KodyPay

Test Automation Engineer

Testing Challenge - Marvel API - v1

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

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

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.

Part One

1. Create a test case for the characters endpoint that ensures every possible character record has all the JSON properties listed below:

GET /v1/public/characters

```
{
    "id": ...,
    "name": ...,
    "description": ...,
    "modified": ...,
    "resourceURI": ...,
    "thumbnail": ...,
    "comics": ...,
    "stories": ...,
    "events": ...,
    "urls": ...
}
```

    a. The values can be empty or null, but the properties must be present;

    b. 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

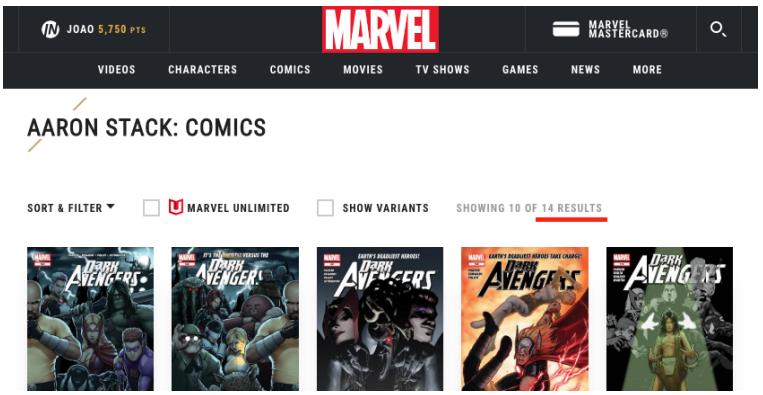
Some characters have a dedicated details web page that can be opened on the browser.
The URL of the details page is a property returned in the “urls” of the character.
See the following example:

GET /v1/public/characters

```
{
    “id”: 1010699,
    “name”: "Aaron Stack",
    “urls”: [
      {
         "type": "detail",
         "url": "http://marvel.com/comics/characters/1010699/aaron_stack"
       },
       ...
    ],
    "comics": {
       "available": 14,
       ...
    }
}
```

1. Create a visual test for a given character id that confirms that the **detail web page** displays the same number of “comics” of the ones returned by the API.

    a. If your test receives the parameter `1010699` you would expect 14 comics available on the page of the URL: http://marvel.com/comics/characters/1010699/aaron_stack

    b. 

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 KodyPay 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)
  - ◆ Share or send access details to your code: tech-interviews@kodypay.com