Cel szkolenia:

This course shows experienced Java programmers how to build RESTful web services using the Java API for RESTful Web Services, or JAX-RS. We first overview the key concepts of REST -- ultimately the thorough and thoughtful use of URLs, HTTP methods, and media types to design and implement scalable and maintainable enterprise services. Then we dive into the elegant JAX-RS standard for building RESTful services, learning how to manage URLs and URL patterns and methods, how to bind input and control response production, and how to manage HTTP entities in popular content types such as XML and JSON.

From here students investigate intermediate features including dependency injection, error handling, and JSR-303 validation, and use Java generics to implement patterns for common operations over an application’s domain classes. We explore sub-resources, the JAX-RS client API, filters and interceptors, and testing techniques, before closing with a summary chapter on REST security that includes implementations of HTTP BASIC security and HMAC signatures.

Learning Objectives

- Understand the advantages of the REST architecture for web services.
- Use JAX-RS to develop simple RESTful services.
- Control dispatching to service methods based on URL patterns and HTTP methods.
- Bind request values to method parameters when expressed as HTTP query parameters, form values, headers, cookies, and more.
- Manage XML and JSON content using XML Schema and JAX -- or without JAXB using leading JAX-RS providers and Reflection-driven entity providers such as MOXy and Jackson.
- Handle error conditions by producing appropriate HTTP responses.
- Use JSR-303 validation for request parameters, headers, and entities.
- Use Java generics to implement REST API patterns for various domain classes.
- Take advantage of lifecycle and context services available to JAX-RS services.
- Organize request-handling methods into sub-resource classes to make REST APIs extensible and maintainable.
- Implement REST clients using the JAX-RS standard API.
- Build filters and interceptors to adapt service endpoint behavior.
- Develop unit tests for JAX-RS services that cover both method code and JAX-RS annotations, using the Jersey test framework.
- Be aware of security concerns for RESTful services and secure services appropriately.

Plan szkolenia:

- Overview of REST and JAX-RS
  - The REST Vision
  - Use of HTTP
  - Use of URIs
  - Use of Content Types
  - CRUD Operations and Business Operations
  - HATEOAS and the Richardson Maturity Model
  - JAX-RS
  - Applications, Resources, and Providers

- Configuration and Lifecycle
  - The JAX-RS Application
  - XML Configuration
  - Annotation-Driven Configuration
  - Applications
  - Root Resource Classes
  - Per-Request vs. Singleton Lifecycle
  - Providers

- Handling Requests
  - The Application Path
  - The @Path Annotation
  - The HTTP Method Annotations
  - Sub-Resource Locators
  - Annotation Inheritance and overriding
@XXXParam Annotations
The @DefaultValue Annotation
Parameter Types
Parameter Converters

Producing Responses
Supported Return Types
The Response Class
Response Entities
Binary Content
Delivering a File

Entity Translation
Entity Parameter and Return Type
Entity Providers
@Consumes and @Produces Annotations
Built-In Entity Providers
Custom Entity Providers

Working with XML and JSON
The JAXB Entity Provider
Driving XML Representations from Schema
Driving JSON Representations with JAXB
JSON Without JAXB: Jackson, MOXy, etc.
CRUD Patterns
Error Handling
Sub-Resources

Dependency Injection
The @Context Annotation
Injectable Types
The Application Subclass
Servlet Configuration and Context
Impact of Lifecycle Policies
Context Providers
Using CDI

Validation and Error Handling
Using Response
Throwing WebApplicationException
Exception Mapping Providers
Selection of Exception Mappers
Java EE Bean Validation
Constraint Annotations
Support for JSR-303
Annotating Method Parameters
Annotating Entity Classes
Error Reporting

Generic Services
Generic Entities
Generic Entity Providers
ParameterizedType
Reflection-Driven Entity Providers
Annotation Inheritance
CRUD Patterns, II
Serialization, Recursion, and Scope
Dynamic Sub-Resources

Working with Databases
Persistence Services
The Java Persistence API
JPA Support for JSR-303
Handling IDs and Keys
Cascading
Caching
Error Handling
Hypermedia Challenges

Sub-Resources
Significance of Sub-Resources
Exposing Sub-Objects
Exposing Collections
Multiple Paths to Resources
Exposing Actions
Using Sub-Resource Locators
Collection vs. Instance Services

The Client API
The Builder Pattern
Client
○ WebTarget
○ Invocation
○ Basic Usage
○ Managing Content Types and Entities
○ Error Handling
○ Registering Providers
○ The Service Locator Pattern
○ Generic Clients

○ Filters and Interceptors
  ○ The Filter Interfaces
  ○ Processing Pattern
  ○ The Request and Response Context Interfaces
  ○ Aborting a Request
  ○ The Interceptor Interfaces
  ○ Adaptive Streams
  ○ Filters on the Client Side
  ○ Interceptor Strategy for Hypermedia

○ Testing
  ○ Testing JAX-RS Services
  ○ Unit Testing and Integration Testing
  ○ Mocking the Container
  ○ The Jersey Test Framework
  ○ Test Configuration
  ○ Mocking Dependencies
  ○ Testing JAX-RS Clients
  ○ Mocking Services

○ Security
  ○ Concerns for RESTful Services
  ○ Authentication and Authorization
  ○ HTTP BASIC and DIGEST
  ○ HTTPS
  ○ Programmatic Security
  ○ SQL Injection
  ○ Cross-Site Request Forgery
  ○ Message-Level Security
  ○ HMACs
Wymagania:

- Strong Java programming skills are essential -- Course 103 Java Programming is excellent preparation.
- Experience with other Java EE standards, especially servlets and JSP, will be very helpful in class, but is not strictly required.

Poziom trudności

|    |    |    |    |

Certyfikaty:

The participants will obtain certificates signed by Capstone Courseware.

Prowadzący:

Authorized Capstone Courseware Trainer.