



Backend Engineer

Coding Challenge

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

We want you to use the Marvel API (see <http://developer.marvel.com/>) to build a **Characters API**:

Part One:

1. Serve an endpoint **/characters** that returns all the Marvel character ids only, in a JSON array of numbers.
 - a. Because Marvel API only returns max 100 records per request, you need to load all of them beforehand with your application, and cache it in memory or file, to efficiently serve your endpoint;
 - b. The request should be something like:

```
GET /characters
```

```
[ 1009718, 1017100, 1009144, 1010699, 1016823, 1009148, 1011334, ... ]
```

2. Serve an endpoint **/characters/{characterId}** that contains the real-time data from the Marvel API **/v1/public/characters/{characterId}**, but containing only the following information about each character: `id`, `name`, `description`, `thumbnail`

```
GET /characters/1009718
```

```
{
  "id": 1009718,
  "name": "Wolverine",
  "description": "Born with super-human senses and the power to heal from almost any wound, Wolverine was captured by a secret Canadian organization and given an unbreakable skeleton and claws. Treated like an animal, it took years for him to control himself. Now, he's a premiere member of both the X-Men and the Avengers.",
  "thumbnail": {
    "path": "http://i.annihil.us/u/prod/marvel/i/mg/2/60/537bcaef0f6cf",
    "extension": "jpg"
  }
}
```

You'll need to sign up for Marvel developer API key at <http://developer.marvel.com> (free)

Once you have a key then the API documentation is at <http://developer.marvel.com/docs>

Part Two:

Lets add more value to the Characters API.

3. Enable a translated version to another language of the character's "description".
 - a. Accept a query parameter with the language ISO-639-1 code: **/characters/{characterId}?language={languageCode}**
 - b. Use any translation service of your choice, it can be an API or library, but the goal is to execute the translations in real-time
4. Create a Swagger specification for your Characters API that can be viewed with Swagger UI or imported to Postman

Constraints

- We're looking for a running server that can be accessed on **http://localhost:8080/**
- 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

Also write a markdown README.md with instructions on how to:

- Install any dependencies, files or environment variables your code requires
- How to build and run your server

Send your source code in a compressed file to: interview@yapily.com