



# Red Hat build of Apicurio Registry 2.5

## 迁移 Apicurio Registry 部署

从 Apicurio Registry 版本 1.1 迁移到 2.5



## Red Hat build of Apicurio Registry 2.5 迁移 Apicurio Registry 部署

---

从 Apicurio Registry 版本 1.1 迁移到 2.5

## 法律通告

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 摘要

本指南描述了 Apicurio Registry 版本 2.x 的主要更改，并解释了如何将现有 Apicurio Registry 版本 1.1 部署迁移到版本 2.5。

---

# 目录

<b>前言</b> .....	<b>3</b>
使开源包含更多	3
对红帽文档提供反馈	3
<b>第 1 章 从 APICURIO REGISTRY 1.1 迁移到 2.X</b> .....	<b>4</b>
1.1. 新数据存储选项	4
1.2. 新的 V2 REST API	4
1.3. 重构的 JAVA 客户端库	4
<b>第 2 章 迁移 APICURIO REGISTRY 数据</b> .....	<b>5</b>
<b>第 3 章 迁移 APICURIO REGISTRY 客户端应用程序</b> .....	<b>6</b>
<b>附录 A. 使用您的订阅</b> .....	<b>8</b>
访问您的帐户	8
激活订阅	8
下载 ZIP 和 TAR 文件	8



# 前言

## 使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。详情请查看 [CTO Chris Wright 的信息](#)。

## 对红帽文档提供反馈

我们感谢您对我们文档的反馈。

要改进，创建一个 JIRA 问题并描述您推荐的更改。提供尽可能多的详细信息，以便我们快速解决您的请求。

### 前提条件

- 您有一个红帽客户门户网站帐户。此帐户可让您登录到 Red Hat Jira Software 实例。如果您没有帐户，系统会提示您创建一个帐户。

### 流程

1. 单击以下链接：[创建问题](#)。
2. 在 **Summary** 文本框中输入问题的简短描述。
3. 在 **Description** 文本框中提供以下信息：
  - 找到此问题的页面的 URL。
  - 有关此问题的详细描述。  
您可以将信息保留在任何其他字段中的默认值。
4. 点 **Create** 将 JIRA 问题提交到文档团队。

感谢您花时间来提供反馈。

## 第 1 章 从 APICURIO REGISTRY 1.1 迁移到 2.X

Apicurio Registry 2.x 包括了与之前 Apicurio Registry 1.1 发行版本有变化的新功能。本节论述了 Apicurio Registry 1.1 和版本 2.x 之间的主要变化。

由于 2.x 中的具有破坏性的变化，因此没有自动升级，需要执行一个迁移过程。这包括将现有 registry 中的数据移到新 registry 中。您还必须查看现有的 registry 客户端应用程序并更新其配置以满足新要求。

当迁移到 2.x 版本时，您必须考虑以下主要更改：

### 1.1. 新数据存储选项

Apicurio Registry 1.1 中的现有 registry 数据存储选项(**流**、**jpa** 和 **infinispan**)已被版本 2.x 中的新存储选项(**sql** 和 **kafkasql**)替代。这些新存储选项提供更强大的、性能和维护的 Apicurio Registry 部署。

有关如何使用您选择的存储选项部署 Apicurio Registry 2.x 的详情，请参阅在 [OpenShift 上安装和部署红帽构建的 Apicurio Registry](#)。

### 1.2. 新的 V2 REST API

Apicurio Registry 2.x 包含一个新的 REST API，支持工件组，并改进了长期可维护性。Apicurio Registry 仍然支持原始 registry v1 REST API 和兼容性 API，如 Confluent 和 IBM schema registry API。Apicurio Registry 现在实现了 CNCF 云事件规格中提供的 Schema Registry 规格。

### 1.3. 重构的 JAVA 客户端库

- Apicurio Registry Java 客户端类在版本 2.x 中提供，名为 **apicurio-registry-client** 的不同 Maven 模块。
- Kafka 客户端序列化器和反序列化器(SerDes)类在三个不同的 Maven 模块中包括在版本 2.x 中，每个支持的数据格式都有一个：Apache Avro、Protobuf 和 JSON Schema。现在，您只能在不拉取的、不涉及的依赖软件包的情况下使用您想要的模块。

