



Open Liberty 2021

Release Notes for Open Liberty 21.0.0.5 on Red Hat OpenShift Container Platform

Release Notes for Open Liberty 2021 on Red Hat OpenShift Container Platform

Open Liberty 2021 Release Notes for Open Liberty 21.0.0.5 on Red Hat OpenShift Container Platform

Release Notes for Open Liberty 2021 on Red Hat OpenShift Container Platform

Legal Notice

Copyright © 2020 IBM Corp

Code and build scripts are licensed under the Eclipse Public License v1 Documentation files are licensed under Creative Commons Attribution-NoDerivatives 4.0 International (CC BY-ND 4.0)

Abstract

These release notes contain the latest information about new features, enhancements, fixes, and issues contained in Open Liberty 2021 on Red Hat OpenShift Container Platform release.

Table of Contents

CHAPTER 1. FEATURES	3
1.1. RUN YOUR APPS USING 21.0.0.5	3
1.1.1. LDAP connection support for Kerberos authentication	3
1.2. BUILD MULTIPART PAYLOADS FOR YOUR JAX-RS CLIENT AND SERVICES	5
1.3. SUPPORT FOR JAVA SE 16	5
CHAPTER 2. RESOLVED ISSUES	7
CHAPTER 3. FIXED CVES	8
CHAPTER 4. KNOWN ISSUES	9

CHAPTER 1. FEATURES

Open Liberty 21.0.0.5 comes complete with a new Kerberos authentication method, allowing for the use of an LDAP user registry to easily and quickly authorize connections originating from the same source. Also included is the ability to create and exchange multipart payloads using JAX-RS clients and services.

In [Open Liberty 21.0.0.5](#):

- [LDAP connection support for Kerberos authentication](#)
- [Build multipart payloads for your JAX-RS client and services](#)
- [Support for Java SE 16](#)

1.1. RUN YOUR APPS USING 21.0.0.5

If you're using [Maven](#), here are the coordinates:

```
<dependency>
  <groupId>io.openliberty</groupId>
  <artifactId>openliberty-runtime</artifactId>
  <version>21.0.0.5</version>
  <type>zip</type>
</dependency>
```

Or for [Gradle](#):

```
dependencies {
  libertyRuntime group: 'io.openliberty', name: 'openliberty-runtime', version: '[21.0.0.5,)'
}
```

Or if you're using Docker:

```
FROM open-liberty
```

1.1.1. LDAP connection support for Kerberos authentication

LDAP bind operations are used to authenticate clients (and the users or applications behind them) to the directory server. This establishes an authorization identity that is used for subsequent operations that are processed on that connection, and specifies the LDAP protocol version that the client uses. Before this update, the **LdapRegistry** element supported binding either anonymously or by using simple authentication with a user (**bindDN**) and password (**bindPassword**). This update adds an option to bind to LDAP: GSSAPI/Kerberos. Kerberos is an authentication mechanism that allows a client and service to mutually authenticate by a Key Distribution Center (KDC). In Open Liberty 21.0.0.5, you can use either a Kerberos credential cache (ccache) or a Kerberos keytab file.

To update an **LdapRegistry** to use the GSSAPI/Kerberos option, you can set the bind authentication mechanism type using the new **LdapRegistry** attribute, **bindAuthMechanism**:

```
bindAuthMechanism="GSSAPI"
```

You also need the Kerberos principal or Service Principal Name:

```
krb5Principal="user1@EXAMPLE.COM"
```

If you are using a Kerberos credential cache (ticket cache or ccache) to store the Kerberos credentials, add the ticket cache file name to the **LdapRegistry** with the new attribute, **krb5TicketCache**:

```
krb5TicketCache="${server.config.dir}/security/krb5-user1.cc"
```

If you are using a custom Kerberos configuration file (krb.conf or krb.ini), set the file name using the global Kerberos configuration element:

```
<kerberos configFile="${server.config.dir}/security/krb5.conf"/>
```

If you are using a Kerberos keytab file to store encrypted keys for principals, set the file using the global Kerberos configuration element:

```
<kerberos keytab="${server.config.dir}/security/krb5.keytab"
configFile="${server.config.dir}/security/krb5.conf"/>
```

If the Kerberos configuration file is not defined, Open Liberty will attempt to resolve by using the JDK default locations and the operating system default locations.

For the Kerberos credentials, the locations are checked in the following order: the ticket cache (if provided), the configured keytab file, and finally the JDK default location.

