



I'm not robot



Continue

Swagger ui dist

👉🏻 Want to get an easy open source contribution? Check out our Good First Issue label. 🏴‍☠️ are you looking for the old version of Swagger UI? See branch 2.x. This repository publishes three different NPM modules: swagger-ui is a traditional npm module for use in single-page applications capable of solving dependencies (via Webpack, Browserify, etc.). swagger-ui-dist is a dependency-free module that includes everything you need to serve the Swagger user interface in a server-side project, or a single-page application that can't solve the npm module dependencies. swagger-ui-react is the Swagger UI packaged as a React component for use in React applications. We strongly suggest you use swagger-ui instead of swfaronnade-ui-dist if you build a single-page application, since swagonnade-ui-dist is significantly larger. If you're looking for ol' HTML/JS/CSS, download the latest version and copy the contents of the folder/dist to your server. Compatibility The OpenAPI specification has been the subject of 5 revisions since its initial inception in 2010. Compatibility between the Swagger user interface and the OpenAPI specification is as follows: Documentation Use Customization Overview Plugin API Development of custom layout Development Contributor Integration Tests You'll need JDK version 7 or more as shown here Integration Tests can be run locally with npm run e2e - make sure you don't run a development server. Swagger UI browser support works in the latest versions of Chrome, Safari, Firefox and Edge. Known Questions To help with migration, here are the problems currently known with 3.X. This list will be updated regularly and will not include features that have not been implemented in previous versions. Only a portion of the previously supported settings are available. The JSON form editor is not implemented. Support for the Format collection is partial. l0n (translations) is not implemented. Relative trajectory support for external files is not implemented. Security Contact Please disclose any security-related problems or vulnerabilities by sending a security@swagger.io, instead of using public problem tracking. Page 2 Swagger UI is part of the Swagger project. The Swagger project allows you to produce, view and consume your own RESTful services. No proxy or third-party services are required. Do it your way. Swagger UI is a non-addictive collection of HTML, JavaScript and CSS assets that dynamically generate beautiful documentation and a sandbox from a Swagger-compatible API. Because the Swagger user interface has no dependencies, you can host it in any server environment, or on your local machine. What is Swagger? Swagger™'s goal is to define a standard and linguistic agnostic interface to REST APIs that allows humans and computers to discover and understand the capabilities of the service without access to source code, documentation or network traffic inspection. When properly defined via Swagger, a consumer can and interact with the remote service with minimal implementation logic. Similar to what the interfaces have done for lower-level programming, Swagger removes guesswork by calling the service. Check out Swagger-Spec for more information about the Swagger project, including additional libraries with support for other languages and more. Compatibility The OpenAPI specification has been the subject of 4 revisions since its initial inception in 2010. The compatibility between swagger-ui and the OpenAPI specification is as follows: Swagger UI Version Release Date OpenAPI Spec compatibility Notes Status 2.2.10 2017-01-04 1.1, 1.2, 2.0 tag v2.2.10 2.1.5 2016-07-20 1.1, 1.2, 2.0 tag v2.1.5 2.0.24 2014-09-12 1.1, 1.2 tag v2.0.24 1.0.13 2013-03-08 1.1, 1.2 tag v1.0.13 1.0.1 2.1.011-10 1.0, 1.1 tag v1.0.1 Note: Support for versions over 1.2 is limited and you may experience problems that we cannot improve and support. We strongly recommend that you upgrade your specification to the latest version. How to use it Download You can use the swagger-ui code AS-IS! No need to build or recompile - just clone this repo and use the pre-built files in the dist folder. If you like swagger-ui as it is, stop here. Swagger UI browser support works in all persistent-leaf desktop browsers (Chrome, Safari, Firefox). The Internet Explorer support is version 8 (IE8) and above. Build You can rebuild swagger-ui on your own to modify it or just so you can say you've done it. To do this, follow these steps: Make sure you have 4.0 or more nodes installed, and on your classpath. Windows users: Please install Python before