Масштабируемая отказоустойчивая архитектура на основе серверов Apache HTTPD и Tomcat

Оглавление

[Масштабируемая отказоустойчивая архитектура на основе серверов Apache HTTPD и Tomcat 1](#_Toc447134563)

[1. Введение 3](#_Toc447134564)

[2. Настройка HTTPD 3](#_Toc447134565)

[2.1 Создание Windows-сервиса для HTTPD 4](#_Toc447134566)

[2.1.1 Коннектор mod\_jk 4](#_Toc447134567)

[3. Установка и настройка Tomcat 4](#_Toc447134568)

[3.1 Установка 4](#_Toc447134569)

[3.2 Настройка 4](#_Toc447134570)

[3.2.1 Внешние переменные 4](#_Toc447134571)

[3.2.2 Коннектор tomcat-jk2 4](#_Toc447134572)

[3.2.3 Настройка server.xml 5](#_Toc447134573)

[3.2.4 Настройка источников данных, пулов соединений 5](#_Toc447134574)

[3.3 Создание Windows-сервисов для Tomcat 5](#_Toc447134575)

[3.3.1 Проверка работоспособности сервера CAS 7](#_Toc447134576)

[4. Настройка переменных окружения 9](#_Toc447134577)

[5. Дистрибутивы, библиотеки 9](#_Toc447134578)

[5.1 Дистрибутив Apache Tomcat 9](#_Toc447134579)

[5.2 Коннектор Tomcat-JK2 9](#_Toc447134580)

[5.3 Дистрибутив Apache HTTPD 9](#_Toc447134581)

[5.4 Коннектор MOD\_JK 9](#_Toc447134582)

[6. Приложение 1. Настройка HTTPD. 9](#_Toc447134583)

[6.1 Настройка кэширования 9](#_Toc447134584)

[6.2 Настройки, связанные с безопасностью сервера приложений 10](#_Toc447134585)

[6.2.1 Скрытие названия и версии сервера приложений 10](#_Toc447134586)

[7. Приложение 3.Пример настройки файла *server.xml* в Tomcat. 10](#_Toc447134587)

[8. Приложение 4.Пример настройки файла mod\_jk.conf 13](#_Toc447134588)

[9. Приложение 5 14](#_Toc447134589)

[10. Ссылки 14](#_Toc447134590)

# Введение

Документ содержит основные положения по настройке системы серверов Apache HTTP Server (HTTPD) + Tomcat.



Рисунок 1.Общая схема взаимодействия серверов Apache HTTPD, Tomcat и СУБД

Архитектура предполагает простую и удобную возможность вертикального и горизонтального масштабирования функционала приложений, разрабатываемых в стандарте J2EE, при увеличении интенсивности запросов к системе и (или) увеличении одновременно работающего функционала.

Apache HTTPD выполняет функции:

* Reverse proxy для экземляров серверов Tomcat, обеспечивая обработку внешних запросов по стандартному порту 80;
* Балансировку нагрузки на экземпляры серверов Tomcat;
* Эффективный доступ к статическим ресурсам
* TODO

# Настройка HTTPD

Настройка сервера HTTPD для взаимодействия с Tomcat включает:

* Настройку файла *mod\_jk.conf*
* Настройку файла *%APACHE\_HOME%/conf/httpd.conf*

Файл *mod\_jk.conf* создаётся в ручном режиме и размещается в директории *%TOMCAT\_HOME%/conf*  (предлагаемый режим, когда *mod\_jk.conf* автоматически создаётся при запуске Tomcat, работает некорректно).

В *п.* приведён пример **ручной** настройки, обеспечивающей работу с встроенными примерами приложений Tomcat)

В конце файла *httpd.conf* необходимо подключить файл *mod\_jk.conf*:

…

*# Include mod\_jk's specific configuration file*

*Include %TOMCAT\_HOME%/conf/mod\_jk.conf*

…

В приложении [Приложение1.Настройка кэширования](#_Настройка_кэширования) приведён перечень общих умолчательных настроек Apache HTTPD.

