



Zachodniopomorski  
Uniwersytet  
Technologiczny  
w Szczecinie



Wydział  
Informatyki



KATEDRA INŻYNIERII OPROGRAMOWANIA

<http://kio.wi.zut.edu.pl/>

*INŻYNIERIA OPROGRAMOWANIA*



# Zaawansowane Techniki Programowania Java

## #07 : JAX-WS (*[javax.xml.ws](http://javax.xml.ws)*)

*Prowadzący:*

*Krzysztof Kraska*

*email: [kkraska@wi.zut.edu.pl](mailto:kkraska@wi.zut.edu.pl)*

---

Szczecin, 15 maja 2018 r.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Objectives

After completing this lesson, you should be able to:

- Describe the basic structure of a Simple Object Access Protocol (SOAP) message and how it is encapsulated by transport mechanisms
- Explain how WSDL defines a web service, including its message representation and transport mechanism
- Create a JAX-WS web service
- Create a JAX-WS client by using Java SE
- Create a JAX-WS client by using Java EE



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 15 maja 2018 r.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Reasons for Using SOAP

The SOAP web services specification defines an interoperable, platform-independent means for component interaction. The SOAP web service requirements include:

- Decoupling message representation from transport mechanisms
- Supporting extensible frameworks

ORACLE

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 15 maja 2018 r.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Extensible Message Representation Simple Object Access Protocol



ORACLE

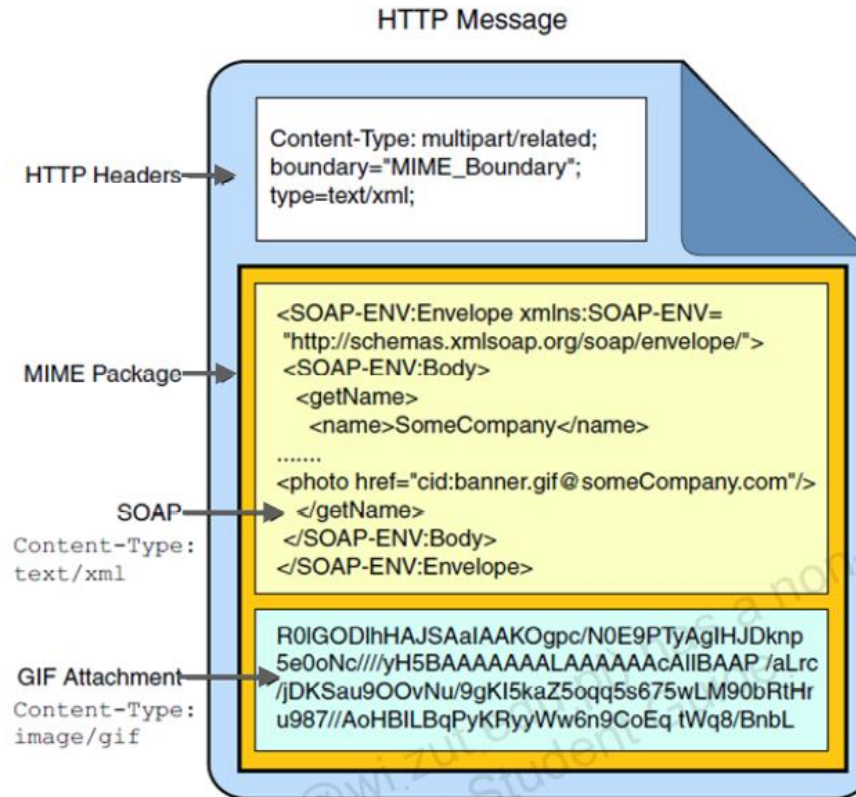
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 15 maja 2018 r.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## SOAP over HTTP



ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

HTTP is not the only transport supported by SOAP, but it is the most common.

Szczecin, 15 maja 2018 r.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Raw SOAP/HTTP Request

```
Accept: text/html, image/gif, image/jpeg, */*; q=.2
Connection: Keep-Alive
Content-Length: 206
Content-Type: text/xml; charset=utf-8
Host: localhost:7001
SOAPAction: ""
User-Agent: Oracle JAX-RPC 1.1
<env:Envelope
  xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <env:Body>
    <getHello xmlns="http://ou/">
      <arg0 xmlns="">matt</arg0>
    </getHello>
  </env:Body>
</env:Envelope>
```

ORACLE

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 15 maja 2018 r.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Raw SOAP/HTTP Response

```
HTTP/1.1 200 OK
Content-type: text/xml; charset=utf-8

<?xml version='1.0' encoding='UTF-8'?>
<S:Envelope
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getHelloResponse xmlns:ns2="http://ou/">
      <return>Hello matt!</return>
    </ns2:getHelloResponse>
  </S:Body>
</S:Envelope>
```

ORACLE

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 15 maja 2018 r.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## WSDL

Web Services Description Language (WSDL) is an XML-based interface definition language that is used for describing a web service.

