Red Hat build of Apicurio Registry 2.4

Release Notes for Apicurio Registry 2.4

What's new in Red Hat build of Apicurio Registry

Last Updated: 2023-11-07
What's new in Red Hat build of Apicurio Registry
Abstract

Describes the Red Hat build of Apicurio Registry product and provides the latest details on what’s new in this release.
# Table of Contents

**PREFACE** ................................................................. 3  
MAKING OPEN SOURCE MORE INCLUSIVE ............................. 3  

**CHAPTER 1. APICURIO REGISTRY RELEASE NOTES** .................. 4  
1.1. APICURIO REGISTRY INSTALLATION OPTIONS ..................... 4  
1.2. APICURIO REGISTRY SUPPORTED PLATFORMS ...................... 4  
1.2.1. Supported integration with other products ...................... 4  
1.2.2. Operator metadata versions ................................... 5  
1.3. APICURIO REGISTRY NEW FEATURES ............................... 5  
  Apicurio Registry core new features ............................... 5  
  Apicurio Registry Operator new features ......................... 7  
  Apicurio Registry user documentation and examples .......... 8  
1.4. APICURIO REGISTRY DEPRECATED FEATURES .................... 8  
  Apicurio Registry core deprecated features .................... 8  
  Apicurio Registry Operator deprecated features ................ 9  
1.5. UPGRADING AND MIGRATING APICURIO REGISTRY DEPLOYMENTS 9  
  1.5.1. Updating 2.x client dependencies ....................... 9  
  1.5.2. Upgrading from Apicurio Registry 2.x on OpenShift ...... 10  
  1.5.3. Migrating from Apicurio Registry 1.1 on OpenShift ...... 10  
1.6. APICURIO REGISTRY RESOLVED ISSUES .......................... 10  
  Apicurio Registry Operator resolved issues .................... 12  
1.7. APICURIO REGISTRY RESOLVED CVES ............................. 12  
1.8. APICURIO REGISTRY KNOWN ISSUES .............................. 13  
  Apicurio Registry core known issues ............................ 13  
  Apicurio Registry Operator known issues ...................... 14  

**APPENDIX A. USING YOUR SUBSCRIPTION** ......................... 15  
Accessing your account ............................................. 15  
Activating a subscription ......................................... 15  
Downloading ZIP and TAR files .................................... 15
PREFACE

MAKING OPEN SOURCE MORE INCLUSIVE
Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright’s message.
CHAPTER 1. APICURIO REGISTRY RELEASE NOTES

Red Hat build of Apicurio Registry 2.4 is provided as a General Availability release. Apicurio Registry is a data store for standard event schemas and API designs, and is based on the Apicurio Registry open source community project.

**NOTE**

Red Hat build of Apicurio Registry is the new product name for Red Hat Integration Service Registry. Red Hat build of Apicurio Registry 2.x and Red Hat Integration Service Registry 2.x are functionally identical.

You can use Apicurio Registry to manage and share the structure of your data using a web console, REST API, Maven plug-in, or Java client. For example, client applications can dynamically push or pull the latest schema updates to or from Apicurio Registry without needing to redeploy. You can also create optional rules to govern how Apicurio Registry content evolves over time. These rules include validation of content, integrity of artifact references, and backwards or forwards compatibility of schema or API versions.

### 1.1. APICURIO REGISTRY INSTALLATION OPTIONS

You can install Apicurio Registry on OpenShift with either of the following data storage options:

- PostgreSQL database
- Red Hat AMQ Streams

For more details, see [Installing and deploying Red Hat build of Apicurio Registry on OpenShift](#).

### 1.2. APICURIO REGISTRY SUPPORTED PLATFORMS

Apicurio Registry 2.4 supports the following core platforms:

- Red Hat OpenShift Container Platform 4.14, 4.13, 4.12, 4.11
- Red Hat OpenShift Service on AWS
- Microsoft Azure Red Hat OpenShift
- PostgreSQL 15, 14, 13, 12
- Red Hat AMQ Streams 2.5, 2.2
- OpenJDK 17, 11

For more details, see the following article:

- [Supported Configurations for Red Hat build of Apicurio Registry](#)

### 1.2.1. Supported integration with other products

Apicurio Registry 2.4 also supports integration with the following products:

- Red Hat Single Sign-On (RH-SSO) 7.6
1.2.2. Operator metadata versions

For details on the corresponding Service Registry Operator metadata versions used to install and deploy Apicurio Registry, see the following article:

- Red Hat Integration - Service Registry Operator metadata versions.

1.3. APICURIO REGISTRY NEW FEATURES

Apicurio Registry 2.4 includes the following new features:

Apicurio Registry core new features

Artifact references improvements

- **Artifact reference integrity rule**: You can now apply a new artifact-specific rule and global rule to enforce the integrity of artifact references when creating or updating artifacts. For all artifact types, the rule detects duplicate artifact references and prevents references to nonexistent artifacts. For a subset of artifact types (Apache Avro, Google Protobuf, OpenAPI, and AsyncAPI), the rule ensures that all artifact references are mapped.

- **Automatic artifact reference detection in Maven plug-in**: For a subset of artifact types (Apache Avro, Google Protobuf, and JSON Schema), the Maven plug-in can now automatically identify and configure all of the artifact references for a given artifact when creating or updating artifacts.

- **Visualization of artifact references in the web console**: You can now view both inbound and outbound references for an artifact version, in a new References tab. The REST API now supports the retrieval of both inbound and outbound references. Previously, the REST API retrieved outbound references only, and the web console did not display references.

- **Artifact references are now treated as content**: This release now considers artifact references when calculating the content hash and ID, and when checking compatibility of an artifact version. If you upload the same schema content with different references, you create a new artifact version.

Client SDK generation for OpenAPI

This release adds a new web console feature to enable users to generate a client SDK from an OpenAPI artifact. This feature uses Kiota from Microsoft to generate the SDK. The feature runs in your browser only, and cannot be automated by using an API. For more information, see https://github.com/microsoft/kiota.

Artifact version deletion

This release adds a new REST API operation, and a new Delete artifact version setting in the web console, to enable you to delete artifact versions by using the REST API. Previously, artifact versions were immutable; you could deprecate or disable them, but not delete them. However, it is sometimes necessary to delete an artifact version, despite the semantic implications. For example, you might need to delete an artifact version for regulatory or policy reasons.

Core Registry REST API improvements

- **Version comments API**: You can now use the REST API to add comments to artifact versions. Managing comments is not yet available in the web console.
- **Export API supports application/json in the Accept header**: You can now send `application/json` as the value of the **Accept** header in a call to the `/admin/export` API endpoint. Previously, the only way to return an `application/json` formatted response was by using the **forBrowser** query parameter. Now you can use either the query parameter or the **Accept** header.

- **Group management**: You can now optionally manage artifact groups directly by using the REST API, rather than implicitly as part of artifact creation.

**Confluent Compatibility REST API improvements**

- **Updated support for Confluent Compatibility API**: Added support for version 7 of the Confluent Schema Registry API. Apicurio Registry now supports both v6 and v7, at different endpoints.

- **Customizable maximum number of Subjects** : You can now use the `registry.ccompat.max-subjects` dynamic configuration property to customize the maximum number of Subjects returned by the `ccompat` API.

- **Added header for group customization**: When using the Confluent Schema Registry API v7, you can now use the `X-Registry-GroupId` header to specify an artifact group. Without using this header, all artifacts are assumed to be located in the default group.

**Support for authentication in Microsoft Azure Active Directory**

Added support for using Active Directory authentication in Microsoft Azure for access to the Core Registry REST API and web console. For more details, see the Apicurio blog post on **Securing Apicurio Registry 2.4.x using Azure Active Directory**.

**Support for HTTPS passthrough ingress**

Provides configuration variables to avoid redirect issues when deploying Apicurio Registry with HTTPS passthrough ingress:

- **REGISTRY_URL_OVERRIDE_HOST** - Override the hostname used to generate externally-accessible URLs.

- **REGISTRY_URL_OVERRIDE_PORT** - Override the port used to generate externally-accessible URLs.

  These host and port overrides are useful when deploying Apicurio Registry with HTTPS passthrough ingress or route. In these cases, the request URL and port that are reused for redirection do not belong to the actual external URL used by the client, because the request is proxied. The redirection then fails because the target URL is not reachable.

**Other changes**

- **Configure scope of Client Credentials flow for basic authentication**: Apicurio Registry uses the Client Credentials flow to request access tokens on behalf of users, when they use basic authentication. This feature allows users to configure the **scope** request parameter.

- **Serializer/deserializer artifact resolution with content hash**: SerDe classes can now resolve artifact coordinates by using the content hash of the schema.

- **Maven plug-in option to minify Avro**: When registering an Avro schema by using the Maven plug-in, you can now minify the content prior to registration.

