

Hello!

Kyle Shockey
Software Engineer @ SmartBear



kyshoc

Any Way You Want It

Extending Swagger UI for Fun & Profit

GitHub, Inc. github.com/swagger-api/swagger-js/

swagger-api / swagger-js

Code Issues Pull requests Projects Wiki Insights

Javascript library to connect to swagger-enabled APIs via browser or nodejs <http://swagger.io>

swagger swagger-js swagger-api rest rest-api javascript openapi-specification

2,117 commits 18 branches 167 releases 170 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

shockey housekeeping: drop `v` from GitHub Release titles (#1388) Latest commit c2e635c 2 days ago

.github housekeeping: remove Q&A section from feature template (#1341) 2 months ago

docs docs: add `Default Content-Type values` section (#1284) 5 months ago

examples housekeeping: use `js-yaml` fork (#1353) 2 months ago

release housekeeping: drop `v` from GitHub Release titles (#1388) 2 days ago

src fix: pad with `0` when percent-encoding characters (via #1386) 2 days ago

test-webpack fix: decouple resolver from `function.prototype.name` (#1351) 2 months ago

test fix: pad with `0` when percent-encoding characters (via #1386) 2 days ago

.babelrc Don't mix import and module.exports (#1318) 3 months ago

.editorconfig in with the new 2 years ago

.eslintrc fix: drop browser global whitelisting from test harnesses (#1383) 7 days ago

.gitignore housekeeping: write new version identifier to filesystem during relea... 9 days ago

.npmignore in with the new 2 years ago

LICENSE Happy new year 9 months ago

README.md fix: another dummy commit (#1381) 9 days ago

package.json release: v3.8.21 2 days ago

webpack.bundle.config.js export umd bundle to browser, for compat reasons 2 years ago

webpack.check.js in with the new 2 years ago

webpack.common.js Add failing test for cyclic allOf + \$ref in resolve subtree (#1275) 5 months ago

webpack.config.js in with the new 2 years ago

webpack.test.config.js Give example code for how to use the `allowUnsafeExports` option (#1251) 2 months ago

Display a menu

GitHub, Inc. github.com/swagger-api/swagger-ui/

Unwatch 624 Unstar 12,751 Fork 5,900

Code Issues 229 Pull requests 17 Projects 0 Wiki Insights

Swagger UI is a collection of HTML, Javascript, and CSS assets that dynamically generate beautiful documentation from a Swagger-compliant API. <http://swagger.io>

swagger swagger-ui swagger-api swagger-js rest rest-api openapi-specification oas openapi

3,928 commits 37 branches 150 releases 303 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

shockey	housekeeping: drop `v` from GitHub Release titles (via #4882)	Latest commit 8062a9e 2 days ago
.github	housekeeping: remove Q&A section from feature template (#4708)	2 months ago
dev-helpers	improvement: prevent loading resources from third party CDN (via #4598)	2 months ago
dist	release: v3.19.0	2 days ago
docs	Added anchors to configuration parameters (#4711)	2 months ago
release	housekeeping: drop `v` from GitHub Release titles (via #4882)	2 days ago
src	fix: tolerate callback parameter values in ParameterRow (via #4873)	4 days ago
swagger-ui-dist-package	Maintain backwards compatibility with absolutePath	a year ago
test	housekeeping: test fixture file path mistake (via #4862)	2 days ago
.agignore	add .agignore	11 months ago
.babelrc	housekeeping: bundle size reductions (#4713)	2 months ago
.dockerignore	feature(docker-image): allow custom nginx port within container (#4086)	8 months ago
.editorconfig	Create .editorconfig	2 years ago
.eslintrc	meta: introduce Cypress end-to-end testing (via #4827)	24 days ago
.gitattributes	Set EOL of docker-run.sh to LF	a year ago
.gitignore	meta: introduce Cypress end-to-end testing (via #4827)	24 days ago
.npmignore	improve(npm-module): include standalone preset in `swagger-ui` (#4101)	8 months ago
.travis.yml	housekeeping: use bundlesize (#4701)	2 months ago
CONTRIBUTING.md	docs(contributing): add merging pull requests section (#4076)	8 months ago
Dockerfile	fix(docker): make shell script executable (via #4876)	2 days ago

