# 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-scriptcommand 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 theChainlink Functions Hardhat Starter Kit orChainlink Functions Toolkit NPM package .

To use theChainlink Functions Playground , enter any source code, arguments, and secrets you would like to use. Click theRun codebutton to view the output.

## Chainlink Functions Hardhat Starter Kit

This repository comes preconfigured withHardhat and theChainlink Functions Toolkit NPM package , allowing you to quickly get started with Functions.

To simulate:

1. In a terminal, clone thefunctions-hardhat-starter-kit repository and change to thefunctions-hardhat-starter-kitdirectory.

gitclone https://github.com/smartcontractkit/functions-hardhat-starter-kit&&\cdfunctions-hardhat-starter-kit 2. Runnpm installto install the dependencies.

npminstall 3. For simulation, you don't need to set up the environment variables. Runnpx hardhat functions-simulate-scriptto simulate thecalculation-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 usesEuler's number to compute the total amount after interest.

npx hardhat functions-simulate-scriptResult:

secp256k1 unavailable, reverting to browser version

Response returned by script during local simulation: 1003757

note

The JavaScript source code is set up in theFunctions-request-config.js file. To simulate your own JavaScript source code, modifysourceto replace./calculation-example.jswith the relative path to your file.

## Chainlink Functions Toolkit NPM package

Follow theSimple Computation guide to learn how to import theChainlink Functions Toolkit NPM package into your JavaScript project to simulate and execute a Functions request. You can read theExamine the code section for a detailed description of the code example.