#### 其他资源

- 有关 v2 REST API 的详情，请参阅 [Registry REST API 文档](#)。



## 第 2 章 迁移 APICURIO REGISTRY 数据

将数据迁移到 Apicurio Registry 2.x 需要从现有 1.1 部署导出所有数据，并将其导入到新的 2.x 部署中。如果您使用 Apicurio Registry 作为 Kafka 应用程序的模式 registry，则数据迁移至关重要，因为每个 Kafka 消息传输存储在 Apicurio Registry 中的 schema 的全局标识符。此标识符必须在 registry 数据迁移过程中保留。

Apicurio Registry 2.x 提供了一个 API，用于批量导入/导出 registry 部署中的所有数据，这样可保证从现有 registry 导入数据时保留所有标识符。导出 API 下载自定义 **.zip** 文件，其中包含您的工件的所有信息。导入 API 接受此 **.zip**，并将所有工件加载到单个批处理中的 registry 中。

Apicurio Registry 1.1 不提供导入/导出 API。但是，版本 2.x 提供与 Apicurio Registry 1.1 兼容的导出工具，以导出 **.zip**，您可以导入到 2.x registry。此工具使用通用现有 API 来导出 registry 中的所有内容。但是，它的性能小于 2.x 导出 API，并且仅在从 1.1 registry 导出时使用。

### 前提条件

- 运行您要从中导出的 1.1 服务器的 Apicurio Registry 实例，以及您要导入的 2.x 服务器。
- 从红帽客户门户网站下载 [Apicurio Registry exportV1 工具](#)。这是您可以在命令行中运行的 Java 应用程序。

### 流程

1. 使用 **exportV1** 工具导出 Apicurio Registry 1.1 中的所有数据。这会在当前目录中生成 **registry-export.zip** 文件：

```
java -jar apicurio-registry-utils-exportV1-2.5.10.Final-redhat-00001.jar http://old-registry.my-company.com/api
```

2. 使用导入 API 将 **.zip** 文件导入到 Apicurio Registry 2.x 中：

```
curl -X POST "http://new-registry.my-company.com/apis/registry/v2/admin/import" \  
-H "Accept: application/json" -H "Content-Type: application/zip" \  
--data-binary @registry-export.zip
```

3. 运行以下命令并比较 count 字段，检查所有工件是否已导入到新的 2.x registry 中：

```
curl "http://old-registry.my-company.com/api/search/artifacts"
```

```
curl "http://new-registry.my-company.com/apis/registry/v2/search/artifacts"
```

### 其他资源

- 有关导入/导出 REST API 的详情，请查看 [红帽构建的 Apicurio Registry 用户指南](#)。
- 有关从版本 1.x 迁移到 2.x 的导出工具的详情，请参阅 [1.x 版本的 Apicurio Registry 导出工具](#)。

## 第 3 章 迁移 APICURIO REGISTRY 客户端应用程序

您必须检查现有的 Apicurio Registry 客户端应用程序，以确保 Maven 依赖项和 Java 客户端配置满足版本 2.x 的新要求。例如，这包括 Apicurio Registry Java REST 客户端库或 Kafka 客户端序列化器(Serdes)库的新 Maven 依赖项。您还必须使用新的 registry v2 API 路径更新 Java 应用程序配置。

### 前提条件

- 使用 SerDe 现有的 Apicurio Registry 1.1 Java 客户端应用程序或 Kafka 客户端生成者和消费者 Java 应用程序

### 流程

1. 如果使用 Apicurio Registry Java REST 客户端，您必须更改 Apicurio Registry Java 客户端库的 Maven 依赖项，该库已在版本 2.x 中重新打包：

```
<dependency>
  <groupId>io.apicurio</groupId>
  <artifactId>apicurio-registry-client</artifactId>
  <version>2.5.10.Final-redhat-00001</version>
</dependency>
```

