

HostGator Latin America

Backend Code Challenge

The challenge

Create a Web API that queries the Cat API: <https://docs.thecatapi.com>

The API must allow the search for Cat Breeds by name. The API should cache the results from the Cat API into a local MySQL database. If the cat breed is not found in the local MySQL database, it will query the cat API. The only endpoint that must be implemented is the GET to /breeds

Constraints

1. Use the RESTful standard;
2. Use the proper HTTP verbs and statuses;
3. You can choose between either stack:
 - a. PHP 7 (with Composer) and Slim Framework <https://github.com/slimphp/Slim> *
 - b. Golang (with Go Modules) and Gin <https://github.com/gin-gonic/gin> ;
4. The API must implement authentication with JWT;
5. Feel free to use any third-party library;
6. The project must be on a private Github repository (we will send you the user to share with);
7. The project must not be committed at once. We want to see your history.
8. The application must be covered by unit tests and API tests (integration tests);
9. The database schema must be created via migrations;
10. The application must be well documented (comments, README, Endpoints, flow diagrams, architecture diagrams...);
11. Gain extra points if you do it using Golang :-)

If you find any blocker while solving the challenge, feel free to apply any solution you find necessary to complete it;

** This means you should not use entire Frameworks like Laravel, Cake or Symfony.*



Tests

These are the tests that will be done against your API

1. POST <http://localhost:XX/login>
 - a. body: '{"password":"@#\$RF@!718", "username":"admin"}'
2. GET <http://localhost:XX/breeds?name=sib>

This endpoint will be called twice, the second time it **should not** try to access the Cats API (because the result will be cached in the local database).

3. GET <http://localhost:XX/breeds?name=a>
4. GET <http://localhost:XX/breeds?potato=yes>
5. GET <http://localhost:XX/breeds?name=sib>

This time, we will block external access (**the code won't be able to reach the Cat API**).

6. GET <http://localhost:XX/breeds?name=sib>

This time, we will change the database credentials (**the code won't be able to reach the local database**).

Extras

1. Create a Docker and Docker compose environments for the app or deploy the app in a free public web server;
2. Create CI/CD Pipelines (Jenkins, CircleCI, GitLab, Travis, etc);
3. Document the API using a tool such as Swagger;
4. Implement a healthcheck endpoint;