The following example shows how to configure the **LdapRegistry** element using a ticket cache and custom Kerberos config file:

```
<kerberos keytab="${server.config.dir}/security/krb5.keytab"
configFile="${server.config.dir}/security/krb5.conf"/>

<ldapRegistry id="LDAP" realm="SampleLdapADRealm" host="ldap_hostname" port="389"
ignoreCase="true" baseDN="DC=example,DC=com" bindAuthMechanism="GSSAPI"
krb5Principal="user1@EXAMPLE.COM" krb5TicketCache="${server.config.dir}/security/krb5-
user1.cc" ldapType="Custom" />
```

The following example shows how to configure an LDAP Registry using a keytab and custom Kerberos config file:

```
<kerberos keytab="${server.config.dir}/security/krb5.keytab"
configFile="${server.config.dir}/security/krb5.conf" />

<ldapRegistry id="LDAP" realm="SampleLdapADRealm" host="ldap_hostname" port="389"
ignoreCase="true" baseDN="DC=example,DC=com" bindAuthMechanism="GSSAPI"
krb5Principal="user1@EXAMPLE.COM" ldapType="Custom" />
```

For more information on **LdapRegistry**, see the [LDAP User Registry documentation](#).

To enable this new function in your app, add the LDAP User Registry 3.0 feature to your **server.xml** file:

```
<featureManager>
  <feature>ldapRegistry-3.0</feature>
</featureManager>
```


1.2. BUILD MULTIPART PAYLOADS FOR YOUR JAX-RS CLIENT AND SERVICES

Often, a RESTful service or client will need to send multiple disparate pieces of data in the same request, for example, uploading a resume/CV with a picture and text for name and address. This is usually done using **multipart/form-data**.

While Liberty currently has APIs to enable users to *receive* multipart/form-data payloads, it does not have any APIs to enable users to *send* multipart payloads - until now. With the new **AttachmentBuilder** API, users can now send multipart requests from their JAX-RS clients or send multipart payloads as responses from their JAX-RS resources.

To send a multipart request, you will need to enable the **jaxrs-2.0** or **jaxrs-2.1** feature. Presumably, if you are already using JAX-RS, one of these features will already be enabled. To send a multipart payload, you must send an instance of **List<Attachment>**. Each object in that list represents a single attachment part. Here is an example of creating and sending a multipart request from a JAX-RS client:

```
List<Attachment> attachments = new ArrayList<>();

attachments.add(AttachmentBuilder.newBuilder("blogPost")
    .inputStream(new FileInputStream("/path/to/yesterdaysBlogPost.xml"))
    .fileName("myRenamedBlogPost.asciidoc")
    .contentType("text/asciidoc")
    .contentId("myBlogPostID")
    .header("X-PriorityLevel", "Medium")
    .build());

attachments.add(AttachmentBuilder.newBuilder("file1")
    .inputStream("some.xml", new FileInputStream("/path/to/myPicture.png"))
    .contentType("image/png")
    .build());

attachments.add(AttachmentBuilder.newBuilder("authorName")
    .inputStream(new ByteArrayInputStream("John Doe".getBytes()))
    .build());

Response response = client.target(BLOG_SITE_URI)
    .request()
    .post(Entity.entity(attachments, MediaType.MULTIPART_FORM_DATA));
```

For more information visit:

- [multipart/form-data RFC 7578](#)

1.3. SUPPORT FOR JAVA SE 16

Any official Java SE 16 release from [AdoptOpenJDK](#), [Oracle](#), or other OpenJDK vendor will work with Open Liberty. Java SE 16 is not a long-term supported release, with standard support scheduled to end in September 2021.

Keep in mind, Eclipse OpenJ9 [typically offers faster startup times](#) than Hotspot.

The primary features added in this release include:

- [JEP 396](#) Strongly Encapsulate JDK Internals by Default

- [JEP 395](#) Records
- [JEP 386](#) Alpine Linux Port
- [JEP 376](#) Concurrent Thread-Stack Processing
- [JEP 397](#) Sealed Classes (Second Preview)
- [JEP 338](#) Vector API (Incubator)

For more information on downloading a version of Java 16, see [AdoptOpenJDK.net](#), [Eclipse.org](#) or [OpenJDK.java.net](#).

For working with the **server.env** file in Open Liberty, see the **Configuration Files** section of the Open Liberty [Server Configuration Overview documentation](#).

For more information on new features available in Java 16, see [OpenJDK](#).

CHAPTER 2. RESOLVED ISSUES

See the [Open Liberty 21.0.0.5 issues that were resolved for this release](#) .

CHAPTER 3. FIXED CVEs

For a list of CVEs that were fixed in Open Liberty 21.0.0.5, see [security vulnerabilities](#).

CHAPTER 4. KNOWN ISSUES

See the [list of issues that were found but not fixed during the development of 21.0.0.5](#) .