При необходимости для каждого из экземпляров Tomcat могут быть выставлены дополнительные настройки.

## Создание Windows-сервиса для HTTPD

Windows-сервис используемой версии HTTPD создаётся командой

***httpd -k install***,

которая запускается из директории *%APACHE\_HOME%/bin/*

### Коннектор mod\_jk

mod\_jk – коннектор, поддерживающий взаимодействие Apache HTTPD-Tomcat по протоколу AJP со стороны HTTPD.

Она добавляется в директорию %APACHE\_HOME%/modules.

Скачать можно здесь (см.).

# Установка и настройка Tomcat

Используемый дистрибутив Tomcat [см.Дистрибутив Tomcat] ориентирован на работу в 64-разрядной Windows, для её работы необходима соответствующая 64-разрядная версия Java 1.7+.

## Установка

Установка Tomcat выполняется копированием из дистрибутива [см.Дистрибутив Tomcat]

## Настройка

### Внешние переменные

Внешние переменные для Tomcat определяются в файле *%TOMCAT\_HOME%/bin/setEnv.bat* (здесь и далее используется нотация Windows), который автоматически вызывается в самом начале работы*.*

Для Tomcat используемой версии необходима версия Java 1.7+. Для неё в ***setEnv.bat*** задаётся переменная *JAVA\_HOME*, а директория *%JAVA\_HOME%/bin* прописывается в переменной *PATH*, например:

*set JAVA\_HOME=D:/Program Files/Java/jdk1.8.0\_45*

*set PATH=%JAVA\_HOME%/bin;% PATH%*

### Коннектор tomcat-jk2

Tomcat-jk2 – библиотека (jar), реализующая коннектор, поддерживающий взаимодействие Apache HTTPD-Tomcat по протоколу AJP со стороны Tomcat.

Она добавляется в директорию %TOMCAT\_HOME%/lib.

Скачать можно здесь (см. Библиотека Tomcat-JK2).

### Настройка server.xml

См.пример настройки в

### Настройка источников данных, пулов соединений

Настройка пулов соединений для источников данных Tomcat выполняется в файле *%TOMCAT\_HOME%/conf/context.xml*.

В общем случае для каждого realm-а используется отдельный источник данных, работающий в отдельной схеме БД. Ниже приводится пример настройки пула соединений для источника данных: *jdbc/JEPDS* , настроенного на Oracle:

…

*<Context>*

*…*

*<Resource*

*name="jdbc/ JEPDS "*

*auth="Container"*

*type="javax.sql.DataSource"*

*driverClassName="oracle.jdbc.OracleDriver"*

*url="jdbc:oracle:thin:@//host:1521/DB1"*

*username="dbschema1"*

*password=" welcome "*

*maxActive="100"*

*maxIdle="30"*

*maxWait="10000"*

*jmxEnabled="true" />*

*…*

*<Context>*

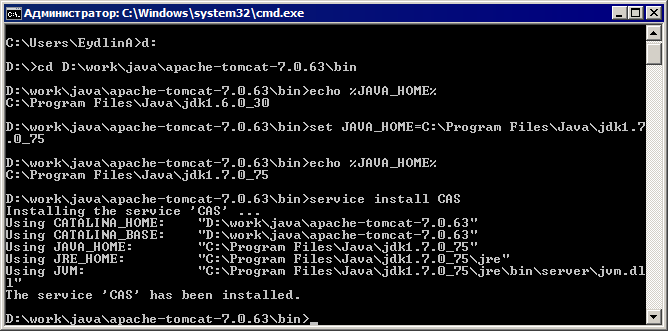
## Создание Windows-сервисов для Tomcat

TODO: Актуализировать, заменить CAS на Tomcat\_1.

