Open Liberty

Masterclass: Hands-on Liberty for Developers

Ready to get your hands dirty?

Online exercises: https://ibm.biz/OpenLibertyMasterclass

The Masterclass is Open Sourced at:
https://github.com/OpenLiberty/open-liberty-masterclass

Follow the instructions in the ”Before you begin” section of the README.md to:

- Install Pre-requisites
- Prime Maven and Docker Caches

Mastersclass hands-on content

- The Application
- Module 1: Build
- Module 2: Dev Mode
- Module 3: Application APIs
- Module 4: Server Configuration
- Module 5: Externalizing Configuration
- Module 6: Integration Testing
- Module 7: Docker
  - Overriding Dev Server Configuration
- Module 8: Testing in Containers
- Module 9: Support Licensing
- Conclusion

Brief History

Open Liberty

Liberty Goals

Open Liberty

- Efficient
- Simple to use
- Consistency
- Just enough runtime
- End of migration
- Agile-ready

Liberty

- First shipped in WAS 8.5 in 2012
  - Servlet + JSP + JPA
- Web Profile 6 in 2014
- Java EE 7 in 2016 – first commercial product to certify
- Java EE 8 in 2018 – first to certify
- Jakarta EE 8 in 2019 – first to certify
- Eclipse MicroProfile – first to deliver 1.0-1.4, 2.0-2.1, 3.0

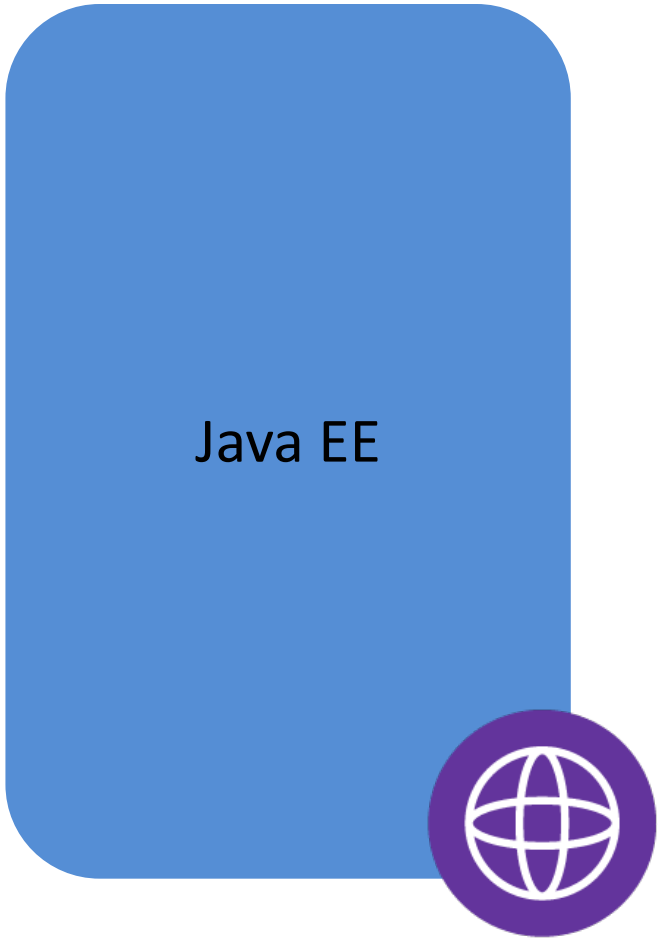

Docs
Get Started
Support
Fork the code
openliberty.io
Launched September 2017
Open Liberty
An IBM Open Source Project
A lightweight open framework for building fast and efficient cloud-native Java microservices.
Build cloud-native apps and microservices while running only what you need. Open Liberty is the most flexible server runtime available to Java™ developers in this solar system.
Get Open Liberty

Liberty Architecture

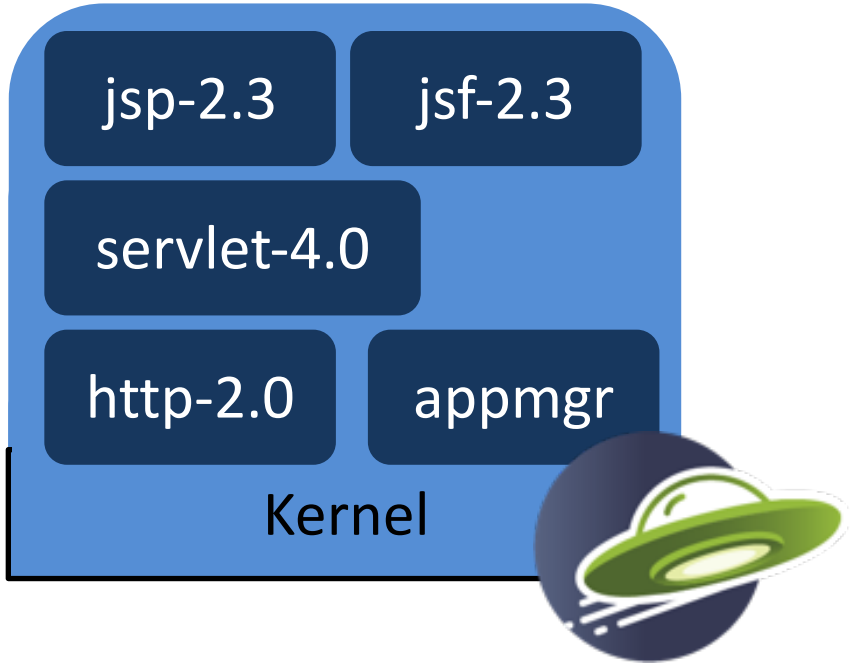
Fit for purpose

Open Liberty

- You control which features are loaded into each server instance




```
<feature>jsf-2.3</feature>
```



