Watcher

This is a simple util application written to assist me with developement and testing in a test environment with mutiple applications on a single Tomcat.

I have written this because, as a developer I had no access to Tomcat. I had to go to sys admin for each log, configuration etc. But with this application I can check everything I need via web browser and without bothering anyone else.

It should be used only in test environment!

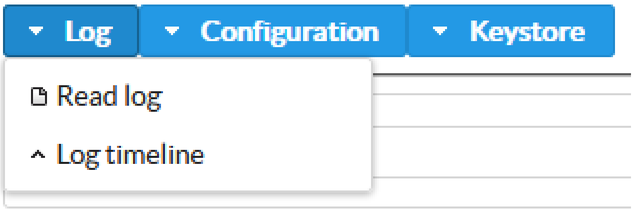
Author: Anđelko Spevec

Year: 2017

Contents

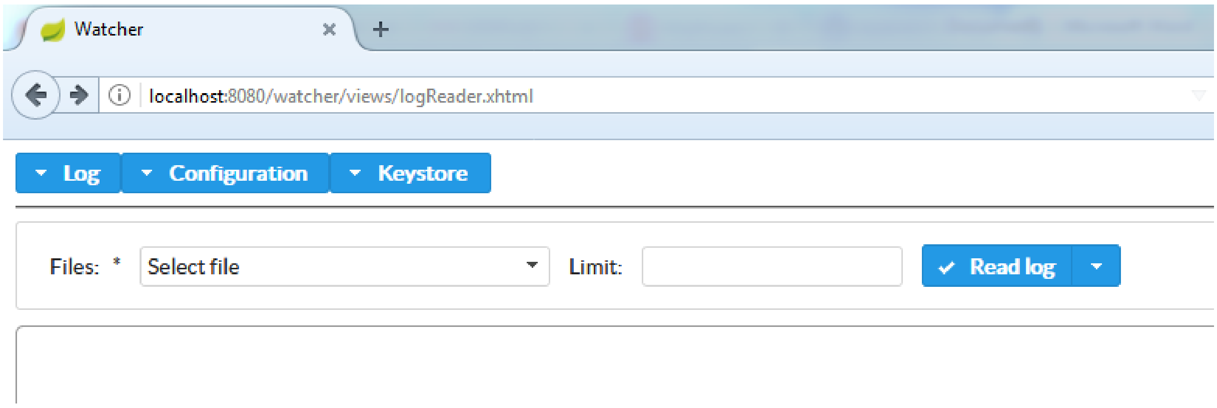
Log

This module is used for reading of log files from Tomcat server and their representation.



Reading log files

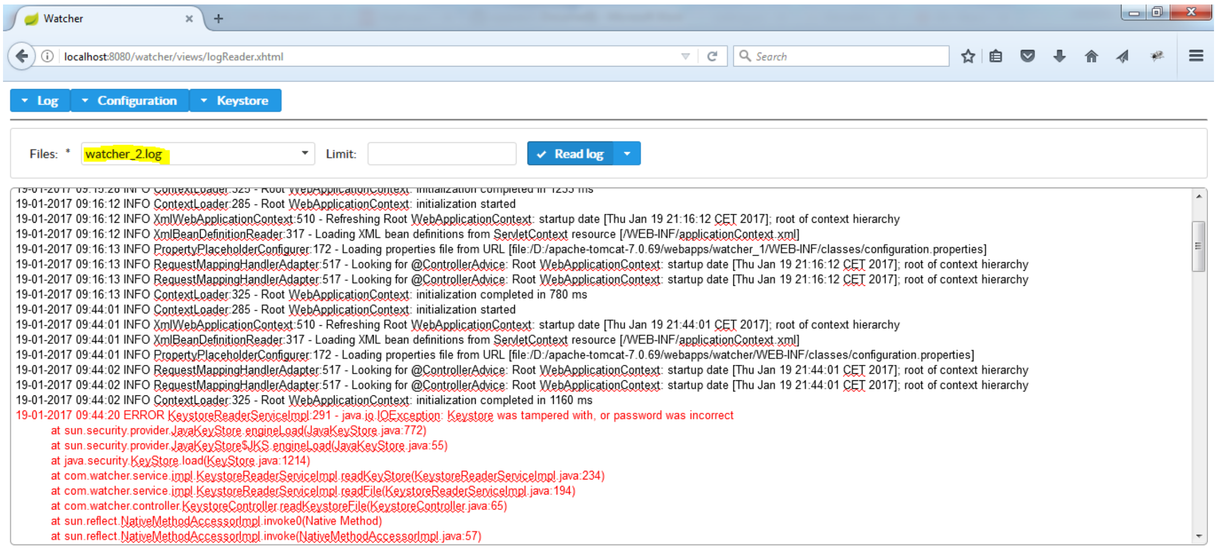
URL: http://localhost:8080/watcher/views/logReader.xhtml