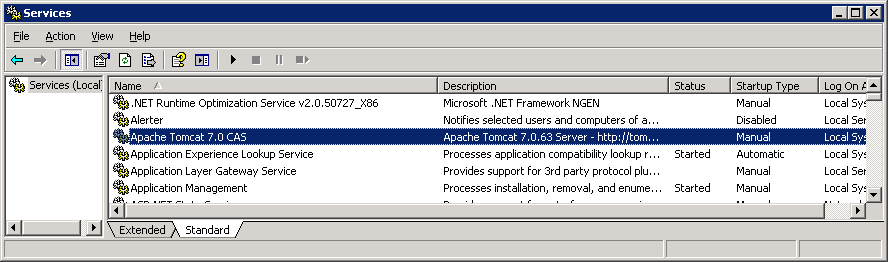
Для того, чтобы создать Windows-сервис для Tomcat, например, с именем *Tomcat\_1*, нужно перейти в директорию *<TomcatHome>/bin*,открыть в нём консольное окно и последовательно выполнить команды:

*set java\_home=<TomcatJavaHome>*

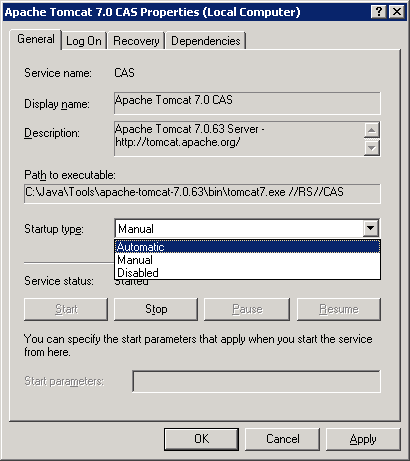
*service install Tomcat\_1*



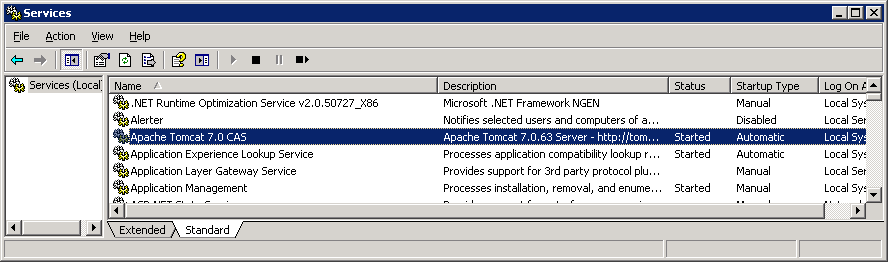
После этого в окне *Services* в списке сервисов должен появиться элемент с названием *Apache Tomcat 7.0 CAS* (в некоторых версиях - просто *CAS*):



Далее необходимо настроить сервис на автоматический запуск:

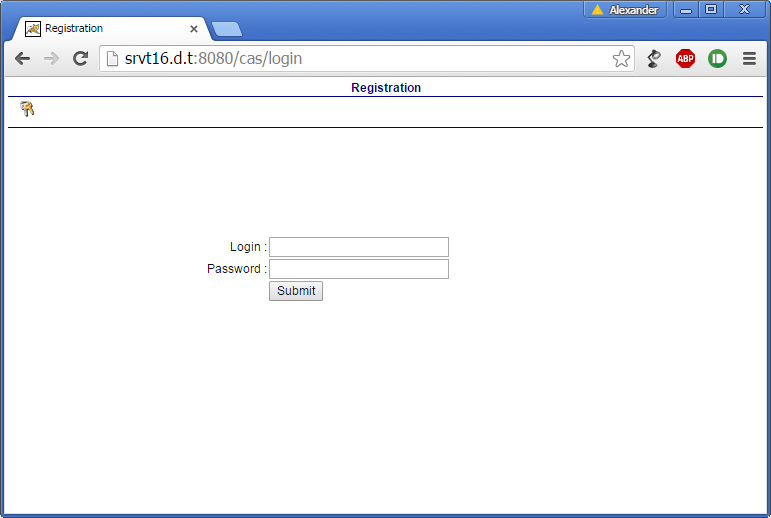


После запуска сервиса в окне *Services* CAS должен иметь вид:

