Hortonworks DataFlow

Schema Registry User Guide

(November 9, 2017)

docs.hortonworks.com

Hortonworks DataFlow: Schema Registry User Guide

Table of Contents

1. Integrating Schema Registry

1.1. Integrating with NiFi

1.1.1. Understanding NiFi Record Based Processing

The RecordReader and RecordWriter Controller Services and Processors that allow you convert events from one type (json, xml, csv, Avro) to another (json, xml, csv, Avro). These controller services use the Schema Registry to fetch the schema for the event to do this conversion.

NiFi includes the following RecordReader and RecordWriter processors:

- ConsumeKafkaRecord_0_10 1.2.0
- ConvertRecord
- PublishKafkaRecord_0_10
- PutDatabaseRecord
- QueryRecord
- SplitRecord

NiFi also includes the following Record based Controller Services:

- HortonworksSchemaRegistry
- AvroRecordSetWriter
- CSVRecordSetWriter
- FreeFormTextRecordSetWriter
- JsonRecordSetWriter
- ScriptedRecordSetWriter

More Information

HCC Article on Apache NiFi record based processing

1.1.2. Setting up the HortonworksSchemaRegistry Controller Service

About This Task

To configure Schema Registry to communicate with NiFi dataflows, the first thing you must do is tell NiFi about the Schema Registry instance with which you want to communicate. You do this from the NiFi UI, using the HortonworksSchemaRegistry Controller Service.

Prerequisites

You have already installed Schema Registry.

Steps

1. From the Global Menu, click **Controller Settings** and select **Controller Services** tab.
2. Click the + icon to display the **Add Controller Service** dialog.
3. Use the Filter box to search for HortonworksSchemaRegistry and click **Add**.
4. Click the Edit icon to display the **Configure Controller Service** dialog.
5. Provide the Schema Registry URL with which you want NiFi to communicate and click **Apply**.

Tip

If you are running an Ambari-managed cluster, you can find this value in the Streaming Analytics Manager Service in Ambari for the configuration property called `registry.url`. The URL looks similar to `http://$REGISTRY_SERVER:7788/api/v1`.

6. Enable this HortonworksSchemaRegistry by clicking the Enable icon, selecting the **Scope**, and clicking **Enable**.

1.1.3. Adding and Configuring Record Reader and Writer Controller Services

About This Task

NiFi provides Record Reader and Writer Controller Services to support record-based processing. These Controller Services are new services that allows you convert events from one type (JSON, XML, CSV, Avro) to another. These Controller Services use the Schema Registry to fetch the schema for the event to do this conversion. Before using these new Controller Services, you must configure them for use with Schema Registry.

You can configure Controller Services either globally, before you have created a Process Group, or at any time, on a per-Process Group basis.

Steps for Adding Controller Services globally

1. To access Controller Services configuration dialog for gobal configuration, click the **Global Menu** at the top right of your canvas, and select **Controller Settings**.
2. Click the + icon to display the **NiFi Settings** dialog.
3. Use the **Filter** box to search for the Controller Service you want to add, select that service, and click **Add**.

Steps for Adding Controller Services Per Process Group

1. Click on your Process Group, and then right-click anyway on your canvas.
2. Click **Configure** to display the **Process Group Configuration** dialog.
3. Click the **Controller Services** tab, and then click + to display the **Add Controller Service** dialog.
4. Use the **Filter** box to search for the Controller Service you want to add, select that service, and click **Add**.

Steps for Configuring Record Reader and Writer Controller Services for Integration with Schema Registry

1. From the **Process Group Configuration** view, click the **Edit** icon from the right-hand column. This displays the **Configure Controller Service** dialog.
2. Click the **Properties** tab.
3. The Schema Access Strategy specifies how to obtain the schema using for interpreting FlowFile data. To ensure integration with Schema Registry, configure Schema Access Strategy with one of the following two values:
   - HWX Schema Reference Attributes – The NiFi FlowFile is given a set of 3 attributes to describe the schema:
     - schema.identifier
     - schema.version
     - schema.protocol.version
   - HWX Content-Encoded Schema Reference – Each NiFi FlowFile contains a reference to a schema stored in Schema Registry. The reference is encoded as a single byte indicating the protocol version, 8 bytes indicating the schema identifier and 4 bytes indicating the schema version.
4. The Schema Write Strategy specifies how the schema for a record should be added to FlowFile data. To ensure integration with Schema Registry, configure Schema Write Strategy with either HWX Schema Reference Attributes orHWX Content-Encoded Schea Reference.