- Provides an interface that exposes web services
- Maps web services to underlying programs and software systems
- Describes how to connect to and use a web service

ORACLE

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 15 maja 2018 r.



# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Structure of a WSDL File

**<definitions>: Root WSDL Element**

**<types>: What data types will be transmitted?**

**<message>: What exact information is expected?**

**<portType>: What operations (functions) will be supported?**

**<binding>: How will the messages be transmitted on the wire?  
What SOAP-specific details are there?**

**<service>: Define the collection of ports that make up the service and where  
is the service located?**

**ORACLE®**

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 15 maja 2018 r.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

Note that there are differences in the structure of a WSDL file between version 1.1 and 2.0 of the WSDL specification:

WSDL 1.1	WSDL 2.0
Service	Service
Port	Endpoint
Binding	Binding
Porttype	Interface
Operation	Operation
Message	n/a
Types	Types

Messages were removed in WSDL 2.0. In 1.1, the message defined the typing information of the binding. In WSDL 2.0, the schema handles all typing.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## WSDL: Sample Structure

```
<?xml version='1.0' encoding='UTF-8'?>
<definitions xmlns:wsu="http://docs.oasis-
  open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
  1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy"
  xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://ou/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://ou/" name="HelloService">
  <types><!-- ... --></types>
  <message><!-- ... --></message>
  <portType><!-- ... --></portType>
  <binding><!-- ... --></binding>
  <service><!-- ... --></service>
</definitions>
```

The WSDL file for a SOAP service can be downloaded by making an HTTP GET request to the URL of the web service with a `wSDL` query parameter, for example:

<http://localhost:7001/HelloWS/HelloService?wSDL>

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## JAX-WS

Java API for XML Web Services (JAX-WS) is a technology for building web services and clients that communicate by using XML.

- Web service operations are transmitted as SOAP messages (XML files) over HTTP.
- The JAX-WS API hides SOAP complexity from the application developer.
- On the server side, web service operations are defined by using annotations and POJOs.
- Client programs are easy to code.
- With JAX-WS, clients and web services are platform-independent.
  - A JAX-WS client can access a web service that is not running on the Java Platform and vice versa.

ORACLE

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The JAX-WS API is a Java API for creating SOAP-based web services.

Szczecin, 15 maja 2018 r.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## JAX-WS Development Approaches

- WSDL First
  - Generate web service artifacts by using the information contained in a WSDL file.
- Code First
  - Create a service endpoint interface (SEI) or value classes as Java source files, and then use them as inputs to generate the associated WSDL descriptor and other portable artifacts.

ORACLE

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 15 maja 2018 r.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Benefits of a Code First Approach

A code first approach to providing a web service interface to an existing enterprise application offers several advantages:

- You can use the quickest development path.
- It is a natural approach, especially when business logic has already been implemented.
- You can map existing domain models directly to WSDL with little effort.

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 15 maja 2018 r.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Web Service Creation Process

To create a JAX-WS web service, perform the following steps:

1. Create a web or an enterprise application with NetBeans.
2. Create a class to represent your service.
3. Annotate the class with `@WebService`.
4. Build and deploy your project.
  - A WSDL file is automatically generated for you.
5. Create a web service client by referring to the WSDL URL.

ORACLE

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 15 maja 2018 r.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Simple Greeter Web Service

```
@WebService
public class Greeter {
    public final String worldGreeting = "Hello World!";

    public Greeter() { }

    @WebMethod
    public String greetWorld() {
        return worldGreeting;
    }
}
```

Must be annotated with @WebService

Must have a no-arg constructor

Use this annotation to define a web service method.

ORACLE

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 15 maja 2018 r.



# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

The JAX-WS requirements on the Java class are as follows:

- Must be annotated with `javax.jws.WebService`
- Must not be declared `final`
- Must not be `abstract`
- Must have a default `public` no argument constructor
- Must not have a `finalize()` method

**Note:** The annotations, `@WebService` and `@WebMethod`, are actually part of another specification, JSR 181, Web Services Metadata.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Testing a Web Service

- Using the WebLogic Server, the URL to test a web service has the following format:  
`http://localhost:7001/appName/webService?Tester`
  - Operations with complex types are not supported.
- You can use any HTTP tool that can submit a POST request with a custom body:
  - Firefox RESTClient
  - cURL
- You can use JDeveloper's HTTP Analyzer to test a web service.
- You can use SoapUI for a dedicated SOAP web service testing application.