Liberty Zero Migration

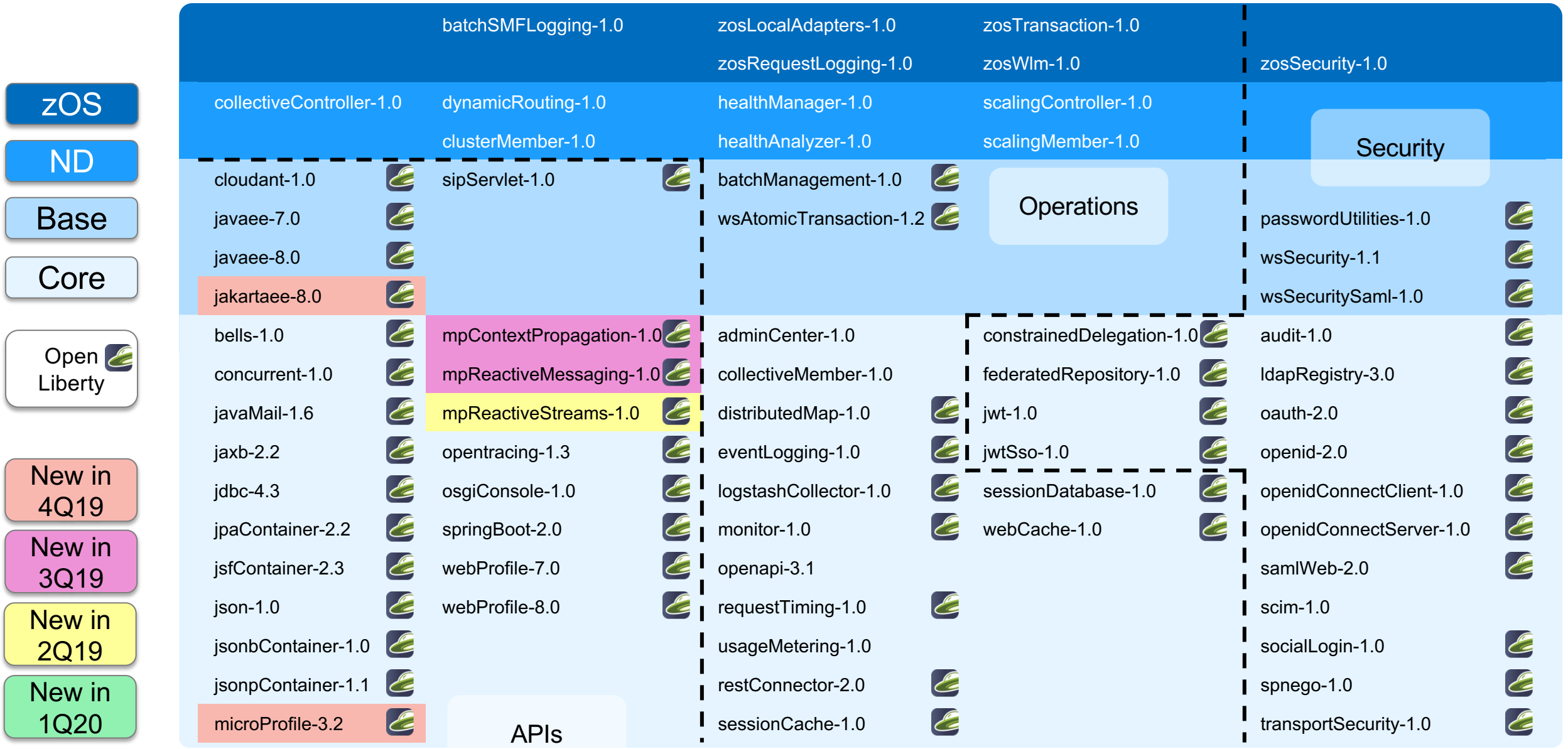
Open Liberty

- Zero config migration
  - Write once, run forever
- Zero migration for apps
  - No behavior changes in existing features
  - New behaviors in new features
- Choose your Java
  - Java 12, 11, 8
  - AdoptOpenJDK
  - IBM
  - OpenJDK
  - Oracle

