آشنایی با swagger

تدوین: مهندس محبوبه مهدی زاده

تاریخ آخرین ویرایش: ۱۸ خرداد ۱۳۹۹

اداره سامانه های کاربردی – مرکز فاوای دانشگاه فردوسی مشهد

Swaggerچیست؟

APIها امروزه دنیای مدرن وب را تحت تاثیر خود قرار داده‌اند. در واقعAPI ها با نمایان ساختن بخشی از عملکرد و اطلاعات نهفته موجود در برنامه‌ها، تعامل آن‌ها با محیط خارج را ممکن می‌کنند.

هراندازه یک داکیومنت مربوط به API قوی باشد به همان اندازه آن API نیز خوب خواهد بود. یک APIخوب زمانی که مردم نحوه استفاده از آن را ندانند؛ می‌تواند به ابزاری بلا استفاده و غیرمفید تبدیل شود و اینجاست که مشخص می‌شود چرا مستندسازی می‌تواند برای یک API تجازی ضروری باشد. اما ایجاد و حفظ یک داکیومنت خوب که درک آن آسان، تعامل با آن لذت‌بخش ونیز مورد رضایت مصرف‌کننده باشد؛ می‌تواند به‌شدت چالش‌برانگیز باشد. با این‌که ایجادمستندات عالی نیاز به صبر و تلاش دارد، اما این امر دلالت مستقیم بر ماندگاری و مقبولیت API دارد. با انتخاب ابزار مستندسازی مناسب، می‌توانیم داکیومنت‌های قابل‌فهم‌تری ایجاد کنیم.

Swagger یک الگوی توصیفی برای تولید و داکیومنت کردن Web APIهاست که بعدها به یک استاندارد، به نامOpenAPI Specification تبدیل شد.

به زبان ساده، یک استاندارد برای تعریف API نوشته شده است و Swagger ابزارهای لازم برای این کار را در اختیار قرار میدهد، درواقع شما بوسیله‌ی یک ساختار JSONیا YAML معین می کنید کهAPI های شما چه روت (Route) هایی دارد، چه ورودی هایی دریافت می کند و خروجی چه مسیری خواهد بود.

علاوه بر این وقتی یک استاندارد تعیین میشود، ابزارهای بسیاری تولید می‌شوند که زبان یکدیگر را درک می‌کنند. تعدادی از این ابزارها عبارتنداز:

Swagger Editor:

اولین ادیتور اپن‌سورسِ کاملا اختصاصی برای APIهای مبتنی بر OpenAPI می‌باشد. این ابزار یک محیط در اختیار کاربر قرار می‌هد که در آن می‌تواند مسیرهای موجود در API را تعیین کند، توضیحات آن را بنویسد و مشخص کند که چه پارامترهایی باید برای آن ارسال شود و خروجی آن چه خواهد بود. شما می‌توانید API خود را در این ادیتور طراحی، توصیف و مستندسازی کنید.

Swagger UI:

این ابزار یک فایل معتبر Swagger را دریافت می‌کند و توضیحات آن را نمایش می‌دهد و امکان تست نیز وجود دراد. در واقع این ابزار، مستندات API را نمایش و تست آن را انجام میدهد.

Swagger CodeGen:

یک تولیدکننده کد(code generator) اپن سورس است که شامل مجموعه ابزارهایی است که این امکان را به کاربر میدهد که از یک فایل Swagger (تعریف شده با RESTful API) کدهای سمت سرور (server stubs) و یا سمت کلاینت (client SDK) را تولید کند.

Swagger فقط یک الگوی توصیفی برای نوشتن API هست و کدی نوشته نمی‌شود، فقط تعریف می کنید که API به چه شکلی خواهد بود. اصولاً این موضوع با برنامه نویسی Backend ارتباط پیدا می کند و ارتباط مستقیمی با زبان‌هایی مثل اندروید ندارد.

محصولاتی که تحت وب هستند، همیشه یک بخش سرور و یک بخش کلاینت دارند. کلاینت ها با سرور ارتباط برقرار می کنند، که این کلاینت ها میتوانند وبسایت یا اپ موبایل یا دسکتاپ و یا همه‌ی اینا باشند.

ضرورت وجود ابزارهایی مثل Swagger زمانی بیشتر نمایان می‌شود که ایجاد API و کلاینت آن را به صورت موازی انجام دهیم. با داشتن یک فایلswagger می‌توان در Swagger UI کل مسیرهای API را به همراه توضیحات و نحوه‌ی استفاده از آن را مشاهده نمود، و نیز API را تست کرد، یعنی بدون نوشتن کد، می‌توان API را کاملاً تست کرد و در صورت تمایل با استفاده از ابزار CodeGen تمامی فایل‌های آماده برای استفاده در اندروید یا جاوا اسکریپت و... را ایجاد کرد. و این موضع بسیار کار تیمی را بهبود می‌بخشد.

