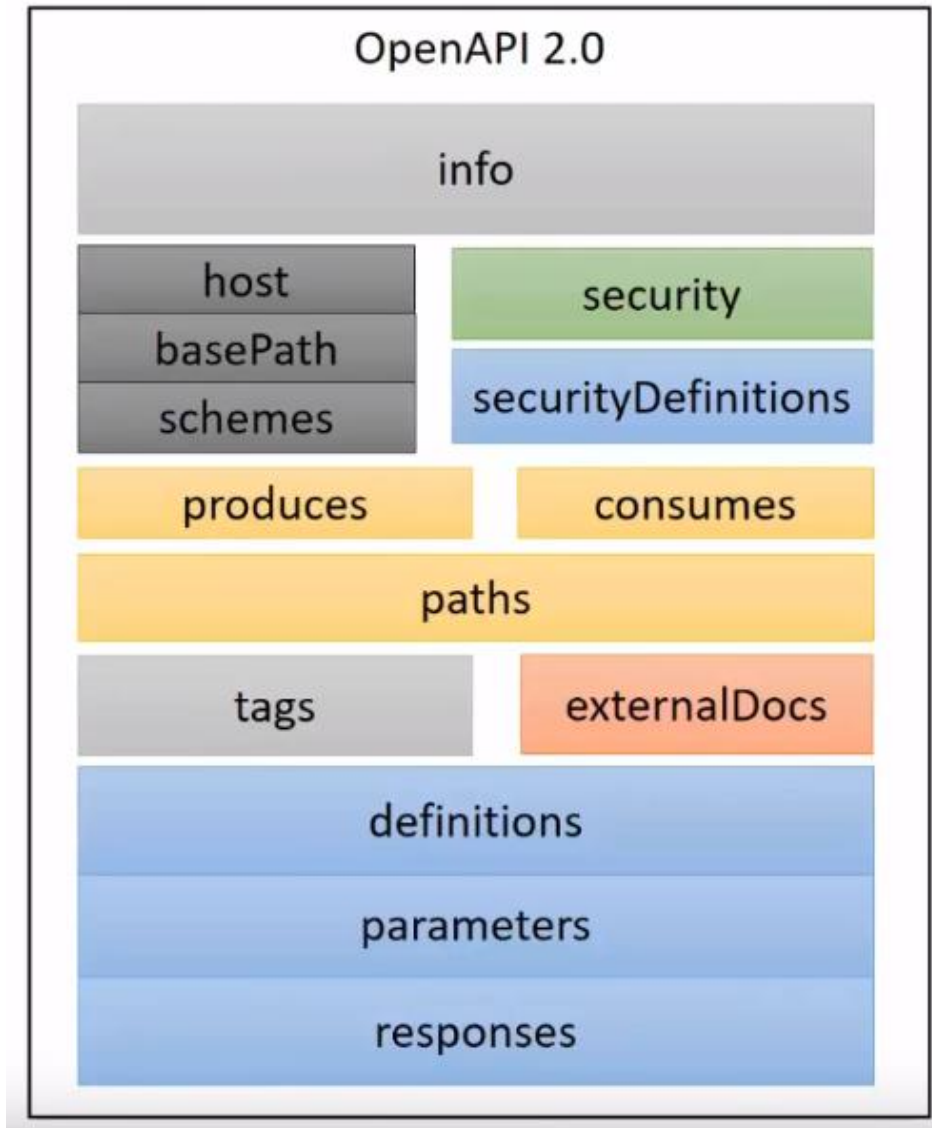
<https://petstore.swagger.io>



O Swagger é um conjunto de ferramentas, de desenvolvedor de APIs baseados na OpenAPI.

<https://swagger.io>

OpenAPI Specification (OAS) 2.0



Originalmente conhecido como Swagger Specification, é uma especificação para produzir uma documentação interativa, no modelo OpenAPI de serviços de uma API REST.

Porque 2.0?

Mais popular versão no mercado atualmente.