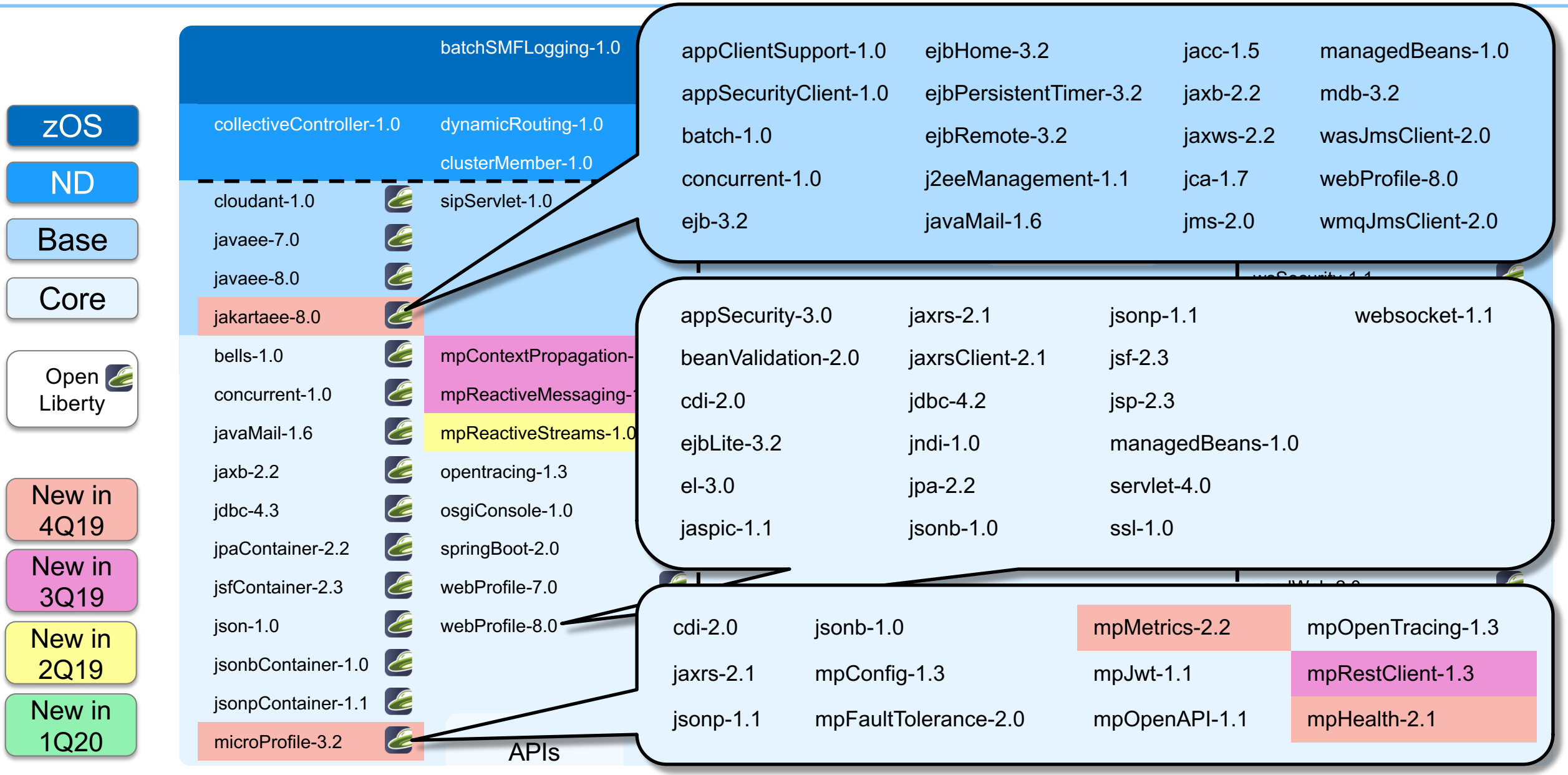
Features

Open Liberty

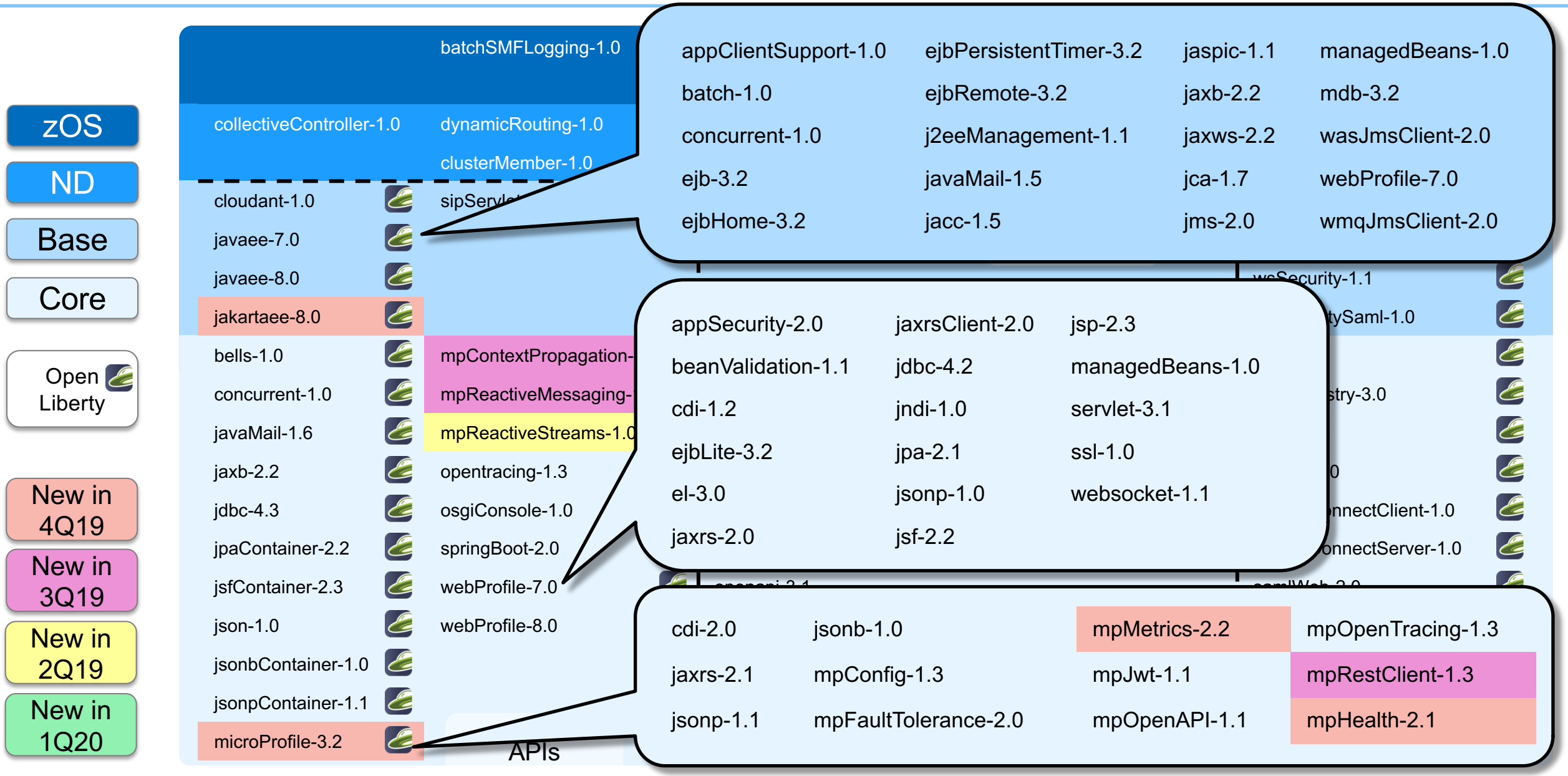
Periodic Table of Liberty (20.0.0.3)