ORACLE

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 15 maja 2018 r.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Web Service URLs

The default URL for a web service is based on the class name of the service.

- POJO endpoints: The default URL is `http://host:port/{app}/{classname}Service`.
  - Use of `@WebService(serviceName="mypath")` results in: `http://host:port/{app}/mypath`.
- EJB endpoints: The default URL is `http://host:port/{classname}/{classname}Service`.
  - Use of `@WebService(serviceName="mypath")` results in: `http://host:port/{classname}/mypath`.

ORACLE

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## JAX-WS Requirements for Web Service Methods

- Must be `public`
  - By default, every public method in the class will be a part of the web service.
- Must not be `static` or `final`
- Must have JAXB-compatible parameters and return types
  - Parameters and return types must not implement the `java.rmi.Remote` interface.

ORACLE

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 15 maja 2018 r.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Generating the WSDL

To publish an SEI, there must be a WSDL per service and JAXB classes for each SOAP message.

- JAX-WS can deliver WSDL and message classes dynamically. The WSDL is available at:

<http://host:port/path/to/service?WSDL>.

Obtaining the WSDL from a running service is preferred.

However, if you need to generate a WSDL without running the SEI:

- Use the annotation processor in `jaxws-tools.jar`.

```
javac -processor  
com.sun.tools.ws.processor.modeler.annotation.WebServiceApp
```

ORACLE

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 15 maja 2018 r.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Creating a JAX-WS Client

- The JAX-WS Reference Implementation (RI) is included in Java SE.
- The basic SOAP web services can be accessed by a Java SE client.
- Use NetBeans to generate client artifacts.
  - All you need is the web service's WSDL.
    - Right-click the package and select **New > Web Service from WSDL**.
  - Optionally use `wsimport` (similar to `xjc`) to generate client artifacts from the command line.

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 15 maja 2018 r.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Classes Produced from WSDL

Running `wsimport` or using the wizard in NetBeans on the Greeter WSDL (one port with one operation) produces six `.java` files.

- `GreetWorld.java`: The JAXB-annotated class that is used to marshall and unmarshal the SOAP request body
- `GreetWorldResponse.java`: The JAXB-annotated class that is used to marshall and unmarshal the SOAP response body
- `Greeter.java`: The JAX-WS port class
- `GreeterService.java`: The JAX-WS service class
- `ObjectFactory.java`: The JAXB factory
- `package-info.java`: The JAXB package-level annotations

ORACLE

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Local WSDL Files

When configuring a web service client reference in NetBeans, two local copies of the WSDL file are made.

- `MyProject/xml-resources/`
  - You must switch to the Files tab to inspect this directory.
  - This copy is used when running `wsimport` every time you build your project.
- `MyProject/src/META-INF/wsdl/`
  - This is packaged with the application so that a copy of the WSDL need not be downloaded when initializing the client.
  - Clients can improve performance by using this copy, but by default, it is not used.

ORACLE

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 15 maja 2018 r.



# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Calling a Web Service

After the JAX-WS artifacts are generated, calling a web service is fairly easy.

```
public class Main {  
    public static void main(String[] args) {  
        GreeterService service = new GreeterService();  
        Greeter port = service.getGreeterPort();  
        System.out.println("The Message is: " +  
                           port.greetWorld());  
    }  
}
```

The Service subclass  
that loads the WSDL.  
This is a port factory.

The port with  
methods that match  
the operations

Call the web service.

ORACLE

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 15 maja 2018 r.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Java EE Web Service Clients

When creating an EE client, you probably will perform the same steps every time. If your client is a managed component that supports injection of resources, you can obtain the service or port via injection.

```
@WebServiceRef(wsdlLocation = "WEB-INF/wsdl/localhost_7001/SOAPWS/GreeterService.wsdl")
private GreeterService service;
```

Optional  
performance-  
enhancing attribute

```
@WebServiceRef(wsdlLocation = "WEB-INF/wsdl/localhost_7001/SOAPWS/GreeterService.wsdl")
private Greeter port;
```

ORACLE

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 15 maja 2018 r.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Custom WSDL Method Name

Web services require unique names for each method. Use the `@WebService` annotation to specify a different service name:

```
@WebMethod(operationName="GreetFirst")
public String greetName(String name){
    return englishGreeting + name;
}

@WebMethod(operationName="GreetFull")
public String greetName(String first, String last){
    return englishGreeting + first + " " + last;
}
```

ORACLE

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 15 maja 2018 r.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Custom WSDL Operation Parameter Names

The default names of method parameters are `arg0`, `arg1`, and so on:

```
<xs:complexType name="GreetFirst">
  <xs:sequence>
    <xs:element name="arg0" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

To modify the element names used for method parameters, use the `@WebParam` annotation.

