Firebolt documentation
Getting started
Architecture overview
Integrations
Account and user management
Using the SQL workspace
Using the CLI
Working with engines
Working with tables
Using indexes
Loading data
Working with partitions
Working with semi-structured data
Exporting query results
Developing with Firebolt
Node.js
Python
REST API
SQLAlchemy
JDBC
.NET
Go
General reference
SQL commands
SQL functions

Using the JDBC driver

Firebolt provides a type 4 JDBC driver to connect to Firebolt from Java applications such as Tableau, DBeaver, and others. The driver is released as open source software using a permissive Apache 2 license and can be browsed, forked, downloaded, and contributed to through its GitHub repository.

Download the JAR file

The Firebolt JDBC driver is provided as a JAR file and requires Java 1.8 or later.

Download the Firebolt JDBC driver JAR file from the GitHub Releases page.

Adding the Firebolt JDBC driver as a Maven dependency

If you are using Apache Maven, you can configure and build your projects to use the Firebolt JDBC driver to connect to your Firebolt resources. To do this, add the JDBC driver as a dependency in your project **pom.xml** file by including a link to the Firebolt Maven repository.

See below for an example pom.xml file:

> **Note**
>
> Be sure to replace <version>0.00</version> with the latest (highest) version number. You can identify the latest version by viewing the version history in the Firebolt Maven repository.

```
<!-- pom.xml  -->

<project ...>
    ...
    <repositories>
                ...
                <repository>
                        <id>repsy</id>
                        <name>Firebolt Private Maven Repository on Repsy</name>
                        <url>https://repo.repsy.io/mvn/firebolt/maven</url>
                </repository>
                ...
        </repositories>
        ...
<dependency>
                <groupId>com.firebolt</groupId>
                <artifactId>firebolt-jdbc</artifactId>
                <version>0.00</version>
       </dependency>
</project>
```

Adding the Firebolt JDBC driver as a gradle dependency

> **Note**
>
> Be sure to replace <version>0.00</version> with the latest (highest) version number. You can identify the latest version by viewing the version history in the Firebolt Maven repository.

```
/* build.gradle */

repositories {
    mavenCentral()
    maven {
      url 'https://repo.repsy.io/mvn/firebolt/maven'
    }
}

dependencies {
    implementation 'com.firebolt:firebolt-jdbc:0.00'
}
```

Connecting to Firebolt with the JDBC driver

Connection details are provided to the Firebolt JDBC driver in a connection string. The string has the following format:

```
jdbc:firebolt://api.app.firebolt.io/<database>?<connection_params>
```

<database>
The name of the Firebolt database to connect to.

<connection_params>
A list of connection parameters following the standard URL query string format.

Here is an example of a connection string:

```
jdbc:firebolt://api.app.firebolt.io/my_database?
user=me%40mycompany.com&password=verysecurepassword123&engine=my_database_general_purpose&buffer_size=1000000&
```

> **Note**
>
> Since the connection string is a URI, make sure to percent-encode any reserved characters or special characters used in parameter keys or parameter values.

Available connection parameters

The table below lists the available connection parameters that can be added to the Firebolt JDBC connection string. All parameter keys are case-sensitive.

Parameter key	Data type	Default value	Range	Description
user	STRING	No default value		The email address associated with your Firebolt user. **Required.**
password	STRING	No default value		The password used for connecting to Firebolt. **Required.**
account	STRING	No default value		Your Firebolt account name.
database	STRING	No default value		Specifies the name of the database to connect to. Takes precedence over the database name provided as a path parameter.
client_buffer_size	INT	65536	1 to 2147483647	Specifies the maximum amount of RAM in bytes that Firebolt uses to cache HTTP messages. The default value is acceptable for a broad range of applications. You might try reducing this value if your engine has issues related to inadequate memory. A buffer that is too small will create a bottleneck. A larger buffer is unlikely to improve throughput performance.
buffer_size	INT	65536	1 to 2147483647	Specifies the buffer used by the driver to read the response from the Firebolt API, in bytes.
connection_timeout_millis	INT	60000	-2147483648 to 2147483647	Specifies the amount of time in milliseconds to wait to establish a connection before the connection is considered failed. A timeout value of zero is interpreted as an infinite timeout. A negative value is interpreted as undefined (system default if applicable).
max_connections_per_route	INT	500	1 to 2147483647	Specifies the maximum number of connections per route.
max_connections_total	INT	10000	1 to 2147483647	Specifies the maximum total number of connections.
socket_timeout_millis	INT	0	-2147483648 to 2147483647	Specifies the socket timeout in milliseconds. This is the timeout for waiting for data – the maximum period of inactivity between two consecutive data packets. A timeout value of zero is interpreted as an infinite timeout. A negative value is interpreted as undefined (system default if applicable).
ssl	BOOLEAN	true	true or false	When set to true, connections use SSL / TLS certificates. This parameter also determines the port used by the driver. If true, it uses port 443. If false, it uses port 80.
ssl_mode	STRING	strict	strict or none	When set to strict, the certificate is validated to ensure it is correct. If set to none, no certificate verification is used.
ssl_certificate_path	STRING	No default value		The absolute file path for the SSL root certificate.
time_to_live_millis	INT	60000	-2147483648 to 2147483647	Specifies the maximum lifespan of connections. A value of 0 or less leaves this parameter undefined (which means that the connection can be kept alive indefinitely).

System settings as connection parameters

In addition to the parameters specified above, any system setting can be passed as a connection string parameter. For example, if you wanted to set a custom CSV delimiter, your connection string would be as follows:

```
jdbc:firebolt://api.app.firebolt.io/my_database?format_csv_delimeter=%7C&<other_connection_params>
```

Applying system settings using SET

In addition to passing system settings as connection string parameters, any system setting can be passed to Firebolt as a SET command in SQL. Multiple SET statements can passed at once as long as they immediately follow one after another separated by semicolons, as shown in the following example.

```
SET input_format_csv_allow_single_quotes = 0;
SET format_csv_delimiter = |;
```

Full reference documentation

Complete reference documentation for the classes and methods implemented in the Firebolt JDBC driver can be found here.