

JDBC Driver for Hive Query Engine

Learn how to use Treasure Data's [JDBC](#) (Java Database Connectivity) driver, which enables you to use Treasure Data with a standard JDBC interface.

- [Download](#)
- [Usage](#)
- [JDBC Parameter Precedence](#)
- [List of JDBC Configurations Parameters](#)
- [JDBC URL examples](#)
- [Query Execution Internals](#)

Treasure Data JDBC Driver (td-jdbc) is an open-source project and is available from the Treasure Data [td-jdbc repository](#).

Download

maven central 0.5.10

- [JDBC Driver Download](#)
- Use `td-jdbc-(version)-jar-with-dependencies.jar`
- For Maven projects, use the following dependency setting:

```
<dependency>
  <groupId>com.treasuredata</groupId>
  <artifactId>td-jdbc</artifactId>
  <version>(version)</version>
</dependency>
```

Usage

Here is a sample code for counting the number of rows in `www_access` table in `sample_dataset` database. First, create a `java.sql.Connection` object using JDBC address `jdbc:td://api.treasuredata.com/(database name)`, then create `java.sql.Statement` objects and call `executeQuery(sql)`, etc. to run queries:

```
Properties props = new Properties();
props.setProperty("user", "(your account e-mail)");
props.setProperty("password", "(your password)");

// Alternatively, you can use API key instead of user and password
// props.setProperty("apikey", "(your API key)");

// For using SSL connection (default is false)
// props.setProperty("useSSL", "true");

// To run Hive jobs specify "type" parameter. The default is "presto"
// props.setProperty("type", "hive");

Connection conn = DriverManager.getConnection("jdbc:td://api.treasuredata.com/sample_datasets", props);
Statement st = conn.createStatement();
try {
  ResultSet rs = st.executeQuery("SELECT count(1) FROM www_access");
  while (rs.next()) {
    int count = rs.getInt(1);
    System.out.println("result = " + count);
  }
  rs.close();
}
finally {
  st.close();
  conn.close();
}
```

JDBC Parameter Precedence

To configure JDBC connection parameters, you can use URL parameters, Properties object or System properties. The precedence of these properties is:

1. Environment variable (only for TD_API_KEY parameter)
2. System properties
3. Properties object passed by *DriverManager.getConnection(jdbc_url, Properties)*
4. JDBC URL parameters (e.g., *jdbc:td://api.treasuredata.com/mydb?type=hive;useSSL=true*), separated by semi-colon.

If your environment defines TD_API_KEY variable, td-jdbc uses it. For the other properties, System properties have the highest priority.

List of JDBC Configurations Parameters

You must provide **apikey** property or both **user** (your account e-mail) and **password** for the authentication:

key	default value	description
apikey		API key to access Treasure Data. You can also set this via TD_API_KEY environment variable.
user		Account e-mail address (unnecessary if apikey is set)
password		Account password (unnecessary if apikey is set)
type	presto	Query engine. hive, presto or pig
useSSL	false	Use SSL encryption for accessing Treasure Data
httpproxyhost		Proxy host (optional) e.g., "myproxy.com"
httpproxyport		Proxy port (optional) e.g., "80"
httpproxyuser		Proxy user (optional)
httpproxypassword		Proxy password (optional)

If both **user** and **password** are given, td-jdbc uses this pair instead of **apikey**.

JDBC URL examples

When some SQL/BI tool has no functionality to set these properties, use URL parameters. For example, here is an example to set **useSSL** parameter in the URL:

```
jdbc:td://api.treasuredata.com/<db_name>;useSSL=true
```

To access a database named "sample_db" in your account:

```
jdbc:td://api.treasuredata.com/sample_db;useSSL=true
```

You can choose the query engine [Hive](#) or [Presto](#) (default) by the parameter:

```
jdbc:td://api.treasuredata.com/sample_db;useSSL=true;type=hive  
jdbc:td://api.treasuredata.com/sample_db;useSSL=true;type=presto
```

To connect Treasure Data through a proxy server, specify the following proxy settings:

- httpproxyhost
- httpproxyport
- httpproxyuser
- httpproxypassword

For example:

```
jdbc:td://api.treasuredata.com/testdb;httpproxyhost=myproxy.com;httpproxyport=myport;httpproxyuser=myusername;  
httpproxypassword=mypassword
```

Query Execution Internals

When running a query (e.g. `SELECT`), the driver submits a job request to Treasure Data. `td-jdbc` periodically monitors the job progress and fetches the result after the job completion.

For `INSERT` statement, `td-jdbc` buffers the data into local memory, then flushes it to Treasure Data every 5 minutes, so there will be a delay until your data becomes accessible in Treasure Data.