1.1.4. Using Record-Enabled Processors

About This Task

Record-enabled Processors allow you to use convert data between formats by specifying Controller Services for record reading and record writing. This streamlines your dataflows and inproves overall performance.

Steps

1. From the NiFI UI, drag the Processor icon onto your canvas to display the **Add Processor** dialog.

2. Use the **Filter** box to find the Processor you want to add. Available record-enabled processors are:
   - ConsumeKafkaRecord_0_10
   - ConvertRecord
   - PublishKafkaRecord_0_10
   - PutDatabaseRecord
   - QueryRecord
   - SplitRecord
3. Select the Processor you want, and click **Add**.
4. Right-click the Processor on the canvas, and select **Configure** to display the **Configure Processor** dialog.
5. Click the **Properties** tab and select a Controller Service value for the **Record Reader** and **Record Writer** values.
6. Click **OK** and then **Apply**.

1.2. Integrating with Kafka

You can integrate Schema Registry with Kafka in one of two ways, depending on your use case.

1.2.1. Integrating Kafka and Schema Registry Using NiFi Processors

About This Task

If you are using an Ambari-managed HDF cluster with Schema Registry, NiFi, and Kafka installed, you can use NiFi Processors to integrate Schema Registry with Kafka.

Steps

1. Integrate NiFi with Schema Registry.
2. Build your NiFi dataflow.
3. At the point in your dataflow where you want to either consume from a Kafka topic, or publish to a Kafka topic, add one of the following two Processors:
   - ConsumeKafkaRecord_0_10
   - PublishKafkaRecord_0_10
4. Configure your Kafka Processor with the following information:

- Kafka Brokers – Provide a comma-seperated list of Kafka Brokers you want to use in your dataflow.
- Topic Name – The name of the Kafka topic to which you want to publish or from which you want to consume data.
- Record Reader – Provide the Controller Service you want to use to read incoming FlowFile records.
- Record Writer – Provide the Controller Service you want to use to serialize record data before sending it to Kafka.

1.2.2. Integrating Kafka and Schema Registry

About This Task

If you are running HDF without NiFi, integrate your Kafka Producer and Consumer manually. To do this you must add a dependency on the Schema Registry Serdes, and update the Kafka Producer and Kafka Consumer configuration files.

Steps to Add a Schema Registry Serdes Dependency

1. Add the following text to `schema-registry-serdes`:

```
<dependency>
      <groupId>com.hortonworks.registries</groupId>
      <artifactId>schema-registry-serdes</artifactId>
</dependency>
```

Steps to Integrate the Kafka Producer

1. Add the following text to the Kafka Producer configuration:

```
config.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, bootstrapServers);
config.putAll(Collections.singletonMap(SchemaRegistryClient.Configuration.
SCHEMA_REGISTRY_URL.name(), props.get(SCHEMA_REGISTRY_URL)));
config.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, StringSerializer.
class.getName());
config.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG,
 KafkaAvroSerializer.class.getName());
```

2. Edit the above text with values for the following properties:
   - `schema.registry.url`
   - `key.serializer`
   - `value.serializer`

Steps to Integrate the Kafka Consumer

1. Add the following text to the Kafka Consumer configuration:

```
config.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG, bootstrapServers);
config.putAll(Collections.singletonMap(SchemaRegistryClient.Configuration.
SCHEMA_REGISTRY_URL.name(), props.get(SCHEMA_REGISTRY_URL)));
```

```
config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.
class.getName());
config.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG,
 KafkaAvroDeserializer.class.getName());
```

2. Edit the above text with values for the following properties:

   - `schema.registry.url`
   - `key.deserializer`
   - `value.deserializer`

1.3. Integrating with Stream Analytics Manager

About This Task

Integrating with Stream Analytics Manager (SAM) is primarily a task you perform on SAM. Perform the integration using the SAM Ambari Configs, either during installation or at any point afterwards.

