

服务器管理与维护

项目6 Web服务器的配置与管理

任务10 端口和主页设置



目录

Contents

01 修改tomcat服务器的web服务端口

02 修改网站的主页

03 总结

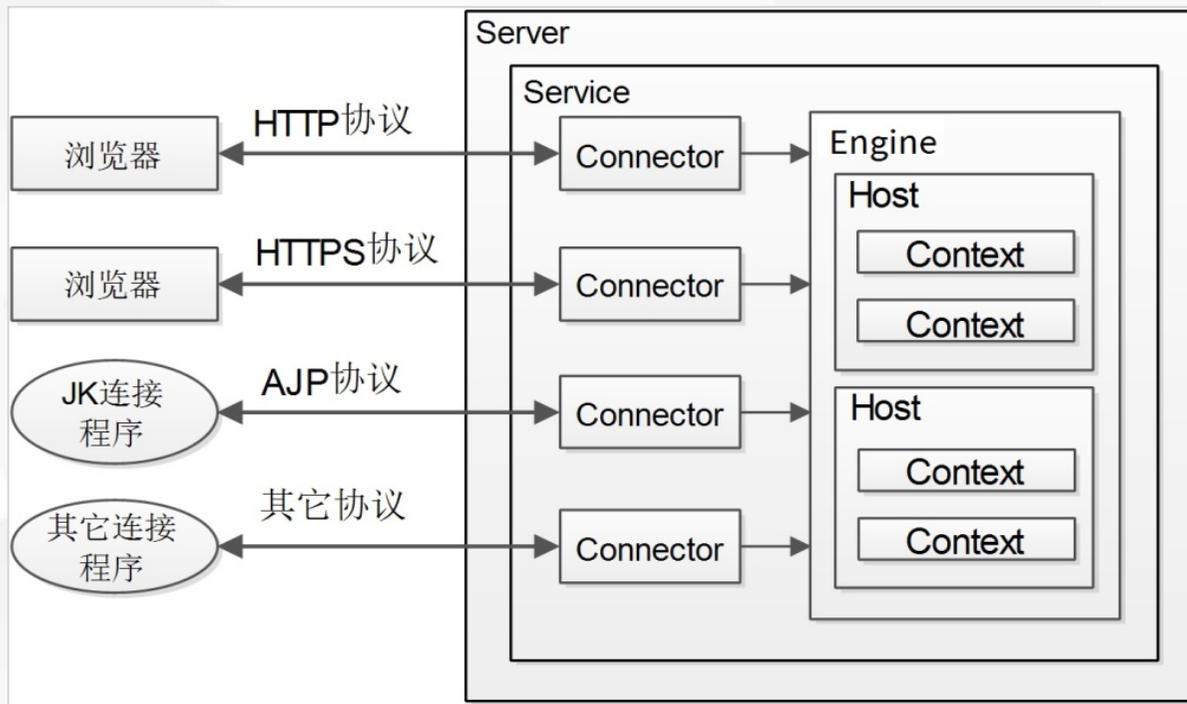
04 学习检验

01

修改tomcat服务器的web服务端口

修改tomcat服务器的web服务端口

上节课访问tomcat主页时，输入的网址是http://localhost:8080，说明该服务的运行端口是8080，如何修改这个端口呢？首先观察Tomcat服务器的组织结构：



修改tomcat服务器的web服务端口

最顶层的元素是Server，代表Tomcat服务器，它的配置信息通过Tomcat根目录下的conf子目录中的server.xml文件来储存。

例如：C:\apache-tomcat-8.5.41\conf\server.xml。

用IE浏览器打开server.xml,可以看出它是一个树形结构，我们可以点击左边红色的加号展开节点，或点击减号收拢节点。

修改tomcat服务器的web服务端口

```
C:\apache-tomcat-8.5.41\conf\server.xml
C:\apache-tomcat-8.5.41\c... x

<?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 shutdown="SHUTDOWN" port="8005">
  <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 pathname="conf/tomcat-users.xml" factory="org.apache.catalina.users.MemoryUserDatabaseFactory" description="User database that can be
updated and saved" type="org.apache.catalina.UserDatabase" auth="Container" name="UserDatabase"/>
</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="8080" redirectPort="8443" connectionTimeout="20000" protocol="HTTP/1.1"/>
  <!-- A "Connector" using the shared thread pool-->
  <!-- <Connector executor="tomcatThreadPool" port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" /> -->
  <!-- Define a 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="8443" protocol="org.apache.coyote.http11.Http11NioProtocol" maxThreads="150" SSLEnabled="true"> <SSLHostConfig> <Certificate
```

