



Open Liberty 2020

Release Notes for Open Liberty 20.0.0.9 on Red Hat OpenShift Container Platform

Release Notes for Open Liberty 2020 on Red Hat OpenShift Container Platform

Open Liberty 2020 Release Notes for Open Liberty 20.0.0.9 on Red Hat OpenShift Container Platform

Release Notes for Open Liberty 2020 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 2020 on Red Hat OpenShift Container Platform release.

Table of Contents

CHAPTER 1. FEATURES	3
1.1. RUN YOUR APPS USING 20.0.0.9	3
1.2. EXPERIMENT WITH A THIRD-PARTY GRAPHQL CLIENT API	3
1.3. DEVELOP MICROPROFILE GRAPHQL QUERIES AND MUTATIONS FASTER WITH THE BUILT-IN GRAPHIQL USER INTERFACE	4
1.4. SIGNIFICANT BUGS FIXED IN THIS RELEASE	5
1.4.1. Change your SSL config without restarting your server	5
1.4.2. Occasional error during Arquillian testing	5
1.4.3. Include element not being parsed with featureUtility	5
1.4.4. MicroProfile GraphQL Exception allowlist now up and running	6
CHAPTER 2. RESOLVED ISSUES	7
CHAPTER 3. FIXED CVES	8
CHAPTER 4. KNOWN ISSUES	9

CHAPTER 1. FEATURES

With Open Liberty 20.0.0.9 you can now experiment with the type-safe SmallRye GraphQL Client API, and more easily write and run GraphQL queries and mutations using a built-in GraphiQL user interface. Plus there has been a number of significant bug fixes.

In [Open Liberty 20.0.0.9](#):

- [Experiment with a third-party GraphQL client API](#)
- [Built-in GraphiQL user interface for faster queries and mutations](#)
- [Significant bugs fixed in this release](#)

1.1. RUN YOUR APPS USING 20.0.0.9

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

```
<dependency>
  <groupId>io.openliberty</groupId>
  <artifactId>openliberty-runtime</artifactId>
  <version>20.0.0.9</version>
  <type>zip</type>
</dependency>
```

Or for [Gradle](#):

```
dependencies {
  libertyRuntime group: 'io.openliberty', name: 'openliberty-runtime', version: '[20.0.0.9,)'
}
```

Or if you're using Docker:

```
FROM open-liberty
```

1.2. EXPERIMENT WITH A THIRD-PARTY GRAPHQL CLIENT API

[MicroProfile GraphQL](#) has only been available in Open Liberty for a few months now, and it is already a hit! That said, there are lots of things we'd like to do to improve it - and make it more complete. One of those things is a Client API. While the official Client API is not expected until the next release of MicroProfile GraphQL, you can now experiment with the type-safe SmallRye GraphQL Client API. SmallRye is the underlying implementation of MicroProfile GraphQL, and you can access some of its above-and-beyond-the-spec features by adding the "third-party" API type visibility to your application, like so:

```
<application name="MyGraphQLApp" location="MyGraphQLApp.war">
  <classloader apiTypeVisibility="+third-party"/>
</application>
```

Now you can access SmallRye GraphQL APIs like the type-safe client! Note that these APIs may change in future releases as SmallRye is constantly evolving. For more information please visit [SmallRye GraphQL project](#) - in Open Liberty 20.0.0.9, we are using SmallRye GraphQL 1.0.7. The SmallRye

GraphQL Client APIs are very similar to [MicroProfile Rest Client](#) in that it uses an interface to invoke remote methods in a type-safe manner. For example, suppose we want a client that can invoke a query of all super heroes in a given location. We would create a query interface like this:

```
@GraphQLClientApi
interface SuperHeroesApi {
    List<SuperHero> allHeroesIn(String location);
}
```

Where **SuperHero** on the client side looks like this:

```
class SuperHero {
    private String name;
    private List<String> superPowers;
}
```

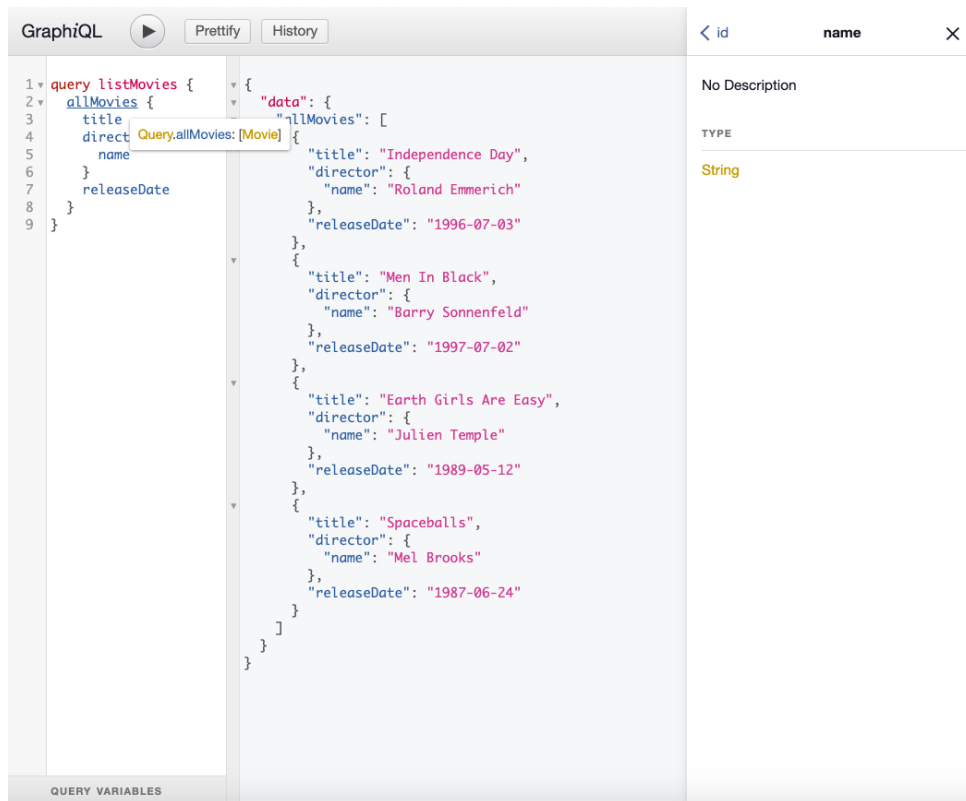
The **SuperHero** entity might contain dozens more fields on the server side, but if we're only interested in the hero's name and super powers, then we only need those two fields in our client-side class. Now we can invoke the query with some code like this:

```
SuperHeroesApi api = GraphQLClientBuilder.newBuilder().build(SuperHeroesApi.class);
List<SuperHero> heroesOfNewYork = api.allHeroesIn("NYC");
```