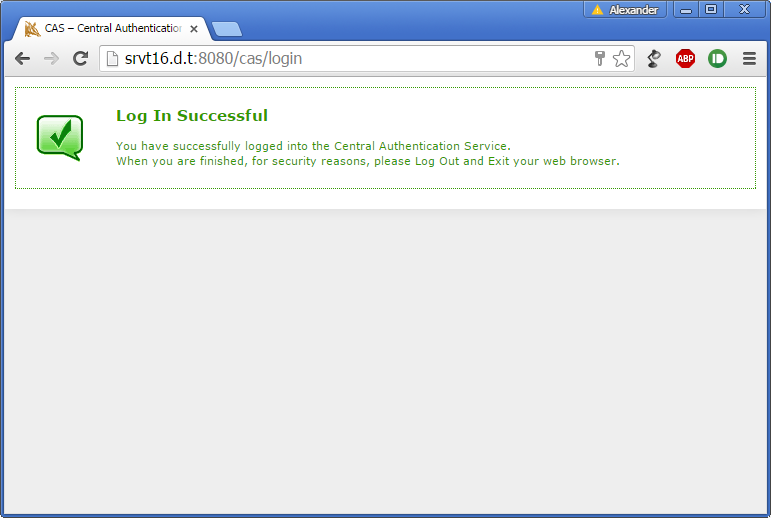


### Проверка работоспособности сервера CAS

Для проверки работоспособности установленного сервера CAS в окне браузера нужно зайти по адресу *<host>:8080/cas/login* (для ICA - *<host>:8080/cas-ica/login*). Должна появиться страница вида:



После ввода правильных значений логин/пароль должна появиться страница:



Для удаления сервиса Tomcat используется команда

*service uninstall CAS*

# Настройка переменных окружения

TODO

# Дистрибутивы, библиотеки

В состав дистрибутивов системы входят:

* Дистрибутив *Apache Tomcat*
* Библиотека-коннектор *Tomcat\_JK2*
* Дистрибутив Apache HTTPD
* Коннектор *mod\_jk*

## Дистрибутив Apache Tomcat

[apache-tomcat-7.0.68-windows-x64](http://apache-mirror.rbc.ru/pub/apache/tomcat/tomcat-7/v7.0.68/bin/apache-tomcat-7.0.68-windows-x64.zip)

## Коннектор Tomcat-JK2

Скачивается отсюда: [tomcat-jk2-5.0.28.jar](http://www.java2s.com/Code/JarDownload/tomcat-jk2/tomcat-jk2-5.0.18.jar.zip)

## Дистрибутив Apache HTTPD

[Дистрибутив httpd-2.2.31-x86-r3.zip](http://de.apachehaus.com/downloads/httpd-2.2.31-x86-r3.zip)

## Коннектор MOD\_JK

Скачивается отсюда: [tomcat-connectors-1.2.40-windows-i386-httpd-2.2.x.zip](http://apache-mirror.rbc.ru/pub/apach...s-1.2.40-windows-i386-httpd-2.2.x.zip)

# Установка приложений JepRia на Tomcat

Web-приложения, оформленные war-файлом или структурированные согласно стандарту J2EE, устанавливаются на Tomcat простым копированием в директорию *%TOMCAT\_HOME%/webapps*.

## Автоматизированная установка приложений с помощью Ant

Скрипт сборки приложений JepRia – *build.xml* – позволяет выполнять на Tomcat автоматизированную установку/удаление (deploy/undeploy) приложений средствами Ant при помощи целей (targets) *tomcat.deploy* и *tomcat.undeploy*.

## Используемые библиотеки

При установке/удалении приложений используются библиотеки:

*%TOMCAT\_HOME%/lib/catalina-ant.jar*

*%TOMCAT\_HOME%/lib/tomcat-coyote.jar*

*%TOMCAT\_HOME%/lib/tomcat-util.jar*

В процессе настройки автоматизированной установки необходимо убедиться, чтобы директория *%ANT\_HOME%/lib* не содержала иных версий этих библиотек.