Files: * Select file

Limit:

From *Files* dropdown menu you can choose from log files that are in Tomcat logs folder. After you select a file, you can click on *Read log* and content of file will be shown in editor below.



Files: * watcher_2.log

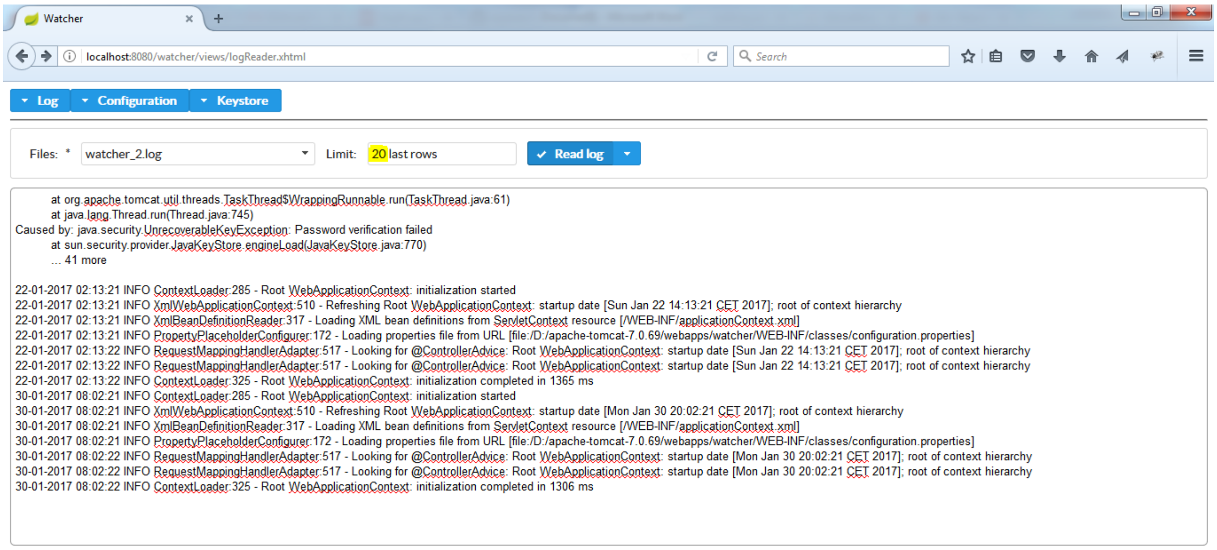
Limit:

```
19-01-2017 09:15:28 INFO ContextLoader:325 - Root WebApplicationContext: initialization completed in 1233 ms
19-01-2017 09:16:12 INFO ContextLoader:285 - Root WebApplicationContext: initialization started
19-01-2017 09:16:12 INFO XmlWebApplicationContext:510 - Refreshing Root WebApplicationContext: startup date [Thu Jan 19 21:16:12 CET 2017]; root of context hierarchy
19-01-2017 09:16:12 INFO XmlBeanDefinitionReader:317 - Loading XML bean definitions from ServletContext resource [/WEB-INF/applicationContext.xml]
19-01-2017 09:16:13 INFO PropertyPlaceholderConfigurer:172 - Loading properties file from URL [file:/D:/apache-tomcat-7.0.69/webapps/watcher_1/WEB-INF/classes/configuration.properties]
19-01-2017 09:16:13 INFO RequestMappingHandlerAdapter:517 - Looking for @ControllerAdvice: Root WebApplicationContext: startup date [Thu Jan 19 21:16:12 CET 2017]; root of context hierarchy
19-01-2017 09:16:13 INFO RequestMappingHandlerAdapter:517 - Looking for @ControllerAdvice: Root WebApplicationContext: startup date [Thu Jan 19 21:16:12 CET 2017]; root of context hierarchy
19-01-2017 09:16:13 INFO ContextLoader:325 - Root WebApplicationContext: initialization completed in 780 ms
19-01-2017 09:44:01 INFO ContextLoader:285 - Root WebApplicationContext: initialization started
19-01-2017 09:44:01 INFO XmlWebApplicationContext:510 - Refreshing Root WebApplicationContext: startup date [Thu Jan 19 21:44:01 CET 2017]; root of context hierarchy
19-01-2017 09:44:01 INFO XmlBeanDefinitionReader:317 - Loading XML bean definitions from ServletContext resource [/WEB-INF/applicationContext.xml]
19-01-2017 09:44:01 INFO PropertyPlaceholderConfigurer:172 - Loading properties file from URL [file:/D:/apache-tomcat-7.0.69/webapps/watcher/WEB-INF/classes/configuration.properties]
19-01-2017 09:44:02 INFO RequestMappingHandlerAdapter:517 - Looking for @ControllerAdvice: Root WebApplicationContext: startup date [Thu Jan 19 21:44:01 CET 2017]; root of context hierarchy
19-01-2017 09:44:02 INFO RequestMappingHandlerAdapter:517 - Looking for @ControllerAdvice: Root WebApplicationContext: startup date [Thu Jan 19 21:44:01 CET 2017]; root of context hierarchy
19-01-2017 09:44:02 INFO ContextLoader:325 - Root WebApplicationContext: initialization completed in 1160 ms
19-01-2017 09:44:20 ERROR KeystoreReaderServiceImpl:291 - java.io.IOException: Keystore was tampered with, or password was incorrect
	at sun.security.provider.JavaKeyStore.engineLoad(JavaKeyStore.java:772)
	at sun.security.provider.JavaKeyStore$JKS.engineLoad(JavaKeyStore.java:55)
	at java.security.KeyStore.load(KeyStore.java:1214)
	at com.watcher.service.impl.KeystoreReaderServiceImpl.readKeyStore(KeystoreReaderServiceImpl.java:234)
	at com.watcher.service.impl.KeystoreReaderServiceImpl.readFile(KeystoreReaderServiceImpl.java:194)
	at com.watcher.controller.KeystoreController.readKeystoreFile(KeystoreController.java:65)
	at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
	at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
```