Periodic Table of Liberty (20.0.0.3)



Periodic Table of Liberty (20.0.0.3)



Build

Module 1

Support for Maven and Gradle

- Manage full server and application lifecycle through the two most popular build technologies.
- Maven
  - liberty-maven-plugin
  - liberty-archetype-*
  - Docs & Source: https://github.com/OpenLiberty/ci.maven
- Gradle
  - liberty-gradle-plugin
  - Docs & Source: https://github.com/OpenLiberty/ci.gradle
- All the artefacts you need in Maven Central
  - https://search.maven.org/search?q=io.openliberty
  - https://search.maven.org/search?q=com.ibm.websphere

Dev Mode

Module 2

Open Liberty

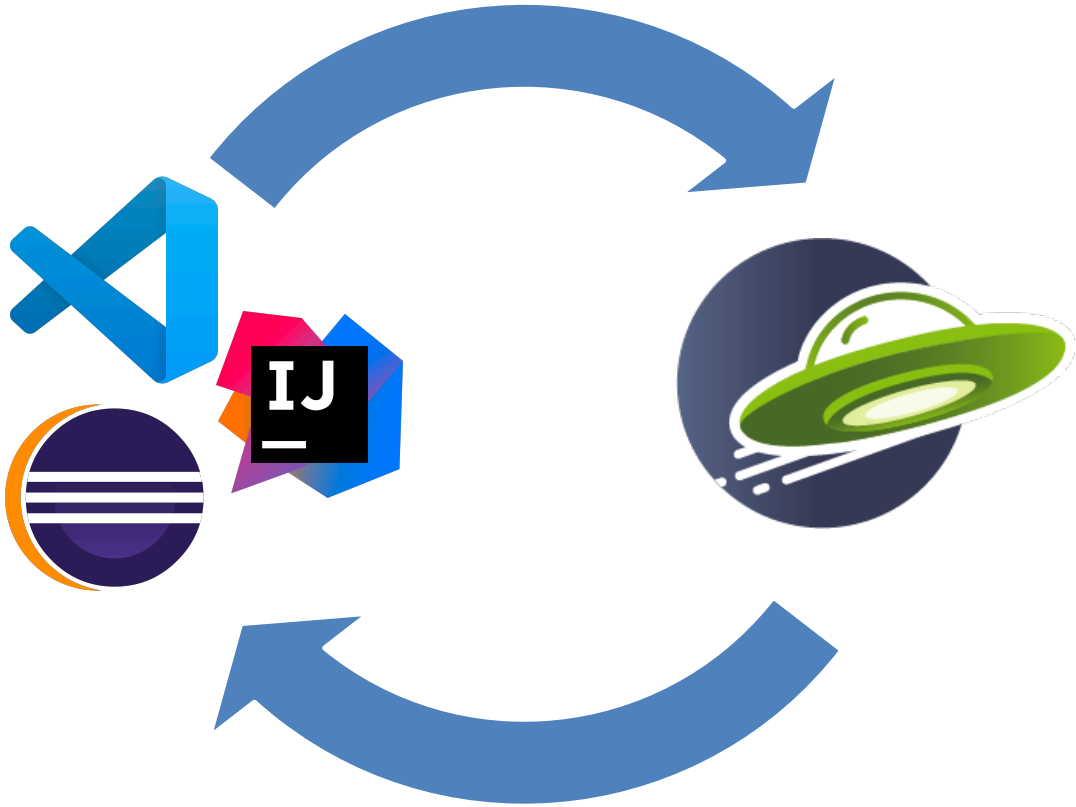
dev mode

Open Liberty

- Boosts developer productivity
- Immediate feedback for code and config changes
- No re-build necessary
- Hot deployment, testing and debugging