修改tomcat服务器的web服务端口

点击Server节点左边的减号，将节点收拢，可以看到Server节点是整个配置文档的根节点。在节点中包含shutdown和port属性，表示通过8005端口发送大写的“SHUTDOWN”指令，可以关闭服务器。



```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE for
regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may
License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable
the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
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. Document
+ <Server shutdown="SHUTDOWN" port="8005">
```

修改tomcat服务器的web服务端口

展开Server节点，找到Service子节点，该节点表示一个Web服务，名叫“Catalina”，是一个风景秀丽的小岛的名字。



```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE
regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you
License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicabl
the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or im
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. Docum
- <Server shutdown="SHUTDOWN" port="8005">
  <Listener className="org.apache.catalina.startup.VersionLoggerListener"/>
  <!-- Security listener. Documentation at /docs/config/listeners.html <Listener className="org.apache.catalina.secu
  <!--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>
    <!-- A "Service" is a collection of one or more "Connectors" that share a single "Container" Note: A "Service" is not i
    subcomponents such as "Valves" at this level. Documentation at /docs/config/service.html -->
    + <Service name="Catalina">
  </Server>
```

修改tomcat服务器的web服务端口

展开Service节点，下面包含两个名叫Connector的子节点，代表网络连接，客户端可以通过这两个连接来访问服务。

```
- <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
  <!-- A "Connector" represents an endpoint by which requests are received and responses are returned. Docum
  Connector: /docs/config/http.html Java AJP Connector: /docs/config/ajp.html APR (HTTP/AJP) Connector: /doc
  port 8080 -->
  <Connector port="8080" redirectPort="8443" connectionTimeout="20000" protocol="HTTP/1.1"/>
  <!-- A "Connector" using the shared thread pool-->
  <!-- <Connector executor="tomcatThreadPool" port="8080" protocol="HTTP/1.1" connectionTimeout="20000'
  <!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443 This connector uses the NIO implementation. The def
  the APR/native library and the useOpenSSL attribute of the AprLifecycleListener. Either JSSE or OpenSSL style
  SSLImplementation selected. JSSE style configuration is used below. -->
  <!-- <Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol" maxThreads="150" SSL
  certificateKeystoreFile="conf/localhost-rsa.jks" type="RSA" /> </SSLHostConfig> </Connector> -->
  <!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443 with HTTP/2 This connector uses the APR/native impl
  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" SSL
  className="org.apache.coyote.http2.Http2Protocol" /> <SSLHostConfig> <Certificate certificateKeyFile="con
  rsa-cert.pem" certificateChainFile="conf/localhost-rsa-chain.pem" type="RSA" /> </SSLHostConfig> </Conne
  <!-- Define an AJP 1.3 Connector on port 8009 -->
  <Connector port="8009" redirectPort="8443" protocol="AJP/1.3"/>
  <!-- An Engine represents the entry point (within Catalina) that processes every request. The Engine impleme
  headers included with the request, and passes them on to the appropriate Host (virtual host). Documentation
  <!-- You should set jvmRoute to support load-balancing via AJP ie : <Engine name="Catalina" defaultHost="lo
```

