**RunKit Embed**

**Embed node.js on any website**

It’s easy to embed RunKit on any website. The embedded version of a RunKit notebook has a single executable code cell. It will grow and shrink vertically to fit its content, and fill its container horizontally. You can have as many instances as you like on a given page.

You can also embed an existing notebook using [OEmbed](https://runkit.com/docs/embed#oembed).

**Embedded Example**

There are two ways to get RunKit on your page, described below. In both cases, you want to include:   
<script src="https://embed.runkit.com"></script>

**1. Attaching RunKit to an existing element**

This is the easiest way to embed RunKit. Just include one script tag, and let it know which element contains the source code you want to attach to with the data-element-id attribute.



<script src="https://embed.runkit.com" data-element-id="my-element"></script>

​

<!-- anywhere else on your page -->

<div id="my-element">// GeoJSON!

var getJSON = require("async-get-json");

​

await getJSON("https://storage.googleapis.com/maps-devrel/google.json");</div>

**2. Programmatically creating a notebook**

You can also create an embedded notebook programmatically. Just include the same script tag, then call RunKit.createNotebook.



<script src="https://embed.runkit.com"></script>

<div id="my-element"></div>

<script>var notebook = RunKit.createNotebook({

// the parent element for the new notebook

element: document.getElementById("my-element"),

// specify the source of the notebook

source: "// GeoJSON!\nvar getJSON = require(\"async-get-json\");\n\nawait getJSON(\"https://storage.googleapis.com/maps-devrel/google.json\");"

})</script>

**Creation Options**

In addition to the ones mentioned above, there are some additional options available when creating a notebook. Most are available using both embed methods, but some are only available when creating a notebook programatically.

Create a notebook that can not be edited or run

If "endpoint", the notebook will be run as a [server endpoint](https://runkit.com/docs/endpoint) and a link to it will be exposed in the embed.

Provide a semver range that the node engine should satisfy, e.g. 4.0.x or > 6.9.2

**env / data-env-\***

An array of environment variables that will be made available in the execution environment (under process.env). These should be strings in the form "name=value". When using the data-env- form, only letters and numbers are allowed, and they will be uppercased (e.g. data-env-foo will become process.env.FOO)

**title**

The title of the notebook when it is saved on RunKit.

**minHeight**

The minimum height of the embed. This value is 130px by default.

**packageTimestamp**

The timestamp in UTC milliseconds to recreate the state of package availability. No packages published to npm after this time are available in this embed, which is useful if you want to freeze the conditions for reproducing a bug, or guaranteeing that versions of a package and its dependencies are loaded. This value is set to the time the embed is created by default.

**preamble**

You can put setup code in the preamble so it doesn’t pollute the main embed. This is useful when using RunKit embeds for your documentation: in the preamble you can require your package, then use it in the main embed.

**onLoad**

A function that will be called when the embed has fully loaded. The function will be passed a reference to the notebook.

**onEvaluate**

A function that will be called any time the notebook is evaluated.

**Notebook API**

The notebook object (returned by createNotebook) also exposes an API for interacting with it, with the following methods:

**notebook.getSource(callback)**

Pass a callback, which will be called with the current source of the notebook.

**notebook.setSource(source, callback)**

Pass a string to set the notebook’s source. The callback is optional.

**notebook.getPreamble(callback)**

Pass a callback, which will be called with the current source of preamble of the embed.

**notebook.setPreamble(source, callback)**

Pass a string to set the embed preamble’s source. The callback is optional.

**notebook.evaluate(callback)**

Force the notebook to evaluate. The callback is optional, and only represents that the request to evaluate has succeeded, but not that evaluation is finished.

**notebook.getShareableURL(callback)**

Pass a callback, which will be called with the shareable URL of the embed.

**Bindings**

* [React Component](https://github.com/runkitdev/react-runkit)
* [Ember Component](https://github.com/runkitdev/ember-runkit)
* [Angular Component](https://github.com/runkitdev/angular-runkit)

**OEmbed**

**Embed existing notebooks with OEmbed.**

Our OEmbed endpoint is available at: https://embed.runkit.com/oembed

We support only the JSON format for requests. See the [OEmbed documentation](http://oembed.com/) for information about how to implement the protocol. RunKit does support auto-discovery of notebooks.

**Dynamic Height**

RunKit notebooks are highly interactive, and as a result they want to change height in response to user interaction. If you're using RunKit with [Embed.ly](http://embed.ly), using their [Platform.js](http://docs.embed.ly/docs/platformjs) library will make sure everything works smoothly. If you're using OEmbed directly, you'll want to include this script, or something similar, on your page:



<script>

window.addEventListener('message', function(e) {

if (e.origin !== "https://runkit.com")

return;

​

try {

var data = JSON.parse(e.data);

} catch (e) {

return false;

}

​

This code is not RunKit specific, and it's how many sites implement this behavior. It's a simple protocol for an iframe to message its parent requesting the height of the frame be changed.

**Examples**

RunKit embeds are used on many websites for [documentation](https://facebook.github.io/immutable-js/docs/#/Map), to provide examples in a [blog post](https://forge.autodesk.com/blog/runkit-nodejs-playground-forge-sample), or to let visitors [run a code snippet](https://stripe.com). Check out these sites using RunKit embeds to enhance their pages: