****

**Who are you?**

You are an experienced PHP backend developer, with good knowledge in setting up a PHP development environment with Linux/Nginx/PHP/MySQL stack and developing an exercise within a few days, ideally in just one weekend or less.

**What is the purpose of the exercise?**

To evaluate your skills in the following areas:

* Setting up a small PHP project, ideally without major usage of a bundled framework, but instead usage of composer packages when needed. Any framework components can be used separately via Composer as well.
* Setting up a basic MySQL database & database migration query for a single table database, connected to an ORM package, and developing a basic CRUD application, connected to a standard RESTful JSON API
* HTTP connection to 3rd party API using good practices via Guzzle
* Following general PSR standards and good code documentation & commenting

**Exercise Description:**

* Please code a RESTful API to store, retrieve, delete, update phone book items.
* Each phone book item should have at least the following fields:
  + First name (required)
  + Last name
  + Phone number (required) - must be validated based on some standard, e.g. +12 223 444224455
  + Country code - country code should be validated via <https://api.hostaway.com/countries>
  + Timezone name - should be validated via <https://api.hostaway.com/timezones>
  + insertedOn (required) - DateTime type
  + updatedOn (required) - DateTime type
* In every insert or update, a call should be sent to the given API endpoints to get list of countries or timezones for validation, and proper error should be thrown if it’s invalid
* Exceptions should be handled properly, specially upon validation or HTTP call issues
* Different layers of application shall be separated when necessary
* Proper design patterns shall be used when necessary
* Results should be possible to be retrieved by ID, or as total results, or by searching parts of the name

In order to show your skills and get ahead of other candidates, you can work on one or many of the following bonus items:

* **Bonus 1:** you can implement pagination in the RESTful API result, which includes total result, support for number of items per page, and offset
* **Bonus 2:** add OAuth 2 authentication for accessing the CRUD endpoints
* **Bonus 3:** you can also add logging of the errors, and caching of external API calls
* **Bonus 4:** if you have a working code on Docker or Vagrant with proper provisioning script, we appreciate it

**Tip**: we are using the Phalcon framework (https://phalconphp.com/en/), and we love The PHP League products (https://thephpleague.com), so feel free using them in your project :)

**Please submit the result by filling your application here:**

[**https://hostaway.recruitee.com/o/middle-php-developer**](https://hostaway.recruitee.com/o/middle-php-developer)