swagger-compare Documentation

Release 0.1.0

Kevin Johnson

Feb 19, 2019

Contents:

swagger-compare aims to make the job of communicating api changes easier. The current phase of this project deals only with endpoints that have been marked as deprecated. By making use of the x-replaced-by and x-remove-on swagger extensions a clear, and automated diff summary can be generated from the baseline swagger, and the new swagger document.

give it a try!

```
docker run kjjuno/swagger-compare https://raw.githubusercontent.com/kjjuno/swagger-
→compare/master/test/petstore.yaml https://raw.githubusercontent.com/kjjuno/swagger-
→compare/master/test/petstore-new.yaml
```

cli examples

swagger-compare is available to install from npm

```
npm install -g swagger-compare
```

swagger-compare is also available on Docker Hub

```
docker pull kjjuno/swagger-compare
```

1.1 Quick Start

Let's say you have an api hosted at https://my-api/swagger/v1/sagger.json

You have a build server that has just built something from your `master` branch and now you want to compare the new swagger to the deployed swagger.

Add this line to your build script:

```
swagger-compare https://my-api/swagger/v1/sagger.json ./wwwroot/v1/swagger.json
```

You should get the following type of output

```
version: 1.0.0
deprecated:
  paths:
    /v1/pet:
      post:
        x-replaced-by:
          path: /v2/pet
          verb: post
        x-remove-on: undefined
      put:
        x-replaced-by: undefined
```

```
      x-remove-on: 3/4/2019
  /v2/pet:
    post:
      x-replaced-by:
        path: /v3/pet
        verb: post
      x-remove-on: 2/13/2019
```

Take a look at *swagger-extensions* to learn more about the x-replaced-by and x-remove-on extensions.

1.2 Usage

```
swagger-compare [options] <baseline> <new>
{version}

arguments:
  baseline        path or url to baseline swagger document
  new             path or url to new swagger document

options:
  -f,--format     default: yaml. Must be on of [yaml|json]
  --help          display this help page
  --version       display the version of swagger-compare

Any paths marked as deprecated that show up only in the <new>
document will be included in the report.
```

code examples

swagger-compare is available to install from npm. To use it in your project install the module and save it to your package.json file.

```
npm install --save swagger-compare
```

2.1 Quick Start

```javascript
// require libraries to load a swagger spec
const fs      = require('fs');
const yaml    = require('js-yaml');
const swagger = require('swagger-compare');

// read swagger spec as text from the file system.
var baselineText = fs.readFileSync(baselineFile, 'utf8');
var newText = fs.readFileSync(newFile, 'utf8');

// parse text into an object representation of the swagger spec
var baselineDoc = yaml.safeLoad(baselineText);
var newDoc = yaml.safeLoad(newText);

// compare the two swagger specs.
var summary = swagger.compare(baselineDoc, newDoc);

// display the diff summary to the console
console.log(yaml.safeDump(summary));
```

Take a look at *swagger-extensions* to learn more about the x-replaced-by and x-remove-on extensions.

Swagger Extensions

This tool makes use of the x-replaced-by and x-remove-on swagger extensions. Without applying these to your swagger you will only be able to detect that an endpoint has been deprecated, but you will not be able to easily tell what should be used instead.

3.1 x-replaced-by

The x-replaced-by extension gives extra context so that consumers of your api will know not only which endpoints have been deprecated, but which endpoints should be used instead.

3.2 x-remove-on

The x-remove-on extension communicated the date when a deprecated endpoint is scheduled to be removed. This helps consumers know when they need to have migrated to the new endpoint to avoid an outage.

Writing Documentation

Welcome to swagger-compare! Documentation for this project is hosted graciously by https://readthedocs.org

Helping to improve the documentation is a great place to start if you are looking to contribute to this project. Read the Getting Started guide on https://readthedocs.org to get familiar with the documentation system.

4.1 Be Clear

As much as possible you should strive for clear, unambiguous documentation. Use proper english, and strive for concise explanations.

4.2 Use Examples

Wherever possible try to use examples that show command line usage, or code. Make sure any examples used actually work. There is nothing more frustrating than following a tutorial that doesn't compile, or is out of date.

4.3 Bugs in the documentation

If you should find any cases in this documentation where there are typos, incorrect examples, out dated statements, or any other "bug" in the documentation please submit an issue to the GitHub issues site. Or, consider submitting a pull request!