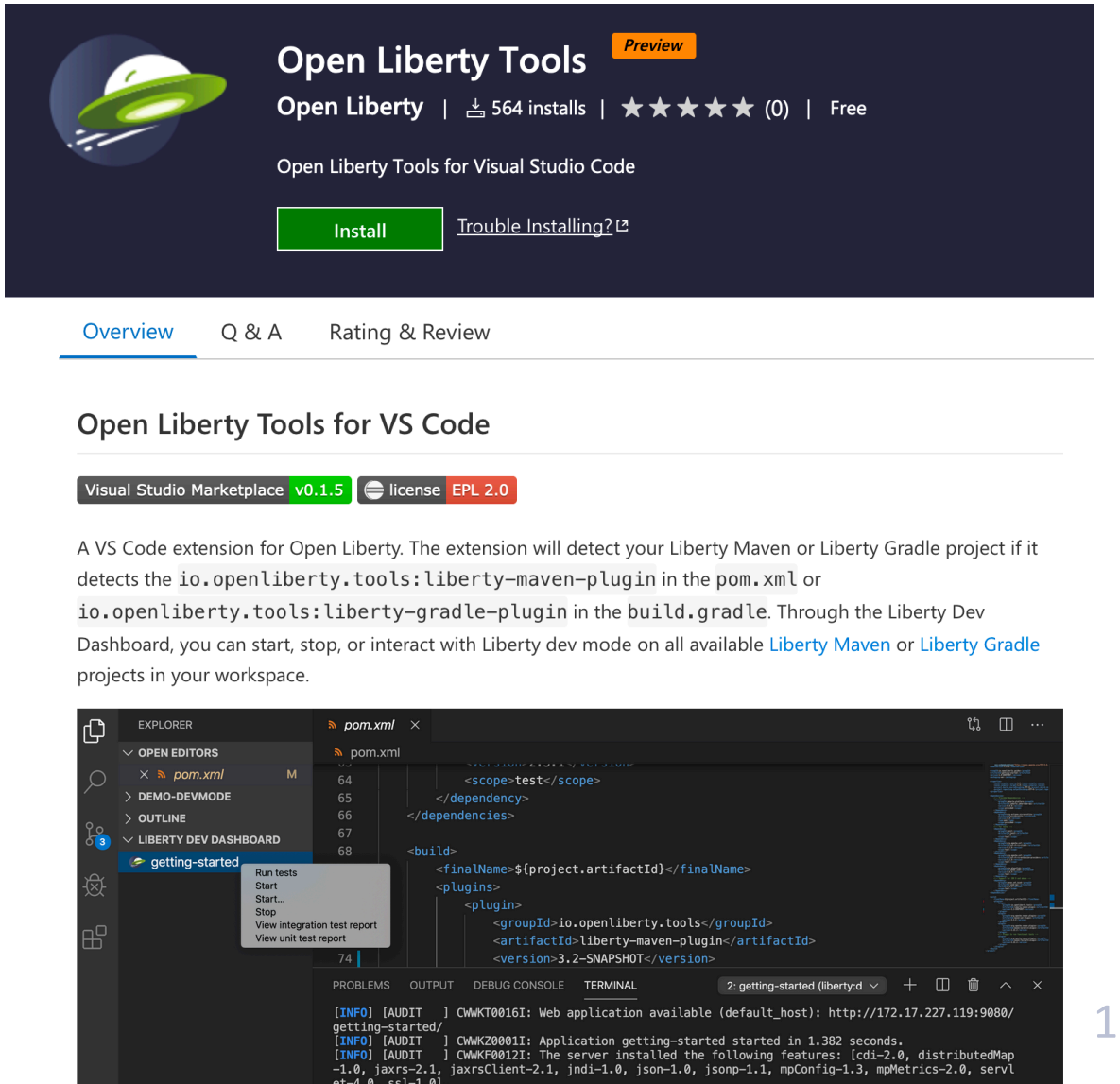
mvn liberty:dev

gradle libertyDev



Open Liberty Tools for VS Code

- Launch dev mode easily to build with Liberty from popular editor
- Auto detect Liberty projects
- View unit and integration test reports



Application APIs

Module 2

Java EE 7

appClientSupport-1.0
batch-1.0
concurrent-1.0
ejb-3.2
ejbHome-3.2
ejbPersistentTimer-3.2
ejbRemote-3.2
j2eeManagement-1.1
javaMail-1.5
jacc-1.5
jaspic-1.1
jaxb-2.2
jaxws-2.2
jca-1.7
jms-2.0
managedBeans-1.0
mdb-3.2
wasJmsClient-2.0
webProfile-7.0
wmqJmsClient-2.0

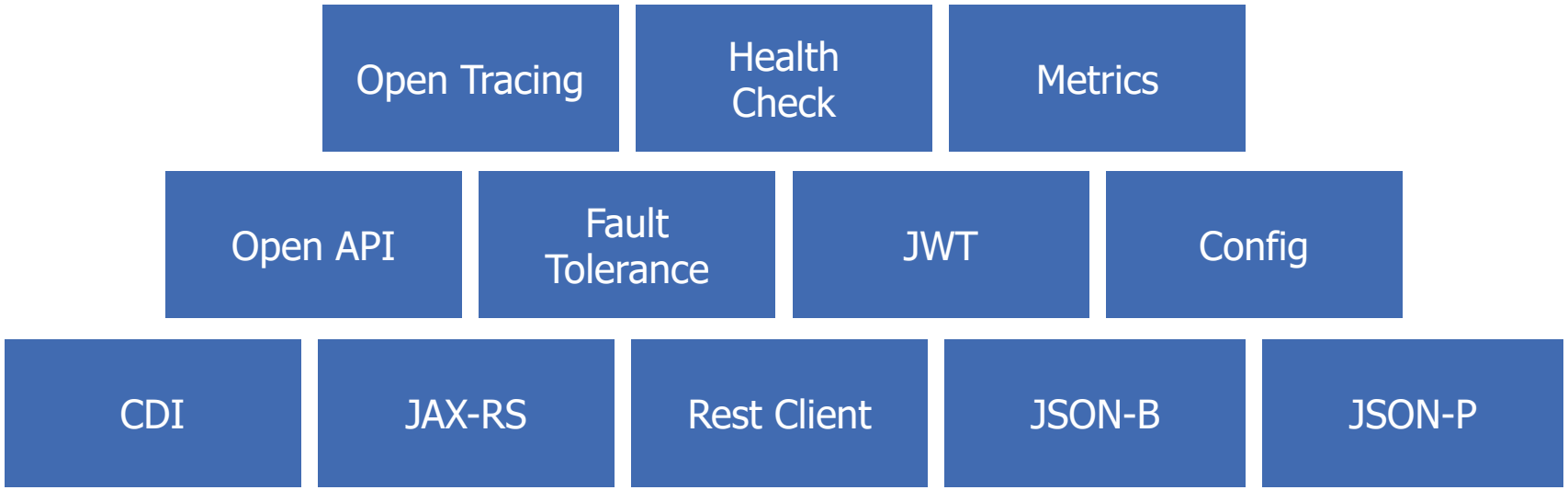
appSecurity-2.0
beanValidation-1.1
cdi-1.2
ejbLite-3.2
el-3.0
jaxrs-2.0
jaxrsClient-2.0
jdbc-4.2
jndi-1.0
jpa-2.1
jsonp-1.0
jsf-2.2
jsp-2.3
managedBeans-1.0
servlet-3.1
ssl-1.0
websocket-1.1

