



## QA API Technical Test

Welcome to Ito World's QA API technical test exercise!

We would like to see your approach to testing REST API with an automated test framework.

The API in question is the open-source API provided by [ExchangeRate-API](#)

[ExchangeRate-API](#) is an open-source currency conversion API, that returns JSON object with exchange rates from the base currency to all the supported currency codes.

### Pre-requisites

#### Create an account

To get access to the API Key, you will need to create an account.

URL: <https://app.exchangerate-api.com/sign-up>

The API key can be seen on the [Dashboard](#) page.

Any issues with creating an account please contact: [meenu.atrey@itoworld.com](mailto:meenu.atrey@itoworld.com)

#### API Endpoint

Once you have an account - you can now access the API and the documentation

#### Acceptance Criteria

1. Create an automated test framework in: Python, Java, JavaScript or C# for the ExchangeRate API.
2. Add automated test coverage for the [Standard Requests](#) endpoint and [Pair Conversions](#) endpoint. Please note, for Pair conversion the endpoint doesn't work with /AMOUNT. Please use the endpoint: <https://v6.exchangerate-api.com/v6/YOUR-API-KEY/pair/EUR/GBP>
  - a. Create a public github account if you don't have it previously. Create a repo in your account and make it shared.
  - b. Create test framework under the shared repo.
  - c. Write a function which enables user to enter three parameters- input currency type, input currency value and output currency type. Function should output the currency value in output currency type.
  - d. Write test coverage for both APIs response and test coverage/error capturing for above function. The coverage doesn't need to cover all scenarios. A few will suffice.
  - e. Your approach to different types of test coverage against both the APIs and the function written is important.
3. Include a README with following information:
  - a. Instructions on how to setup the test framework.
  - b. Instructions to execute the tests.
  - c. Have you found any bugs / unexpected behaviours?

**Additional Information**

1. What we're looking for is your approach to designing and creating a small test framework.
2. How would we interpret the test results clearly?
3. Consider how we can add test data to existing test cases.
4. Your approach to writing a clean and well documented code.
5. How simple would it be to maintain the code in the test framework?

**Submitting the technical test**

1. You have one week to complete this technical exercise.
2. Once ready, please submit your repo's link to: [meenu.atrey@itoworld.com](mailto:meenu.atrey@itoworld.com) with cc to: [gary.hawkes@itoworld.com](mailto:gary.hawkes@itoworld.com) , [chiara.hallam@itoworld.com](mailto:chiara.hallam@itoworld.com)
  - a. You will be expected to demo and discuss this exercise at the interview.
3. Any further questions, please contact: [meenu.atrey@itoworld.com](mailto:meenu.atrey@itoworld.com) or [gary.hawkes@itoworld.com](mailto:gary.hawkes@itoworld.com)