Converting Swagger to RAML

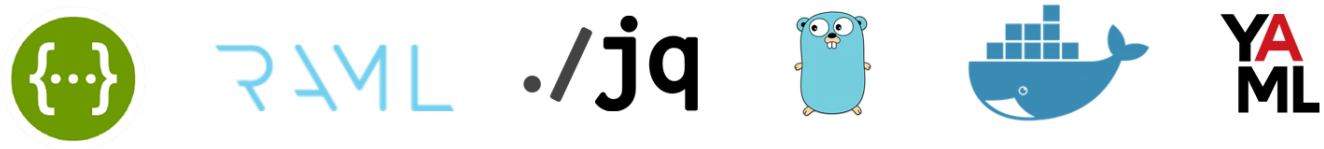
Mitch Dresdner

Table of Contents

Converters and tooling examples

Summary

This article will help you experiment with some basic conversions between Swagger and RAML and explore some tooling using some cool technologies.



JSON tooling and conversion experiments

- Standalone NPM converter
- Conversion using a Docker container
- Converting YAML to JSON with a Go utility
- Using jq for XPath like querying of JSON

As you can see from the list above we have our work cut out for us, let's roll up our sleeves and get to work!

Open API Specification (OAS)

Swagger *aka* **OAS**

Swagger and OAS will be used interchangably in this tutorial.

Conversion examples

This article assumes some basic familiarity with modeling JSON API's using Swagger and RAML, for those needing a bit of a refresher, please refer to the links below.

Refresher on Swagger and RAML

- Click for RAML tutorial

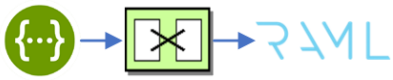
- Click for Swagger tutorial

My earlier DZone article Zero Code REST With json-server has instructions to help you install and get started with NPM and HTTPie if you're not aleady familiar. With those packages installed, we can add the other necessary components.

In the steps below we'll be installing the necessary dependencies for getting started.

Mulesoft OAS RAML Converter

OAS RAML Converter is a Mulesoft project which you can use for OAS/RAML conversions. We'll clone the git project, install dependencies, build the javascript project and manually run the converter.



Clone repo and build converter

```
git clone https://github.com/mulesoft/oas-raml-converter.git

npm install
npm run build
```

With the OAS RAML Converter installed, lets run a conversion from Swagger to RAML. In our examples we'll use the Petstore *swagger.json*, so lets download that next.

```
http --download http://petstore.swagger.io/v2/swagger.json

node lib\bin\converter.js --from OAS20 --to RAML ./swagger.json > petstore.raml
```

Proxy support

--proxy=http:http://proxy.foo.bar:80

If you're behind a firewall, you may need to add the proxy switch above to the HTTPie command line, with your proxy server address.

If all went well, the swagger.json was converted into petstore.raml.

Mulesoft hosts an online conversion website if you prefer not to install the local Javascript

component.

If you would like to install the converter globally run the commands below.

```
npm install -g oas-raml-converter

oas-raml-converter --help
```

Converting between Swagger and OAS.

```
oas-raml-converter --from OAS20 --to RAML ./path/to/swagger.json
oas-raml-converter --from OAS20 --to RAML ./path/to/swagger.json > petstore.raml

oas-raml-converter --from RAML --to OAS20 ./path/to/petstore.raml > swagger.json
```

Be sure to use *lib|bin|converter.js* if you didn't install oas-raml-converter globally.

Mulesoft Docker converter

Mulesoft also has a Docker version of the converter that you can learn more about here

Cloning the repository and starting the Docker container.

```
git clone https://github.com/mulesoft/oas-raml-converter-service

docker build -t oas-raml-converter:0.1 .

docker run -i -p 3000:3000 -t oas-raml-converter:0.1 /bin/bash
```

With the docker container running and firewall port 3000 open if you're running remote, lets convert our earlier *swagger.json* file into RAML.

Using the Docker converter to convert OAS to RAML

```
http POST YOUR_DOCKER_IP_ADDRESS:3000/swagger/to/raml Content-type:text/plain
@swagger.json > petstore.raml
```

Tooling examples

Using JSON Query (jq)

JSON Query is descibed as a flexible, light weight command line processor for performing Xpath like queries on JSON data. After using the link above to install it, lets run some simple queries on our *swagger.json* file.

```
./jq
```

Verify jq was properly installed

```
jq --help
```

Display current version and commandline options

When you use the Windows *type* or Linux *cat* command to display **swagger.json**, you'll notice that the entire file is on a single line. To pretty print the file we can use the *jq* identity function.

Pretty print swagger.json

```
jq . swagger.json
```

You can also use jq to extract snippets of JSON or perform a myriad of mathematical and utility functions on the data See jq manual here.

Extract a JSON snippet

```
jq ".tags" swagger.json
```

Extract an element

```
jq ".tags[0].name" swagger.json
```

Escape special characters to handle JSON paths

```
jq ".paths.\"/pet\".post" swagger.json
```

Produces a JSON snippet of the /pet URI for a POST operation. The slash is a special character in *jq* and needs to be escaped with the quotes.

Converting YAML to JSON

Sometimes you may find that you have the YAML version of a Swagger API specification which you need the JSON equivalent for in order to generate RAML. Here's a nifty conversion utility written in Golang, which can be used to generate the JSON schema.

You'll need to clone the git repository to your *GOPATH*, build and install. For a quick start guide to getting up and running with Go, see this turorial here.

Cloning the repository and installing the Golang converter.

```
cd %GOPATH%\src

go get -u github.com/wakeful/yaml2json

cd yaml2json

go build

go install

yaml2json -version
```

GOPATH\bin will need to be in your PATH

Converting a YAML file to JSON

```
yaml2json PATH_TO_YOUR\file.yaml | jq . > PATH_TO_YOUR\file.json
```

There we have it, in this example we pipe the output to JSON Query using an identity function to prettify the output, then redirect the output to our new JSON file.

This concludes our brief examples with conversions and tooling.

I hope you enjoyed reading this article as much as I have enjoyed writing it, i'm looking forward to your comments!

About the Author:

Mitch Dresdner is a Senior Mule Consultant at TerraThink