نظارت و مدیریت API های مختلف نیازمند کار زیادی می باشد .از دیگر مزایای استفاده از ابزارهای swagger در هنگام کار با Web API ها می‌توان به موارد زیر اشاره نمود:

تست و debug کردن API ها نیاز به عیب یابی دارد: انجام این کار برای برنامه های موبایل خیلی سخت و طاقت فرسا است . IDE ها (محیطهای توسعه یکپارچه) هم برای راحت تر کردن این قضیه وجود دارد ولی به مناسب Debugg کردن API ها نمی‌باشد.

تعیین منبع مشکلات از طریق عملکرد کاربر در همان لحظه: برنامه نویسان دوست دارند استفاده API ها را در یک برنامه مشاهده کنند. مثلا زمانی که کاربر در هنگام استفاده از یک برنامه با خطایی مواجه می شود، برنامه نویس دوست دارد بداند چه تقاضایی باعث بوجود آمدن چنین خطایی شده و به چه دلیل .

تولید پاسخهای سفارشی از WebAPI : علی‌الخصوص زمانی که برنامه و API در یک خط موازی حرکت می‌کنند یا زمانی که یک ویژگی خاص جدید نیازمند بروزرسانی API می ماند تا قابل استفاده شود.

می‌توان اینطور نتیجه گرفت که:

«برای تسریع استفاده از APIهای تحت وب برنامه نویسان نیازمند نمونه های استفاده از آن APIها، مستندات منسجم و با مفهوم و هر آن چیزی هستند که در صرفه جویی وقتشان کمکشان کند. برای همین نیازمندی، ساختارهایی همانند Swaggerموجود است که ثابت کرده بسیار کارامد هستند.»

Swagger Editor

Swagger Editor یک ادیتور open source برای طراحی، تعریف و مستندسازی APIهای RESTful در swagger specification می باشد. سواگر ادیتور به صورت رایگان و آنلاین در editor.swagger.io قابل دسترس می‌باشد. سورس کد آن نیز در gethub موجود است:

GitHub: https://github.com/swagger-api/swagger-editor

این ادیتور در هر مرورگری کار میکند. در مرورگر اجرا می‌شود و به طور کامل client-side javascript ساخته می‌شود و نگرانی بابت ذخیره اطلاعات API در سرورهای سواگر نمی‌باشد. در این ادیتور ذخیره‌سازی ابری وجود ندارد، بنابراین در صورت نیاز باید اطلاعات را به صورت لوکال ذخیره کرد. از سواگر ادیتور می توان به صورت لوکال یا از نسخه تحت وب آن استفاده نمود.

پیش نیازها برای نصب لوکال Swagger Editor

موارد زیر قبل از دانلود و اجرای Swagger Editor باید بر روی سیستم نصب شود:

Nodejs

هنگامی که nodejs با موفقیت نصب شد، بایستی npm نیز به صورت کامل نصب شود که می‌توان با استفاده از دستور زیر در محیط CMD این کار را انجام داد:

```
npm install;
```

اگر سیستم عامل ویندوز است:

میتوان از nodejs.msi برای نصب nodejs استفاده کرد. فایل installer یا همان msi سه عمل را انجام میدهد. ابتدا برنامه nodejs را داخل یک پوشه کپی می کند(مسیر انتخابی). در گام دوم npm را نصب می‌کند. و در آخر، آدرس nodejs را در متغیر محیطی PATH قرار می دهد تا بتوان از node در command prompt استفاده کرد(اگر بعد از نصب سیستم reboot شود بهتر است).

در صورت استفاده از فایلexe باید آن را در پوشه ای ذخیره کرد و بعد از آن، آدرس node را در path ایجاد و در نهایت npm را به صورت دستی نصب نمود(در اینجا باید فایل zip مربوط به npm را در پوشه ای که nodejs قرار دارد extract کنیم)

