

# Starburst Presto JDBC Query Progress Monitor

JDBC applications can use the Query Progress Monitor feature to listen on SQL statement execution progress.

The following interfaces are exposed for use with this feature:

- **IStatementProgressProvider** – An interface implemented by the driver's statement class to expose a method for the calling application to register a listener on the query execution progress. The interface also exposes a method for clearing a listener that had previously been registered.
- **IPRQueryProgressMonitor** – Defines the interface for the listener to register against an IStatementProgressProvider. The interface has an accept method that takes an object which implements the IPRQueryStats interface. The calling JDBC application is responsible for providing an implementation that allows the IPRQueryProgressMonitor interface to work with the IPRQueryStats passed in via the accept method as desired. During query execution the accept method may be called more than once by the driver to pass in the status of an executing query.
- **IPRQueryStats** – A collection of information representing the current progress of an executing query. The driver is responsible for providing implementation for this interface. When the calling JDBC application registers a listener (IPRQueryProgressMonitor) for a statement, the driver deserializes the progress information from the server during query execution into an object implementing the IPRQueryStats interface, and passes that object to the listener via the accept method.
- **IPRStageStats** – A collection of information representing a query execution stage in the Presto server. This is a part of the overall query progress, and an object implementing this interface may be nested inside objects implementing IPRQueryStats and IPRStageStats. The driver is responsible for providing implementation for the IPRStageStats interface.
- **IPRBasicQueryStats** – Captures query progress information that is common between IPRQueryStats and IPRStageStats. Both IPRQueryStats and IPRStageStats extend this interface. The driver is responsible for providing implementation for this interface.

For further information regarding these interfaces and particularly what statistics are available for retrieval, please refer to the javadocs or the interfaces themselves.

Below is a basic code example of how to use this feature:

```
import com.starburst.presto.dataengine.IPRQueryProgressMonitor;
import com.starburst.presto.dataengine.IPRQueryStats;

public class QueryProgresMonitor implements IPRQueryProgressMonitor
{
    @Override
    public void accept(IPRQueryStats statistics)
    {
        // Handle the query statistics as desired.
        System.out.println(statistics.getState());
    }
}

import com.teradata.presto.dataengine.IStatementProgressProvider;

public class TestClass
{
    // Create the Connection object.
    DriverManager.getConnection(
        "jdbc:presto://localhost:8080",
        "user",
        "password")

    // Create the Statement object.
    Statement statement = conn.createStatement();

    // Create the listener object that will implement the accept() method.
    QueryProgresMonitor queryProgressMonitor = new QueryProgresMonitor();

    // Unwrap the java.sql.Statement object to an implementation of
    // IStatementProgressProvider so the listener may be set in the driver.
    // This allows for statistics updates to notify the listener.
    statement.unwrap(IStatementProgressProvider.class)
        .setProgressMonitor(queryProgressMonitor);

    // Execute the query.
    ResultSet result = statement.executeQuery("select * from example_table");
}
```