Designing and Documenting APIs - OpenAPI(Swagger) vs RAML

Introduction

Documentation will set you free.

Whether your API is for internal, private use, or for extensive public interaction, a thorough design documentation can make or break an API.

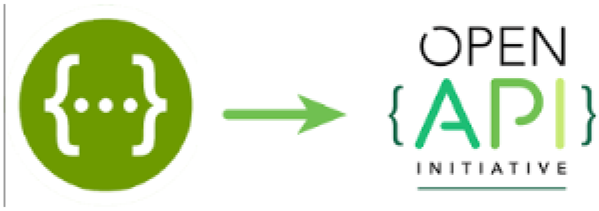
It’s not just a tool for developers, either — proper API documentation can drive API adoption more than any other tool within the API provider’s grasp.

Swagger (now Open API Specification - OAS) and **RAML** are two of the most popular specifications for developing APIs on the market right now.

- But which one is better for us?

- What is the difference between them?

OpenAPI Specification



Of the many API documentation and specification formats, **Swagger** is certainly one of the most popular.

It allows you to design , describe, produce, visualize, and consume RESTful web services.

It also has a strong ecosystem to libraries to generate code to provide (server side) and consume (client side) in many technical languages and format.

Swagger's popularity stems from its simplicity. From its concise documentation to its to its ease of readability for machines and humans alike, Swagger is a framework that has been made very easy to utilize.

Also there is Swagger UI , which is a dependency-free collection of *HTML*, *Javascript*, and *CSS* assets that dynamically generate beautiful documentation from a Swagger-compliant API.

Starting January 1st 2016 the Swagger Specification has been donated to to the Open API Initiative (OAI) and has been renamed to the OpenAPI Specification (OAS). The Open API specification is backed by the industry leaders like Google, Microsoft and IBM.

RAML

RAML, or **RESTful API Modeling Language**, is an YAML-based language for describing RESTful APIs.

However, RAML can be used for describing non-REST APIs as well.

Like Swagger, with RAML it is also possible to generate source code for client and server and comprehensive user documentation.

Comparison

	Swggaer/OAS	**RAML**
Open Source/Free	Yes	Yes
Major Sponsor	Smartbear	Mulesoft
Format	Yaml/Json	Yaml
Industry Standard Initiative	Open API Initiative has an open governance model around the Swagger Specification under the Linux Foundation.	None
Codegen	Both Server side and client side .NET, Go, Haskell, Java, JavaScript, Node.js, PHP, Python, Ruby, Scala, Clojure, Coldfusion, D, Eiffel, Erlang, Groovy, and Typescript	Both Server Side and Client Side. .NET, Go, Haskell, Java, JavaScript, Node.js, PHP, Python, Ruby, Scala, Elixer and Pearl
Html Gen	Swagger UI	raml2html
Editor	Online and offline	Offline. Using Atom/API Workbench
API test	Can execute api calls from the documentation	API tests from documentation is possible using 3rd party plugin
Top-Down (Design and then generate code)	Yes	Yes
Bottom-Up (Write API code and then reverse engineer Documentation)	Yes	Yes
Reusable traits (like pagination/ Auth)	No	Yes
API Management tool support	APIgee, IBM API Connect, Ndgit, Mulesoft	Mulesoft
Ecosystem/Open source support	Very very popular. More than 13K repositories on GitHub	A rich ecosystem, but not as accepted as Swagger. 1.5K repositories on Github.
Community support	Over 16K question and answers on https://stackexchange.com	A little less than 1K in questions and answers on https://stackexchange.com

Conclusion

As you can see, both specifications described and compared above are quite strong, market leaders and mostly match for like for like features.

However though, swagger is much more popular, more widely accepted by larger players, has a richer ecosystem, and finally supported by a throbbing tech community.

In the past, companies that use Mulesoft anypoint has chosen RAML over swagger, however, that reason will also go away with Mulesoft also joined the OpenAPI initiative and has started supporting OpenAPI/swagger in the latest version.

In fact, with Mulesoft joining the OpenAPI initiative, it's most likely that two standards will now converge to become OpenAPI Specification.

And that's make Swagger as a clear winner for an organization starting their API journey.

External Reading

https://swagger.io/blog/news/mulesoft-joins-the-openapi-initiative/

https://stackshare.io/stackups/raml-vs-swagger-ui

https://apievangelist.com/2015/03/30/quantifying-the-community-around-the-swagger-api-specification/

https://blogs.mulesoft.com/dev/api-dev/open-api-raml-better-together/

https://dzone.com/articles/rest-api-documentation-part-2-1

https://blog.vsoftconsulting.com/blog/is-raml-or-swagger-better-for-building-apis