Display a menu

Display a menu

swagger-api / swagger-editor

Code Issues Pull requests Projects Wiki Insights

Swagger Editor <http://editor.swagger.io>

swagger swagger-editor swagger-ui rest-api rest openapi-specification

3,062 commits 12 branches 298 releases 67 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

swaggerhub-bot release: v3.6.11 Latest commit `badc456` 2 days ago

File/Folder	Description	Time Ago
<code>.github</code>	housekeeping: remove Q&A section from feature template (#1825)	2 months ago
<code>build-tools</code>	fix(docker-releases): trim leading `v` from image names (#1759)	4 months ago
<code>dev-helpers</code>	Add favicons to dev-server html, for consistency	a year ago
<code>dist</code>	release: v3.6.11	2 days ago
<code>docs</code>	feature: `getEditorMetadata` helper (#1789)	3 months ago
<code>release</code>	housekeeping: drop `v` from GitHub Release titles (via #1884)	2 days ago
<code>src</code>	fix: force HTTPS download session links for 2.0 io generator (#1873)	23 days ago
<code>swagger-editor-dist-package</code>	Correct path for swagger-editor.js	10 months ago
<code>test</code>	fix: force HTTPS download session links for 2.0 io generator (#1873)	23 days ago
<code>.agignore</code>	Add agignore for searching	a year ago
<code>.babelrc</code>	Feature/add unit test to editor (#1764)	3 months ago
<code>.editorconfig</code>	Create .editorconfig	2 years ago
<code>.eslintrc</code>	fix: missing production dependencies (#1820)	2 months ago
<code>.gitignore</code>	housekeeping: automated release via `release-it` (via #1881)	3 days ago
<code>.npmignore</code>	Add travis config	2 years ago
<code>.travis.yml</code>	Update .travis.yml (#1828)	2 months ago
<code>Dockerfile</code>	`ico` -> `png` (#1694)	6 months ago
<code>LICENSE</code>	Happy new year 🎉	9 months ago
<code>README.md</code>	docs: README touchups (#1772)	4 months ago
<code>sunrise-icon</code>	housekeeping: add Cypress e2e test runner (via #1874)	22 days ago

- Familiar with Swagger UI?

- Familiar with Swagger UI?
- Used Swagger UI in a project?

- Familiar with Swagger UI?
- Used Swagger UI in a project?
- Familiar with React?

- Familiar with Swagger UI?
- Used Swagger UI in a project?
- Familiar with React?
- Written a plugin for Swagger UI?

✨ *The Plugin System* ✨

✨ The Plugin System ✨

- The heart of **Swagger UI** is the plugin system.
All built-in functionality the user sees is
provided by a plugin in the core project.

✨ The Plugin System ✨

- The heart of **Swagger UI** is the plugin system.
All built-in functionality the user sees is
provided by a plugin in the core project.
- **Swagger UI** is built with React.js and Redux.

✨ The Plugin System ✨

- The heart of **Swagger UI** is the plugin system.
All built-in functionality the user sees is
provided by a plugin in the core project.
- **Swagger UI** is built with React.js and Redux.
- Plugins can provide new or override existing:

✨ The Plugin System ✨

- The heart of **Swagger UI** is the plugin system.
All built-in functionality the user sees is provided by a plugin in the core project.
- **Swagger UI** is built with React.js and Redux.
- Plugins can provide new or override existing:
 - React components

✨ The Plugin System ✨

- The heart of **Swagger UI** is the plugin system.
All built-in functionality the user sees is provided by a plugin in the core project.
- **Swagger UI** is built with React.js and Redux.
- Plugins can provide new or override existing:
 - React components
 - Redux state logic (actions/reducers/selectors)

✨ The Plugin System ✨

- The heart of **Swagger UI** is the plugin system.
All built-in functionality the user sees is provided by a plugin in the core project.
- **Swagger UI** is built with React.js and Redux.
- Plugins can provide new or override existing:
 - React components
 - Redux state logic (actions/reducers/selectors)
 - Helper functions

✨ The Plugin System ✨

- The heart of **Swagger UI** is the plugin system.
All built-in functionality the user sees is provided by a plugin in the core project.
- **Swagger UI** is built with React.js and Redux.
- Plugins can provide new or override existing:
 - React components
 - Redux state logic (actions/reducers/selectors)
 - Helper functions
- Building a plugin requires understanding how to work with this architecture - let's take a look