Especificação atual

<https://github.com/OAI/OpenAPI-Specification>

Especificação 2.0 completa

<https://swagger.io/docs/specification/2-0>

Referência seguida no SwagDoc

<https://github.com/OAI/OpenAPI-Specification/blob/master/versions/2.0.md>

Documentação interativa

<http://openapi-map.apihandyman.io/?version=2.0>

Json Schema Specification

```
▼ {  
  employee  
  
  ▼ {  
    description: Employee request data  
    id integer($int64)  
      The employee identification code.  
    name* string  
      maxLength: 80  
      The employee full name.  
    phone string  
      The employee phone number.  
    hireDate string($yyyy-MM-dd)  
      The employee hire date.  
    salary number  
      The employee gross salary.  
    address  
      ▼ {  
        description: The employee full address.  
        description string  
          The employee address description.  
        city string  
          The employee address city.  
        region string  
          The employee address region.  
        country string  
          The employee address country.  
        postalCode string  
          The employee address postal code.  
      }  
  }  
}
```

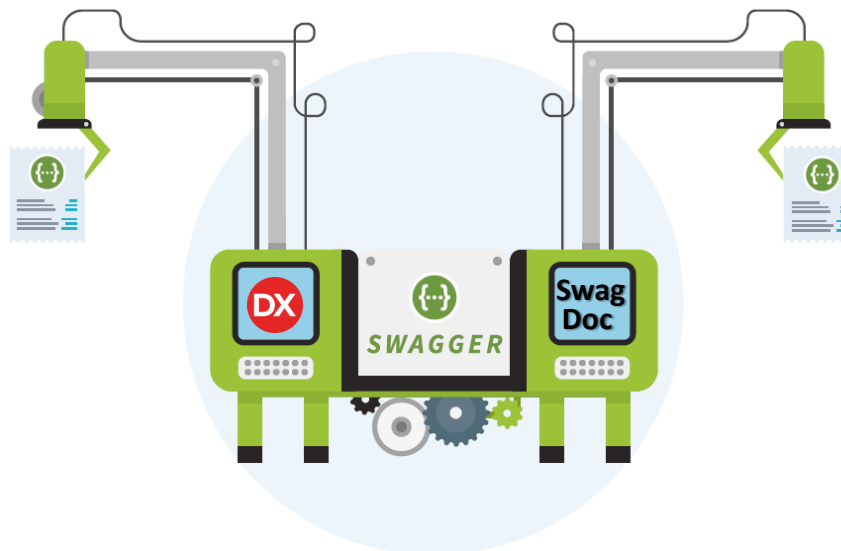


O esquema JSON é um vocabulário que permite anotar e validar documentos JSON.

<http://json-schema.org>

Swagger

```
1 {
2   "swagger": "2.0",
3   "info": {
4     "version": "v1",
5     "title": "Sample API",
6     "description": "Sample API Description",
7     "termsOfService": "http://www.apache.org/licenses/LICENSE-2.0",
8     "contact": {
9       "name": "Marcelo Jaloto",
10      "email": "marcelojaloto@gmail.com",
11      "url": "https://github.com/marcelojaloto/SwagDoc"
12    },
13    "license": {
14      "name": "Apache License - Version 2.0, January 2004",
15      "url": "http://www.apache.org/licenses/LICENSE-2.0"
16    }
17  },
18  "host": "localhost",
19  "basePath": "/api",
20  "schemes": [
21    "http"
22  ],
23  "consumes": [
24    "application/json"
25  ],
26  "produces": [
27    "application/json"
28  ],
29  "paths": {
30    "/employees": {
31      "get": {
32        "description": "Returns a employee list.",
33        "operationId": "{A9A8D343-00EB-402A-8248-94BDA7B6ECD4}",
34        "tags": [
35          "Employee"
36        ],
37        "responses": {
38          "200": {
39            "description": "Successfully returns data",
40            "schema": {
41              "type": "object",
42              "properties": {
43                "employees": {
44                  "type": "array",
45                  "description": "The employee datas.",
46                  "items": {
47                    "type": "object",
48                    "properties": {
```



O que é SwagDoc?