Remember that this client API is not official, but the official API (in MicroProfile GraphQL 1.1) will be based on this - think of this as a sneak peak!

1.3. DEVELOP MICROPROFILE GRAPHQL QUERIES AND MUTATIONS FASTER WITH THE BUILT-IN GRAPHIQL USER INTERFACE

Open Liberty now sports a built-in [GraphiQL](#) user interface. This web-based UI allows you to write and execute GraphQL queries and mutations in real time with advanced editing features like command completion, query history, schema introspection, etc. To enable the UI, you must first [write and deploy a MicroProfile GraphQL application](#). Then add this line to your server.xml: **<variable name="io.openliberty.enableGraphQLUI" value="true" />** You can access the UI using a web browser by accessing your GraphQL app's context root and adding **/graphql-ui**. For example, suppose we use the default port (9080) and our app is named **myGraphQLApp**, we would access the UI at: <http://localhost:9080/myGraphQLApp/graphql-ui>. Resolved in issue [#13201](#)



1.4. SIGNIFICANT BUGS FIXED IN THIS RELEASE

We've spent some time fixing bugs. The following sections describe just some of the issues we resolved in this release. If you're interested, here's the full list of [fixed bugs in 20.0.0.9](#).

1.4.1. Change your SSL config without restarting your server

If you use the JAX-RS Client in your application to access SSL-secured RESTful services, then you will likely have a key store and/or trust store configured. These SSL settings enable you to ensure you are communicating with the endpoint you expect - and that nobody else is listening in! These settings also enable SSL-based client authentication. Whatever you use these SSL settings for, if you need to change them, then that meant that you would need to restart your server so that the clients could pick up the new settings. Well, that is a thing of the past! Starting in 20.0.0.9, JAX-RS clients can now dynamically adjust to changes in the SSL configuration. This should rapidly improve app development and deployment! Resolved in issue [#13027](#).

1.4.2. Occasional error during Arquillian testing

During Arquillian testing there was a bug that would throw an occasional **ArrayIndexOutOfBoundsException** in **JaspiServiceImpl.getDescription**. This problem was caused by registering the internal **AuthConfigProvider** for JSR 375. There are two threads registering two instances of the provider. Additional synchronization has been added so that a single internal provider is registered. Resolved in issue [#11504](#).

1.4.3. Include element not being parsed with featureUtility

This bug was caused where the include tag in **server.xml** was being ignored by **featureUtility**. The **include** element can be used in a **server.xml** file to consolidate configuration from additional separate **server.xml** files. The bug was fixed by handling the parsing with a more appropriate XML method. Resolved in issue [#13138](#).

1.4.4. MicroProfile GraphQL Exception allowlist now up and running

As an open source product, we love hearing feedback from Open Liberty users. We especially love comments like this: "Hello, I am using microprofile-graphql on openliberty and everything goes well..." and even the rest of that sentence (taken from Open Liberty [#13036](#)): "except for the exception whitelisting mechanism via microprofile config described here."

Our MicroProfile GraphQL feature has only been generally available for a few months, so it's great to see that users are using it - and it's exciting to see that they are already exploring the "dark corners" (exception handling, etc.). While we hate to find out that we let a bug slip through the cracks, we're always eager to fix them when they do. If you find an issue or would like to suggest an enhancement that would make your experience with Open Liberty better, please let us know! You can always reach us by [opening an issue on GitHub](#) or on Twitter at [@OpenLibertyIO](#). We're also available for chatting online using Gitter at: <https://gitter.im/OpenLiberty/help> and <https://gitter.im/OpenLiberty/developer-experience>. We can't wait to hear from you!

CHAPTER 2. RESOLVED ISSUES

See the [Open Liberty 20.0.0.9 issues that were resolved for this release](#) .

CHAPTER 3. FIXED CVEs

For a list of CVEs that were fixed in Open Liberty 20.0.0.9, see [security vulnerabilities](#).

CHAPTER 4. KNOWN ISSUES

See the [list of issues that were found but not fixed during the development of 20.0.0.9](#) .