安装：

wget <https://mirrors.tuna.tsinghua.edu.cn/apache/tomcat/tomcat-8/v8.5.57/bin/apache-tomcat-8.5.57.tar.gz>

下载不了就去<https://mirrors.tuna.tsinghua.edu.cn/apache/tomcat> 查看最新镜像

解压：tar -zxvf apache-tomcat-8.5.57.tar.gz

修改配置文件：server.xml

<?xml version="1.0" encoding="UTF-8"?>

<!--

Licensed to the Apache Software Foundation (ASF) under one or more

contributor license agreements. See the NOTICE file distributed with

this work for additional information regarding copyright ownership.

The ASF licenses this file to You under the Apache License, Version 2.0

(the "License"); you may not use this file except in compliance with

the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

-->

<!-- Note: A "Server" is not itself a "Container", so you may not

define subcomponents such as "Valves" at this level.

Documentation at /docs/config/server.html

-->

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

<Listener className="org.apache.catalina.startup.VersionLoggerListener" />

<!-- Security listener. Documentation at /docs/config/listeners.html

<Listener className="org.apache.catalina.security.SecurityListener" />

-->

<!--APR library loader. Documentation at /docs/apr.html -->

<Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />

<!-- Prevent memory leaks due to use of particular java/javax APIs-->

<Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />

<Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />

<Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />

<!-- Global JNDI resources

Documentation at /docs/jndi-resources-howto.html

-->

<GlobalNamingResources>

<!-- Editable user database that can also be used by

UserDatabaseRealm to authenticate users

-->

<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>

<!-- A "Service" is a collection of one or more "Connectors" that share

a single "Container" Note: A "Service" is not itself a "Container",

so you may not define subcomponents such as "Valves" at this level.

Documentation at /docs/config/service.html

-->

<Service name="Catalina">

<!--The connectors can use a shared executor, you can define one or more named thread pools-->

<!--

<Executor name="tomcatThreadPool" namePrefix="catalina-exec-"

maxThreads="150" minSpareThreads="4"/>

-->

<!-- A "Connector" represents an endpoint by which requests are received

and responses are returned. Documentation at :

Java HTTP Connector: /docs/config/http.html

Java AJP Connector: /docs/config/ajp.html

APR (HTTP/AJP) Connector: /docs/apr.html

Define a non-SSL/TLS HTTP/1.1 Connector on port 8080

-->

<Connector port="80" protocol="HTTP/1.1"

connectionTimeout="20000"

URIEncoding="UTF-8"

redirectPort="443" />

<!-- A "Connector" using the shared thread pool-->

<!--

<Connector executor="tomcatThreadPool"

port="8080" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" />

-->

<!-- Define an SSL/TLS HTTP/1.1 Connector on port 8443

This connector uses the NIO implementation. The default

SSLImplementation will depend on the presence of the APR/native

library and the useOpenSSL attribute of the

AprLifecycleListener.

Either JSSE or OpenSSL style configuration may be used regardless of

the SSLImplementation selected. JSSE style configuration is used below.

-->

<Connector port="443" protocol="org.apache.coyote.http11.Http11NioProtocol"

maxThreads="150" SSLEnabled="true">

<SSLHostConfig>

<Certificate certificateKeystoreFile="/opt/tomcat/cert/2996929\_newayue.com.pfx" certificateKeystoreType="PKCS12" certificateKeystorePassword="GnkbzM7i" />

</SSLHostConfig>

</Connector>

<!-- Define an SSL/TLS HTTP/1.1 Connector on port 8443 with HTTP/2

This connector uses the APR/native implementation which always uses

OpenSSL for TLS.

Either JSSE or OpenSSL style configuration may be used. OpenSSL style

configuration is used below.

-->

<!--

<Connector port="8443" protocol="org.apache.coyote.http11.Http11AprProtocol"

maxThreads="150" SSLEnabled="true" >

<UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol" />

<SSLHostConfig>

<Certificate certificateKeyFile="conf/localhost-rsa-key.pem"

certificateFile="conf/localhost-rsa-cert.pem"

certificateChainFile="conf/localhost-rsa-chain.pem"

type="RSA" />

</SSLHostConfig>

</Connector>

-->

<!-- Define an AJP 1.3 Connector on port 8009 -->

<Connector port="8009" protocol="AJP/1.3" redirectPort="443" />

<!-- An Engine represents the entry point (within Catalina) that processes

every request. The Engine implementation for Tomcat stand alone

analyzes the HTTP headers included with the request, and passes them

on to the appropriate Host (virtual host).

Documentation at /docs/config/engine.html -->

<!-- You should set jvmRoute to support load-balancing via AJP ie :

<Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">

-->

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

<!--For clustering, please take a look at documentation at:

/docs/cluster-howto.html (simple how to)

/docs/config/cluster.html (reference documentation) -->

<!--

<Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>

-->

<!-- Use the LockOutRealm to prevent attempts to guess user passwords

via a brute-force attack -->

<Realm className="org.apache.catalina.realm.LockOutRealm">

<!-- This Realm uses the UserDatabase configured in the global JNDI

resources under the key "UserDatabase". Any edits

that are performed against this UserDatabase are immediately

available for use by the Realm. -->

<Realm className="org.apache.catalina.realm.UserDatabaseRealm"

resourceName="UserDatabase"/>

</Realm>

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

unpackWARs="true" autoDeploy="true">

<!-- SingleSignOn valve, share authentication between web applications

Documentation at: /docs/config/valve.html -->

<!--

<Valve className="org.apache.catalina.authenticator.SingleSignOn" />

-->

<!-- Access log processes all example.

Documentation at: /docs/config/valve.html

Note: The pattern used is equivalent to using pattern="common" -->

<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"

prefix="localhost\_access\_log" suffix=".txt"

pattern="%h %l %u %t &quot;%r&quot; %s %b" />

</Host>

</Engine>

</Service>

</Server>

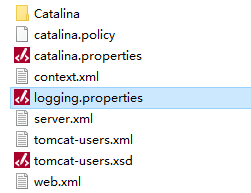
启动时报错：org.apache.jasper.servlet.TldScanner.scanJars At least one JAR was scanned for TLDs yet contained no TLDs

进入位于Tomcat的/ conf目录中的context.xml并添加：

<JarScanner scanClassPath="false"/>

1.在conf下的logging.properties 中添加

org.apache.jasper.servlet.TldScanner.level = FINE 重新启动项目 错误就会消失



2.在conf下的Catalina.properties 中添加

tomcat.util.scan.StandardJarScanFilter.jarsToSkip = \*\*\*\*  
项目就可以启动成功!!!

# Tomcat启动时一直卡在org.apache.catalina.startup.HostConfig.deployWAR Deploying web application archive

解决方法：

1. 打开$JAVA\_HOME/jre/lib/security/java.security这个文件
2. 将securerandom.source=file:/dev/random 修改为 securerandom.source=file:/dev/urandom