React.js

React.js

- Everything is a component

React.js

- Everything is a component
- An application is formed by building a tree of components, similar to how you'd structure a tree of HTML tags

React.js

- Everything is a component
- An application is formed by building a tree of components, similar to how you'd structure a tree of HTML tags
- Components take inputs called props, provided by the entity using the component

React.js

- Everything is a component
- An application is formed by building a tree of components, similar to how you'd structure a tree of HTML tags
- Components take inputs called props, provided by the entity using the component
- Components return outputs of:

React.js

- Everything is a component
- An application is formed by building a tree of components, similar to how you'd structure a tree of HTML tags
- Components take inputs called props, provided by the entity using the component
- Components return outputs of:
 - HTML

React.js

- Everything is a component
- An application is formed by building a tree of components, similar to how you'd structure a tree of HTML tags
- Components take inputs called props, provided by the entity using the component
- Components return outputs of:
 - HTML
 - more components

React.js

- Everything is a component
- An application is formed by building a tree of components, similar to how you'd structure a tree of HTML tags
- Components take inputs called props, provided by the entity using the component
- Components return outputs of:
 - HTML
 - more components
 - a mix of both

React.js

- Everything is a component
- An application is formed by building a tree of components, similar to how you'd structure a tree of HTML tags
- Components take inputs called props, provided by the entity using the component
- Components return outputs of:
 - HTML
 - more components
 - a mix of both
- Simply put: components are functions that ultimately return HTML to be rendered on a page.

Redux

Redux

- **Redux stores the state of an application**
(i.e., anything that can change)

Redux

- **Redux stores the state of an application (i.e., anything that can change)**
- **State is one big JavaScript object**

Redux

- **Redux stores the state of an application** (i.e., anything that can change)
- **State** is one big JavaScript object
- **Action creators:** fns exposed to the app that trigger state changes by creating actions

Redux

- **Redux stores the state of an application** (i.e., anything that can change)
- **State** is one big JavaScript object
- **Action creators:** fns exposed to the app that trigger state changes by creating actions
- **Actions:** JS objects that describe a change to be made

Redux

- **Redux** stores the state of an application (i.e., anything that can change)
- **State** is one big JavaScript object
- **Action creators:** fns exposed to the app that trigger state changes by creating actions
- **Actions:** JS objects that describe a change to be made
- **Reducers:** fns that define how to make state changes
(oldState, action) => newState

Redux

- **Redux** stores the state of an application (i.e., anything that can change)
- **State** is one big JavaScript object
- **Action creators:** fns exposed to the app that trigger state changes by creating actions
- **Actions:** JS objects that describe a change to be made
- **Reducers:** fns that define how to make state changes
(oldState, action) => newState
- **Selectors:** fns that get data from the current state

Modifying Existing Behavior

✨ Modifying Existing Behavior ✨

- **Problem #1:** one size doesn't fit all, especially in tooling

✨ Modifying Existing Behavior ✨

- **Problem #1:** one size doesn't fit all, especially in tooling
- **Problem #2:** two users who want opposite things can both be contextually correct

✨ Modifying Existing Behavior ✨

- **Problem #1:** one size doesn't fit all, especially in tooling
- **Problem #2:** two users who want opposite things can both be contextually correct
- **Problem #3:** adding feature flags for every possible behavior becomes a maintenance nightmare

✨ Modifying Existing Behavior ✨

- **Problem #1:** one size doesn't fit all, especially in tooling
- **Problem #2:** two users who want opposite things can both be contextually correct
- **Problem #3:** adding feature flags for every possible behavior becomes a maintenance nightmare
- **Solution:** empower users with general interfaces that can override default behavior

“How can I auto-close the authentication popup after the user successfully logged in?

It's a bit annoying when developing and you have to close it manually.”

