

build.gradle

```
allprojects {
    apply plugin: 'maven'

    group = 'sample'
    version = '1.0'
}

subprojects {
    apply plugin: 'java'

    sourceCompatibility = 1.8
    targetCompatibility = 1.8
    tasks.withType(JavaCompile) {
        options.encoding = 'UTF-8'
    }

    repositories {
        mavenLocal()
        mavenCentral()
    }
}
```

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>sample</groupId>
    <artifactId>ejb</artifactId>
    <version>1.0</version>
    <packaging>pom</packaging>

    <name>Sample EAR Project</name>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
    </properties>

    <modules>
        <module>ejb-ejb</module>
        <module>ejb-war</module>
        <module>ejb-ear</module>
    </modules>
</project>
```

ejb-ear/build.gradle

```
apply plugin: 'ear'
apply plugin: 'liberty'

description = 'EAR Module'

buildscript {
    repositories {
        mavenCentral()
        maven {
            name = 'Sonatype Nexus Snapshots'
            url = 'https://oss.sonatype.org/content/repositories/snapshots/'
        }
    }
    dependencies {
        classpath 'net.wasdev.wlp.gradle.plugins:liberty-gradle-plugin:2.6.3'
    }
}

dependencies {
    deploy project(':ejb-ejb')
    deploy project(path:':ejb-war', configuration:'archives')
    testCompile group: 'commons-httpclient', name: 'commons-httpclient', version:'3.1'
    testCompile group: 'junit', name: 'junit', version:'4.12'
    libertyRuntime group: 'com.ibm.websphere.appserver.runtime', name: 'wlp-javaee7', version: '18.0.0.1'
}

ear {
    archiveName = baseName + '.' + extension
}

liberty {
    server {
        name = "ejbServer"
        configFile = file('src/main/liberty/config/server.xml')
        apps = [ear]
    }
}

test {
    println 'inside the unit test block'
    reports.html.destination = file("${buildDir}/reports/unit")
    reports.junitXml.destination = file("${buildDir}/test-results/unit")
    exclude '**/it/**'
}

task integrationTest(type: Test) {
    group 'Verification'
    description 'Runs the integration tests.'
    reports.html.destination = file("${buildDir}/reports/it")
    reports.junitXml.destination = file("${buildDir}/test-results/it")
    include '**/it/**'
    exclude '**/unit/**'
}

task printMessageAboutRunningServer {
    doLast {
        println "The server is now running at https://localhost:9443/ejb-war"
        println "To stop the server run 'gradle libertyStop'"
    }
}

clean.dependsOn 'libertyStop'
ear.dependsOn ':ejb-war:war'
check.dependsOn 'integrationTest'
integrationTest.dependsOn 'libertyStart', 'testClasses'
integrationTest.finalizedBy 'libertyStop'
libertyStart.finalizedBy 'printMessageAboutRunningServer'
```

ejb-ear/pom.xml

```
<?xml version='1.0' encoding='utf-8'?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <parent>
    <groupId>sample</groupId>
    <artifactId>ejb</artifactId>
    <version>1.0</version>
  </parent>

  <artifactId>ejb-ear</artifactId>

  <packaging>ear</packaging>

  <name>EAR Module</name>

  <dependencies>
    <dependency>
      <groupId>sample</groupId>
      <artifactId>ejb-ejb</artifactId>
      <version>1.0</version>
      <type>ejb</type>
    </dependency>
    <dependency>
      <groupId>sample</groupId>
      <artifactId>ejb-war</artifactId>
      <version>1.0</version>
      <type>war</type>
    </dependency>
    <dependency>
      <groupId>commons-httpclient</groupId>
      <artifactId>commons-httpclient</artifactId>
      <version>3.1</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.12</version>
      <scope>test</scope>
    </dependency>
  </dependencies>