Git

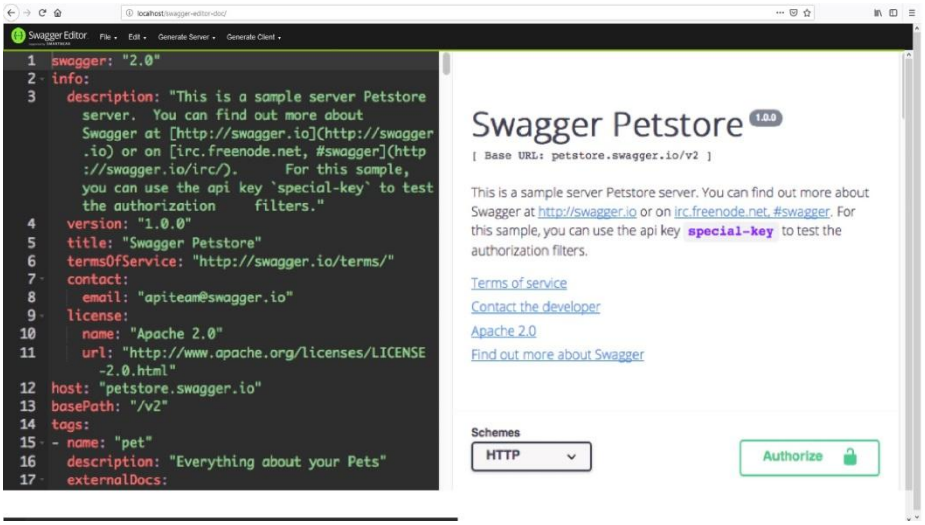
برای clone کردن اطلاعات بایستی git بر روی سیستم نصب شود.

پس از نصب موارد بالا برای clone کردن سواگر، در محیط cmd وارد مسیر hdocs در Xampp یا (www در wamp) می‌شویم و دستور زیر را اجرا می‌کنیم:

```
> git clone https://github.com/swagger-api/swagger-editor.git swagger-editor-doc
```

با این دستور swagger-editor در مسیر موردنظر قرار می‌گیرد. حال می‌توان swagger-editor-doc را اجرا کرد.

زمانی که ادیتور را برای اولین بار اجرا می‌کنیم، در این محیط Swagger Petstore API نمایش داده می‌شود (یک API ساده است که توسط پروژه‌های سواگر برای نمایش قابلیت‌های گسترده OpenAPI specification بکار می‌رود و در سواگر ادیتور به عنوان مثال پیش‌فرض می‌باشد).



همانطور که می‌بینید، swagger editor یک محیط دو قسمتی را نشان می‌دهد:

سمت چپ شامل مشخصات (specification) در قالب YAML است و سمت راست مستندات API تعاملی تولیدشده را نشان می‌دهد. هرگونه ویرایش در سمت چپ ادیتور، بر روی اطلاعات سمت راست راست تاثیرگذار است و تغییرات در همان لحظه در سمت راست نمایان می‌شود.

تعاریف OpenAPI را می‌توان بصورت YAML یا Json مرتب کرد. با این‌که swagger editor هر دو مورد را قبول می‌کند ولی استفاده از YAML را در اولویت قرار می‌دهد و زمانی‌که شما فایل Json را کپی یا Paste می‌کنید از شما درخواست می‌کند تا آن را تبدیل کنید (convert).

یک نوار منو در قسمت بالایی ادیتور وجود دارد که شامل منوهای زیر می‌باشد:

File:

با استفاده از این منو می‌توان فایل OpenAPI را import کرد. این فایل میتواند از فایل‌هایی باشد که بر روی سیستم شما ذخیره است (منوImport File) و یا اینکه می‌تواند یک URL باشد (Import URL). با استفاده از زیرمنوهای Save as YAML و Convert and save as json می‌توان ورژن‌های YAML یا JSON مواردی که در ادیتور تعریف شده است را دانلود کرد. زیرمنو Clear editor صفحه ادیتور را پاک می‌کند.

Edit

در این منو تنها یک زیرمنو (Convert to YAML) وجود دارد که با استفاده از آن می‌توان json را به YAML تبدیل کرد.

Generate Server و Generate Client:

این دو منو، لیستی از انواع مختلف زبان‌های برنامه‌نویسی و فریمورک‌ها را نمایش می‌دهند. اگر بر روی یکی از آن‌ها کلیک کنید، یک فایل zip دانلود می‌شود که شامل بدنه و طرح پروژه برای تولید یا تحلیل API با تعاریف شما است.

این دو ویژگی (Generate Server و Generate Client) در پروژه open source Swagger_Codegen ساخته می‌شود. این بدین معناست که برخلاف مابقی قسمت‌های اپلیکیشن، اگر شما از این دو ویژگی استفاده کنید، تعریف OpenAPI برای پردازش به سرور فرستاده می‌شود و ر روی کلاینت پردازش نمی‌شود.

