



BY VLAD GREGURCO













DEBUGGING









STEP 1: INSTALL XDEBUG EXTENSION

```
FROM phpdockerio/php72-fpm:latest
WORKDIR "/application"

# Install selected extensions and other stuff
RUN apt-get update \
&& apt-get -y --no-install-recommends install php7.2-mysgl \
php7.2-xdebug \
git \
&& apt-get clean; rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/* /usr/share/doc/*
```

- Add php7.2-xdebug extension in Dockerfile for php container
- Now rebuild your containers with command: docker-compose up -d --build
 --force-recreate
- After that extension is installed but not linked to php.









- Not let's connect xdebug and php. First of all we should fine where xdebug extension
 was placed in container. Enter in php container: docker-compose exec php-fpm
 bash
- And execute next command:

```
root@6e838ede488c:/application# find / -name 'xdebug.so'
/usr/lib/php/20151012/xdebug.so
/usr/lib/php/20131226/xdebug.so
/usr/lib/php/20160303/xdebug.so
/usr/lib/php/20170718/xdebug.so
root@6e838ede488c:/application#
```

- Select the last one, for example in this case /usr/lib/php/20170718/xdebug.so
- Now add to php.ini (docker/php-fpm/php-ini-overrides.ini) next config:

```
[xdebug-config]
zend_extension = /usr/lib/php/20170718/xdebug.so
xdebug.remote_enable = on
xdebug.remote_handler = dbgp
xdebug.remote_mode = req
xdebug.remote_host = 172.17.0.1
xdebug.remote_autostart = 0 ;Set to 1 if you want to debug cli
xdebug.remote_connect_back = 0
xdebug.remote_port = 9000
xdebug.remote_port = 9000
xdebug.max_nesting_level = 250
xdebug.profiler_enable = 0
xdebug.profiler_enable_trigger = 1
xdebug.profiler_output_dir = "/tmp"
xdebug.idekey = "PHPSTORM"
```







STEP 3: PHP CONTAINER ENV



Add next ENV variables to docker-compose.yml file

```
php-fpm:
build: docker/php-fpm
container_name: jobeet-php-fpm
working dir: /application

environment:

- XDEBUG CONFIG=idekey=PHPSTORM
- PHP_IDE_CONFIG=serverName=jobeet

volumes:

- .:/application
- ./docker/php-fpm/php-ini-overrides.ini:/etc/php/7.2/fpm/conf.d/99-overrides.ini

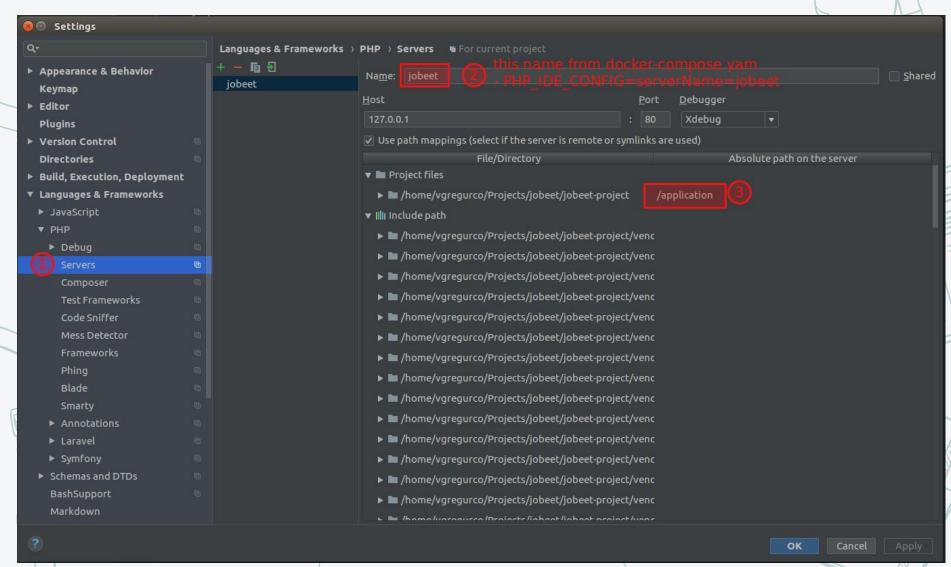
37
```

- Now rebuild your containers with command: docker-compose up -d --build
 - --force-recreate
- Now xdebug should be installed and connected to php. To check it just execute:
 - o docker-compose exec php-fpm bash
 - php -v

```
root@6e838ede488c:/application# php -v
PHP 7.2.5-1+ubuntu16.04.1+deb.sury.org+1 (cli) (built: May 5 2018 04:59:13) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
with Zend OPcache v7.2.5-1+ubuntu16.04.1+deb.sury.org+1, Copyright (c) 1999-2018, by Zend Technologies
with Xdebug v2.6.0, Copyright (c) 2002-2018, by Derick Rethans
```