- *@fouadroumieh, Swagger UI #4469*

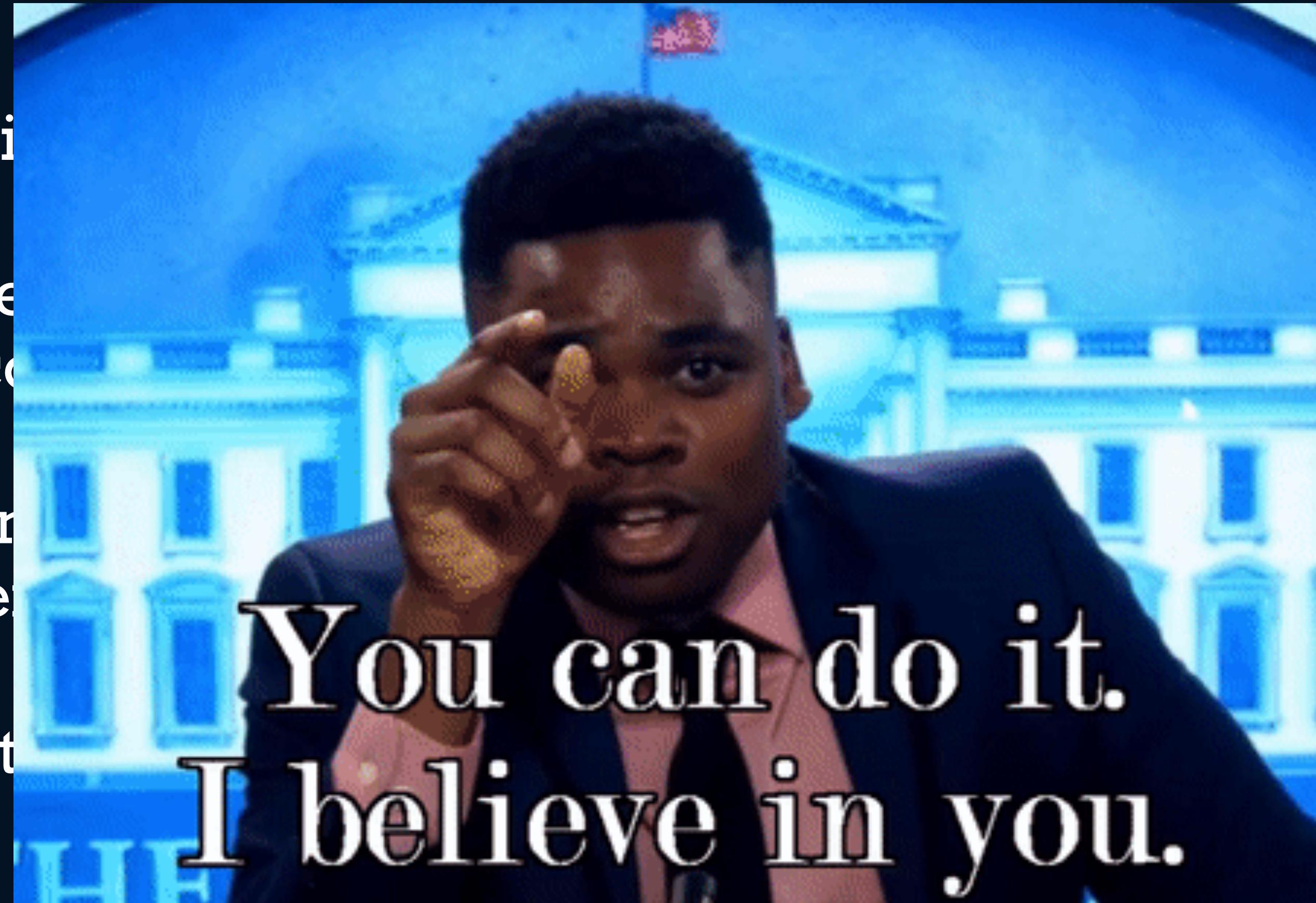
- This is a valid request.

- This is a valid request.
- However, we shouldn't change the default behavior in the core project.

- This is a valid request.
- However, we shouldn't change the default behavior in the core project.
- A config flag (`closeAuthModalOnAuth?`) is not a generally scalable solution.

- This is a valid request.
- However, we shouldn't change the default behavior in the core project.
- A config flag (`closeAuthModalOnAuth?`) is not a generally scalable solution.
- What's a maintainer to do?