É uma biblioteca Delphi que tem como única responsabilidade gerar o arquivo swagger.json, conforme a especificação OpenAPI 2.0 (OAS) e Json Schema, para produzir como principal finalidade, uma documentação de uma API REST, que pode ser carregada na Swagger UI.

Sample API ^{v1}

[Base URL: localhost/api]

Sample API Description

[Terms of service](#)

[Marcelo Jaloto - Website](#)

[Send email to Marcelo Jaloto](#)

[Apache License - Version 2.0, January 2004](#)

Schemes

HTTP

API Employee

GET /employees

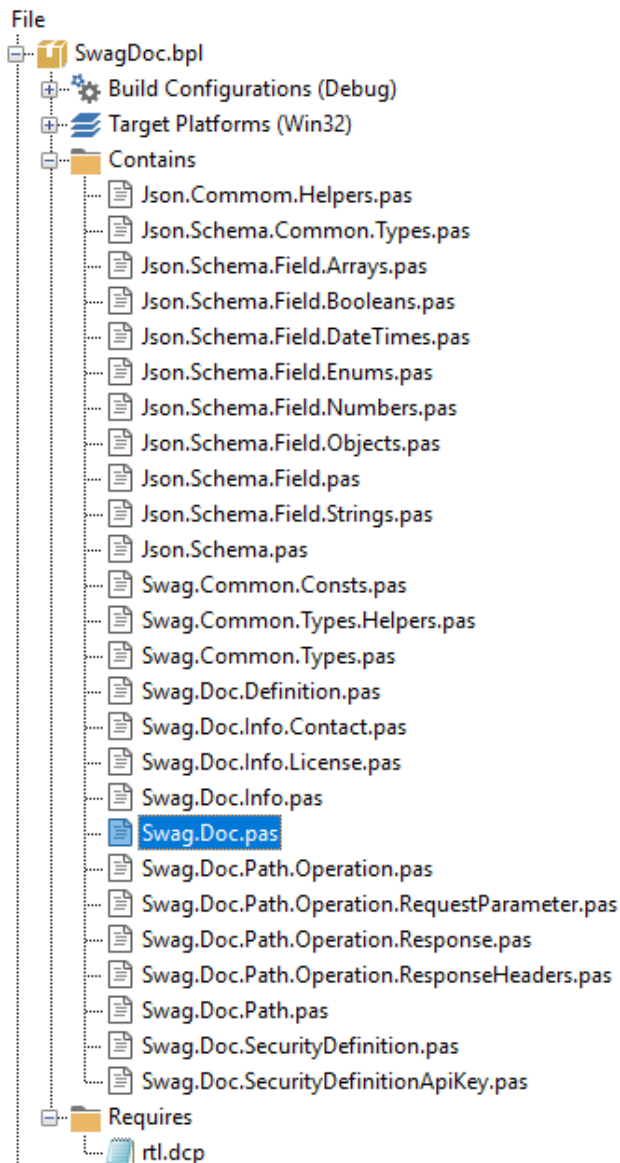
POST /employees

GET /employees/{id}

PUT /employees/{id}

DELETE /employees/{id}

O código fonte SwagDoc



- Biblioteca open source;
- Segue as especificações OpenAPI 2.0 e Json Schema;
- Escrito em inglês para facilitar a colaboração no mundo;
- Documentação da OpenAPI embutida nas classes e atributos;
- Compatível com VCL e Firemonkey;
- Desenvolvido na versão Delphi 10.2 Tokio e CE;
- Provável compatibilidade desde a versão Delphi 2010; (NÃO TESTADO!)
- Pacote nativo RTL (RTTI, JSON, custom attributes, generics) como única dependência;
- Utiliza conceito de “owner” para destruição de objetos;
- A principal classe é a TSwagDoc;

Download aqui:

