Lab: Exploring the Projection Engine

Overview

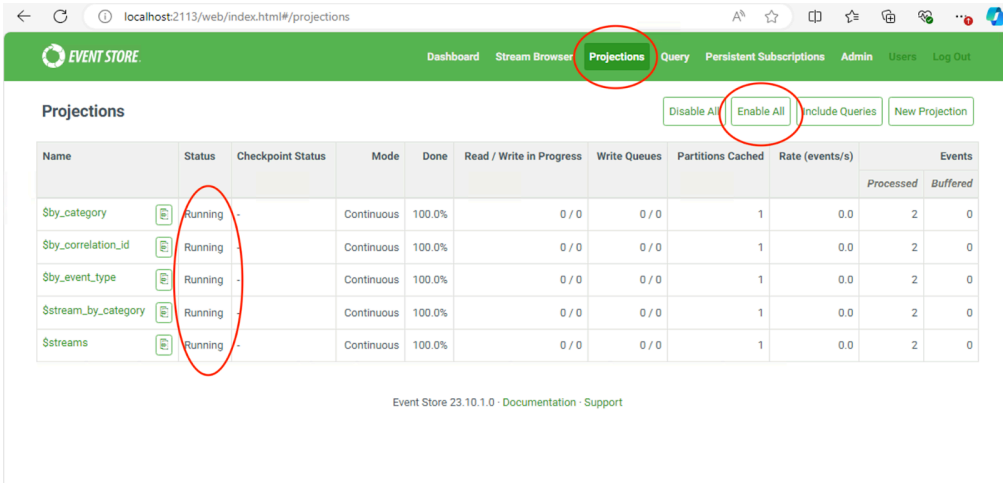
In this lab, you will use the webui to add events that demonstrate the EventStoreDB projection engine.

Notes:

This Lab requires nothing more than a running KurrentDB database and access to the webui. If attempted outside of the context of this course, any EventStoreDB installation with the webui enabled will work, assuming you have credentials for a user with adequate permissions.

Step 1: Verify the Projection Engine is Running

Open the EventStoreDB webui in a browser and verify that the projection engine is running. The relevant sections of the webui are circled in this image.



Step 2: Exploring the “by category” Projection

1. In the stream browser, Select “Add Event.”
2. Add the following three events to demonstrate the $by_category projection.

Event1	Event2	Event3
Stream ID: USA-order1	Stream ID: USA-order2	Stream ID: USA-order3
Event Type: item_sold	Event Type: item_sold	Event Type: item_sold
Data: {“item”: “desk”}	Data: {“item”: “chair”}	Data: {“item”: “table”}

3. View the $ce-usa stream

Any event appended to a stream that starts with “usa-” will be projected into the $ce-usa stream.

Step 3: Exploring the “Event Type” Projection

The Event Type projection was already triggered when we wrote the events in step #2. Each event had the same Event Type, “item_sold.” Those events will be projected into the $et-item_sold stream.

1. View the $et-item_sold stream in the stream browser to verify the events have been projected.

Step 4: Exploring the “Correlation ID” Projection

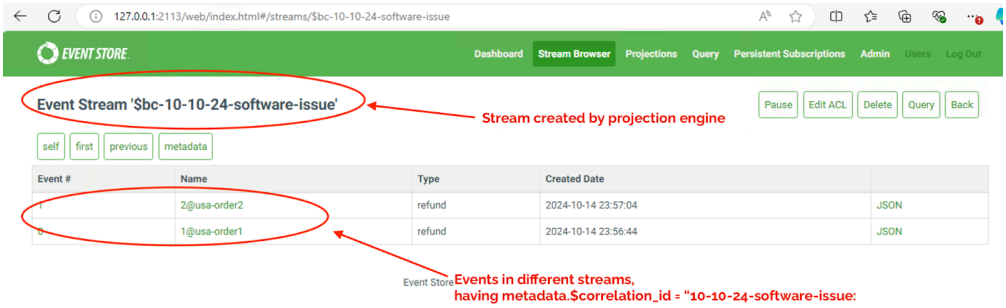
The $by_correlation_id projection is a projection that gathers all events from any stream with the same metadata.$correlation_id value into a stream.

In this step, you will write events with metadata.$correlation_id values.

1. Using the webui, append the following events

Event 1:	Event 2:
Stream ID: usa-order1	Stream ID: usa-order2
Event Type: refund	Event Type: refund
Data: {“refund”:“wrong_item”}	Data:{“refund”:“wrong_item”}
Metadata:{“\(correlationId": "10-10-24-software-issue"} | Metadata:{"\)correlationId”: “10-10-24-software-issue”}	

2. Use the stream browser to verify the presence of these events in the “bc-10-10-24-software-issue” stream that were projected based on the matching $correlation ids in the metadata.



Step 5: Exploring the "Stream By Category" Projection

Given streams with the same "category" where "category" is defined as the string of characters preceding a "-" character in the stream name, there will be a single stream reference event in the Stream by Category projection.

It differs from the "by category" projection in that instead of indexing all events that match the stream category, it provides a single pointer to the stream.

Unfortunately, the stream browser in the webui does not parse the returned event type well enough for it to be useful.

Client code using grpc would be able to render those events.

If you appended two events to the stream usa-order1, and one event to the stream usa-order2, the Stream by Category projection would have two events (a pointer to each stream), and the by category projection would have three events (a pointer to each event).

Step 6: Exploring the Streams Projection

The $streams projection projects all events into the $streams projection.

1. Add an event to a new stream and note the stream's presence in the $streams projection.
2. Add a new event to that stream, and note that you can page through events in the stream browser.

Summary:

Congratulations! You are now familiar with the available system projections in EventStoreDB's projection engine.

Many EventStoreDB users find these projections useful, but you understand they affect throughput. If no projections are in use, a write to EventStoreDB involves a write to the global log, also known as the $all stream, and then a write to the stream index. If projections are used, write amplification occurs as the projection streams are also modified by the write of the event.