**Community-only: Custom artifact types**
Users can potentially extend Apicurio Registry with their own custom artifact types, and remove support for existing types. This feature is available only in the community version of Apicurio Registry, and is not supported.

**NOTE**

To provide this feature, the `ArtifactType` in the REST API was changed from an `enum` to a simple `string`. If you use the REST API client for custom integrations, you might have to edit your code after you upgrade to the newer client.

### Apicurio Registry Operator new features

#### Operator support for HTTPS

Added support for configuring HTTPS connections inside the OpenShift cluster. You can create a Secret containing the certificate and the private key, named `tls.crt` and `tls.key` respectively, and reference the Secret by setting the `spec.configuration.security.https.secretName` field in the `ApicurioRegistry` custom resource definition (CRD) file. The Apicurio Registry Operator can then configure an HTTPS port on the Apicurio Registry `Service` resource. If HTTPS is enabled, you can disable the HTTP connection by setting `spec.security.https.disableHttp` to `true`.

#### Support for manually managed OpenShift resources

You can now disable certain resource types to ensure that the Apicurio Registry Operator does not create or manage those resources, so that you can configure them manually. Manual configuration provides you with greater flexibility in using features that the Apicurio Registry Operator does not currently support. If you disable a resource type, its existing instance is deleted. If you enable a resource type, the Apicurio Registry Operator attempts to find a resource using the app label (for example, `app=example-apicurioregistry`) and start managing it. Otherwise, the Apicurio Registry Operator creates a new instance. You can disable the following resource types in this way:

- Ingress
- NetworkPolicy
- PodDisruptionBudget

#### Improved log level configuration

You can now set the log level for Apicurio Registry and the Apicurio Registry Operator more easily, by using the `spec.configuration.registryLogLevel` field in the `ApicurioRegistry` CRD file. This new field sets the log level for Apicurio application components (excludes non-Apicurio components and libraries), in contrast with the `spec.configuration.logLevel` field, which works for non-Apicurio components and libraries. You can now also set the log level for the Apicurio Registry Operator, by setting the `LOG_LEVEL` environment variable in the Operator `Deployment` resource. Valid `LOG_LEVEL` values are `debug`, `info`, `warn`, and `error`.

#### CORS allowed origins

Servers can use the Cross-Origin Resource Sharing (CORS) mechanism to control whether a response can be shared with the origin of the request. The Apicurio Registry Operator now sets the `CORS_ALLOWED_ORIGINS` environment variable, based on the value of the `spec.deployment.host` and `spec.configuration.security.keycloak.url` fields in the `ApicurioRegistry` CRD file. This environment variable controls the `Access-Control-Allow-Origin` header sent by Apicurio Registry. If you use a custom `Ingress` (for example, to configure HTTPS), you can disable the Operator-managed `Ingress` by setting the `spec.deployment.managedResources.disableIngress` field to `true`, and still set the `spec.deployment.host` field to the appropriate value. If you want to configure a
fully customized CORS policy, you can use the `spec.deployment.env` field to set the `CORS_ALLOWED_ORIGINS` environment variable manually.

Configuring Apicurio Registry deployment by using a pod template

The ApicurioRegistry CRD file now contains the `spec.deployment.podTemplateSpecPreview` field as a Technology Preview feature. The `spec.deployment.podTemplateSpecPreview` field has the same structure as the `spec.template` field in an OpenShift Deployment resource (the `PodTemplateSpec` struct). With some restrictions, the Apicurio Registry Operator forwards the data from this field to the corresponding field in the Apicurio Registry Deployment resource. This feature provides greater configuration flexibility, without requiring the Apicurio Registry Operator to natively support each use case.

**IMPORTANT**

Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope.

Apicurio Registry user documentation and examples

The documentation library has been updated with the new features available in version 2.4:

- Installing and deploying Red Hat build of Apicurio Registry on OpenShift
- Migrating Red Hat build of Apicurio Registry deployments
- Red Hat build of Apicurio Registry User Guide
- Apicurio Registry v2 core REST API documentation

The open source demonstration applications have also been updated:

- https://github.com/Apicurio/apicurio-registry-examples

1.4. APICURIO REGISTRY DEPRECATED FEATURES

Apicurio Registry core deprecated features

- **Confluent Schema Registry API version 6 (compatibility API)**: Apicurio Registry currently supports two versions of the Confluent Schema Registry API on separate endpoints: version 6 and version 7. Use of version 6 is deprecated; the v6 API endpoint will be removed in a future release. Ensure that you replace all references to the v6 endpoint with references to the v7 API endpoint.