<https://github.com/marcelojaloto/SwagDoc/tree/master/Source>

Demonstração de Exemplos

Sample API publicada no SwaggerHub
<https://app.swaggerhub.com/apis-docs/swagdoc/sample-api/v1>
OpenAPI 2.0 Exemplos
<https://swagger.io/docs/specification/2-0/adding-examples>

```
Sample Api with Documentatation Swagger using SwagDoc
Generate

"put":
{
  "description":"Updates a employee.",
  "operationId":"(28E989FB-0225-40BD-A97E-8D1EA80D09AF)",
  "tags":
  [
    "Employee"
  ]
  "parameters":
  [
    {
      "in":"path",
      "name":"id",
      "required":true,
      "type":"integer"
    }
    {
      "in":"body",
      "name":"employee",
      "required":true,
      "schema":
      {
        "$ref":"#/definitions/employee"
      }
    }
  ]
  "responses":
  {
    "200":
    {
      "description":"Successfully updates data",
      "schema":
      {
        "type":"object",
        "properties":

```

Teste no Swagger Editor

Swagger Editor

```
put:
  description: Updates a employee.
  operationId: '(28E989FB-0225-40BD-A97E-8D1EA80D09AF)'
  tags:
  - Employee
  parameters:
  - in: path
    name: id
    required: true
    type: integer
  - in: body
    name: employee
    required: true
    schema:
      $ref: '#/definitions/employee'
  responses:
  '200':
    description: Successfully updates data
    schema:
      type: object
      properties:
        employees:
          type: object
          description: Employee request data
          required: true
          properties:
            id:
              type: integer
              description: The employee identification code.
              format: int64
            name:
              type: string
              description: The employee full name.
              maxLength: 80
            phone:
              type: string
              description: The employee phone number.
            hireDate:
              type: string
              description: The employee hire date.
              format: yyyy-MM-dd
            salary:
              type: number
              description: The employee gross salary.
            address:
              type: object
              description: The employee full address.
              properties:
                description:
                  type: string
                  description: The employee address description.
                city:
                  type: string
                  description: The employee address city.
                region:
                  type: string
                  description: The employee address region.
                country:
                  type: string
                  description: The employee address country.
                postalCode:

```

GET /employees

POST /employees

GET /employees/{employeeId}

PUT /employees/{employeeId}

Updates a employee.

Parameters

Name	Description
id * required	
integer	
(path)	
employee * required	
(body)	

Edit Value | Model

```
{
  "name": "Marcelo Jaloto",
  "phone": "10 123456789",
  "hireDate": "2010-01-15",
  "salary": 1000000,
  "address": {
    "description": "Rua Delphi, 123",
    "city": "Florianópolis",
    "region": "Santa Catarina",
    "country": "Brasil",
    "postalCode": "98.765-432"
  }
}
```

Cancel

Parameter content type

application/json

Execute

<https://editor.swagger.io>

<https://github.com/marcelojaloto/SwagDoc/tree/master/Demos/SampleApi>

Deploy dos arquivos Swagger

Passo a passo:

1- Baixe os arquivos do Swagger UI na pasta Deploy do GitHub do SwagDoc;

2- Gere o arquivo swagger.json com SwagDoc;

3- Copie os arquivos Swagger UI mais o arquivo swagger.json para pasta de deploy do servidor web;

4- Edite o arquivo index.html, procure pelo atributo "url" e altere o valor para o local web onde encontra-se o arquivo swagger.json;

5- Abra o browser e teste sua documentação Swagger carregando o arquivo index.html no local web.