following the guidelines below for node-gyp reconstruction to run. npm install npm run build You should see the distribution under the dist folder. Open ./dist/index.html to launch the Swagger user interface into a Development browser Use npm run serve to make a new generation, watch for changes, and serve the result to . Running with Docker docker shoot fanaggerapi/swagger-ui docker run -p 80:8080 fanaggerapi /swagger-ui Build using Docker To build swfaronnade-ui using a Docker container: docker build -t swagagger-ui-builder. docker run -p 80:8080 swagon-ui-builder This will start Swagger UI at . Use Once you open the Swagger user interface, it will load the Swagger Petstore service and display its APIs. You can enter your own server url and click 'Explorer' to view the API. Personalize You can choose to customize the interface Swagger for your organization. Here's a look at what's in its various directories: dist: Contains a distribution that you can deploy on a server or load from your local machine. dist/lang: The lib location swagger: Contains javascript dependencies that boast-ui depends on node_modules: Contains node modules that swagger-ui uses for its development. src src/hand/templates: handlebar models used to make swfaronnade-ui src/hand/html: html files, some images and css src/hand/javascript: main code SwaggerUI To use swagger-ui you should take a look at the source of swfaronnade-ui page.html and and it. This essentially requires you to instantiate a SwaggerUI object and the call load() on it as below: var swaggerUi = new SwaggerUI({ url: ' ', dom_id: 'swagger-ui-container' }); swaggerUi.load(); Settings Description of the url's name The url pointing to swagger.json (Swagger 2.0) or the list of resources (previous versions) according to OpenAPI Spec. permissions An object of authorization to pass to swagger-js. Setting it here will trigger the inclusion of any custom permission or signature logic when extracting the swaggering description file. Note that the object structure must be key: AuthorizationObject - spec A JSON object describing the OpenAPI specification. When used, the url setting will not be parsed. This is useful for testing manually generated specifications without hosting them. Works for Swagger 2.0 specs only. validatorUrl By default, Swagger-UI tries to validate the specifications against swagger.io's online validator. You can use this setting to set a different validator URL, for example for locally deployed validators (Badge Validator). Putting it null will disable the validation. This setting is relevant to Swagger 2.0 specs only. dom_id id of a dom element within which SwaggerUI will put the user interface for swagger. booleanValues SwaggerUI makes boolean data types like a dropdown. By default, it provides a real and false chain as possible choices. You can use this setting to change the dropdown values to be something else, for example 0 and 1 by setting booleanValues again Array(0, 1). docExpansion controls the display of the API list. It can be defined as none (default), list (shows operations for each resource), or complete (fully expanded: displays operations and their details). apisSorter Apply a kind to the API/tags list. It can be alpha (sort by name) or a function (see Array.prototype.sort()) to find out how the sorting function works). By default is the command returned by the unchanged server. operationsSorter Apply a kind to the operating list of each API. It can be alpha (sorting by alphanumeric paths), method (sort by HTTP method) or a function (see Array.prototype.sort()) to find out how the sorting function works). By default is the command returned by the unchanged server. defaultModelRendering Controls how models are displayed when the API is first rendered. (The user can always change the rendering of a given model by clicking on the links 'Model' and 'Model Schema'.) It can be defined as model or scheme, and the default is scheme. onComplete This is a parameter recall that can be passed to each request, use the headers as below: swaggerUi.api.clientAuthorizations.add(key, new SwaggerClient.ApiKeyAuthorization('XXXX', header)). Note! You can pass several header params on a single request, just use unique names for them (key is used in the example above). Location and translation Location files are in the lang directory. Note that language files and translator are not included in SwaggerUI by default. You have to add them manually. To enable the translation, you need to attach the next two lines to your Swagger's index.html (or another entry point you use) The first-line script is a translator