- This is the case
- However, the common behavior in the classroom is
- A common gene
- What is the



```
const CloseAuthModalAfterAuthorizationPlugin = (system) => {  
}  
}
```

```
const CloseAuthModalAfterAuthorizationPlugin = (system) => {
  const closeModalAndProxyAction = (ori, args) => {
    setTimeout(() => {
      // use setTimeout to close the modal on the next tick
      system.authActions.showDefinitions(false)
    }, 0)
    return ori(...args)
  }
}
```

```
const CloseAuthModalAfterAuthorizationPlugin = (system) => {
  const closeModalAndProxyAction = (ori, args) => {
    setTimeout(() => {
      // use setTimeout to close the modal on the next tick
      system.authActions.showDefinitions(false)
    }, 0)
    return ori(...args)
  }

  return {
    ...
  }
}
```

```
const CloseAuthModalAfterAuthorizationPlugin = (system) => {
  const closeModalAndProxyAction = (ori, args) => {
    setTimeout(() => {
      // use setTimeout to close the modal on the next tick
      system.authActions.showDefinitions(false)
    }, 0)
    return ori(...args)
  }

  return {
    statePlugins: {

    }
  }
}
```

```
const CloseAuthModalAfterAuthorizationPlugin = (system) => {
  const closeModalAndProxyAction = (ori, args) => {
    setTimeout(() => {
      // use setTimeout to close the modal on the next tick
      system.authActions.showDefinitions(false)
    }, 0)
    return ori(...args)
  }

  return {
    statePlugins: {
      auth: {
        ...
      }
    }
  }
}
```

```
const CloseAuthModalAfterAuthorizationPlugin = (system) => {
  const closeModalAndProxyAction = (ori, args) => {
    setTimeout(() => {
      // use setTimeout to close the modal on the next tick
      system.authActions.showDefinitions(false)
    }, 0)
    return ori(...args)
  }

  return {
    statePlugins: {
      auth: {
        wrapActions: {
          ...
        }
      }
    }
  }
}
```

```
const CloseAuthModalAfterAuthorizationPlugin = (system) => {
  const closeModalAndProxyAction = (ori, args) => {
    setTimeout(() => {
      // use setTimeout to close the modal on the next tick
      system.authActions.showDefinitions(false)
    }, 0)
    return ori(...args)
  }

  return {
    statePlugins: {
      auth: {
        wrapActions: {
          authorize: (ori) => (...args) => {

            },
            authorizeOauth2: (ori) => (...args) => {

              }
            }
          }
        }
      }
    }
}
```

```
const CloseAuthModalAfterAuthorizationPlugin = (system) => {
  const closeModalAndProxyAction = (ori, args) => {
    setTimeout(() => {
      // use setTimeout to close the modal on the next tick
      system.authActions.showDefinitions(false)
    }, 0)
    return ori(...args)
  }

  return {
    statePlugins: {
      auth: {
        wrapActions: {
          authorize: (ori) => (...args) => {
            return closeModalAndProxyAction(ori, args)
          },
          authorizeOAuth2: (ori) => (...args) => {
            return closeModalAndProxyAction(ori, args)
          }
        }
      }
    }
  }
}
```

<https://awywii-close-auth-modal.surge.sh>

“I want to display only tagged groups of operations, without any of Swagger UI’s top menus or definition metadata blocks.”