## Используемые переменные

В файле *deploy.properties* необходимо определить переменные *DEPLOYMENT\_PATH, LOGIN, PASSWORD*, например:

…

*# Параметры установки для локального Tomcat*

*DEPLOYMENT\_PATH=http://localhost:8080/manager/text*

*LOGIN=admin*

*PASSWORD=admin*

…

Кроме этого должны быть определены переменные *TOMCAT\_HOME* и *JDBC\_HOME*, например:

*# Директория установки Tomcat*

*TOMCAT\_HOME=D:/Java/Tools/apache-tomcat-7.0.68-windows-x64*

*# Директория размещения библиотек JDBC*

*JDBC\_HOME=D:/Java/Tools/BinRepo/com/oracle*

## Создание администратора Tomcat

Автоматизированная установка выполняется от имени специального пользователя - администратора Tomcat, которого необходимо создать, определив его в файле *%TOMCAT\_HOME%/config/tomcat-users.xml* следующим образом (пароль, разумеется, вариабелен):

*<?xml version='1.0' encoding='utf-8'?>*

*<tomcat-users>*

*<role rolename="manager-gui"/>*

*<role rolename="manager-script"/>*

*<role rolename="manager-jmx"/>*

*<role rolename="manager-status"/>*

*<user username="admin" password="admin" roles="manager-gui,manager-script,manager-jmx,manager-status"/>*

*</tomcat-users>*

# Приложение 1. Настройка HTTPD.

## Настройка кэширования

Для корректного кэширования и гарантированного обновления клиентских частей приложений при установке приложений на Tomcat необходимо установить следующие настройки кэширования в конфигурационном файле *httpd.conf* :

После правила запрета доступа к файлам, начинающихся с .ht:

*<Files ~ "^\.ht">*

*Order allow,deny*

*Deny from all*

*</Files>*

Добавить правила:

*#*

*# Accurate caching of JepRia applications.*

*#*

*<Files \*.nocache.\*>*

*ExpiresActive on*

*ExpiresDefault "now"*

*Header* ***merge*** *Cache-Control "public, max-age=0, must-revalidate"*

*</Files>*

*<Files \*.cache.\*>*

*ExpiresActive on*

*ExpiresDefault "now plus 1 year"*

*</Files>*

Полезные ссылки по настройке кэширования для GWT-приложений:

<http://www.gwtproject.org/doc/latest/DevGuideCompilingAndDebugging.html#perfect_caching>

<http://seewah.blogspot.ru/2009/02/gwt-tips-2-nocachejs-getting-cached-in.html>

## Настройки, связанные с безопасностью сервера приложений

### Скрытие названия и версии сервера приложений

Для того, чтобы скрыть версию и наименование сервера приложения, необходимо выполнить следующие действия в конфигурационном файле httpd.conf:

* 1. **Загрузить модуль mod\_security:**

В секции Dynamic Shared Object (DSO) Support прописать:

LoadModule security\_module  modules/ApacheModuleSecurity.dll

В секции добавления модулей (ClearModuleList):

AddModule mod\_security.c

* 1. **Установить значение опции ServerSignature:**

ServerSignature Off

* 1. **Установить значение опции SecServerSignature**

SecServerSignature “Unknown”

*Примечание 1*: Вместо “Unknown” может быть любое не пустое значение – фактически данное значение будет возвращаться в заголовке ответа “Server”

*Примечание 2:*  В том случае, если значение SecurityTokens по какой-то причине перекрыто, необходимо установить его в Full

# Приложение 3.Пример настройки файла *server.xml* в Tomcat.

Ниже приводится пример файла server.xml Tomcat, настроенного на взаимодействие с Apache HTTPD. Изменённые/добавленные фрагменты выделены ***жирным шрифтом***:

*<?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" />*

*<!--Initialize Jasper prior to webapps are loaded. Documentation at /docs/jasper-howto.html -->*