All events and their stacktrace that are SEVERE, ERROR, WARN, DEBUG log level are shown in different color for easy viewing.

Reverse reading of log files

Sometimes you want to read just last X lines of a log file. For that you can fill *Limit field* and put in a number of last rows to read.



Files: * watcher_2.log

Limit: 20 last rows

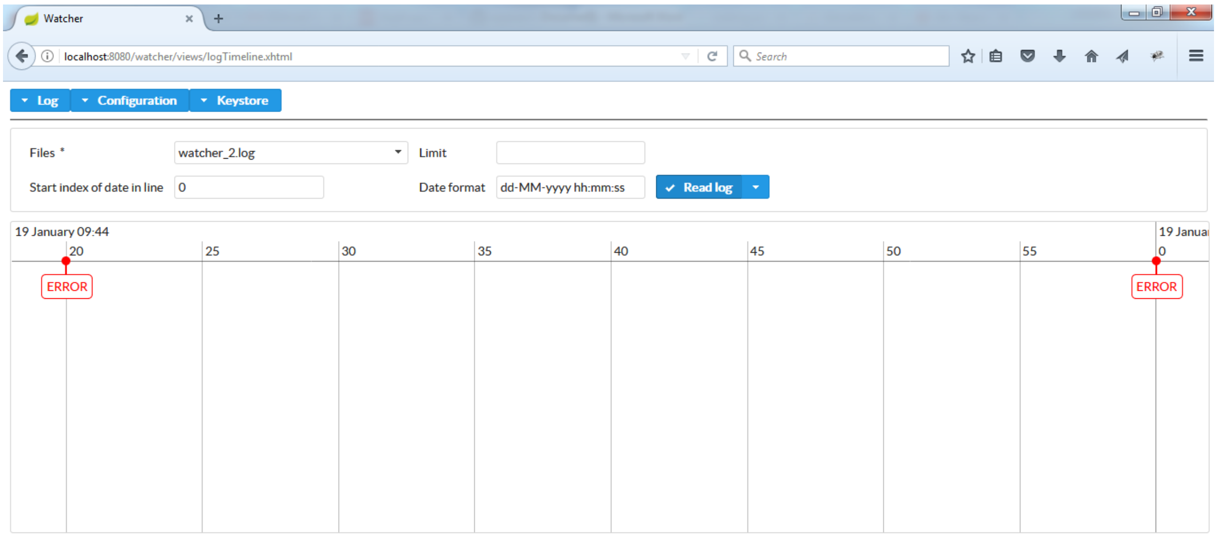
```
        at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)
        at java.lang.Thread.run(Thread.java:745)
Caused by: java.security.UnrecoverableKeyException: Password verification failed
        at sun.security.provider.JavaKeyStore.engineLoad(JavaKeyStore.java:770)
        ... 41 more

22-01-2017 02:13:21 INFO ContextLoader:285 - Root WebApplicationContext: initialization started
22-01-2017 02:13:21 INFO XmlWebApplicationContext:510 - Refreshing Root WebApplicationContext: startup date [Sun Jan 22 14:13:21 CET 2017]; root of context hierarchy
22-01-2017 02:13:21 INFO XmlBeanDefinitionReader:317 - Loading XML bean definitions from ServletContext resource [/WEB-INF/applicationContext.xml]
22-01-2017 02:13:21 INFO PropertyPlaceholderConfigurer:172 - Loading properties file from URL [file:/D:/apache-tomcat-7.0.69/webapps/watcher/WEB-INF/classes/configuration.properties]
22-01-2017 02:13:22 INFO RequestMappingHandlerAdapter:517 - Looking for @ControllerAdvice: Root WebApplicationContext: startup date [Sun Jan 22 14:13:21 CET 2017]; root of context hierarchy
22-01-2017 02:13:22 INFO RequestMappingHandlerAdapter:517 - Looking for @ControllerAdvice: Root WebApplicationContext: startup date [Sun Jan 22 14:13:21 CET 2017]; root of context hierarchy
22-01-2017 02:13:22 INFO ContextLoader:325 - Root WebApplicationContext: initialization completed in 1365 ms
30-01-2017 08:02:21 INFO ContextLoader:285 - Root WebApplicationContext: initialization started
30-01-2017 08:02:21 INFO XmlWebApplicationContext:510 - Refreshing Root WebApplicationContext: startup date [Mon Jan 30 20:02:21 CET 2017]; root of context hierarchy
30-01-2017 08:02:21 INFO XmlBeanDefinitionReader:317 - Loading XML bean definitions from ServletContext resource [/WEB-INF/applicationContext.xml]
30-01-2017 08:02:21 INFO PropertyPlaceholderConfigurer:172 - Loading properties file from URL [file:/D:/apache-tomcat-7.0.69/webapps/watcher/WEB-INF/classes/configuration.properties]
30-01-2017 08:02:22 INFO RequestMappingHandlerAdapter:517 - Looking for @ControllerAdvice: Root WebApplicationContext: startup date [Mon Jan 30 20:02:21 CET 2017]; root of context hierarchy
30-01-2017 08:02:22 INFO RequestMappingHandlerAdapter:517 - Looking for @ControllerAdvice: Root WebApplicationContext: startup date [Mon Jan 30 20:02:21 CET 2017]; root of context hierarchy
30-01-2017 08:02:22 INFO ContextLoader:325 - Root WebApplicationContext: initialization completed in 1306 ms
```

For reverse reading of log files, highlighting of log events is not implemented at the moment.

Timeline representation of log files

This option is used to read log file and to show every log event of SEVERE, ERROR,WARN,DEBUG log level in a timeline.