  <build>
    <finalName>${project.artifactId}</finalName>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-ear-plugin</artifactId>
        <version>2.10.1</version>
        <configuration>
          <version>7</version>
          <fileNameMapping>no-version</fileNameMapping>
          <generateApplicationXml>true</generateApplicationXml>
        </configuration>
      </plugin>
      <plugin>
        <groupId>net.wasdev.wlp.maven.plugins</groupId>
        <artifactId>liberty-maven-plugin</artifactId>
        <version>2.6.3</version>
        <configuration>
          <assemblyArtifact>
            <groupId>com.ibm.websphere.appserver.runtime</groupId>
            <artifactId>wlp-javaee7</artifactId>
            <version>17.0.0.2</version>
            <type>zip</type>
          </assemblyArtifact>
          <serverName>ejbServer</serverName>
          <configFile>src/main/liberty/config/server.xml</configFile>
          <appsDirectory>apps</appsDirectory>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

```

        <looseApplication>true</looseApplication>
        <installAppPackages>project</installAppPackages>
    </configuration>
    <executions>
        <execution>
            <id>stop-server-before-clean</id>
            <phase>pre-clean</phase>
            <goals>
                <goal>stop-server</goal>
            </goals>
        </execution>
        <execution>
            <id>create-server</id>
            <phase>prepare-package</phase>
            <goals>
                <goal>install-server</goal>
                <goal>create-server</goal>
            </goals>
        </execution>
        <execution>
            <id>install-apps</id>
            <phase>package</phase>
            <goals>
                <goal>install-apps</goal>
            </goals>
        </execution>
        <execution>
            <id>package-server</id>
            <phase>package</phase>
            <goals>
                <goal>package-server</goal>
            </goals>
        </execution>
        <execution>
            <id>test-start-server</id>
            <phase>pre-integration-test</phase>
            <goals>
                <goal>test-start-server</goal>
            </goals>
        </execution>
        <execution>
            <id>test-stop-server</id>
            <phase>post-integration-test</phase>
            <goals>
                <goal>test-stop-server</goal>
            </goals>
        </execution>
    </executions>
</plugin>
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>3.7.0</version>
    <executions>
        <execution>
            <goals>
                <goal>testCompile</goal>
            </goals>
        </execution>
    </executions>
</plugin>
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-failsafe-plugin</artifactId>
    <version>2.20.1</version>
    <executions>
        <execution>
            <goals>
                <goal>integration-test</goal>
                <goal>verify</goal>
            </goals>
        </execution>
    </executions>
</plugin>

```

```

        </plugin>
    </plugins>
</build>
</project>

```

ejb-ejb/build.gradle

```

description = 'EJB Module'

dependencies {
    testCompile group: 'junit', name: 'junit', version: '4.12'
    compileOnly group: 'javax', name: 'javaee-api', version: '7.0'
}

jar {
    archiveName = baseName + '.' + extension
}

```

ejb-ejb/pom.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <parent>
        <groupId>sample</groupId>
        <artifactId>ejb</artifactId>
        <version>1.0</version>
    </parent>

    <artifactId>ejb-ejb</artifactId>
    <packaging>ejb</packaging>

    <name>EJB Module</name>

    <dependencies>
        <dependency>
            <groupId>javax</groupId>
            <artifactId>javaee-api</artifactId>
            <version>7.0</version>
            <scope>provided</scope>
        </dependency>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.12</version>
            <scope>test</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-ejb-plugin</artifactId>
                <version>3.0.0</version>
                <configuration>
                    <ejbVersion>3.1</ejbVersion>
                </configuration>
            </plugin>
        </plugins>
    </build>
</project>

```

ejb-war/build.gradle

