We want to create a small PHP backend application (rest endpoints) to manage the users of our system,

# End points

* Create user:   
  POST: admin/users
* Update user  
  PUT: admin/user/{id}
* Delete user  
   DELETE: admin/user/{id}
* List users   
  GET: admin/users{?filter-conditions}
* Get user   
  Get: admin/user/{id}

# Details

* After creating the user, confirmation email should be sent to the user email
* The user will activate the account by clicking on the link in the email
* All the actions of the users should be logged, so we can track the user's actions

# Technical Details

* You should use MVC
* The endpoints response should be valid json document (use [postman](https://www.getpostman.com/) for testing)
* You should use PHP framework / [Symfony](https://symfony.com/) is the preferred framework
* The user data should be stored in database / Using [doctrine](https://www.doctrine-project.org/)
* Emails should be sent in asynchronous way (ex: using [message queue](https://www.rabbitmq.com/) )
* Your application should use [docker](https://www.docker.com/) : [this course may help](https://www.udemy.com/docker-and-kubernetes-the-complete-guide/)
* The application should be well tested by unit tests: [php-unit](https://phpunit.de/)
* The application should be well tested by integration tests: [behat](http://behat.org/)
* GIT should be used
* Use [nginx](https://www.nginx.com/)

# Bounce points

* You use Hexagonal Architecture: [click here](https://fideloper.com/hexagonal-architecture)
* You use [redis](https://redis.io/)  for caching (performance is very important)

# How to proceed

* Create public git repository
* Implement the task in PHP
* Send us the link

# How we review

* The application will be reviewed
* If you fulfill the requirement, we will add some notes on your code if we found any, please check the notes and make the suggested changes and submit your solution again
* Note that everything should be in containers, I should be able to run the application using one command: **docker-compose run**

Php, message queue, database, Redis … are containers, no need to install anything

Ex: [queue](https://hub.docker.com/_/rabbitmq), [mysql](https://hub.docker.com/_/mysql), [redis](https://hub.docker.com/_/redis)

* Write as much documentation as you can in the readme, the ability to write clearly is important factor
* Code quality
* Testing
* If you decided to use other tools than the ones specified here, please tell us why in the readme

Note

* We do not expect that you know all the above technologies, but those are part of technologies that you will work on, and we want to check your ability to solve complex tasks and your ability to learn new tools and solve complex problems
* Please take your time, learn from tools that you do not know what you do really need
* Feel free to ask questions