Back-end developer test

1 Description

Nowadays loans are common thing, helping people achieve their needs earlier or solve an unexpected issue. For loan issuing firstly we need to receive clients application for a loan.

Your task is to create small solution/service for processing clients applications, where you demonstrate your skill in creating web-services, managing data and showing best practices in code quality.

2 Details

Create an API which is able to provide CRUD operations for the client and application.

2.1 API details

REST API should provide following functionality:

2.1.1 Client API details

- add client
- get client
- get paginates list of clients
- update client
- delete client

Create client payload example:

```
{
      "firstName": "John",
      "lastName": "Doe",
      "email": "john.doe@mail.com",
      "phoneNumber": "+37101234567"
}
```

Where input data validation should be applied:

- firstName: *only latin characters, with size 2-32*
- lastName*: only latin characters, with size 2-32*
- email: *valid email*
- phoneNumber*: valid phone number, compatible with E.164*

2.1.2 Application API details

- add application
- get application
- get paginated list of applications
- update application
- delete application

Create application payload example:

```
{
      "clientId": <client_id>,
      "term": 30,
      "amount": 3000.00,
      "currency": "EUR"
}
```

Where input data validation should be applied:

- clientId: *id of the existing client*
- term: *>=10 and <=30*
- amount: *>= 100.00 and <= 5000.00*
- currency: *EUR*

3 Considerations

- PHP 7+ version should be used.
- Code styling and designing best practices. Think about daily code reviews which you will have to pass (and later about the code you will have to review). Do your best.
- Test coverage. Cover as much functionality as you can with tests.
- You can use any framework you familiar with.
- Use any relational database you familiar with.
- Please send your sources and brief description on how to launch/test your app.

4 Extra points

- Use Symfony or Zend framework.
- Use PostgreSQL as database and Doctrine ORM.
- Use Swagger for API documentation.
- Dockerize your application.