Jakarta EE or Java EE 8

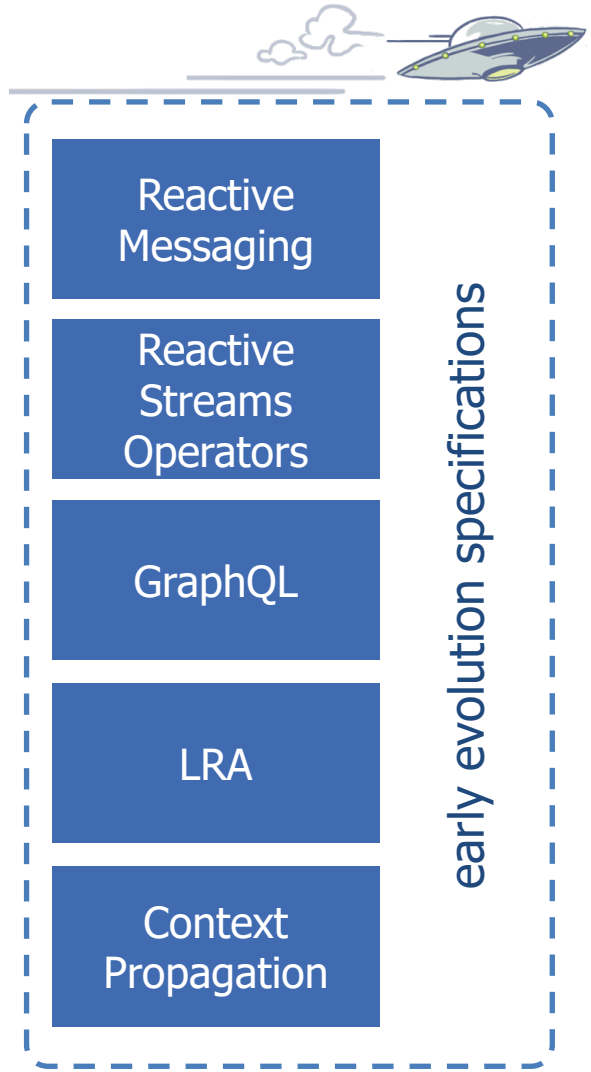
appClientSupport-1.0	ejbHome-3.2	jacc-1.5	managedBeans-1.0
appSecurityClient-1.0	ejbPersistentTimer-3.2	jaxb-2.2	mdb-3.2
batch-1.0	ejbRemote-3.2	jaxws-2.2	wasJmsClient-2.0
concurrent-1.0	j2eeManagement-1.1	jca-1.7	webProfile-8.0
ejb-3.2	javaMail-1.6	jms-2.0	wmqJmsClient-2.0

appSecurity-3.0	jaxrs-2.1	jsonp-1.1	websocket-1.1
beanValidation-2.0	jaxrsClient-2.1	jsf-2.3	
cdi-2.0	jdbc-4.2	jsp-2.3	
ejbLite-3.2	jndi-1.0	managedBeans-1.0	
el-3.0	jpa-2.2	servlet-4.0	
jaspic-1.1	jsonb-1.0	ssl-1.0	

Eclipse MicroProfile



Open Liberty



Eclipse MicroProfile

- Builds on Java EE
- By the Java EE community
- Open Source at Eclipse
- Multiple Implementations

```
@Path(“/”)
public class RestEE {

  @GET
  @Counter
  @Traced
  public String hello() {
    return "Hello MicroProfile!!";
  }
}
```

Server Configuration

Module 4

Simple Config


```
<server>
  <featureManager>
    <feature>jsp-2.3</feature>
  </featureManager>

  <webApplication location="myweb.war" contextRoot="/" />

  <applicationManager autoExpand="true"/>
</server>
```

server.xml

```
-Xmx1g
-Dsystem.prop=value
```

jvm.options

```
WLP_OUTPUT_DIR=/usr/wlp-out/
```

server.env

26

Composing Config


```
<server>
  <httpEndpoint id="defaultHttpEndpoint" host="${host}"
                httpPort="${http}"
                httpsPort="${https}"/>
</server>
```

configDropins/defaults/common-http.xml

```
<server>
  <include location="https://myHost/ports.xml"/>
  <variable name="host" value="${my.host}"/>
  <variable name="http" value="${my.host.http}"/>
  <variable name="https" value="${my.host.https}"/>
</server>
```

