



Software QA Engineer

## Coding Challenge - Marvel API - v2

The Marvel Comics API allows developers to access information about Marvel's vast library of comics.

We want you to build a testing framework that will help our support and engineering team have ongoing confidence in the quality of this API.

There will be 2 exercises:

- Part One is an opportunity to demonstrate your skills by creating a detailed test which proves the possible API requests and expected responses do not regress.
- Part Two requires the implementation of a concrete example expressed as a Cucumber scenario and provided to you by an analyst. This is typical of the way analysts and engineers work together using our Behaviour Driven Development practice.

## Part One

**Imagine that one of our critical products depends on the Marvel API to get the characters data (see <http://developer.marvel.com/>).**

1. Create a test case for the characters endpoint (</v1/public/characters>) that ensures every possible character record has all the JSON properties according to the swagger schema that can be found here:

<https://gateway.marvel.com/docs/public>

Do the necessary assertions to detect regression or changes to the api.

**Note:** This endpoint has pagination and you must iterate all the pages to test every character.

2. Include negative testing for the same characters endpoint. a. Invalid parameters, wrong credentials, etc.

## Part Two

**Provided with the following Cucumber scenario from our analyst, we need to validate the use case.**

1. Create a test case using the [Cucumber](#) scenario below expressed in Gherkin syntax. Use the Marvel API documentation (<https://developer.marvel.com/docs>) to identify the entities mentioned in the scenario steps
2. Use an open source reporting tool to create a test report of your results.
3. Please comment on the suitability of BDD for this type of test.

**Note:** Feel free to use Java or Javascript as your “glue”

### Cucumber Scenario

Feature: Marvel Characters

Scenario Outline: Reconcile the number of comics for each character retrieved using the alternative methods for get /characters and get /characters/{characterId}/comics

Given I have a valid `MarvelAPIKey` credential

When I call get characters

\* with page limit 100

\* with ascending order by name

Then the response includes http status code 200

And includes the <available comics> for each <character-id> and <character-name>

When I call get characters/<character-id>/comics

Then the response includes http status code 200

And includes a list of comics totalling <total-num-comics>

And <available-comics> matches <total-num-comics>

Example:

<character-name>	<character-id>	<available-comics>	<total-num-comics>
"3-D Man"	1011334	12	12
"A-Bomb (HAS)"	1017100	4	4
"AIM"	1009144	52	52

### NOTES

- We're looking for a running test suite that can be executed on the command line

- Use of third party libraries is ok; in the coding interview we'll be asking you about your choices.
- The API keys / secrets should not be stored with the code. See here: <https://support.google.com/cloud/answer/6310037>

## On Completion

→ Write a markdown README.md with:

- Your name
- The project title (the title of this document)
- How to install any dependencies, files or environment variables your code requires
- How to build and run your project for the various test cases

→ Share your source code with the Yapily interview team. Choose the way that works best for you:

- You can host the code on a private git repository (e.g. GitHub, Bitbucket);
- You can compress all the source code;
- Upload the zip file on an online drive (GDrive, Dropbox, etc)