- **Apicurio Registry Core API version 1**: Apicurio Registry support for the original version of the Apicurio Registry Core API (version 1) is now deprecated; the legacy API will be removed in the next major release.
Dynamic log level configuration: The /admin/loggers and /admin/loggers/{logger} API endpoints are now deprecated in the Apicurio Registry Core API (v2). These endpoints will be removed in a future release.

Registry V1 export utility: Apicurio Registry support for the command-line export utility is now deprecated. The export tool, which is used to export data from Apicurio Registry 1.x into a format that can be imported into 2.x, will no longer be released or maintained. All customers should have already upgraded from 1.x to 2.x.

Apicurio Registry Operator deprecated features

- Setting environment variables by editing the Deployment resource: In previous versions, you could set environment variables for Apicurio Registry by directly editing its Deployment resource, which was supported by the Apicurio Registry Operator. Now that you can manage environment variables by using the spec.configuration.env field in the ApicurioRegistry CRD file, the previous procedure is deprecated and the Operator support for it will be removed. Ensure that you use the spec.configuration.env field to set all environment variables that are not set by the Operator.

- Retention of environment variables for features that are not enabled: The Apicurio Registry Operator sets environment variables to enable and configure various features, such as Salted Challenge Response Authentication Mechanism (SCRAM) security when using KafkaSQL storage. When such features are disabled, the Operator currently retains the associated environment variables, which can cause problems. Retention of such environment variables is deprecated, and the Operator support for it will be removed. Ensure that your deployment does not rely on the retention of such environment variables.

- Environment variable precedence: The Apicurio Registry Operator might attempt to set an environment variable that is already explicitly specified in the spec.configuration.env field. If an environment variable has a conflicting value, the value set by the Apicurio Registry Operator takes precedence by default. This behavior will change in the future, to enable users to overwrite most environment variables set by the Operator. Ensure that your deployment does not rely on the original precedence behavior.

1.5. UPGRADING AND MIGRATING APICURIO REGISTRY DEPLOYMENTS

You can upgrade the Apicurio Registry server automatically from Apicurio Registry 2.x to Apicurio Registry 2.4 on OpenShift. There is no automatic upgrade from Apicurio Registry 1.x to Apicurio Registry 2.x, and a migration process is required.

1.5.1. Updating 2.x client dependencies

It is not mandatory to update client dependencies for this release; existing 2.x clients still work with Apicurio Registry 2.4.

However, before the next release of Apicurio Registry, you must update all of your client application dependencies to use the latest Apicurio Registry client version. Client application dependencies include dependencies for Kafka serializers/deserializers (SerDes), Maven plug-in, and Java REST client. For example, to update the Maven dependencies for a Java REST client, specify the version in your pom.xml file as follows:

```xml
<dependency>
  <groupId>io.apicurio</groupId>
  <artifactId>apicurio-registry-client</artifactId>
  <version>2.4.x</version>
</dependency>
```
1.5.2. Upgrading from Apicurio Registry 2.x on OpenShift

You can upgrade from Apicurio Registry 2.x on OpenShift 4.9 to Apicurio Registry 2.4 on OpenShift 4.10 or later. You must upgrade both your Apicurio Registry and your OpenShift versions, and upgrade OpenShift one minor version at a time.

Prerequisites

- You already have Apicurio Registry 2.x installed on OpenShift 4.9.

Procedure

1. In the OpenShift Container Platform web console, click Administration and then Cluster Settings.

2. Click the pencil icon next to the Channel field, and select the next minor candidate version (for example, change from stable-4.9 to candidate-4.10).

3. Click Save and then Update, and wait until the upgrade is complete.

4. If the OpenShift version is less than 4.11, repeat steps 2 and 3, and select candidate-4.11 or later.

5. Click Operators > Installed Operators > Red Hat Integration - Service Registry

6. Ensure that the Update channel is set to 2.x.

7. If the Update approval is set to Automatic, the upgrade should be approved and installed immediately after the 2.x channel is set.

8. If the Update approval is set to Manual, click Install.

9. Wait until the Operator is deployed and the Apicurio Registry pod is deployed.

10. Verify that your Apicurio Registry system is up and running.

Additional resources

- For more details on how to set the Operator update channel in the OpenShift Container Platform web console, see Changing the update channel for an Operator.

1.5.3. Migrating from Apicurio Registry 1.1 on OpenShift

For details on migrating from Apicurio Registry 1.1 to Apicurio Registry 2.x, see Migrating Red Hat build of Apicurio Registry deployments.