URL: http://localhost:8080/watcher/views/logTimeline.xhtml



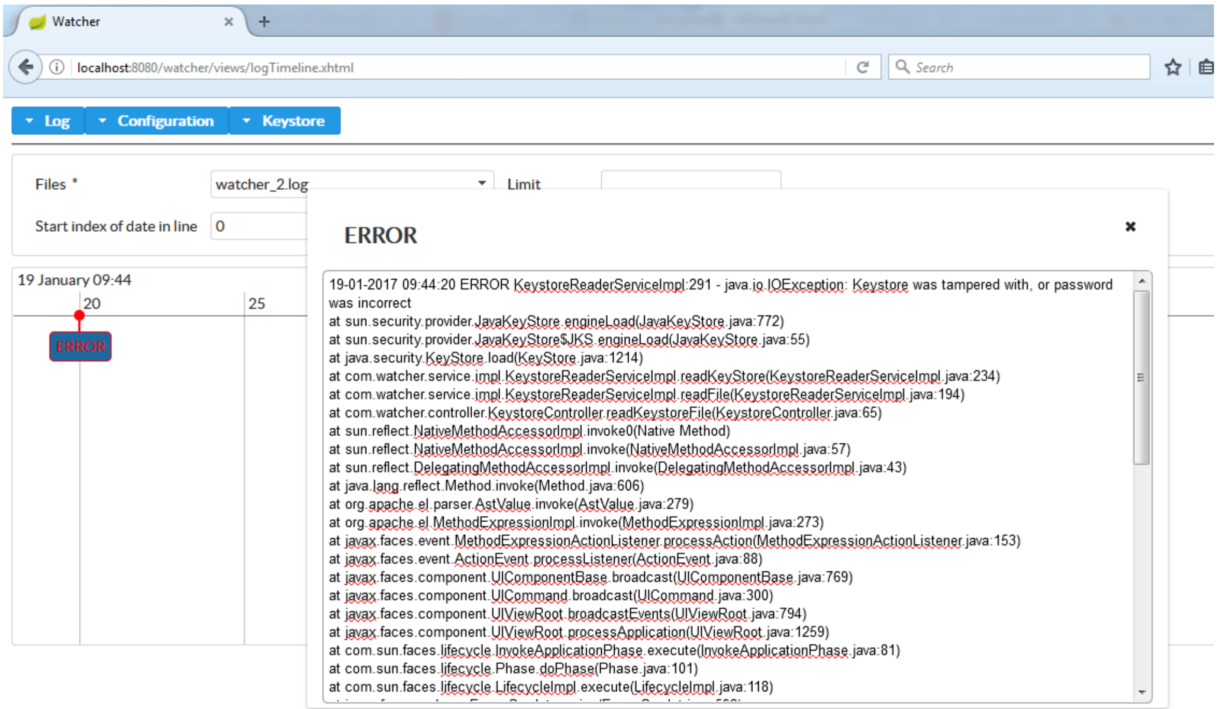
Files * watcher_2.log

Limit

Start index of date in line 0

Date format dd-MM-yyyy hh:mm:ss

Just like normal reading of file, you can choose from list of log files press *Read log*. After that events are shown in timeline. Timeline can be zoomed in/out and moved left/right. You can click on specific event and after that dialog box is shown with full stacktrace for this event.