Steps for Integrating During Installation

1. In the **Customize Services** step, navigate to the **STREAMLINE CONFIG** section of the **Streaming Analytics Manager** tab.
2. Configure **registry.url** to the REST API Endpoint URL for the Registry. The format should be `http://$FQDN_REGISTRY_HOST:$REGISTRY_PORT/api/v1`, where:

   - $FQDN_REGISTRY_HOST – Specifies the host on which you are running Schema Registry
   - $REGISTRY_PORT – Specifies the Schema Registry port number. You can find the Schema Registry port in the **REGISTRY_CONFIG** section of the **Registry** tab.

   For example: `http://FQDN_REGISTRY_HOST:7788/api/v1`

Steps for Integrating After Installation

1. From **Services** pane on the left hand side of Ambari, click **Streaming Analytics Manager** and then click the **Configs** tab.
2. In the **STREAMLINE CONFIG** tab, configure **registry.url** to the REST API Endpoint URL for the Registry. The format should be `http://$FQDN_REGISTRY_HOST:$REGISTRY_PORT/api/v1`, where:

   - $FQDN_REGISTRY_HOST – Specifies the host on which you are running Schema Registry
   - $REGISTRY_PORT – Specifies the Schema Registry port number. You can find the Schema Registry port in the **REGISTRY_CONFIG** section of the **Registry** tab.

   For example: `http://FQDN_REGISTRY_HOST:7788/api/v1`

2. Using Schema Registry

2.1. Adding a New Schema

About This Task

To add a new schema to Schema Registry, provide information about the schema entities, select a compatibility policy and upload the schema text from a file. Schema entities are the collection of information that help you organize and sort your schemas. They include group, version, and informational metadata.

Prerequisites

- Ensure that you understand compatibility policies. Once selected, you cannot change the compatibility policy for a schema.

Steps

1. From the Schema Registry UI, click the + icon.
2. Add the Schema metadata as follows:
   - Name – A unique name for each schema. Used as a key to look up schemas.
   - Description – A short description of the schema.
   - Schema Type – The schema format.

     

     Note

     Avro is currently the only supported type.
   - Schema Group – Allows you to group schemas in any logical order.
   - Compatibility – Sets the compatibility policy for the schema. Once set, this cannot be changed.
3. To allow schema to evolve over time by creating multiple versions, select the **Evolve** checkbox.

   

   Note

   Deselecting **Evolve** means that you can only have one version of a schema.
4. Click **Choose File** to upload a new schema.

More Information

Schema Entities

Compatibility Policies

2.2. Querying Schemas

You can use the Search box at the top of the Schema Registry UI to search for schemas by name, description, or schema text. To search, you can enter a schema name, key words, or any text string to return schemas with that value.

To return to your full list of schemas, clear the search bar and press Enter on your keyboard.

2.3. Evolving Schema

About This Task

You evolve a schema when you create a new version. Schema Registry tracks the changes made to your schema, and stores each set of changes in a seperate version of the schema. When multiple versions exist, you can select which version you want to use. Ensure that you understand compatibility policies, as they determine how you can evolve your schemas.

Prerequesites

- You have selected the **Evolve** checkbox when initially adding the schema.
- You have the schema you want to evolve saved to a file.

Steps

1. Expand the schema you want to edit and select the schema version from which you want to evolve.
2. Click the pencil icon to open the **Edit Version** dialog.
3. Add a description of what has changed in this new version of the schema. You can view the description in the Schema Registry UI to easily understand what has changed in each version of the the schema so we recommend that you add as much detail as you can.
4. Click **Choose File** to upload the schema you want to evolve.
5. Click **Ok.**

More Information

Compatibility Policies