# INFURA Infrastructure Test

This is a take-home test for the INFURA infrastructure team that we would like

you to attempt. You'll find a number of steps to complete below that will

require you to do some coding and configuration; feel free to use the

programming or scripting language(s) you're most comfortable with (although

Python and Go are preferred) and tools, libraries, or frameworks you believe are

best suited to the tasks listed below.

Please submit any code and documentation you write when you feel comfortable with

the result. This test is intended to be completed within a week; should you need

more time, please let us know.

Please also note and let us know how long you worked on the test; this is for

informational purposes only and allows us to make adjustments and improvements

for future applicants (feel free to provide feedback on the test itself too).

1. Register for an [INFURA Project ID](https://infura.io/register)

1. You will have to use this key for subsequent requests to INFURA endpoints,

as briefly shown in the [Choose a network](https://infura.io/docs/gettingStarted/chooseaNetwork) section of the site

2. Create an application that retrieves Ethereum Mainnet transaction and block

data via the INFURA JSON-RPC API from \_https://mainnet.infura.io/v3/[projectId]\_

1. Examples of useful methods include eth\_getTransactionByBlockNumberAndIndex or eth\_getBlockByNumber, but feel free to add any other methods

2. See [the INFURA API docs](https://infura.io/docs)

for a list of supported JSON-RPC methods

3. See [the Ethereum docs](https://eth.wiki/json-rpc/API) for information on the

Ethereum API itself

3. Expose the retrieved transaction and block data via REST endpoints that your

application provides

4. Set up your application to run in a [Docker container](https://www.docker.com)

5. Create a load test for your application

6. Run some load test iterations and document the testing approach and the

results obtained

1. Specify some performance expectations given the load test results:

e.g., this application is able to support X requests per minute

7. Write up a short document describing the general setup of the components

you've put together as well as instructions to run your application

8. \*\*Bonus points\*\*: add unit tests to cover most of the code you've written

9. Submit your application and load test code, as well as any associated

documentation, to the `main` branch of the Github repository we've set up for

this purpose