

Hello, World!

Beginning Spring 5
Chapter 2

Agenda:

1. A Simple „Hello World!” Application
2. Building
3. Testing
4. A simple application with Spring

A simple application

- Spring Framework
- Gradle
- TestNG

Kolejny „Hello World”?

- Skupimy się na:
 - Konfiguracji
 - Narzędziach
 - Składni
 - Springu

Suffering-Oriented Programming

- Nathan Marz - “First make it possible. Then make it beautiful. Then make it fast.”
- Działający
- Zrozumiały
- szybki

Building

Installing gradle
Building the project

- Instalacja gradle w zależności od systemu
- Budowanie projektu

Sposoby instalacji

- Linux

```
sdk install gradle 5.5.1
```

- MacOS

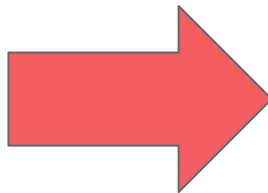
```
brew install gradle
```

- Windows

```
scoop install gradle
```

Tworzenie gradle wrappera

```
> gradle wrapper  
> Task :wrapper  
  
BUILD SUCCESSFUL in 0s  
1 actionable task: 1 executed
```



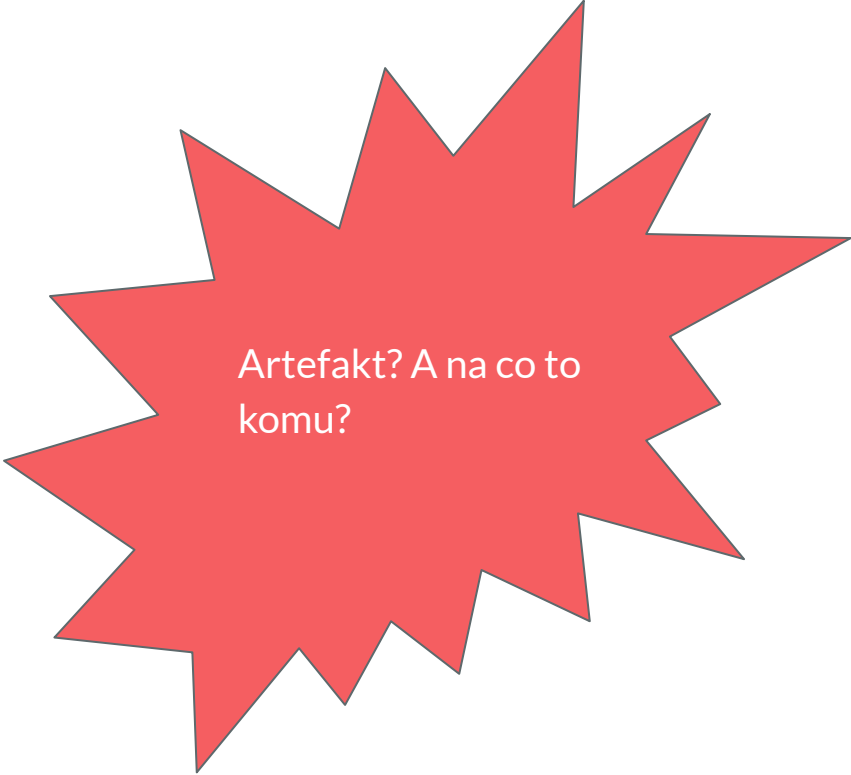
gradlew

gradlew.bat

gradle/

Budowanie projektu

- system oparty na wtyczkach
- budowanie, testowanie i łączenie artefaktów Java



Artefakt? A na co to
komu?

Budowanie projektu

- sekcja wtyczki
- sourceCompatibility i targetCompatibility
- sekcja ext
- Sekcja oznaczona allprojects

```
apply plugin: 'java'
sourceCompatibility = 1.11
targetCompatibility = 1.11

ext {
    springFrameworkVersion = "5.1.5.RELEASE"
    jacksonVersion = "2.9.9"
    testNgVersion = "6.14.3"
}

allprojects {
    apply plugin: 'java'

    repositories {
        jcenter()
        mavenCentral()
    }

    dependencies {
        testImplementation "org.testng:testng:$testNgVersion"
    }

    test {
        useTestNG()
    }
}
```

Budowanie projektu

2

```
chapter9 z common, test,  
jpa i mongodb,
```



```
chapter9common,  
chapter9test
```

3

```
dependencies {  
    compile "org.springframework:spring-core:$springFrameworkVersion"  
    compile "org.springframework:spring-context:$springFrameworkVersion"  
    compile "org.springframework:spring-test:$springFrameworkVersion"  
}
```

1

```
rootProject.name = 'bsg5'  
  
// add inclusions as the chapter sources are followed  
include 'chapter2'  
include 'chapter3'  
include 'chapter4'  
include 'chapter5'  
include 'chapter5common'  
include 'chapter5anno'  
include 'chapter5xml'  
include 'chapter6'  
include 'chapter7'  
include 'chapter8'  
include 'chapter9common'  
include 'chapter9test'  
include 'chapter9jpa'  
include 'chapter9mongodb'  
include 'chapter10'  
include 'chapter10custom'  
include 'chapter10jpa'
```

Budowanie projektu

```
% gradle
Parallel execution is an incubating feature.

> Task :help

Welcome to Gradle 4.6.

To run a build, run gradle <task> ...

To see a list of available tasks, run gradle tasks

To see a list of command-line options, run gradle --help

To see more detail about a task, run gradle help --task <task>

For troubleshooting, visit https://help.gradle.org

BUILD SUCCESSFUL in 0s
1 actionable task: 1 executed
```



Nie ma jeszcze kodu, a już działa!