```
apply plugin: 'war'

jar.enabled = false
description = 'WAR Module'
dependencies {
    testCompile group: 'junit', name: 'junit', version: '4.12'
    providedCompile group: 'javax', name: 'javaee-api', version: '7.0'
    providedCompile project(':ejb-ejb')
}

war {
    archiveName = baseName + '.' + extension
}
```

ejb-war/pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <parent>
        <groupId>sample</groupId>
        <artifactId>ejb</artifactId>
        <version>1.0</version>
    </parent>

    <artifactId>ejb-war</artifactId>
    <packaging>war</packaging>

    <name>WAR Module</name>

    <dependencies>
        <dependency>
            <groupId>javax</groupId>
            <artifactId>javaee-api</artifactId>
            <version>7.0</version>
            <scope>provided</scope>
        </dependency>
        <dependency>
            <groupId>sample</groupId>
            <artifactId>ejb-ejb</artifactId>
            <version>1.0</version>
            <type>ejb</type>
            <scope>provided</scope>
        </dependency>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.12</version>
            <scope>test</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-war-plugin</artifactId>
                <version>3.2.0</version>
                <configuration>
                    <packagingExcludes>pom.xml</packagingExcludes>
                </configuration>
            </plugin>
        </plugins>
    </build>
</project>
```

gradle/wrapper/gradle-wrapper.properties

```
#Wed Oct 11 09:54:19 CDT 2017
distributionBase=GRADLE_USER_HOME
distributionPath=wrapper/dists
zipStoreBase=GRADLE_USER_HOME
zipStorePath=wrapper/dists
distributionUrl=https\://services.gradle.org/distributions/gradle-4.0.2-bin.zip
```

ejb-ear/src/main/liberty/config/server.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="ejb server">

    <!-- Enable features -->
    <featureManager>
        <feature>javaee-7.0</feature>
    </featureManager>

    <!-- Encoded password can be generated using bin/securityUtility -->
    <keyStore password="change1me"/>

    <basicRegistry id="basic" realm="BasicRealm">
        <user name="user1" password="change1me"/>
    </basicRegistry>

    <!-- To access this server from a remote client add a host attribute to the following element, e.g. host="*" -->
    <httpEndpoint id="defaultHttpEndpoint"
        httpPort="9080"
        httpsPort="9443" />

    <enterpriseApplication id="ejb-ear" location="ejb-ear.ear" name="ejb-ear"/>
</server>
```

ejb-ear/src/test/java/wasdev/ejb/it/EndpointIT.java

```
package wasdev.ejb.it;

import static org.junit.Assert.*;
import org.junit.BeforeClass;
import org.junit.Test;
import org.apache.commons.httpclient.HttpClient;
import org.apache.commons.httpclient.HttpStatus;
import org.apache.commons.httpclient.methods.GetMethod;

public class EndpointIT {
    private static String URL;

    @BeforeClass
    public static void init() {
        URL = "http://localhost:9080/ejb-war";
    }

    @Test
    public void testServlet() throws Exception {
        HttpClient client = new HttpClient();

        GetMethod method = new GetMethod(URL);

        try {
            int statusCode = client.executeMethod(method);

            assertEquals("HTTP GET failed", HttpStatus.SC_OK, statusCode);

            String response = method.getResponseBodyAsString(1000);

            assertTrue("Unexpected response body", response.contains("Hello EJB World."));
        } finally {
            method.releaseConnection();
        }
    }
}
```

ejb-ejb/src/main/java/wasdev/ejb/ejb/SampleStatelessBean.java

```
package wasdev.ejb.ejb;

import javax.ejb.Stateless;

@Stateless
public class SampleStatelessBean {

    public String hello() {
        return "Hello EJB World.";
    }
}
```

ejb-war/src/main/java/wasdev/ejb/ejb/web/EJBServlet.java

```
package wasdev.ejb.ejb.web;

import java.io.IOException;
import java.io.PrintWriter;

import javax.ejb.EJB;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import wasdev.ejb.ejb.SampleStatelessBean;

/**
 * A servlet which injects a stateless EJB
 */
@WebServlet({"/", "/ejbServlet"})
public class EJBServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    @EJB
    SampleStatelessBean statelessBean;

    @Override
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws IOException {
        PrintWriter writer = response.getWriter();

        // Call hello method on a stateless session bean
        String message = statelessBean.hello();

        writer.println(message);
    }
}
```


Hello EJB World.