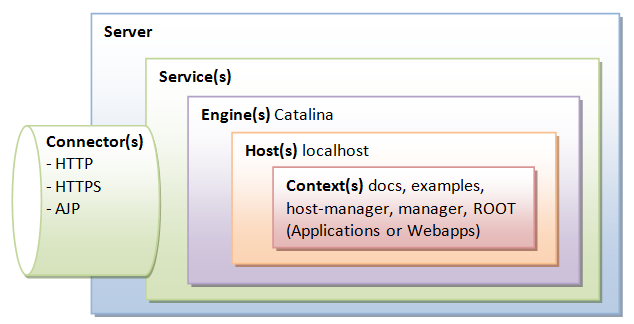
Tomcat Architecture:

* Tomcat is a servlet container that can execute Java Servlet and
* converts JSP and JavaServerFaces (JSF) to Java Servlet.



Tomcat's main configuration file is <CATALINA\_HOME>\conf\server.xml

**Global Naming Resources:**

The <GlobalNamingResources> element defines the JNDI resources, that allows Java software clients to discover and look up data and objects via a name.

<GlobalNamingResources>

<**Resource name**="UserDatabase" auth="Container"

type="org.apache.catalina.UserDatabase"

description="User database that can be updated and saved"

factory="org.apache.catalina.users.MemoryUserDatabaseFactory"

pathname="conf/tomcat-users.xml" />

</GlobalNamingResources>

The default configuration defines a JNDI name called UserDatabase via the <Resource> element.

This is a memory-based database for user authentication loaded from "conf/tomcat-users.xml".

You can define other global resource JNDI such as MySQL database to implement connection pooling.

**Connectors**

A Connector is associated with a TCP port to handle communications between the Service and the clients. The default configuration defines two Connectors:

* HTTP/1.1: Handle HTTP communication and enable Tomcat to be an HTTP server.
  + Clients can issue HTTP requests to the server via this Connector, and receive the HTTP response messages.
* AJP/1.3: Apache JServ Protocol connector to handle communication between Tomcat server and Apache HTTP server.

Engine

It can contains one or more Hosts, known as virtual host.

The Catalina Engine receives HTTP requests from the HTTP connector, and direct them to the correct host based on the hostname/IP address in the request header.

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

**Realm**

A Realm is a database of user, password, and role for authentication.

You can define Realm for any container, such as Engine, Host, and Context, and Cluster.

Types of realm

* **UserDatabaseRealm** : This is default Realm.
  + It uses the JNDI name UserDatabase defined in the GlobalNamingResources.
* **JDBCRealm** : for authenticating users to connect to a relational database via the JDBC driver);
* **DataSourceRealm**: to connect to a DataSource via JNDI;
* **JNDIRealm**: to connect to an LDAP directory; and
* **MemoryRealm**: to load an XML file in memory.

Important configurations:

**Changing the Default "webapps" Directory**

The default directory for deploying web applications is <CATALINA\_HOME>\webapps.

This can be changed by modifying the configuration file "conf\server.xml" <Host> element's "appBase" attribute.

<Host name="localhost" appBase="webapps"

unpackWARs="true" autoDeploy="true"

xmlValidation="false" xmlNamespaceAware="false">

......

</host>

**Automatic Servlet Reload**

To enable automatic servlet reload (whenever a servlet is re-compiled),

* For all web applications : in "<CATALINA\_HOME>\conf\context.xml" , specify <Context reloadable="true">...</Context> or
* For a particular web application.: Set the <Context> element For a particular web application true in "<CATALINA\_HOME>\conf\server.xml"

Enabling automatic servlet reload is handy during application development, but it requires significant runtime overhead to listen to the changes, and is not recommended for production systems. Instead, you could use the "manager" to trigger reloads on demand.

Host:

A Host defines a virtual host under the Engine, which can in turn support many Contexts (webapps).

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

The default configuration define one host called localhost. The appBase attribute defines the base directory of all the webapps, in this case, <CATALINA\_HOME>\webapps. By default, each webapp's URL is the same as its directory name. For example, the default Tomcat installation provides four webapps: docs, examples, host-manager and manager under the webapps directory. The only exception is ROOT, which is identified by an empty string. That is, its URL is http://localhost:8080/.

The unpackWARs specifies whether WAR-file dropped into the webapps directory shall be unzipped. For unpackWARs="false", Tomcat will run the application from the WAR-file directly, without unpacking, which could mean slower execution.

The autoDeploy attribute specifies whether to deploy application dropped into the webapps directory automatically.