Testing



Dodanie zależności testu do projektu

```
apply plugin: 'java'
sourceCompatibility = 1.11
targetCompatibility = 1.11

ext {
    springFrameworkVersion = "5.1.5.RELEASE"
    jacksonVersion = "2.9.9"
    testNgVersion = "6.14.3"
}

allprojects {
    apply plugin: 'java'

    repositories {
        jcenter()
        mavenCentral()
    }

    dependencies {
        testImplementation "org.testng:testng:$testNgVersion"
    }

    test {
        useTestNG()
    }
}
```

Listing 2-6. chapter2/src/test/java/com/bsg5/chapter2/GreeterTest.java

```
package com.bsg5.chapter2;

import org.testng.annotations.Test;
import java.io.ByteArrayOutputStream;

import java.io.PrintStream;
import java.io.UnsupportedEncodingException;
import java.nio.charset.StandardCharsets;

import static org.testng.Assert.assertEquals;

public class GreeterTest {
    @Test
    public void testHelloWorld() {
        Greeter greeter = new HelloWorldGreeter();
        final ByteArrayOutputStream baos = new ByteArrayOutputStream();
        try (PrintStream ps = new PrintStream(baos, true, "UTF-8")) {
            greeter.setPrintStream(ps);
            greeter.greet();
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
        String data = new String(baos.toByteArray(), StandardCharsets.UTF_8);
        assertEquals(data, "Hello, World!");
    }
}
```

Listing 2-8. GreeterTest failure

```
gradle test
Parallel execution is an incubating feature.

> Task :chapter2:test FAILED

Gradle suite > Gradle test > com.bsg5.chapter2.GreeterTest.testHelloWorld
FAILED
    java.lang.AssertionError at GreeterTest.java:26

1 test completed, 1 failed

FAILURE: Build failed with an exception.

* What went wrong:
Execution failed for task ':chapter2:test'.
> There were failing tests. See the report at:
   file:///.../chapter2/build/reports/tests/test/index.html

* Try:
Run with --stacktrace option to get the stack trace. Run with --info or
--debug
option to get more log output. Run with --scan to get full insights.

* Get more help at https://help.gradle.org

BUILD FAILED in 1s
4 actionable tasks: 2 executed, 2 up-to-date
```

A Simple Application with Spring

- Do tej pory klasa Greeter, z metodą greet() bez implementacji
- Konfiguracja beanów - XML.
- Wzorzec Dependency Injection

Implementacja testu bez Springa

```
public void testHelloWorld() {  
    Greeter greeter = new HelloWorldGreeter();  
    final ByteArrayOutputStream baos = new ByteArrayOutputStream();  
    try (PrintStream ps = new PrintStream(baos, true, "UTF-8")) {  
        greeter.setPrintStream(ps);  
        greeter.greet();  
    } catch (UnsupportedEncodingException e) {  
        e.printStackTrace();  
    }  
    String data = new String(baos.toByteArray(), StandardCharsets.UTF_8);  
    assertEquals(data, "Hello, World!");  
}
```

Konfiguracja beanów - XML

```
<bean id="helloGreeter" class="com.bsg5.chapter2.HelloWorldGreeter">  
  <property name="printStream" ref="printStream" />  
</bean>
```

```
<bean id="printStream" class="java.io.PrintStream">  
  <constructor-arg ref="baos"/>  
  <constructor-arg value="true"/>  
  <constructor-arg value="UTF-8"/>  
</bean>
```

```
<bean id="baos" class="java.io.ByteArrayOutputStream" />
```

Refaktoryzacja testu korzystając z beanów

```
@Test
public void testHelloWorld() {
    ApplicationContext context = new ClassPathXmlApplicationContext
        ("/applicationContext.xml");
    Greeter greeter = context.getBean("helloGreeter", Greeter.class);
    ByteArrayOutputStream baos = context.getBean("baos", ByteArray
        OutputStream.class);

    greeter.greet();
    String data = new String(baos.toByteArray(), StandardCharsets.UTF_8);
    assertEquals(data, "Hello, World!");
}
```

Rozszerzenie klasy testowej

Aby klasa testowa poprawnie ładowała kontekst:

- Powinna rozszerzać klasę **AbstractTestNGSpringContextTests**

Jeśli korzysta się z frameworków:

- JUnit 4 - powinna rozszerzać **AbstractJUnit4SpringContextTests**
- JUnit 5 - dodanie adnotacji **@ExtendWith(SpringExtension.class)**

Rezultat

Przy implementacji wyświetlenia w ciele metody greet() "Hello, World!"

```
$ gradle chapter2:test  
BUILD SUCCESSFUL in 7s  
4 actionable tasks: 3 executed, 1 up-to-date
```

Dziękujemy za uwagę!

Bartosz Belski
Piotr Gucwa
Miłosz Momot
Aleksander Kowalczyk

Sebastian Obora
Joanna Gajzler
Aleksandra Duda