- a user on #swagger IRC

```
const OnlyDisplayOperationsPlugin = (system) => {
  return {
    ...
  }
}
```

```
const OnlyDisplayOperationsPlugin = (system) => {
  return {
    components: {
      ...
    }
  }
}
```

```
const OnlyDisplayOperationsPlugin = (system) => {
  return {
    components: {
      OperationsLayout: system.React.createClass({
        render() {
          return (
            <div>
              <h1>Operations Layout</h1>
              <p>This is the operations layout component.</p>
            </div>
          );
        }
      })
    }
  }
}
```

```
const OnlyDisplayOperationsPlugin = (system) => {
  return {
    components: {
      OperationsLayout: system.React.createClass({
        render: function() {
          return (
            <div>
              <h1>Operations Layout</h1>
              <p>This is the operations layout component.</p>
            </div>
          );
        }
      })
    }
  }
}
```

```
const OnlyDisplayOperationsPlugin = (system) => {
  return {
    components: {
      OperationsLayout: system.React.createClass({
        render: function() {
          const Operations = system.getComponent("operations", true)

          return (
            <div>
              <Operations />
            </div>
          )
        }
      })
    }
  }
}
```

```
const OnlyDisplayOperationsPlugin = (system) => {
  return {
    components: {
      OperationsLayout: system.React.createClass({
        render: function() {
          const Operations = system.getComponent("operations", true)

          return <div className="swagger-ui">
            <Operations />
          </div>
        }
      })
    }
  }
}
```

```
const OnlyDisplayOperationsPlugin = (system) => {
  return {
    components: {
      OperationsLayout: system.React.createClass({
        render: function() {
          const Operations = system.getComponent("operations", true)

          return <div className="swagger-ui">
            <Operations />
          </div>
        }
      })
    }
  }
}

const ui = SwaggerUI({
  url: "https://petstore.swagger.io/v2/swagger.json",
  dom_id: '#swagger-ui',
  plugins: [
    OnlyDisplayOperationsPlugin
  ],
  layout: "OperationsLayout"
})
```

```
const OnlyDisplayOperationsPlugin = (system) => {
  return {
    components: {
      OperationsLayout: system.React.createClass({
        render: function() {
          const Operations = system.getComponent("operations", true)

          return <div className="swagger-ui">
            <Operations />
          </div>
        }
      })
    }
  }
}

const ui = SwaggerUI({
  url: "https://petstore.swagger.io/v2/swagger.json",
  dom_id: '#swagger-ui',
  plugins: [
    OnlyDisplayOperationsPlugin ←
  ],
  layout: "OperationsLayout"
})
```

```
const OnlyDisplayOperationsPlugin = (system) => {
  return {
    components: {
      OperationsLayout: system.React.createClass({
        render: function() {
          const Operations = system.getComponent("operations", true)

          return <div className="swagger-ui">
            <Operations />
          </div>
        }
      })
    }
  }
}

const ui = SwaggerUI({
  url: "https://petstore.swagger.io/v2/swagger.json",
  dom_id: '#swagger-ui',
  plugins: [
    OnlyDisplayOperationsPlugin ←
  ],
  layout: "OperationsLayout" ←
})
```

<https://awywi-only-display-operations.surge.sh/>

Extending Functionality

“We want to add an iframe [below each operation] that allows users to look through [our] generated code examples in various SDK languages.”

- @Cloudmersive, *Swagger UI #4806*

```
const CodeExamplesAfterResponsesPlugin = () => {
  return {
    ...
  }
}
```

```
const CodeExamplesAfterResponsesPlugin = () => {
  return {
    wrapComponents: {
      ...
    }
  }
}
```

```
const CodeExamplesAfterResponsesPlugin = () => {
  return {
    wrapComponents: {
      responses: (OriginalComponent, { React }) => (props) => {

        }

      }
    }
}
```

```
const CodeExamplesAfterResponsesPlugin = () => {
  return {
    wrapComponents: {
      responses: (OriginalComponent, { React }) => (props) => {
        return <div>
          <OriginalComponent {...props} />
        </div>
      }
    }
  }
}
```

```
const CodeExamplesAfterResponsesPlugin = () => {
  return {
    wrapComponents: {
      responses: (OriginalComponent, { React }) => (props) => {
        return <div>
          <OriginalComponent {...props} />

          </div>
      }
    }
  }
}
```

```
const CodeExamplesAfterResponsesPlugin = () => {
  return {
    wrapComponents: {
      responses: (OriginalComponent, { React }) => (props) => {
        return <div>
          <OriginalComponent {...props} />
          <iframe src="https://docs.cloudmersive.com" width="100%" height="300px" />
        </div>
      }
    }
  }
}
```

<https://awywi-code-examples-after-responses.surge.sh/>

✨ *Publishing Plugins* ✨

✨ Publishing Plugins ✨

✨ Publishing Plugins ✨

- Building your own plugin is good... publishing a plugin for the world to use is **great**

✨ Publishing Plugins ✨

- Building your own plugin is good... publishing a plugin for the world to use is **great**
- Swagger UI plugins can be packaged as JavaScript modules and distributed on npm

✨ Publishing Plugins ✨

- Building your own plugin is good... publishing a plugin for the world to use is **great**
- Swagger UI plugins can be packaged as JavaScript modules and distributed on npm
- Plugins are straightforward to use, even if you aren't a JavaScript wizard