ERROR

```
19-01-2017 09:44:20 ERROR KeystoreReaderServiceImpl:291 - java.io.IOException: Keystore was tampered with, or password
was incorrect
at sun.security.provider.JavaKeyStore.engineLoad(JavaKeyStore.java:772)
at sun.security.provider.JavaKeyStore$JKS.engineLoad(JavaKeyStore.java:55)
at java.security.KeyStore.load(KeyStore.java:1214)
at com.watcher.service.impl.KeystoreReaderServiceImpl.readKeyStore(KeystoreReaderServiceImpl.java:234)
at com.watcher.service.impl.KeystoreReaderServiceImpl.readFile(KeystoreReaderServiceImpl.java:194)
at com.watcher.controller.KeystoreController.readKeystoreFile(KeystoreController.java:65)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:606)
at org.apache.el.parser.AstValue.invoke(AstValue.java:279)
at org.apache.el.MethodExpressionImpl.invoke(MethodExpressionImpl.java:273)
at javax.faces.event.MethodExpressionActionListener.processAction(MethodExpressionActionListener.java:153)
at javax.faces.event.ActionEvent.processListener(ActionEvent.java:88)
at javax.faces.component.UIComponentBase.broadcast(UIComponentBase.java:769)
at javax.faces.component.UICommand.broadcast(UICommand.java:300)
at javax.faces.component.UIViewRoot.broadcastEvents(UIViewRoot.java:794)
at javax.faces.component.UIViewRoot.processApplication(UIViewRoot.java:1259)
at com.sun.faces.lifecycle.InvokeApplicationPhase.execute(InvokeApplicationPhase.java:81)
at com.sun.faces.lifecycle.Phase.doPhase(Phase.java:101)
at com.sun.faces.lifecycle.LifecycleImpl.execute(LifecycleImpl.java:118)
```

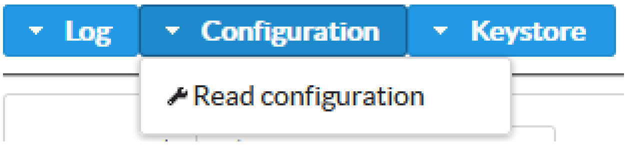
Date format is format used file configuring log4j.properties in parameter:

log4j.appender.file.layout.ConversionPattern=%d{dd-MM-yyyy hh:mm:ss} %-5p %c{1}:%L - %m%n

In parameter above format is dd-MM-yyyy hh:mm:ss and every line starts with date, so field Start index of date field is 0.

