System property

|  |  |
| --- | --- |
| property name | default |
| openfire.lib.dir | ../lib |
| openfireHome | not set |
| developmentMode | false |
| pluginDirs |  |
|  |  |
|  |  |

admin server property

|  |  |
| --- | --- |
| property name | default |
| xmpp.domain |  |
| log.debug.enabled |  |
| xmpp.jmx.enabled | false |
| xmpp.jmx.secure | true |
| xmpp.jmx.port | 1099 |
| xmpp.socket.ssl.storeType | jks, PKCS12 |
| xmpp.socket.ssl.keystore |  |
| xmpp.socket.ssl.keypass |  |
| xmpp.socket.ssl.client.truststore |  |
| xmpp.socket.ssl.trustpass |  |
| xmpp.socket.ssl.truststore |  |
| xmpp.socket.ssl.trustpass |  |
| admin.authorizedJIDs |  |
| cache.clustering.local.class | org.jivesoftware.util.cache.DefaultLocalCacheStrategy |
| cache.clustering.clustered.class | org.jivesoftware.openfire.plugin.util.cache.ClusteredCacheFactory |
| xmpp.server.session.idle | 10 \* 60 \* 1000 ms |
| xmpp.audit.active |  |
| xmpp.audit.message |  |
| xmpp.audit.presence |  |
| xmpp.audit.iq |  |
| xmpp.audit.xpath |  |
| xmpp.audit.filter.xpath <ToDo> |  |
| xmpp.audit.totalsize |  |
| xmpp.audit.filesize |  |
| xmpp.audit.days |  |
| xmpp.audit.logtimeout |  |
| xmpp.audit.logdir |  |
| xmpp.audit.ignore |  |
| xmpp.client.roster.active | true |
| provider.roster.className | org.jivesoftware.openfire.roster.DefaultRosterItemProvider |
| xmpp.privateStorageEnabled | true |
| xmpp.session.conflict-limit | 0 |
| xmpp.socket.heapBuffer | true |
| **xmpp.server.processing.core.threads** | 2 |
| **xmpp.server.processing.max.threads** | 50 |
| **xmpp.server.processing.queue** | 50 |
| **xmpp.client.processing.threads** | 16 |
| **xmpp.client.maxReadBufferSize** | 10MB |
| **xmpp.client\_ssl.processing.threads** | 16 |
| **xmpp.client\_ssl.maxReadBufferSize** | 10MB |
| **xmpp.socket.ssl.algorithm** | TLS |
| xmpp.component.processing.threads | 16 |
| xmpp.multiplex.processing.threads | 16 |
| xmpp.auth.iqauth | true |
| update.lastCheck | 0 (check in every 30s) |
| xmpp.pubsub.enabled | true |
| xmpp.pubsub.service | pubsub |
| xmpp.pubsub.root.nodeID | “” |
| xmpp.pubsub.root.creator | null |
| xmpp.pubsub.multiple-subscriptions | true |
| xmpp.pubsub.sysadmin.jid | null |
| xmpp.pubsub.create.anyone | false |
| xmpp.pubsub.create.jid | null |
| **user.usePlainPassword** |  |
| xmpp.auth.anonymous |  |
|  |  |

plugin property

|  |  |
| --- | --- |
| hazelcast.config.jmx.enabled | false |
|  |  |
|  |  |
|  |  |

openfile.xml

<xmpp><domain></domain></xmpp>

<log><debug><enabled>true</enabled></debug></log>

adminConsole.serverThreads

adminConsole.port

adminConsole.securePort

adminConsole.interface

network.interface

spdy.protocol => npn (Next Protocol Negotiation)

admin.authorizedJIDs

Server Statistics

|  |  |
| --- | --- |
| org.jivesoftware.openfire.cluster | GetBasicStatistics |
| org.jivesoftware.openfire.audit.spi | AuditManagerImpl |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

SSL

http://www.igniterealtime.org/builds/openfire/docs/latest/documentation/ssl-guide.html

http://polinwei.blogspot.tw/2013/02/java-keytoolmicrosoft-active-directory.html

http://www.howtogeek.com/107415/it-how-to-create-a-self-signed-security-ssl-certificate-and-deploy-it-to-client-machines/

gen keystore

1. keytool -storepasswd -keystore keystore
2. keytool -genkey -keystore keystore -alias htc.com -keyalg RSA -keysize 1024 -validity 365
3. keytool -genkey -keystore keystore -alias htc.com -keysize 1024 -validity 365
4. **keytool -delete -keystore keystore -alias rsa**
5. **keytool -delete -keystore keystore -alias dsa**
6. keytool -list  -v -keystore keystore

keytool -genkey -alias **edi-prod**(別名) -keypass **password**(別名密碼) -keyalg **RSA**(演算法) -keysize **1024**(金鑰長度) -validity **365**(有效期，天單位) -keystore **./edi.keystore**(指定生成證書的位置和證書名稱) -storepass **123456**(獲取keystore資訊的密碼)

-Dopenfire.lib.dir="${workspace\_loc:openfire}\target\openfire\lib" -DopenfireHome="${workspace\_loc:openfire}\target\openfire"