The image is a collage of four screenshots illustrating the deployment process:

- Top Left:** A screenshot of the GitHub repository `marcelojaloto / SwagDoc` showing the `Deploy` branch. The file list includes `favicon-16x16.png`, `favicon-32x32.png`, `index.html`, `oauth2-redirect.html`, `readme.txt`, `swagger-ui-bundle.js`, `swagger-ui-bundle.js.map`, `swagger-ui-standalone-preset.js`, `swagger-ui-standalone-preset.js.map`, `swagger-ui.css`, `swagger-ui.css.map`, `swagger-ui.js`, `swagger-ui.js.map`, and `swagger.json`.
- Top Right:** A screenshot of a web browser showing the Swagger UI interface. The address bar is `localhost/api/index.html`. The interface displays a "Sample API" with a "POST /employees" endpoint.
- Bottom Left:** A screenshot of a Windows File Explorer window showing the contents of the `api` directory. The files listed are `favicon-16x16.png`, `favicon-32x32.png`, `index.html`, `oauth2-redirect.html`, `readme.txt`, `swagger.json`, `swagger-ui.css`, `swagger-ui.css.map`, `swagger-ui.js`, `swagger-ui.js.map`, `swagger-ui-bundle.js`, `swagger-ui-bundle.js.map`, `swagger-ui-standalone-preset.js`, and `swagger-ui-standalone-preset.js.map`.
- Bottom Right:** A screenshot of a Notepad window showing the `index.html` file. The code snippet shows the `SwaggerUIBundle` configuration with the `url` attribute updated to `"/api/swagger.json"`.

<https://github.com/marcelojaloto/SwagDoc/tree/master/Deploy>

Seja um colaborador SwagDoc!

- Interesse do SwagDoc por Delphi MVC Framework;



danieleteti commented on 21 Mar

I'm the lead developer of delphimvcframework, and I'd like to add swagger support to the APIs offered by the controllers. Could you add some samples about this project so that we can evaluate It?
Thank you



marcelojaloto commented 10 days ago

Owner + 😊 ...

@danieleteti,
a new full sample is available!

[Download here!](#)

Sample API v1.0



danieleteti commented 9 days ago

+ 😊 ...

Thank you @marcelojaloto
We'll try to integrate it before the upcoming ITDevCon9.
Thank you for the notification.

<https://github.com/marcelojaloto/SwagDoc/issues/2>



Unwatch 16

Unstar 29

Fork 7



- Contribua com o SwagDoc;
- Acompanhe o projeto em Watch;
- Se gostou dê uma Star para o SwagDoc;
- Contribua com ideias e sugestões em uma nova Issue;
- Submeta uma Pull Request (PR) para melhorar o SwagDoc;
- Ou faça uma doação de incentivo para o projeto. :)

Links e Referências

SwagDoc

- <https://github.com/marcelojaloto/SwagDoc>

Exemplos Swagger

- <https://petstore.swagger.io>
- <https://app.swaggerhub.com/apis-docs/swaggerdoc/sample-api/v1>
- <https://idratherbewriting.com/learnapidoc/assets/files/swagger/#/Current%20Weather%20Data/CurrentWeatherData>

Swagger Tools

- <https://swagger.io/tools/swaggerhub>
- <https://editor.swagger.io>

Swagger

- <https://swagger.io>

OpenAPI Specification 2.0

- <https://github.com/OAI/OpenAPI-Specification/blob/master/versions/2.0.md>
- <https://swagger.io/docs/specification/2-0>

Json Schema

- <http://json-schema.org>
- <https://github.com/OAI/OpenAPI-Specification/blob/master/versions/2.0.md#schemaObject>

Referências e tutoriais

- <https://swagger.io/swagger/media/blog/wp-content/uploads/2017/02/Documenting-An-Existing-API-with-Swagger-2.pdf>
- https://idratherbewriting.com/learnapidoc/pubapis_swagger_intro.html
- https://youtu.be/6kwmW_p_Tig (OpenAPI 3.0)



Desenvolvedor Delphi



<https://www.seventh.com.br/trabalhe-conosco>



Empresa Brasileira de Tecnologia





D-Guard (VMS) para
monitoramento de câmeras IP e DVR'S





Sistema D-Guard





OBRIGADO



marcelojaloto@gmail.com



<https://www.facebook.com/marcelojaloto>



<https://twitter.com/marcelojaloto>



<https://www.linkedin.com/in/marcelojaloto>



<https://github.com/marcelojaloto>

Embarcadero

Conference



Marcelo Jaloto



Embarcadero Conference