1.6. APICURIO REGISTRY RESOLVED ISSUES

Table 1.1. Apicurio Registry resolved issues in version 2.4.4
<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registry-3496</td>
<td>Handle Apache Avro complex types in SerDes.</td>
</tr>
<tr>
<td>Registry-3466</td>
<td>Correctly handle non-snake case query parameters in the Python SDK.</td>
</tr>
<tr>
<td>Registry-3458</td>
<td>Fix dynamic basic authentication web console resolution.</td>
</tr>
<tr>
<td>Registry-3315</td>
<td>Registry uses outdated Keycloak context path.</td>
</tr>
</tbody>
</table>

Table 1.2. Apicurio Registry resolved issues in version 2.4.3

<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registry-3307</td>
<td>Enabling Anonymous read access and Artifact owner-only authorization in web console fails with error.</td>
</tr>
<tr>
<td>Registry-3260</td>
<td>Key password not optional when configuring KafkaSQL storage to use SSL.</td>
</tr>
<tr>
<td>Registry-3184</td>
<td>JSON schema compatibility check fails for multipleOf comparison.</td>
</tr>
<tr>
<td>Registry-3143</td>
<td>DownloadReaper error when deploying Apicurio Registry.</td>
</tr>
<tr>
<td>Registry-3096</td>
<td>SearchedGroup.createdBy property should have type String not Object.</td>
</tr>
<tr>
<td>Registry-3088</td>
<td>Serialization error in enum artifact reference and record ID strategy.</td>
</tr>
<tr>
<td>Registry-3086</td>
<td>Cannot upload JSON schemas that reference other schemas with $ref.</td>
</tr>
<tr>
<td>Registry-3080</td>
<td>Compatibility check does not throw exception when empty schema is provided.</td>
</tr>
<tr>
<td>Registry-3014</td>
<td>Upload fails with 422 error when schema has default value of {}.</td>
</tr>
<tr>
<td>Registry-2991</td>
<td>Kafka consumer thread starts before internal database is ready.</td>
</tr>
<tr>
<td>Registry-2955</td>
<td>Redirect filter configured incorrectly.</td>
</tr>
<tr>
<td>Registry-2952</td>
<td>Compatibility rules do not order by version number.</td>
</tr>
<tr>
<td>Registry-2938</td>
<td>SSL configuration does not work unless SASL is enabled.</td>
</tr>
<tr>
<td>Registry-2902</td>
<td>’registry.ccompat.use-canonical-hash’ setting changes schema to canonized form on save.</td>
</tr>
<tr>
<td>Registry-2880</td>
<td>Exception not thrown when wrong credentials passed to auth server.</td>
</tr>
<tr>
<td>Registry-2877</td>
<td>Uploading new version of Protobuf schema fails with NullPointerException.</td>
</tr>
</tbody>
</table>
Apicurio Registry Operator resolved issues

Table 1.3. Apicurio Registry Operator resolved issues in version 2.4.4

<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPT-960</td>
<td>CORS_ALLOWED_ORIGINS is not propagated as environment variable in the resulting OpenShift pod.</td>
</tr>
</tbody>
</table>

Table 1.4. Apicurio Registry Operator resolved issues in version 2.4.3

<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPT-916</td>
<td>Port name not provided in service created by Operator.</td>
</tr>
<tr>
<td>IPT-883</td>
<td>Missing environment variables after deployment destroyed.</td>
</tr>
<tr>
<td>IPT-852</td>
<td>Incorrect method documented for managing environment variables.</td>
</tr>
<tr>
<td>Operator-119</td>
<td>Log level not set correctly.</td>
</tr>
</tbody>
</table>

1.7. APICURIO REGISTRY RESOLVED CVES

The resolved Common Vulnerabilities and Exposures (CVEs) in Apicurio Registry 2.4 are as follows:

Table 1.5. Apicurio Registry resolved CVEs in version 2.4.4.SP1

<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPT-1021</td>
<td>CVE-2023-44487 undertow: HTTP/2: Multiple HTTP/2 enabled web servers are vulnerable to a DDoS attack (Rapid Reset Attack).</td>
</tr>
<tr>
<td>IPT-1020</td>
<td>CVE-2023-44487 netty-codec-http2: HTTP/2: Multiple HTTP/2 enabled web servers are vulnerable to a DDoS attack (Rapid Reset Attack).</td>
</tr>
<tr>
<td>IPT-1019</td>
<td>CVE-2023-39325 CVE-2023-44487 integration-service-registry-operator-container: various flaws.</td>
</tr>
</tbody>
</table>

