# Senior/Regular PHP developer

## Test1:

Using one of modern PHP frameworks (sf2, zf2, Laravel, Yii2) you should implement system for API calls (no frontend part but you can add Swagger or similar tool to explore your api). Your JSON-REST full api (not JSON-RPC) should provide CRUD access to two resources without any security check:

- Post (with fields like: title, body and so on)
- Tags (with only name field) many-to-may relation with posts

Additional methods for API:

- select all posts by tag or tags
- count posts by tag or tags

After any post created system should send email to specified in configuration file Email. After any post removed system should log information about it.

## Requirements

- You have to add documentation so everyone will able to install/integrate/use system e.g. README.md
- Totally OOP
- Host your code on Bitbucket or GitHub

## Requirements(only for **seniors**):

- This system should provide way to add tags to posts as part of REST-full api.
- Your system should be optimised for **high** load that means you have to use some cache tools/engines for queries/responses/etc
- Totally TDD with tests and DI (Dependency injection)
- **Bonuses** if you will use:
    - docker/vagrant
    - DI
    - Queues/Jobs for async calls (e.g. RabbitMQ/Gearman/etc)

## Technologies you can use for solve the test:

- PHP 5.4-5.6
- vagrant/docker
- sf2, zf2, Laravel 5, Yii2
- Doctrine 2/AR/Eloquent ORM
- Swagger
- Postgres/Mysql or MongoDB
- Memcache*/Redis/etc
- composer

- PHPUnit with any mock engine (e.g. mockery)