修改tomcat服务器的web服务端口

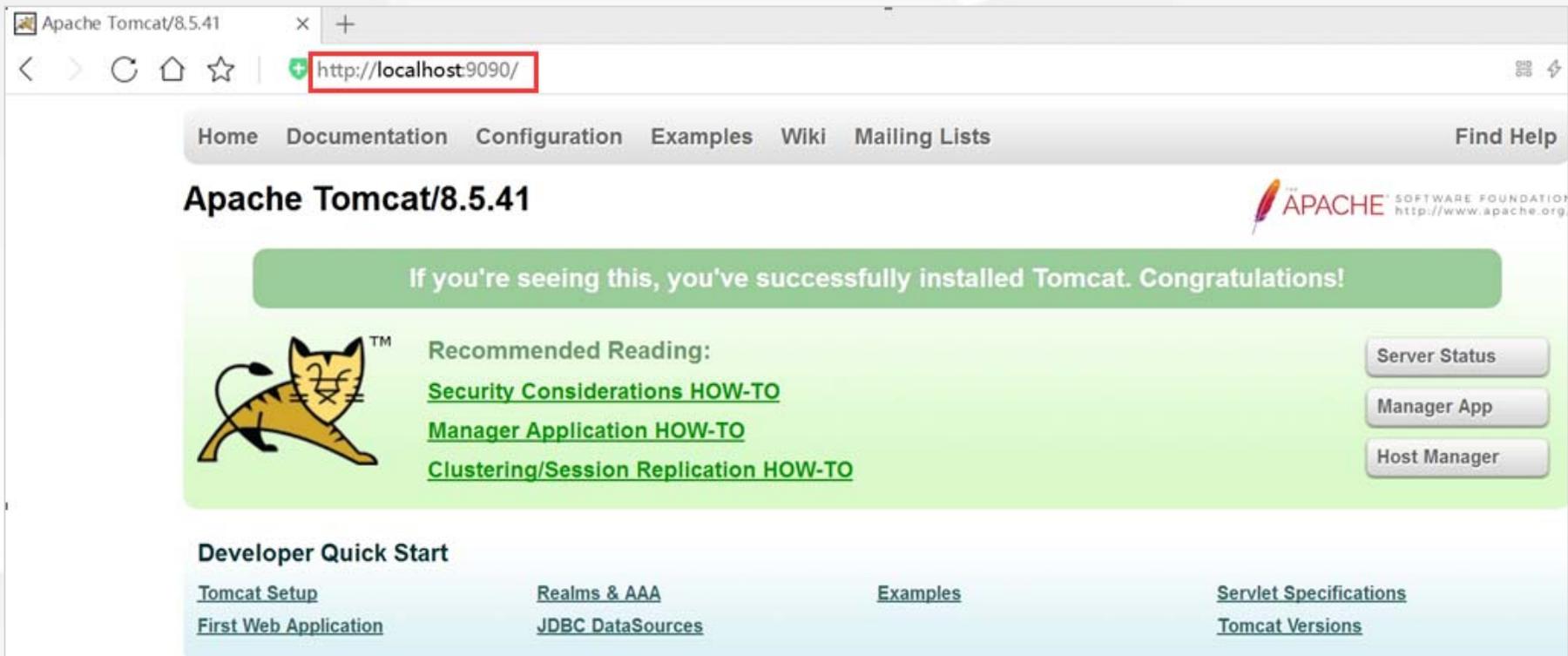
第一个连接的端口号是8080，协议是HTTP1.1，这个连接提供web服务，上一节课中访问主页的地址是http://localhost:8080，里面的端口号就是在这里规定的。如果将端口修改为9090，那么访问主页所用的地址就要变成http://localhost:9090。

修改tomcat服务器的web服务端口

用记事本打开server.xml，找到Connector节点，将port=8080修改为port=9090，保存文件。然后运行bin目录下的startup.bat，重新启动tomcat。

修改tomcat服务器的web服务端口

在浏览器窗口输入http://localhost:9090，打开主页。



02

修改网站的主页

修改网站的主页

当输入网址http://localhost:8080时，浏览器打开了网站的主页，但是这个主页对应的文件在哪里呢？

网站的主页位于C:\apache-tomcat-8.5.41\webapps\ROOT，文件名叫做index.jsp。

Tomcat是怎么找到这个路径的呢？

第一步，找到tomcat的根目录，就是C:\apache-tomcat-8.5.41。

第二步，找到虚拟主机的根目录：

Service中有一个子节点是Engine，它的里面又包含了Host（虚拟主机）子节点。而虚拟主机主机指定的appBase（即程序路径）是webapps，这是一个相对路径，相对于tomcat的根目录，完整的目录就是C:\apache-tomcat-8.5.41\webapps。