لازم به ذکر است که ادیتور swagger ابزاری برای کمک در یادگیری نوشتن OpenAPI است و مستقیما با تعاریف API قابل خواندن برای ماشین (machine-readable) کار می‌کند. این ادیتور، ویژگی‌های استاندارد IDE مثل هایت‌لایت کردن سینتکس، auto complete و اعتبارسنجی (validate) را دارد، اما یک طراح API بصری (visual API designer) نیست و به طور کلی جامعه هدف این ادیتور تنها برنامه‌نویسان می‌باشند و برای افراد غیر برنامه‌نویس مناسب نیست.

ایجاد یک openAPI definition:

برای ایجاد تعریفی از openApi در ادیتور swaager ابتدا باید محیط ادیتور را پاک کنیم(File → Clear editor).

ساختار پایه:

اولین موردی که باید در محیط ادیتور نوشته شود، دستوری است که باید نوع و ورژن specification را مشخص کند.

Swagger: ‘2.0’

آبجکت‌ها:

نام	اجباری	توضیحات
swagger	✓	از نوع رشته. نوع و ورژن specification
info	✓	یک متادیتا درمورد API.
servers	✓	یک آرایه از آبجکت‌های سرورکه اطلاعات ارتباطی با یک سرور هدف را ارائه می‌دهد. اگر server تعریف نشده باشد یا یک آرایه خالی باشد؛ مقدارپیش‌فرض یک آبجکت server با مقدار url ای برابر ./ خواهد بود
paths	✓	عملیات‌های و مسیرهای دردسترس برای API
components		یک المنت برای نگهداری طرح‌های(schemaها) متنوع برای specification
security		یک اعلان از مکانیزم‌های امنیتی که می‌توانند در سراسر API استفاده شوند. لیست مقادیر شامل اشیاء موردنیاز امنیتی جایگزین است که می‌تواند استفاده شود. فقط یک مورد از آبجکت‌های الزامی امنیتی نیاز است تا اجازه درخواست تایید شود. عملیات‌های فردی (individual operation) می‌تواند این تعریف را تغییر دهد(override).
tags		لیستی از تگ‌های استفاده شده توسط specification با متادیتا اضافی. ترتیب تگ‌ها می‌تواند منعکس‌کننده ترتیب آن‌ها بوسیله ابزارهای پارسینگ باشد
externaldoc		داکیومنت خارجی

آبجکت Info:

عموما بهتر است که اطلاعاتی کلی در مورد API در داخل مشخصات گنجانده شود؛ اگر قرار است API به صورت عمومی در دسترس قرار گیرد، از این طریق می‌توان اعتماد کاربر به محصولات کمپانی را افزایش داد . برای اختصاص API metadata، از خصیصه‌های آبجکت Info استفاده می‌کنیم. با استفاده از info می‌توانیم اطلاعات قابل فهم برای کاربر مثل عنوان، توضیحات، اطلاعات تماس و ... را تنظیم کنیم.

خصیصه‌های (property) آبجکت info عبارتند از:

نام	نوع	اجباری	توضیحات
title	string	✓	عنوان اپلیکیشن
Version	string	✓	ورژن(نسخه) API. برای مقداردهی می‌توان از ورژن معنایی مثل ۱,۰,۰ استفاده کرد یا رشته دلخواهی مثل 0.99-beta بکار برد
description	string		توضیحات API. متنی دلخواه در قالب html یا markdown
termsOfService	string		لینک به صفحه‌ای که شرایط خدمات توضیح داده شده است. مقدار این فیلد باید به فرمت URL باشد
contact	Contact object		اطلاعات تماس که شامل خصیصه‌های name، email و url می‌باشد. name: نام یا اطلاعات متنی. اگر از این خصیصه به تنهایی کاربردی نداردشود. اگر email یا url مقداردهی شده باشند در لینکی که با آدرس email یا url ایجاد می‌کند از مقدار name به عنوان hypertext لینک استفاده می‌شود و اگر name مقداردهی نشده باشد از مقدار پیش‌فرض آن یعنی the developer استفاده می‌شود Email: مقداری با فرمت پست الکترونیک دریافت می‌کند. این فیلد به صورت لینک است که با کلیک بر روی آن امکان ارسال نامه به ایمیل ثبت شده وجود دارد url : آدرس url که لینک آن در خروجی نمایش داده می شود.
license	License object		شامل خصیصه‌های license و url می‌باشد. name: نام license که در صورت استفاده از url به عنوان hypertext لینک به کار می‌رود. url: آدرس url مربوط به شرح license

externalDocs	لینکی به یک داکیومنت خارجی(در صورت وجود) که شامل خصیصه‌های description و url است. Code یا ابزارهای تولید مستندات می‌تواند description را به عنوان متن لینک استفاده کند. description: توضیح url: آدرس

آبجکت server :

یک شیء که یک سرور را نشان می دهد.