```
@WebMethod(operationName="GreetFirst")
public String greetName(
    @WebParam(name="firstName") String name) {
    return englishGreeting + name;
}
```

ORACLE

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 15 maja 2018 r.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Viewing SOAP Messages

Sometimes, it helps to dump all the SOAP messages exchanged by the server to the standard output.

- JAX-WS includes a generic framework, JAX-WS Handlers, which can be used to dump the messages.
- Use a system property to dump all JAX-WS messages.  
`-Dcom.sun.xml.ws.transport.http.HttpAdapter.dump=true`

ORACLE

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 15 maja 2018 r.

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Multithreading the Endpoint Publisher

In production mode, the Endpoint publisher would need to handle concurrent requests so that several pending requests could be processed at the same time<sup>\*)</sup>.

```
package ch01.ts;

import javax.xml.ws.WebMethod;
import javax.xml.ws.WebService;
import javax.xml.ws.soap.SOAPBinding;
import javax.xml.ws.soap.SOAPBinding.Style;

@WebService
@SOAPBinding(style = Style.RPC)
public interface TimeServer {
    @WebMethod
    String getTimeAsString();

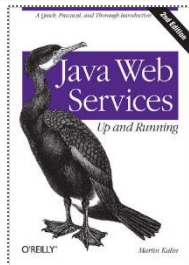
    @WebMethod
    long getTimeAsElapsed();
}
```

```
package ch01.ts;

import java.util.Date;
import javax.xml.ws.WebService;

@WebService(endpointInterface="ch01.ts.TimeServer")
public class TimeServerImpl implements TimeServer {
    @Override
    public String getTimeAsString() {
        return new Date().toString();
    }

    @Override
    public long getTimeAsElapsed() {
        return new Date().getTime();
    }
}
```



<sup>\*)</sup> Java Web Services: Up and Running by Martin Kalin

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Multithreading the Endpoint Publisher

---

```
package ch01.ts;

import javax.xml.ws.Endpoint;

class TimePublisherMT { // MT for multithreaded
    private Endpoint endpoint;

    public static void main(String[ ] args) {
        TimePublisherMT self = new TimePublisherMT();
        self.create_endpoint();
        self.configure_endpoint();
        self.publish();
    }
    private void create_endpoint() {
        endpoint = Endpoint.create(new TimeServerImpl());
    }
    private void configure_endpoint() {
        endpoint.setExecutor(new MyThreadPool());
    }
    private void publish() {
        int port = 8888;
        String url = "http://localhost:" + port + "/ts";
        endpoint.publish(url);
        System.out.println("Publishing TimeServer on port " + port);
    }
}
```

# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Multithreading the Endpoint Publisher

---

```
package ch01.ts;

import java.util.concurrent.LinkedBlockingQueue;
import java.util.concurrent.ThreadPoolExecutor;
import java.util.concurrent.TimeUnit;
import java.util.concurrent.locks.Condition;
import java.util.concurrent.locks.ReentrantLock;

public class MyThreadPool extends ThreadPoolExecutor {
    private static final int pool_size = 10;
    private boolean is_paused;
    private ReentrantLock pause_lock = new ReentrantLock();
    private Condition unpaused = pause_lock.newCondition();

    public MyThreadPool(){
        super(pool_size,           // core pool size
              pool_size,           // maximum pool size
              0L,                   // keep-alive time for idle thread
              TimeUnit.SECONDS,     // time unit for keep-alive setting
              new LinkedBlockingQueue<Runnable>(pool_size)); // work queue
    }
    ...
}
```



# JAVA API FOR XML (SOAP) WEB SERVICES

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •

## Multithreading the Endpoint Publisher

---

```
// some overrides
@Override
protected void beforeExecute(Thread t, Runnable r) {
    super.beforeExecute(t, r);
    pause_lock.lock();
    try {
        while (is_paused) unpaused.await();
    }
    catch (InterruptedException e) { t.interrupt(); }
    finally { pause_lock.unlock(); }
}

public void pause() {
    pause_lock.lock();
    try { is_paused = true; }
    finally { pause_lock.unlock(); }
}

public void resume() {
    pause_lock.lock();
    try {
        is_paused = false;
        unpaused.signalAll();
    }
    finally { pause_lock.unlock(); }
}
}
```

---

Szczecin, 15 maja 2018 r.

# KONIEC

• ZAAWANSOWANE TECHNIKI PROGRAMOWANIA JAVA •



**DZIĘKUJĘ  
ZA UWAGĘ!!!**

---

Szczecin, 15 maja 2018 r.