Tomcat是怎么找到这个路径的呢？

第三步，找到应用程序根目录：

每个虚拟主机里面可以包含多个应用程序，在tomcat里面叫做Context，如果没有添加Context，主机会默认设定一个，这个应用程序的根目录就位于ROOT中（注意ROOT全大写）。

根据以上三步，就得到网站主页的路径是

C:\apache-tomcat-8.5.41\ webapps\ROOT。

tomcat怎么知道哪个文件是默认的主页呢？

在conf目录下还有一个名叫web.xml的配置文件，这个配置文件会影响所有的应用程序。

里边有这么一段：

```
<welcome-file-list>  
  <welcome-file>index.html</welcome-file>  
  <welcome-file>index.htm</welcome-file>  
  <welcome-file>index.jsp</welcome-file>  
</welcome-file-list>
```

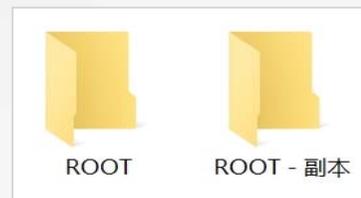
其中welcome-file-list就表示主页列表，tomcat会按照先后顺序依次搜索主页。
在ROOT目录中没有index.html和index.htm，所以就把index.jsp作为主页了。

tomcat怎么知道哪个文件是默认的主页呢？



找准位置后，要修改主页也变得非常简单了。

先把ROOT目录复制一份：



然后将ROOT里面的内容删掉，放入我们自己的主页，主页的文件名按照welcome-file-list中规定的名字，命名为index.html，它的内容包含一个div标签，显示红色的“我的主页”。

```
<div style="font-size:20pt; color:red" align="center">我的主页</div>
```

修改网站的主页

tomcat怎么知道哪个文件是默认的主页呢？

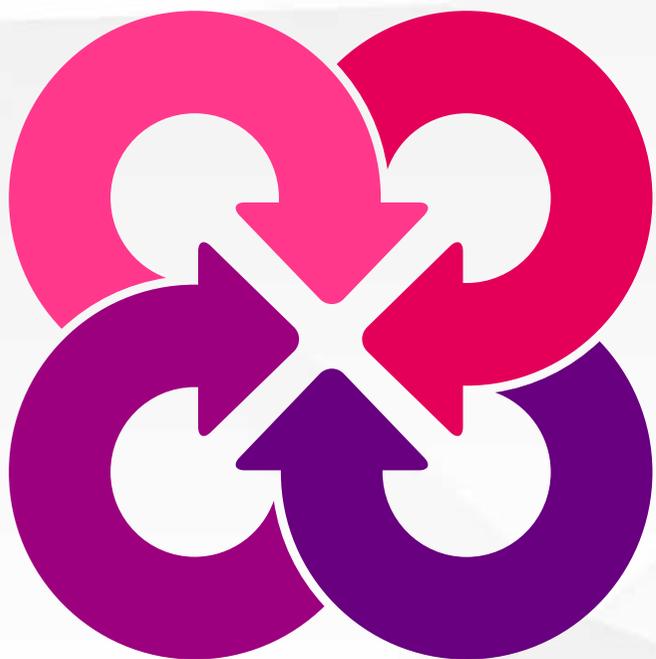
现在刷新浏览器，可以看到主页已经被替换。



03

总结

总结



本课学习了tomcat中服务器的服务端口号设置，以及web应用程序的主页设置，也初步了解了tomcat的关键配置文件和关键路径，接下来的课程将深入学习虚拟主机(Host)、web应用程序(Context)的配置。

05

学习检验



THANKS

服务器管理与维护