فیلدهای ثابت(propertyها)

نام	نوع	اجباری	توضیحات
url	String	✓	یک url به هاست مقصد. این url از متغیرهای سرور پشتیبانی می‌کند و می‌تواند نشبی باشدکه نشان می‌دهد محل هاست با محلی که داکیومنت OpenAPI ذخیره شده است، مرتبط می‌باشد. جایگزینی متغیر زمانی ایجاد می‌شود که یک متغیر در براکت {} نامگذاری شود.
description	String		یک رشته اختیاری برای توصیف هاست طراحی شده با Url. برای نمایش متن از CommonMarl استفاده می‌شود.
variables	Map(string، شی server (variable		یک نگاشت بین نام متغیر و مقدار آن. مقدار برای جایگزینی در قالب url سرور استفاده می‌شود.

Swagger ui:

Swagger ui پارامترهای پیکربندی (configuration) را به چهار روش می‌تواند دریافت کند:

با استفاده از swagger-config.yaml در دایرکتوری ریشه پروژه

آبجکت پیکربندی به صورت یک آرگومان به Swagger ui ارسال می‌شود ((SwaggerUI({.. }

داکیومنت پیکربندی از یک configUrl خاص واکشی(fetch) می‌شود.

آیتم‌های تنظیمات به صورت جفت های کلید/مقدار (key/value) در رشته پرس‌وجو url ارسال می‌شود.

parameters

پارامترهایی با dot هایی در نام آن‌ها، رشته هایی تک هستند که برای تنظیم پارامترهای زیرمجموعه بکار می‌روند و نشانگر یک ساختار تودرتو نیستند.

برای خوانایی بیشتر، پارمترها بوسیله دسته(category) گروهبندی و براساس حروف الفبا مرتب می‌شوند.

نشانه‌گذاری type ها فرمتی مانند زیر دارد:

String = “” : به معنی نوع رشته با مقدار پیش‌فرض “” می‌باشد

String=["a"*, "b", "c", "d"] به این معنی است که نوع رشته می‌تواند مقادیر a،b،c یا d داشته باشد. علامت *مشخص می‌کند که a مقدار پیش‌فرض است.

Core

نام پارامتر	متغیر Docker	نوع	توضیحات
configUrl	CONFIG_URL	String	url برای واکشی (fetch) داکیومنت پیکربندی خارجی
dom_id	DOM_ID	string	اگر domNode مقداردهی نشده، این پارمترالزامیست. شناسه یک عنصر dom که در آن swaggerUi رابط کاربری را برای swaager قرار می دهد.
domNode	---	eleme	اگر dom_id مقداردهی نشده، این پارمتر الزامیست.

		nt	عنصر html dom است که در داخل آن swaggerUi رابط کاربری را برای swaager قرار می دهد. dom_id را overrids می‌کند
spec	spec	Object ={}	یک شی js برای توصیف openAPI Specification. زمانی که از این پارامتر استفاده شود ، پارامتر url بررسی (parse) نمی‌شود. این امر برای تست specificationهایی که بصورت دستی بدون میزبانی (hosting) آنها بدست آمده؛ مفید است
url	URL	String	url که به تعریف API اشاره میکند (عموما swagger.json یا swagger.yml). اگر urls یا spec استقاده شود؛ این پارامتر نادیده گرفته می‌شود.
urls	URLS	Array	یک آرایه از اشیای تعریف API ([{url: "<url1>", name: "<name1>"},{url: "<url2>", name: "<name2>"}]) . بوسیله پلاگین Topbar استفاده می‌شود. وقتی این پارمتر استفاده می‌شود و پلاگین Topbar فعال است، پارامتر url پارس نمی‌شود.Name ها و URL ها باید در تمامی آیتم‌ها در این آرایه یکتا باشند ، به دلیل این‌که به عنوان شناسه‌ها(identifiers) استفاده می‌شوند.
urls.primaryName	URLS_PRIMARY_NAME	String	در مواقعی که از urls استفاده می‌شود، می‌توان از این ساب پارامتر استفاده کرد. اگر این پارمتر با name یکی از مشخصه‌های (spec) ارائه شده در urls منطبق باشد، آن مشخصه در زمان لود Swagger-UI نمایش داده می‌شود.(به طور معمول درحالت پیش‌فرض اولین مشخصه در urls در زمان لودSwagger-UI نمایش می‌یابد)