Table 1.6. Apicurio Registry resolved CVEs in version 2.4.4
<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
</tr>
</thead>
</table>

Table 1.7. Apicurio Registry resolved CVEs in version 2.4.3

<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPT-943</td>
<td>CVE-2023-2976 guava: Insecure temporary directory creation.</td>
</tr>
<tr>
<td>IPT-890</td>
<td>CVE-2021-46877 jackson-databind: Possible DoS if using JDK serialization to serialize JsonNode.</td>
</tr>
<tr>
<td>IPT-880</td>
<td>CVE-2022-3510 protobuf-java: Message-Type Extensions parsing issue leads to DoS.</td>
</tr>
<tr>
<td>IPT-879</td>
<td>CVE-2022-3509 protobuf-java: Textformat parsing issue leads to DoS.</td>
</tr>
<tr>
<td>IPT-876</td>
<td>CVE-2023-28867 graphql-java: crafted GraphQL query causes stack consumption.</td>
</tr>
<tr>
<td>IPT-856</td>
<td>CVE-2022-3782 keycloak: path traversal via double URL encoding.</td>
</tr>
<tr>
<td>IPT-825</td>
<td>CVE-2022-40152 woodstox-core: woodstox to serialise XML data was vulnerable to Denial of Service (DoS) attacks.</td>
</tr>
</tbody>
</table>

1.8. APICURIO REGISTRY KNOWN ISSUES

The following known issues apply in Apicurio Registry 2.4.x:

Apicurio Registry core known issues

Registry-3413 - Legacy REST API date formats enabled by default

For maximum compatibility and for easier upgrades from older versions of Apicurio Registry, the date format used in the REST API is not compliant with OpenAPI standards (because of a bug in older versions).

Before the next release of Apicurio Registry, you must upgrade all of your client applications to use the latest client version. The next release will fix the date format bug, which will result in older clients no longer being compatible with the REST API.

To update your REST API to be OpenAPI compliant, you can fix the date format bug in this version of Apicurio Registry as follows:
1. Update all of your client applications to version **2.4.4.SP1-redhat-00001**, as described in [Updating 2.x client dependencies](#).

2. Set the following environment variable to the value shown:

   ```
   REGISTRY_APIs_V2_DATE_FORMAT=yyyy-MM-dd'T'HH:mm:ss'Z'
   ```

**IPT-814** - Apicurio Registry logout feature incompatible with RH-SSO 7.6

In RH-SSO 7.6, the `redirect_uri` parameter used with the logout endpoint is deprecated. For more details, see the [RH-SSO 7.6 Upgrading Guide](#). Because of this deprecation, when Apicurio Registry is secured by using the RH-SSO Operator, clicking the **Logout** button displays the **Invalid parameter: redirect_uri** error.

For a workaround, see [https://access.redhat.com/solutions/6980926](https://access.redhat.com/solutions/6980926).

**IPT-701** - CVE-2022-23221 H2 allows loading custom classes from remote servers through JNDI

When Apicurio Registry data is stored in AMQ Streams, the H2 database console allows remote attackers to execute arbitrary code by using the JDBC URL. Apicurio Registry is not vulnerable by default and a malicious configuration change is required.

**Apicurio Registry Operator known issues**

**Operator-42** - Autogeneration of OpenShift route might use wrong base host value

If multiple `routerCanonicalHostname` values are specified, autogeneration of the Apicurio Registry OpenShift route might use a wrong base host value.
Apicurio Registry is provided through a software subscription. To manage your subscriptions, access your account at the Red Hat Customer Portal.

Accessing your account

1. Go to access.redhat.com.
2. If you do not already have an account, create one.
3. Log in to your account.

Activating a subscription

1. Go to access.redhat.com.
2. Navigate to My Subscriptions.
3. Navigate to Activate a subscription and enter your 16-digit activation number.

Downloading ZIP and TAR files
To access ZIP or TAR files, use the customer portal to find the relevant files for download. If you are using RPM packages, this step is not required.

1. Open a browser and log in to the Red Hat Customer Portal Product Downloads page at access.redhat.com/downloads.
2. Locate the Red Hat Integration entries in the Integration and Automation category.
3. Select the desired Apicurio Registry product. The Software Downloads page opens.
4. Click the Download link for your component.

Revised on 2023-11-07 12:54:51 UTC