configDropins/overrides/ports.xml

27

Externalizing Configuration

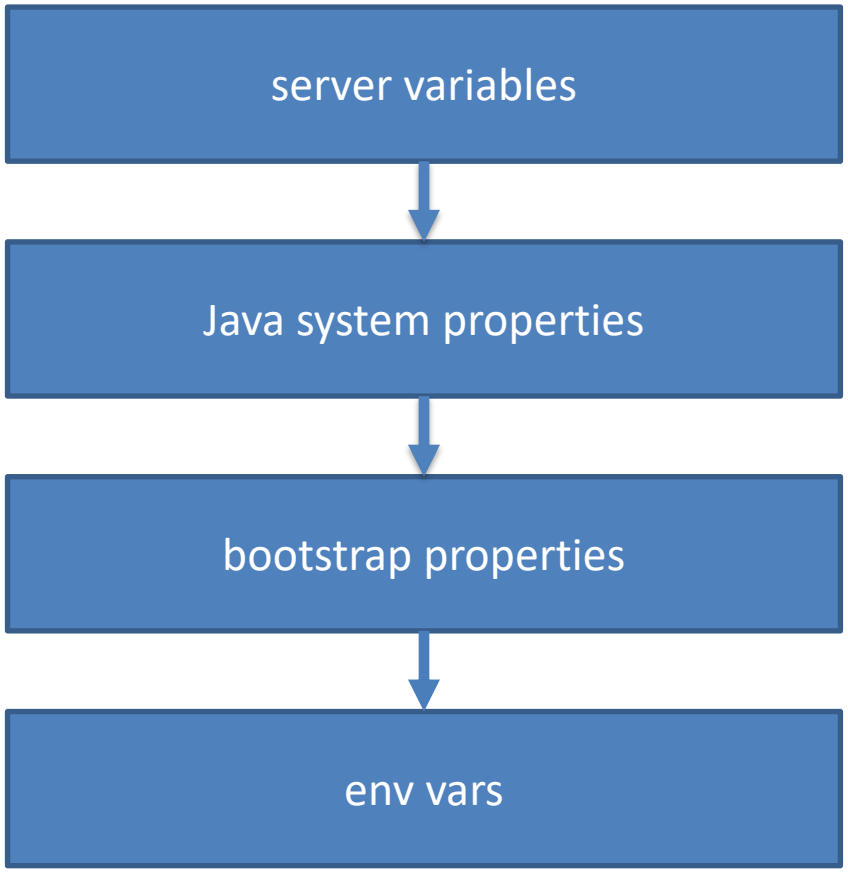
Module 5

Open Liberty

Variable Resolution

Open Liberty

- Liberty variable resolution happens top to bottom



Externalizing Config


```
<server>
  <httpEndpoint id="defaultHttpEndpoint" host="*"
                httpPort="${env.SERVER_HTTP_PORT}"
                httpsPort="${env.SERVER_HTTPS_PORT}"/>
</server>
```

configDropins/defaults/common-http.xml

30

Eclipse MicroProfile

- Inject Configuration
- Sourced from
  - Properties file
  - Environment
  - Java system properties

```
@Path(“/”)
public class RestEE {

  @Inject
  @ConfigProperty(“name”)
  private String name;

  @GET
  @Counter
  @Traced
  public String hello() {
    return "Hello " + name;
  }
}
```

Testing

Module 6

Open Liberty

Testing

- Use maven-surefire-plugin to run unit test
  - Use CDI dependency injection to make your code easy to mock & unit test
- Use maven-failsafe-plugin to run integration test
- Use Liberty Arquillian integration to run tests on a Liberty server
- Use MicroShed Testing for system testing in Containers (Module 8)

Docker

Module 7

Open Liberty

Liberty in Containers

app and config

+

liberty and java

+

os config

+

=

portable container

Hosted Cloud

- IBM Cloud Kubernetes Service
- Azure Kubernetes Service
- Google Kubernetes Engine
- Amazon Elastic Kubernetes Service
- Jelastic

Private Cloud

- Red Hat Open Shift Container Platform
- Pivotal Kubernetes Service
- Pivotal Cloud Foundry

```
FROM open-liberty
ADD myapp.war /config/dropins/myapp.war
```

Making the most of Docker

Open Liberty

Testing in Containers

Module 8

MicroShed Testing

Open Liberty

- Integration tests that are easy to setup, write, and run
- Test your apps the same way they run in production...in Containers
- Can run multiple containers on same network (e.g. test DB integration)
- https://microshed.github.io/

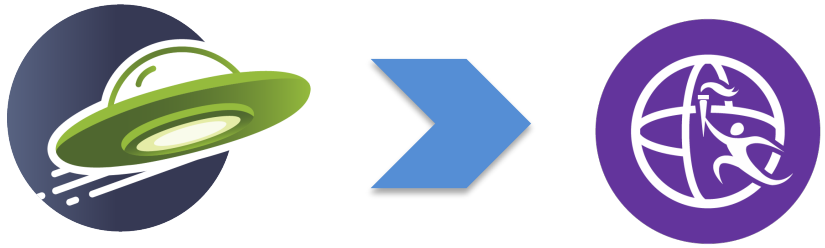
```
@MicroShedTest
public class MyTest {

    // Search for Dockerfile.
    // Start app in Container.
    // Wait for Container before running tests.
    @Container
    public static MicroProfileApplication app
        = new MicroProfileApplication()
                .withAppContextRoot("/myservice");

    // Inject JAX-RS REST Client
    @Inject
    public static MyService mySvc;

    // A test method like any other
    @Test
    public void testMyService() {
        ...
    }
}
```

Support Licensing

Module 9

Open Liberty

Buy WebSphere Liberty



	Monthly VPC (List Price)
WebSphere ND	US$341
WebSphere	US$91.20
Liberty Core	US$45

https://www.ibm.com/cloud/websphere-application-server/pricing

Licensing Options

Open Liberty

You are not alone...

Whether you're an enterprise or a smaller crew, there's Open Liberty support for all

IBM Cloud Pak for Applications

I'm an IBM customer wanting to modernize to OpenShift.

IBM Support

WebSphere Application Server

I just want support for Open Liberty.

IBM Support

Red Hat Runtimes

I'm a Red Hat customer wanting to deploy into OpenShift.