-Xbootclasspath/p:${workspace\_loc:openfire}\target\openfire\lib\npn-boot.jar

npn-boot.jar with java 1.75

// Load core module

// Load standard modules

**~~loadModule(PresenceSubscribeHandler.class.getName());~~**

**~~loadModule(PresenceUpdateHandler.class.getName());~~**

~~loadModule(IQOfflineMessagesHandler.class.getName());~~

loadModule(IQPEPHandler.class.getName());

loadModule(IQPEPOwnerHandler.class.getName());

loadModule(MulticastDNSService.class.getName());

loadModule(IQSharedGroupHandler.class.getName());

loadModule(AdHocCommandHandler.class.getName());

~~loadModule(DefaultFileTransferManager.class.getName());~~

~~loadModule(FileTransferProxy.class.getName());~~

~~loadModule(MediaProxyService.class.getName());~~

loadModule(PubSubModule.class.getName());

loadModule(IQDiscoInfoHandler.class.getName());

loadModule(IQDiscoItemsHandler.class.getName());

loadModule(MultiUserChatManager.class.getName());

// Load this module always last since we don't want to start listening for clients

// before the rest of the modules have been started

loadModule(ConnectionManagerImpl.class.getName());

Some sample JID's:

* user@example.com
* user@example.com/home
* example.com

~~start~~

~~initialize~~

~~locateOpenfire~~

~~verifyHome~~

~~isStandAlone~~

~~Terminator~~

~~ShutdownHookThread~~

shutdownServer

~~package org.jivesoftware.openfire.starter~~

<query xmlns='http://jabber.org/protocol/disco#info'>

<feature var='urn:xmpp:carbons:2'/>

</query>

~~loadModule(PrivateStorage.class.getName());~~

// Load core modules

~~loadModule(SessionManager.class.getName());~~

~~loadModule(PacketRouterImpl.class.getName());~~

~~loadModule(IQRouter.class.getName());~~

~~loadModule(MessageRouter.class.getName());~~

~~loadModule(PresenceRouter.class.getName());~~

~~loadModule(PacketDelivererImpl.class.getName());~~

~~loadModule(TransportHandler.class.getName()); 不用看~~

~~loadModule(OfflineMessageStrategy.class.getName());~~

~~loadModule(OfflineMessageStore.class.getName());~~

~~loadModule(VCardManager.class.getName());~~

~~loadModule(IQBindHandler.class.getName());~~

~~loadModule(IQSessionEstablishmentHandler.class.getName());~~

~~loadModule(IQAuthHandler.class.getName());~~

~~loadModule(IQPingHandler.class.getName());~~

~~loadModule(IQPrivateHandler.class.getName());~~

~~loadModule(IQRegisterHandler.class.getName());~~

~~loadModule(IQRosterHandler.class.getName());~~

~~loadModule(IQTimeHandler.class.getName());~~

~~loadModule(IQEntityTimeHandler.class.getName());~~

~~loadModule(IQvCardHandler.class.getName());~~

~~loadModule(IQVersionHandler.class.getName());~~

~~loadModule(IQLastActivityHandler.class.getName());~~

~~loadModule(IQOfflineMessagesHandler.class.getName());~~

~~loadModule(IQPEPHandler.class.getName());~~

~~loadModule(IQPEPOwnerHandler.class.getName());~~

~~loadModule(MulticastDNSService.class.getName());~~

~~loadModule(IQSharedGroupHandler.class.getName());~~

~~loadModule(IQPrivacyHandler.class.getName());~~

~~loadModule(DefaultFileTransferManager.class.getName());~~

~~loadModule(FileTransferProxy.class.getName());~~

~~loadModule(IQDiscoInfoHandler.class.getName());~~

~~loadModule(IQMessageCarbonsHandler.class.getName());~~

~~loadModule(MediaProxyService.class.getName());~~

~~loadModule(UpdateManager.class.getName());~~

~~loadModule(FlashCrossDomainHandler.class.getName());~~

loadModule(ClearspaceManager.class.getName()); 不用看

~~loadModule(ConnectionManagerImpl.class.getName());~~

~~loadModule(RoutingTableImpl.class.getName());~~

~~loadModule(AuditManagerImpl.class.getName());~~

~~loadModule(RosterManager.class.getName());~~

~~loadModule(MulticastRouter.class.getName());~~

~~loadModule(PacketTransporterImpl.class.getName());~~

loadModule(PresenceSubscribeHandler.class.getName());

loadModule(PresenceManagerImpl.class.getName());

loadModule(PresenceUpdateHandler.class.getName());

loadModule(InternalComponentManager.class.getName());

loadModule(AdHocCommandHandler.class.getName());

loadModule(PubSubModule.class.getName());

loadModule(IQDiscoItemsHandler.class.getName());

loadModule(MultiUserChatManager.class.getName());

http://xmpp.org/rfcs/rfc6121.html

http://xmpp.org/registrar/namespaces.html

privateStorage 私人東西

privateList 白名單 黑名單