2. 在 Java 客户端应用程序中，您必须更改 registry URL 配置，从指向现有的 v1 API 路径到新的 v2 路径。例如：

```
public class ClientExample {

    private static final RegistryRestClient client;

    public static void main(String[] args) throws Exception {
        // Create a registry client
        String registryUrl = "https://new-registry.my-company.com/apis/registry/v2";
        RegistryClient client = RegistryClientFactory.create(registryUrl);
    }
}
```

您可以在 [红帽构建的 Apicurio Registry 用户指南](#) 中找到有关 Java 客户端的更多详细信息。

3. 如果使用 Apicurio Registry SerDes 库，您必须更改 Maven 依赖项，这些依赖项已在版本 2.x 中重新打包。在 Apicurio Registry 1.1 中，SerDes 库只由一个 Maven 依赖项提供：

```
<dependency>
  <groupId>io.apicurio</groupId>
  <artifactId>apicurio-registry-utils-serde</artifactId>
  <version>1.3.2.Final-redhat-00002</version>
</dependency>
```

在 Apicurio Registry 2.x 中，SerDes 库被分成三个 Maven 依赖项，每个支持的数据格式都有一个：**avro**、**protobuf** 和 **json** 模式，具体取决于您的用例：

```
<dependency>
  <groupId>io.apicurio</groupId>
  <artifactId>apicurio-registry-serdes-avro-serde</artifactId>
  <version>2.5.10.Final-redhat-00001</version>
```

```

</dependency>
<dependency>
  <groupId>io.apicurio</groupId>
  <artifactId>apicurio-registry-serdes-protobuf-serde</artifactId>
  <version>2.5.10.Final-redhat-00001</version>
</dependency>
<dependency>
  <groupId>io.apicurio</groupId>
  <artifactId>apicurio-registry-serdes-jsonschema-serde</artifactId>
  <version>2.5.10.Final-redhat-00001</version>
</dependency>

```

4. 在 Kafka 生成者和消费者 Java 应用程序中，您必须更改 registry URL 配置，将其从指向现有的 v1 API 路径改为指向新的 v2 路径。例如：

*现有 registry v1 API 路径：*

```

props.putIfAbsent(AbstractKafkaSerDe.REGISTRY_URL_CONFIG_PARAM, "http://old-registry.my-company.com/api");

```

*新 registry v2 API 路径：*

```

props.putIfAbsent(SerdeConfig.REGISTRY_URL, "http://new-registry.my-company.com/apis/registry/v2");

```

重构的 SerDes 库还包括其他对配置属性的重要更改。如需有关 SerDes 配置的更多详细信息，请参阅 [红帽构建的 Apicurio Registry 用户指南](#)。

## 其他资源

- 有关详细配置示例，请参阅 [Apicurio Registry 示例应用程序](#)。

## 附录 A. 使用您的订阅

Apicurio Registry 通过软件订阅提供。要管理您的订阅，请访问红帽客户门户中的帐户。

### 访问您的帐户

1. 转至 [access.redhat.com](https://access.redhat.com)。
2. 如果您还没有帐户，请创建一个帐户。
3. 登录到您的帐户。

### 激活订阅

1. 转至 [access.redhat.com](https://access.redhat.com)。
2. 导航到 **My Subscriptions**。
3. 导航到 **激活订阅** 并输入您的 16 位激活号。

### 下载 ZIP 和 TAR 文件

要访问 ZIP 或 TAR 文件，请使用客户门户网站查找下载的相关文件。如果您使用 RPM 软件包，则不需要这一步。

1. 打开浏览器并登录红帽客户门户网站 **产品下载页面**，网址为 [access.redhat.com/downloads](https://access.redhat.com/downloads)。
2. 在 **Integration** 和 **Automation** 类别中找到 **Red Hat Integration** 条目。
3. 选择所需的 Apicurio Registry 产品。此时会打开 **Software Downloads** 页面。
4. 单击组件的 **Download** 链接。

更新于 2024-05-15