

# Configuring a datasource connection

This guide covers how to configure a JNDI datasource connection to your database. With this type of connection, Confluence asks the application server (Tomcat) for your database connection information.

If you'd prefer to use a JDBC connection see the guide for your database:

- [Database Setup for PostgreSQL](#)
- [Database Setup for SQL Server](#)
- [Database Setup For MySQL](#)
- [Database Setup for Oracle](#)

Direct JDBC is the most common way to connect Confluence to your database and is the easiest method when it comes time to upgrade Confluence.

## On this page:

- [New Confluence installation](#)
- [Existing Confluence installation](#)
- [Upgrading Confluence with a datasource](#)

## Related pages:

- [Database JDBC Drivers](#)

## New Confluence installation

The Confluence setup wizard will only provide an option to use a datasource if it detects one in your Tomcat configuration. If you want to use a datasource, follow the steps below.

### 1. Stop Confluence

In the Confluence setup wizard, you'll be prompted to choose your database. At this point, you should:

1. Use `bin/stop-confluence.sh` or `bin/stop-confluence.bat` to stop Confluence.
2. Back up the following files, in case you need to revert your changes:
  - `<installation-directory>/conf/server.xml`
  - `<installation-directory>/confluence/WEB-INF/web.xml`
  - `<home-directory>/confluence.cfg.xml`

### 2. Add your database driver

Copy your database driver into the `<installation-directory>/libdirectory`.

Here's where to find the driver for your database:

- **PostgreSQL:** bundled with Confluence at `<installation-directory>/confluence/WEB-INF/lib/postgresql-x.x.x.jar`
- **Microsoft SQL Server:** bundled with Confluence at `<installation-directory>/confluence/WEB-INF/lib/mssql-jdbc-x.x.x.x.jar`
- **MySQL:** head to [Database JDBC Drivers](#) to download the driver
- **Oracle:** head to [Database JDBC Drivers](#) to download the driver

### 3. Configure the datasource in Tomcat

Next, add the datasource configuration to Tomcat.

1. Edit `<installation-directory>/conf/server.xml`
2. Find the following lines:

```
<Context path="" docBase="../confluence" debug="0" reloadable="true">
<!-- Logger is deprecated in Tomcat 5.5. Logging configuration for Confluence is
specified in confluence/WEB-INF/classes/log4j.properties -->
```

3. Insert the following `DataSourceResourceElement` for your specific database directly after the lines above (inside the `Context` element, directly after the opening `<Context.../>` line, before `Manager`).

```
<Resource name="jdbc/confluence" auth="Container" type="javax.sql.DataSource"
  username="<database-user>"
  password="<password>"
  driverClassName="org.postgresql.Driver"
  url="jdbc:postgresql://<host>:5432/<database-name>"
  maxTotal="60"
  maxIdle="20"
  validationQuery="select 1"/>
```

```
<Resource name="jdbc/confluence" auth="Container" type="javax.sql.DataSource"
  username="<database-user>"
  password="<password>"
  driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"
  url="jdbc:sqlserver://<host>:1433;database=<database-name>"
  maxTotal="60"
  maxIdle="20"
  validationQuery="select 1"/>
```

If you're using Confluence 6.3 or earlier, you'll need to specify the JTDs driver for SQL Server. See [Configuring a datasource connection](#) in our 6.3 documentation for a sample configuration.

```
<Resource name="jdbc/confluence" auth="Container" type="javax.sql.DataSource"
  username="<database-user>"
  password="<password>"
  driverClassName="com.mysql.jdbc.Driver"
  url="jdbc:mysql://<host>:3306/<database-name>?useUnicode=true&characterEncoding=utf8"
  maxTotal="60"
  maxIdle="20"
  defaultTransactionIsolation="READ_COMMITTED"
  validationQuery="Select 1"/>
```

```
<Resource name="jdbc/confluence" auth="Container" type="javax.sql.DataSource"
  driverClassName="oracle.jdbc.OracleDriver"
  url="jdbc:oracle:thin:@<host>:1521:<SID>"
  username="<database-user>"
  password="<password>"
  connectionProperties="SetBigStringTryClob=true"
    accessToUnderlyingConnectionAllowed="true"
  maxTotal="60"
  maxIdle="20"
  maxWaitMillis="10000"/>
```

See [how to find your Oracle URL](#).

Replace <database-user>, <password>, <host> and <database-name> (or <SID> for Oracle) with details of your own database. You may also need to change the port, if your database server is not running on the default port.

4. Configure the connection pool and other properties. See the [Apache Tomcat 8 Datasource documentation](#) for more information.

