

Red Hat Application Development II: Implementing Microservice Architectures

Code: JB283R
URL: [View Online](#)

Building on Red Hat Application Development I: Programming in Java EE (JB183), the introductory course for Java EE application development, Red Hat Application Development II: Implementing Microservice Architectures (JB283) emphasizes learning architectural principles and implementing microservices in Java EE, primarily based on MicroProfile with WildFly Swarm and OpenShift.

Skills Gained

- Deploy and monitor microservice-based applications.
- Implement a microservice with MicroProfile.
- Implement unit and integration tests for microservices.
- Use the config specification to inject data into a microservice.
- Create a health check for a microservice.
- Implement fault tolerance in a microservice.
- Secure a microservice using the JSON Web Token (JWT) specification.

Who Can Benefit

- This course is designed for Java developers.

Prerequisites

- Attend Red Hat Application Development I: Programming in Java EE (JB183) or demonstrate equivalent experience
- Be proficient in using an integrated development environment such as Red Hat® Developer Studio or Eclipse
- Experience with Maven is recommended, but not required

Course Details

Describe microservice architectures

- Describe components and patterns of microservice-based application architectures.

Deploy microservice-based applications

- Deploy portions of the course case study applications on an OpenShift cluster.

Implement a microservice with MicroProfile

- Describe the specifications in MicroProfile, implement a microservice with some of the specifications, and deploy it to an OpenShift cluster.

Test microservices

- Implement unit and integration tests for microservices.

Inject configuration data into a microservice

- Inject configuration data from an external source into a microservice.

Create application health checks

- Create a health check for a microservice.

Implement fault tolerance

- Implement fault tolerance in a microservice architecture.

Develop an API gateway

- Describe the API gateway pattern and develop an API gateway for a series of microservices.

Secure microservices with JWT

- Secure a microservice using the JSON Web Token specification.

Monitor microservices

- Monitor the operation of a microservice using metrics, distributed tracing, and log aggregation.