and script src-lang/translator.js type 'text/javascript' /script. The js second is your language lexemes. If you want to annex support for the new language, you just need to create lang/your_lang.js and fill it in as it is done in existing files. To annex the new lexeme for translation, you need to do two things: lexeme in the language file. Example of a new line: new sentence: translation of the new sentence. Mark this lexeme in source.html with the data-sw-translate attribute. Example of modified source: 'anyHtmlTag data-sw-translate' /anyHtmlTag, or 'anyHtmlTag data-sw-translate value=new sentence'. For now, only html inside, title attribute and value attribute will be translated. Translated. OR support: How to deal with Can't read from the server. There may not be the appropriate original access control settings. CORS is a technique to prevent websites from doing bad things with your personal data. Most browsers - JavaScript toolkits not only support CORS, but apply it, which has implications for your API server that supports Swagger. You can read on CORS here: . There are two cases where no action is required for CORS support: swagger-ui is hosted on the same server as the application itself (same host and port). The application is located behind a proxy that allows the required CORS headers. This can already be covered within your organization. Otherwise, the CORS support must be enabled for: Your Swagger docs. For Swagger 2.0, it's the swagger.json and all the \$refed external documents, and for the previous version, these are the resource list and API reporting files. For the Try it now button to work, CORS must also be enabled on your API endpoints. CORS support test You can check CORS support with one of three techniques: lock your API and inspect the headers. For example: \$ curl -I HTTP/1.1 200 OK Date: Sat, 31 Jan 2015 23:05:44 GMT Access-Control-Allow-Origin: Access-Control-Allow-Methods: GET, POST, DELETE, PUT, PATCH, OPTIONS Access-Control-Allow-Headers: Content-Type, api_key, Authorization Content-Type: application/json Content-Length: 0 This tells us that the list of petstore resources supports OPTIONS, and the following headers: Content Type, api_key, Authorization. Try swagger-ui from your file system and look at the debugging console. If CORS isn't activated, you'll see something like this: XMLHttpRequest can't load . There are no 'Access-Control-Allow-Origin' headers on the requested resource. The zero origin is therefore not allowed to access it. Swagger-UI cannot easily display this error state. Using the web. Keep in mind that this will show a positive result, even if access-control-allow-headers is not available, which is still needed for Swagger-UI to function properly. CORS Activation The CORS activation method depends on the server and/or frame you use to host your application, provides information on how to activate horns in some common web servers. Other servers/frameworks can provide you with information on how to activate it specifically in their case of use. Cors and Header Parameters Swagger allows you to easily send headers as settings for requests. The names of these headers must be support in your CORS configuration. Based on our example above: Access-Control-Allow-Headers: Content-Type, api_key, Authorization Only headers with these names will be allowed to be sent by Swagger-UI. How to Improve It Create Your Own Flag-api/Fanaronnade-ui To Share Your Changes, submit a traction request. Change the log Please see versions for the change log. Copyright License 2016 SmartBear Software Under the Apache license, version 2.0 (the license); you cannot use this file unless in accordance with the license. You can obtain a copy of the license at apache.org/licenses/LICENSE-2.0 Unless applicable law requires it or accepts in writing, software distributed under the license is distributed on an AS IS basis, without WARRANTIES OR WHAT NATURE, whether express or implied. See the license for the specific language governing licenses and limitations under the license. Licence.

[smith_wesson_sd9ve_holster.pdf](#) , [normal_5f92149f5ffb8.pdf](#) , [velocidad media ejercicios resueltos.pdf](#) , [starbucks_barista_championship_study_guide_2019](#) , [kik_app_download.apk](#) , [moviraninanovubijilaf0.pdf](#) , [all_instagram_photos_iphone](#) , [blackberry_curve_8900_whatsapp_free](#) , [premam_movie_download_in_tamil_dubbed_hd](#) , [matlab_for_engineers.pdf](#) , [normal_5f872c8d5971e.pdf](#) , [verbs_adverbs_prepositions_conjunctions_and_interjections_worksheets.pdf](#) , [las_3_preguntas_jorge_bucay_descargar](#) ,