STEP 5: MAPPING INFORMATION

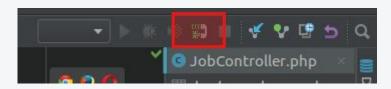




STEP 6: USAGE

- Install xdebug extension in your browser. For chrome:
 https://chrome.google.com/webstore/detail/xdebug-helper/eadndfjplgieldjb
 igjakmdgkmoaaaoc
- Activate it on page of your site
- Put breakpoints

start listen xdebug in PhpStorm







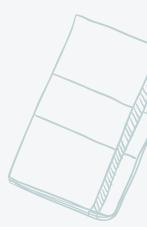




- > https://github.com/gregurco/jobeet/tree/xdebug (code from presentation)
- > https://blackfire.io/
- https://www.getpostman.com/
- https://symfony.com/doc/4.0/components/stopwatch.html
- https://chrome.google.com/webstore/detail/xdebug-helper/eadndfjplgieldjb igjakmdgkmoaaaoc

















TESTING











- Enter in php container: docker-compose exec php-fpm bash
- Run: php bin/phpunit
- This command for the first time will install all dependencies and run tests
- Create your first test in file: /tests/Controller/JobControllerTest.php

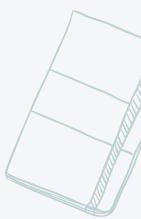
Run this test: **php bin/phpunit**



- > https://github.com/gregurco/jobeet/tree/tests (code from presentation)
- https://symfony.com/doc/4.0/testing.html
- https://symfony.com/doc/4.0/components/dom_crawler.html
- https://github.com/liip/LiipFunctionalTestBundle
- https://github.com/SymfonyTest
- https://knpuniversity.com/screencast/phpunit

















DEPLOYING











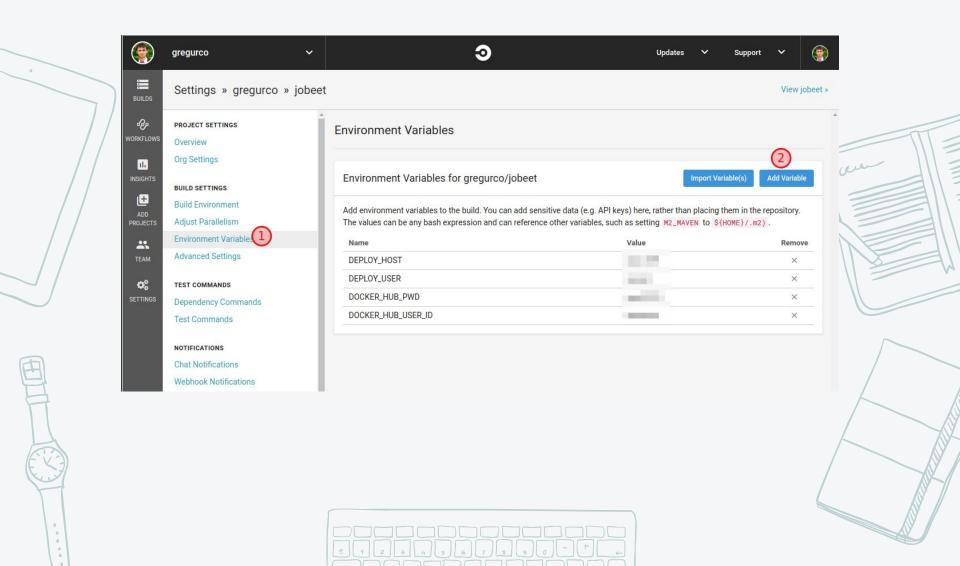
- Create file .circleci/config.yml from example: https://github.com/gregurco/jobeet/blob/circle_ci/.circleci/config.yml
 - All my changes for CI system you can see here (in Files Changed tab):
 https://github.com/gregurco/jobeet/compare/day9...circle_ci
- Create repository on: https://hub.docker.com/
- Create account on: https://circleci.com
- Connect CircleCI with your github account and select there your repository
- Now after each commit build will be triggered
- Setup on the server docker and docker-compose and clone project repository in some folder
- Create on the server ssh key for CircleCI and copy private key to CircleCI. More info here: https://blog.couchbase.com/continuously-deploying-nodejs-application-circleci/



STEP 2: CIRCLECT ENV VARIABLES



• Setup env variables in CircleCI (credentials from DockerHub and Server):





- > https://github.com/gregurco/jobeet/tree/circle_ci (code from presentation)
- https://circleci.com/
- https://travis-ci.org/
- https://www.heroku.com/

























- https://github.com/gregurco/jobeet-tutorial
- https://github.com/gregurco/jobeet
- https://habr.com/hub/php/
- https://knpuniversity.com/
- http://fabien.potencier.org/
- > https://matthiasnoback.nl/
- > [EN] https://leanpub.com/a-year-with-symfony
- > [RU] https://leanpub.com/a-year-with-symfony-ru
- https://www.youtube.com/channel/UCjmwHCZ-qh3ro7hHTQhqYQg

Twitter accounts:

- https://twitter.com/symfony
- > https://twitter.com/seldaek
- https://twitter.com/KnpUniversity
- https://twitter.com/goetas_asmir
- https://twitter.com/matthiasnoback
- https://twitter.com/Ocramius





