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

*<!-- 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" />*

***~~<Listener className="org.apache.jk.config.ApacheConfig" modJk="D:/Tools/Apache24/modules/mod\_jk.so" workersConfig="D:/Java/Tools/apache-tomcat-8.0.33/conf/jk/workers.properties" />~~***

*<!-- 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 (blocking & non-blocking)*

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

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

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

*-->*

*<Connector port="8080"* ***URIEncoding="UTF-8****" protocol="HTTP/1.1"*

*connectionTimeout="20000"*

*redirectPort="8443" />*

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

*<!--*

*<Connector executor="tomcatThreadPool"*

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

*connectionTimeout="20000"*

*redirectPort="8443" />*

*-->*

*<!-- Define a SSL HTTP/1.1 Connector on port 8443*

*This connector uses the BIO implementation that requires the JSSE*

*style configuration. When using the APR/native implementation, the*

*OpenSSL style configuration is required as described in the APR/native*

*documentation -->*

*<!--*

*<Connector port="8443" protocol="org.apache.coyote.http11.Http11Protocol"*

*maxThreads="150" SSLEnabled="true" scheme="https" secure="true"*

*clientAuth="false" sslProtocol="TLS" />*

*-->*

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

*<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />*

*<!-- 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">*

***<Listener className="org.apache.jk.config.ApacheConfig" append="true" forwardAll="false" modJk="D:/Tools/Apache24/modules/mod\_jk.so" />***

*<!-- 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>*

# Приложение 4.Пример настройки файла mod\_jk.conf

Пример настройки приводится для обеспечения запуска примеров Tomcat:

*########## Auto generated on Wed Mar 30 19:21:54 MSK 2016##########*

*<IfModule !mod\_jk.c>*

*LoadModule jk\_module "D:/Apache22/modules/mod\_jk.so"*

*</IfModule>*

*JkWorkersFile "D:/Java/Tools/apache-tomcat-7.0.68-windows-x64/conf/jk/workers.properties"*

*JkLogFile "D:/Java/Tools/apache-tomcat-7.0.68-windows-x64/logs/mod\_jk.log"*

*JkLogLevel emerg*

*# Select the log format*

*JkLogStampFormat "[%a %b %d %H:%M:%S %Y]"*

*# JkOptions indicate to send SSL KEY SIZE,*

*JkOptions +ForwardKeySize +ForwardURICompat -ForwardDirectories*

*# JkRequestLogFormat set the request format*

*JkRequestLogFormat "%w %V %T"*

*# Send everything for context /ws to worker ajp13*

*JkMount /ws ajp13*

*JkMount /ws/\* ajp13*

*# Send everything for context /examples to worker ajp13*

*JkMount /examples ajp13*

*JkMount /examples/\* ajp13*

# Приложение 5



Рисунок . Снижение нагрузки с Tomcat посредством переноса ответственности за статические ресурсы на HTTPD.



Рисунок .Схема балансировки нагрузки.

# Ссылки

1. [Apache HTTPD, mod\_jk, Apache Tomcat, Linux](http://programador.ru/apache-mod_jk-tomcat/)
2. [Using Apache HTTP Server on Microsoft Windows](http://httpd.apache.org/docs/current/platform/windows.html#down)
3. [Installing Tomcat 6 with Java 6 and Apache 2.2 on Windows Server 2003 x64](http://www.anchor.com.au/hosting/dedicated/tomcat_with_apache_on_windows_x64)
4. Полезный ресурс: [Installing Apache and Tomcat and Mysql on Windows 7](http://boldinventions.com/index.php?option=com_content&view=article&id=95%3Ainstalling-apache-and-tomcat-and-mysql-on-window-7&catid=34%3Acategory-electronics-articles&Itemid=53)
5. Помогло при настройке *mod\_jk.conf*: [How To Configure Tomcat to work with Apache](http://www.ntu.edu.sg/home/ehchua/programming/howto/apacheplustomcat_howto.html)