GitHub, Inc. github.com/shockey/swagger-ui-plugin-cornify

shockey / **swagger-ui-plugin-cornify**

Code Issues Pull requests Projects Wiki Insights Settings

Cornify plugin for Swagger-UI v3

Edit

swagger swagger-ui unicorns rainbows Manage topics

16 commits 1 branch 10 releases 1 contributor MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

shockey Update README.md Latest commit 96f99f6 on Oct 27, 2017

src	Release 2.1.1	11 months ago
.gitignore	Init	11 months ago
.npmignore	Release 1.0.1	11 months ago
.release-it.json	Release 2.0.0-beta.2	11 months ago
LICENSE	Update LICENSE	11 months ago
README.md	Update README.md	11 months ago
package.json	Release 2.1.1	11 months ago
webpack.config.js	Release 2.0.0-beta.1	11 months ago

README.md

npm package 2.1.1

swagger-ui-cornify

Cornify plugin for Swagger-UI v3.

This is a lighthearted example of how to create an external plugin for Swagger-UI. This project has a minimal Webpack build pipeline that creates bundles for browser consumption.

The plugin is available on npm as `swagger-ui-cornify`, and minified release copies are attached to each [GitHub Release](#).

If you use the script directly, the module will be exposed on the global scope as `SwaggerUICornify`.

Display a menu

NPM, Inc. www.npmjs.com/package/swagger-ui-cornify

swagger-ui-cornify

2.1.1 • Public • Published a year ago

Readme Admin 1 Dependencies 0 Dependents 10 Versions

npm package 2.1.1

swagger-ui-cornify

Cornify plugin for Swagger-UI v3.

This is a lighthearted example of how to create an external plugin for Swagger-UI. This project has a minimal Webpack build pipeline that creates bundles for browser consumption.

The plugin is available on npm as `swagger-ui-cornify`, and minified release copies are attached to each [GitHub Release](#).

If you use the script directly, the module will be exposed on the global scope as `SwaggerUICornify`.

How to use

Load this plugin (either through npm or directly), and provide it to Swagger-UI when you call it:

```
SwaggerUI({
  url: "http://petstore.swagger.io/v2/petstore.yaml",
  plugins: [
    SwaggerUICornify
  ]
})
```

If/when the API definition you're viewing contains an `x-cornify: true` statement, magic will happen.

Keywords

Display a menu

none

install

```
> npm i swagger-ui-cornify
```

weekly downloads

7

version 2.1.1 license MIT

open issues 0 pull requests 0

homepage github.com repository ⌂ github

last publish a year ago

collaborators

Test with RunKit

Report a vulnerability

The screenshot shows a Mac OS X desktop environment with a dark-themed window. The window title bar reads "unpkg.com/#/" and features standard OS X controls. The main content area has a white background. At the top center is a large, bold, black "UNPKG" logo. Below it is a navigation bar with three items: "Home" (underlined), "Stats", and "About". A short horizontal line is positioned under the "Home" item. Below the navigation is a text block: "unpkg is a fast, global [content delivery network](#) for everything on [npm](#). Use it to quickly and easily load any file from any package using a URL like:" followed by a code snippet "unpkg.com/:package@:version/:file".

Examples

Using a fixed version:

- unpkg.com/react@16.0.0/umd/react.production.min.js
- unpkg.com/react-dom@16.0.0/umd/react-dom.production.min.js

You may also use a [semver range](#) or a [tag](#) instead of a fixed version number, or omit the version/tag entirely to use the [latest tag](#).

- unpkg.com/react@^16/umd/react.production.min.js
- unpkg.com/react/umd/react.production.min.js

If you omit the file path (i.e. use a "bare" URL), unpkg will serve the file specified by the `unpkg` field in `package.json`, or fall back to `main`.

- unpkg.com/d3
- unpkg.com/jquery
- unpkg.com/three

Append a / at the end of a URL to view a listing of all the files in a package.

