



Core Configuration

Server.xml Essentials

<Server> : Root element, defines the entire Tomcat server.

- port** : Shutdown port (e.g., `8005`).
- shutdown** : Shutdown command (e.g., `SHUTDOWN`).

Example:

```
<Server port="8005" shutdown="SHUTDOWN">
```

<Service> : Contains one or more Connectors and a single Engine.

Example:

```
<Service name="Catalina">
```

<Connector> : Defines a port for incoming requests.

- port** : Port number (e.g., `8080`, `8443`).
- protocol** : Protocol (e.g., `HTTP/1.1`, `org.apache.coyote.http11.Http11NioProtocol`).
- connectionTimeout** : Timeout in milliseconds.
- redirectPort** : Port to redirect to for SSL (e.g., `8443`).

Example:

```
<Connector port="8080" protocol="HTTP/1.1"
  connectionTimeout="20000"
  redirectPort="8443" />
```

<Engine> : Represents the Servlet engine.

- name** : Engine name (e.g., `Catalina`).
- defaultHost** : Default virtual host.

Example:

```
<Engine name="Catalina" defaultHost="localhost">
```

<Host> : Represents a virtual host.

- name** : Hostname (e.g., `localhost`, `example.com`).
- appBase** : Application base directory (e.g., `webapps`).
- unpackWARs** : Unpack WAR files on deploy (true/false).
- autoDeploy** : Automatically deploy web applications (true/false).

Example:

```
<Host name="localhost" appBase="webapps"
  unpackWARs="true" autoDeploy="true">
```

<Context> : Represents a web application.

- path** : Context path (e.g., `/`, `/myapp`).
- docBase** : Document base directory or WAR file path.
- reloadable** : Enable auto-reloading on changes (true/false - use with caution in production!).

Example:

```
<Context path="/myapp" docBase="myapp" reloadable="false"/>
```

Context Configuration

Context configuration can be defined in `server.xml` (discouraged for portability) or in individual context XML files under `$CATALINA_BASE/conf/Catalina/localhost/`.

For example, `myapp.xml` for a webapp with context path `/myapp`.

Using context XML files allows for easier deployment and redeployment of web applications without modifying the main server configuration.

Remember to disable auto-deploy and unpackWARs in production environments to avoid unexpected behavior.

Deployment and Management

Web Application Deployment

- WAR File Deployment:** Copy the WAR file to `$CATALINA_BASE/webapps/`. Tomcat automatically deploys it.
- Exploded WAR Deployment:** Copy the exploded WAR directory to `$CATALINA_BASE/webapps/`.
- Context XML Deployment:** Create a context XML file in `$CATALINA_BASE/conf/Catalina/localhost/` (e.g., `myapp.xml`) and specify the `docBase`.

Tomcat Manager Application

- Access:** Access the Tomcat Manager application at `http://<host>:<port>/manager/html`. Requires authentication configured in `$CATALINA_HOME/conf/tomcat-users.xml`.
- Functions:** Deploy, undeploy, start, stop, and reload web applications. View server status and diagnostics.

Command-Line Management

- | | |
|--|-------------------------------|
| <code>shutdown.sh</code> / <code>shutdown.bat</code> | Shuts down the Tomcat server. |
| <code>startup.sh</code> / <code>startup.bat</code> | Starts the Tomcat server. |

Security and SSL Configuration

SSL Configuration

- Keystore Creation:** Use `keytool` to create a keystore file.

```
keytool -genkey -alias tomcat -keyalg RSA -keystore keystore.jks
```

- Connector Configuration:** Configure an SSL Connector in `server.xml`.

```
<Connector port="8443"
protocol="org.apache.coyote.http11.Http11NioProtocol"
maxThreads="200" scheme="https" secure="true"
SSLEnabled="true"
keystoreFile="$catalina.home}/conf/keystore.jks"
keystorePass="changeit" clientAuth="false"
sslProtocol="TLS"/>
```

Key Attributes:

- `keystoreFile`: Path to the keystore file.
- `keystorePass`: Keystore password.
- `clientAuth`: Whether client authentication is required (true/false).
- `sslProtocol`: SSL/TLS protocol.

Security Realms

Tomcat supports various security realms for authentication and authorization. Common realms include:

- `UserDatabaseRealm`: Uses the `tomcat-users.xml` file.
- `JDBCRealm`: Uses a JDBC connection to authenticate against a database.
- `JNDIRealm`: Uses JNDI to retrieve user information.

Troubleshooting

Common Issues

Port Conflicts: Ensure Tomcat ports (8080, 8005, 8443) are not in use by other applications.

Solution: Change the port numbers in `server.xml`.

OutOfMemoryError: Tomcat runs out of memory.

Solution: Increase the JVM heap size in `CATALINA_OPTS` or `JAVA_OPTS` environment variables.

```
export CATALINA_OPTS="-Xms512m -Xmx2048m"
```

Web Application Deployment Failures: Issues during WAR deployment.

Solution: Check Tomcat logs for error messages. Verify the WAR file is valid and the context path is not conflicting with existing applications.

Log Files

- `catalina.out`: General Tomcat output, including startup and shutdown messages.
- `localhost_access_log.txt`: Access logs for the localhost virtual host.
- `manager.log`: Logs for the Tomcat Manager application.
- Web application logs:** Located in the web application's logging directory (e.g., using Log4j or SLF4J).

Log files are typically located in the `$CATALINA_BASE/logs/` directory.