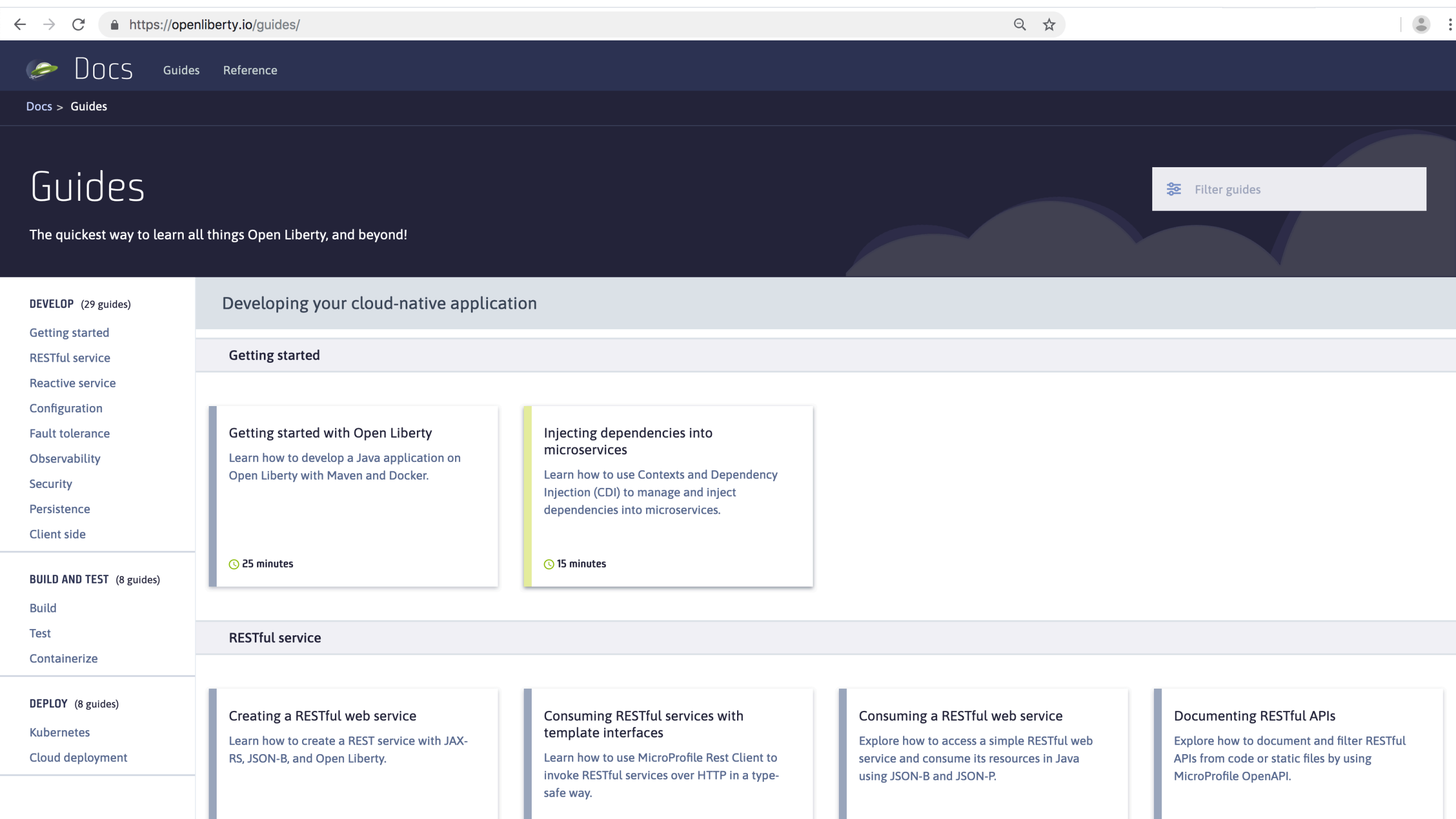
Red Hat Support

Wrap-up

Open Liberty

Mastersclass hands-on content

- Open Liberty Masterclass
  - Table of Contents
  - Before you begin
    - Install Pre-requisites
    - Prime Maven and Docker Caches
  - The Application
  - Module 1: Build
  - Module 2: Dev Mode
  - Module 3: Application APIs
  - Module 4: Server Configuration
  - Module 5: Externalizing Configuration
  - Module 6: Integration Testing
  - Module 7: Docker
    - Overriding Dev Server Configuration
  - Module 8: Support Licensing
  - Conclusion


https://openliberty.io/guides/
Docs
Guides
Reference
Docs > Guides
Guides
The quickest way to learn all things Open Liberty, and beyond!
Filter guides
DEVELOP (29 guides)
Getting started
RESTful service
Reactive service
Configuration
Fault tolerance
Observability
Security
Persistence
Client side
BUILD AND TEST (8 guides)
Build
Test
Containerize
DEPLOY (8 guides)
Kubernetes
Cloud deployment
Developing your cloud-native application
Getting started
Getting started with Open Liberty
Learn how to develop a Java application on Open Liberty with Maven and Docker.
25 minutes
Injecting dependencies into microservices
Learn how to use Contexts and Dependency Injection (CDI) to manage and inject dependencies into microservices.
15 minutes
RESTful service
Creating a RESTful web service
Learn how to create a REST service with JAX-RS, JSON-B, and Open Liberty.
Consuming RESTful services with template interfaces
Learn how to use MicroProfile Rest Client to invoke RESTful services over HTTP in a type-safe way.
Consuming a RESTful web service
Explore how to access a simple RESTful web service and consume its resources in Java using JSON-B and JSON-P.
Documenting RESTful APIs
Explore how to document and filter RESTful APIs from code or static files by using MicroProfile OpenAPI.

Thank You

Open Liberty