- unpkg.com/react/
- unpkg.com/lodash/

Display a menu

```
<html>
```

```
</html>
```

```
<html>
  <link rel="stylesheet" href="//unpkg.com/swagger-ui-dist/swagger-ui.css" />
```

```
</html>
```

```
<html>
  <link rel="stylesheet" href="//unpkg.com/swagger-ui-dist/swagger-ui.css" />

  <div id="swagger-ui"></div>

</html>
```

```
<html>
  <link rel="stylesheet" href="//unpkg.com/swagger-ui-dist/swagger-ui.css" />

  <div id="swagger-ui"></div>

  <script src="//unpkg.com/swagger-ui-dist/swagger-ui-bundle.js"></script>

</html>
```

```
<html>
  <link rel="stylesheet" href="//unpkg.com/swagger-ui-dist/swagger-ui.css" />

  <div id="swagger-ui"></div>

  <script src="//unpkg.com/swagger-ui-dist/swagger-ui-bundle.js"></script>
  <script src="//unpkg.com/swagger-ui-cornify/dist/swagger-ui-cornify.min.js"></script>

</html>
```

```
<html>
  <link rel="stylesheet" href="//unpkg.com/swagger-ui-dist/swagger-ui.css" />

  <div id="swagger-ui"></div>

  <script src="//unpkg.com/swagger-ui-dist/swagger-ui-bundle.js"></script>
  <script src="//unpkg.com/swagger-ui-cornify/dist/swagger-ui-cornify.min.js"></script>

  <script>
    SwaggerUIBundle({
      ...
    })
  </script>
</html>
```

```
<html>
  <link rel="stylesheet" href="//unpkg.com/swagger-ui-dist/swagger-ui.css" />

  <div id="swagger-ui"></div>

  <script src="//unpkg.com/swagger-ui-dist/swagger-ui-bundle.js"></script>
  <script src="//unpkg.com/swagger-ui-cornify/dist/swagger-ui-cornify.min.js"></script>

  <script>
    SwaggerUIBundle({
      url: "https://petstore.swagger.io/v2/swagger.json",
      dom_id: "#swagger-ui"

    })
  </script>
</html>
```

```
<html>
  <link rel="stylesheet" href="//unpkg.com/swagger-ui-dist/swagger-ui.css" />

  <div id="swagger-ui"></div>

  <script src="//unpkg.com/swagger-ui-dist/swagger-ui-bundle.js"></script>
  <script src="//unpkg.com/swagger-ui-cornify/dist/swagger-ui-cornify.min.js"></script>

  <script>
    SwaggerUIBundle({
      url: "https://petstore.swagger.io/v2/swagger.json",
      dom_id: "#swagger-ui",
      plugins: [SwaggerUICornify]
    })
  </script>
</html>
```

If/when the API definition you're viewing contains an `x-cornify: true` statement, magic will happen.

Swagger UI

file:///Users/kyle/Code/ui/dist/index.html

swagger https://petstore.swagger.io/v2/swagger.json Explore

Swagger 1.0.0

This is just a test server Petstore server. You can test it at <http://swagger.io> or on <https://petstore.swagger.io>.
Terms of service: [here](#)
Complete documentation: [here](#)
API spec: [here](#)
First API

Schemes: **HTTPS**

pet Everything about your Pets

POST /pet Add a new pet to the store

PUT /pet Update an existing pet

GET /pet/findByStatus Finds Pets by status

GET /pet/findByTags Finds Pets by tags

GET /pet/{petId} Find pet by ID

POST /pet/{petId} Update an existing pet in the store with form data

DELETE /pet/{petId} Deletes a pet

POST /pet/{petId}/uploadImage uploads an image

The Swagger UI interface is overlaid on a background featuring several whimsical illustrations. At the top left is a pink unicorn standing under a double rainbow with clouds. To the right is a dark purple unicorn head with a single horn. Below the purple unicorn is a large, vibrant rainbow with white clouds. To the right of the rainbow is a green 'Authorize' button with a lock icon. Further down on the right is a smaller illustration of a pink unicorn. At the bottom right is a yellow sun-like character with a face, surrounded by multiple rainbows. The overall theme is magical and playful.

```
<html>
  <link rel="stylesheet" href="//unpkg.com/swagger-ui-dist/swagger-ui.css" />

  <div id="swagger-ui"></div>

  <script src="//unpkg.com/swagger-ui-dist/swagger-ui-bundle.js"></script>
  <script src="//unpkg.com/swagger-ui-cornify/dist/swagger-ui-cornify.min.js"></script>

  <script>
    SwaggerUIBundle({
      url: "https://petstore.swagger.io/v2/swagger.json",
      dom_id: "#swagger-ui",
      plugins: [SwaggerUICornify]
    })
  </script>
</html>
```

<https://github.com/shockey/awywi>

Questions?

Fin