

Simulate your Functions

Before making a Chainlink Functions request from your smart contract, it is always a good practice to simulate the source code off-chain to make any adjustments or corrections.

Currently, there are several options for simulating a request:

- [Chainlink Functions playground](#) .
- [Chainlink Functions Hardhat Starter Kit](#) : Use `thenpx hardhat functions-simulate-script` command to simulate your Functions JavaScript source code.
- [Chainlink Functions Toolkit NPM package](#) : Import this NPM package into your JavaScript/TypeScript project, then use `thesimulateScriptfunction` to simulate your Functions JavaScript source code.

[Chainlink Functions playground](#)

caution

The Chainlink Functions playground can simulate Chainlink Functions within the browser. There may be differences to the real Chainlink Functions environment. For instance, you are limited to vanilla JavaScript and cannot use all the features of Chainlink Functions, such as imports of external modules. To access all the features, use the [Chainlink Functions Hardhat Starter Kit](#) or [Chainlink Functions Toolkit NPM package](#) .

To use the [Chainlink Functions Playground](#) , enter any source code, arguments, and secrets you would like to use. Click the `Run code` button to view the output.

[Chainlink Functions Hardhat Starter Kit](#)

This repository comes preconfigured with [Hardhat](#) and the [Chainlink Functions Toolkit NPM package](#) , allowing you to quickly get started with Functions.

To simulate:

1. In a terminal, clone the [functions-hardhat-starter-kit repository](#) and change to the `functions-hardhat-starter-kit` directory.

```
git clone https://github.com/smartcontractkit/functions-hardhat-starter-kit&&\cdfunctions-hardhat-starter-kit
```

2. Run `npm install` to install the dependencies.

3. For simulation, you don't need to set up the environment variables. Run `npm run hardhat functions-simulate-script` to simulate the [calculation-example.js](#) JavaScript source code. Note: This example calculates the continuously compounding interest for a given principal amount over one month. It takes the principal amount and annual percentage yield (APY) as input arguments and uses [Euler's number](#) to compute the total amount after interest.

`npm run hardhat functions-simulate-script` Result:

`secp256k1 unavailable, reverting to browser version`

Response returned by script during local simulation: 1003757

note

The JavaScript source code is set up in the [Functions-request-config.js](#) file. To simulate your own JavaScript source code, modify `source` to replace `./calculation-example.js` with the relative path to your file.

[Chainlink Functions Toolkit NPM package](#)

Follow the [Simple Computation](#) guide to learn how to import the [Chainlink Functions Toolkit NPM package](#) into your JavaScript project to simulate and execute a Functions request. You can read the [Examine the code section](#) for a detailed description of the code example.