Here are the configuration properties for Tomcat's standard data source resource factory (`org.apache.tomcat.dbcp.dbcp.BasicDataSourceFactory`):

- `driverClassName` - Fully qualified Java class name of the JDBC driver to be used.
- `maxTotal` - The maximum number of active instances that can be allocated from this pool at the same time.
- `maxIdle` - The maximum number of connections that can sit idle in this pool at the same time.
- `maxWaitMillis` - The maximum number of milliseconds that the pool will wait (when there are no available connections) for a connection to be returned before throwing an exception.
- `password` - Database password to be passed to the JDBC driver.
- `url` - Connection URL to be passed to the JDBC driver. (For backwards compatibility, the property `driverName` is also recognized.)
- `user` - Database username to be passed to the JDBC driver.
- `validationQuery` - SQL query that can be used by the pool to validate connections before they are returned to the application. If specified, this query MUST be an SQL SELECT statement that returns at least one row. When a database server reboots, or there is a network failure, all the connections in the connection pool are broken and this normally requires a application server reboot. However, the Commons DBCP (Database Connection Pool) used by Tomcat can validate connections

before issuing them by running a simple SQL query, and if a broken connection is detected, a new one is created to replace it. To do this, you will need to set the "validationQuery" option on the database connection pool.

5. If you plan to use collaborative editing, you'll need to make sure:
  - You're using a supported database driver. Collaborative editing will fail if you're using an unsupported or custom JDBC driver or `driverClassName` in your `datasource`. See [Database JDBC Drivers](#) for the list of drivers we support.
  - Your database connection pool allows enough connections to support both Confluence and Synchrony (which defaults to a maximum pool size of 15)
  - You're using simple username and password authentication for your database.

## 4. Configure the Confluence web application

Configure Confluence to use this `datasource`:

1. Edit `<CONFLUENCE_INSTALLATION>/confluence/WEB-INF/web.xml`.
2. Insert the following element just before `</web-app>` near the end of the file:

```
<resource-ref>
  <description>Connection Pool</description>
  <res-ref-name>jdbc/confluence</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

## 5. Restart Confluence and continue setup process

Now that your `datasource` is configured, you can continue with the setup wizard.

1. Use `bin/start-confluence.sh` or `bin/start-confluence.bat` to restart Confluence
2. Go to <http://localhost:8090> to return to the setup wizard.
3. When prompted choose **My own database(datasource)**.
4. Enter the JNDI name of your `datasource`, for example, `java:comp/env/jdbc/confluence`
5. Follow the prompts to finish setting up Confluence.

## Existing Confluence installation

If you want to switch from using a direct JDBC connection to a `datasource`:

- Stop Confluence.
- Back up the following files, in case you need to revert your changes:
  - `<installation-directory>/conf/server.xml`
  - `<installation-directory>/confluence/WEB-INF/web.xml`
  - `<home-directory>/confluence.cfg.xml`
- Follow the steps above for a new installation and copy your driver and add the `datasource` to the appropriate files. You can find the details of your current database connection in `<home-directory>/confluence.cfg.xml`.
- Edit the `<home-directory>/confluence.cfg.xml` file and remove any line that contains a property that begins with `hibernate`.
- Insert the following at the start of the `<properties>` section.

```
<property name="hibernate.setup"><![CDATA[true]]></property>
<property name="hibernate.dialect"><![CDATA[net.sf.hibernate.dialect.PostgreSQLDialect]]></property>
<property name="hibernate.connection.datasource"><![CDATA[java:comp/env/jdbc/confluence]]></property>
```

```
<property name="hibernate.setup"><![CDATA[true]]></property>
<property name="hibernate.dialect"><![CDATA[net.sf.hibernate.dialect.SQLServerInt1Dialect]]></property>
<property name="hibernate.connection.datasource"><![CDATA[java:comp/env/jdbc/confluence]]></property>
```

```
<property name="hibernate.setup"><![CDATA[true]]></property>
<property name="hibernate.dialect"><![CDATA[com.atlassian.hibernate.dialect.MySQLDialect]]><
/property>
<property name="hibernate.connection.datasource"><![CDATA[java:comp/env/jdbc/confluence]]><
/property>
```

```
<property name="hibernate.setup"><![CDATA[true]]></property>
<property name="hibernate.dialect"><![CDATA[com.atlassian.confluence.impl.hibernate.dialect.
OracleDialect]]></property>
<property name="hibernate.connection.datasource"><![CDATA[java:comp/env/jdbc/confluence]]><
/property>
```

- Restart Confluence.

## Upgrading Confluence with a datasource

If you're upgrading Confluence (manually or using the installer) you will need to:

- Stop Confluence (if you have attempted to start it).
- Copy your database driver into the `<installation-directory>/libdirectory`.
- Edit `<installation-directory>/conf/server.xml` and add your datasource resource.
- Edit `<installation-directory>/confluence/WEB-INF/web.xml` to configure Confluence to use this datasource.

If you forget to do these steps, Confluence will not start up after upgrade and you'll see the following error:

```
HTTP Status 500 - Confluence is vacant, a call to tenanted [public abstract org.hibernate.Session org.
hibernate.SessionFactory.getCurrentSession() throws org.hibernate.HibernateException] is not allowed.
```