

TESTY INTEGRACYJNE Z UŻYCIEM ORG.TESTCONTAINERS

Autor [Tomasz Wyszomirski](#)

CO TO SĄ TESTY INTEGRACYJNE?

"Integration tests determine if independently developed units of software work correctly when they are connected to each other"

Martin Fowler

CO TO JEST DOCKER?

"Docker to taki Vagrant tylko że lepszy"

Mądrość ludowa

GO TO JEST.ORG.TESTCONTAINERS?

"Testcontainers is a Java library that supports JUnit tests, providing lightweight, throwaway instances of common databases, Selenium web browsers, or anything else that can run in a Docker container."

org.testcontainers team



**DLACZEGO TESTY INTEGRACYJE SĄ
POTRZEBNE?**

CO JESZCZE MOGĘ ZROBIĆ?

Właściwie wszystko na co pozwala natywna konsola
dockera
i więcej ...

OCZEKIWANIE NA START KONTENERA

Domyślnie jeżeli wystawiacie jakiś port to kontener
zaczeka aż ten port będzie "up"

Albo coś bardziej zaawansowanego

```
.waitFor(  
    Wait.forHttp("/all")  
        .forStatusCode(200)  
        .forStatusCode(401)  
        .usingTls());
```

Albo

```
.waitFor(Wait.forHealthcheck());
```


ŚLEDZENIE WYJŚCIA KONTENERA

To robiłem na demonstracji

```
Slf4jLogConsumer logConsumer = new Slf4jLogConsumer(LOGGER);  
container.followOutput(logConsumer);
```

Niekoniecznie musi być STDOUT

```
container.followOutput(toStringConsumer, OutputType.STDERR);
```

Można reagować na wyjście

```
consumer.waitUntil(frame ->  
    frame.getUtf8String().contains("STARTED"), 30, TimeUnit.SECONDS)
```

TRYBY UŻYCIA

- Pojedyncze kontenery (tak jak ja to robię)
- Docker Compose (odpala się cały plik compose)
- Dockerfile (ciekawa opcja, pozwala dynamicznie tworzyć obrazy)

DEDYKOWANE KONTENERY

Cassandra

Couchbase

Influxdb

Oracle-XE

Kafka

Spock

ElasticSearch

SqlServer

CIEKAWY KONTENERY

MockServer

<http://www.mock-server.com/>

Jeżeli komponenty rozmawiają po HTTP pozwala fajnie i wygodnie mockować interakcje

Selenium

<https://www.seleniumhq.org/>

Gotowy kontener z zainstalowanymi webdriverami i selenium, fajny ficzer: robi screeny i nagrywa filmiki

DZIĘKUJE

https://github.com/twyszomirski/lightning_talks