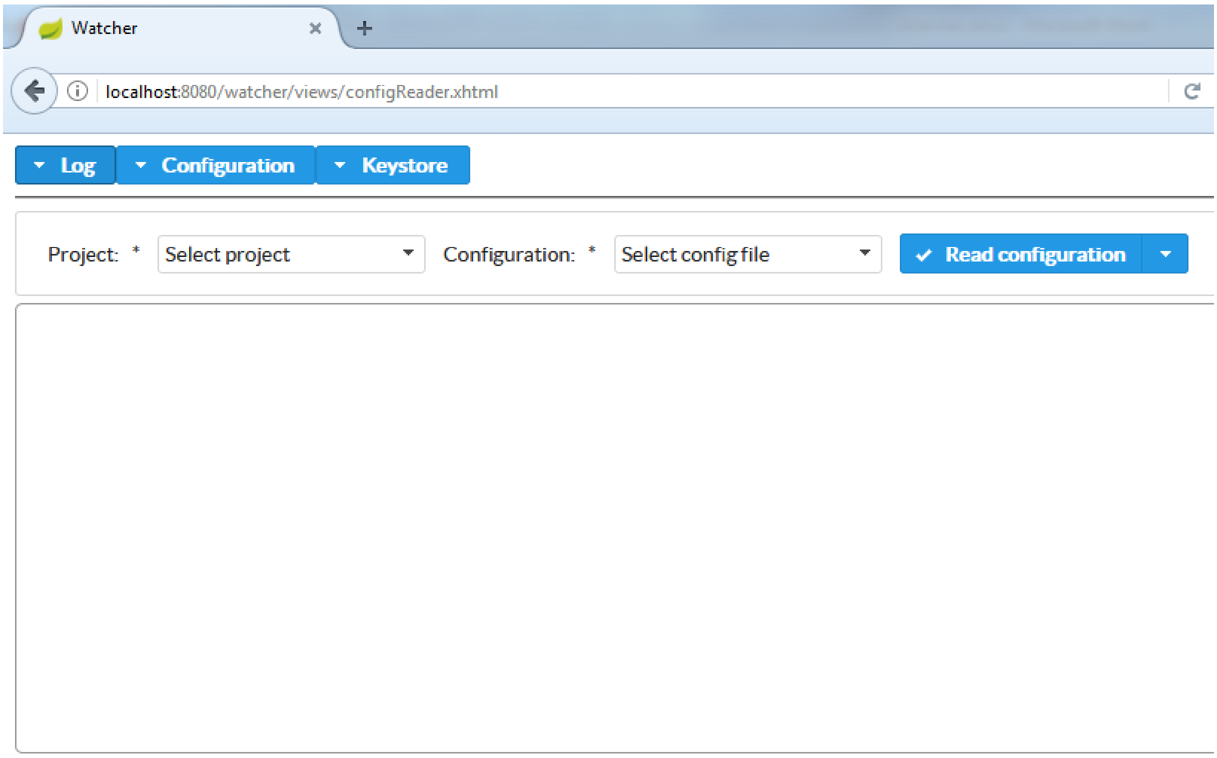
Configuration

This module is used for reading of configuration files of projects from Tomcat server and their representation.

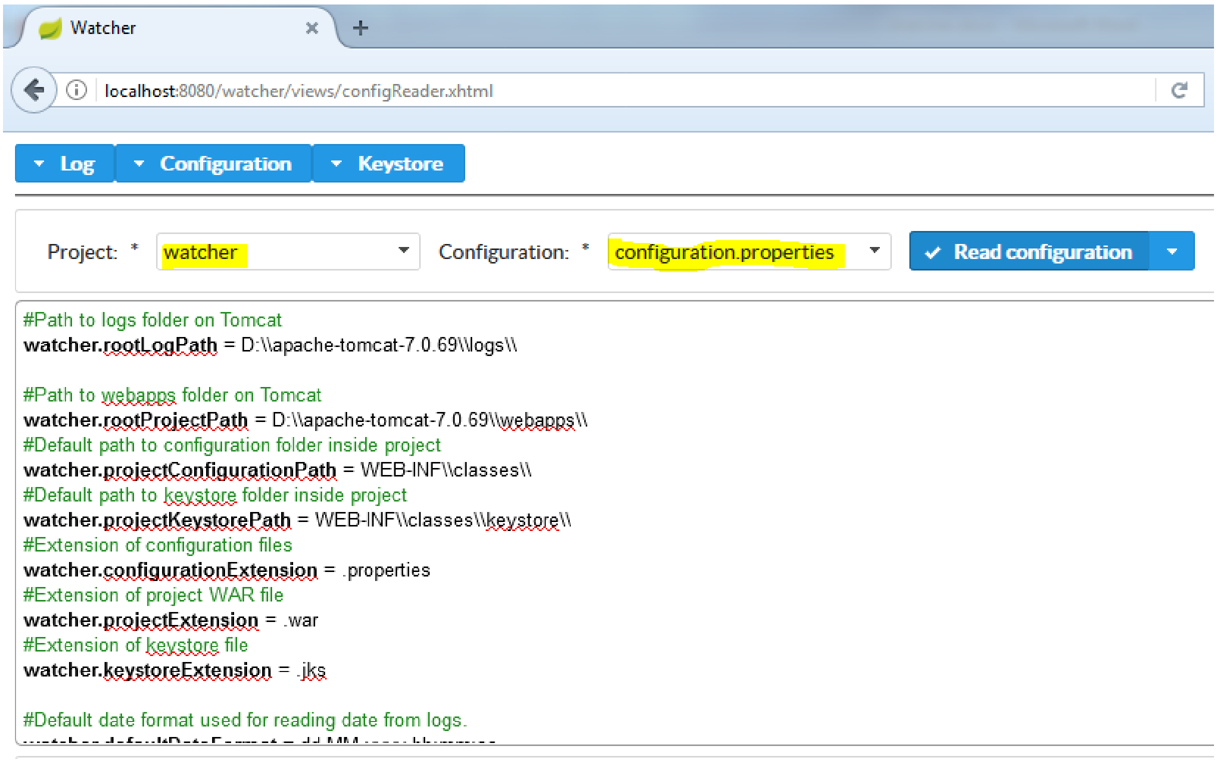


Reading configuration

URL: http://localhost:8080/watcher/views/configReader.xhtml



For reading configuration, you must first choose project from *Project* drowdown menu. After project is selected, list of configuration files from that project is updated. Now you can select configuration file from *Configuration* dropdown menu and click *Read configuration.*



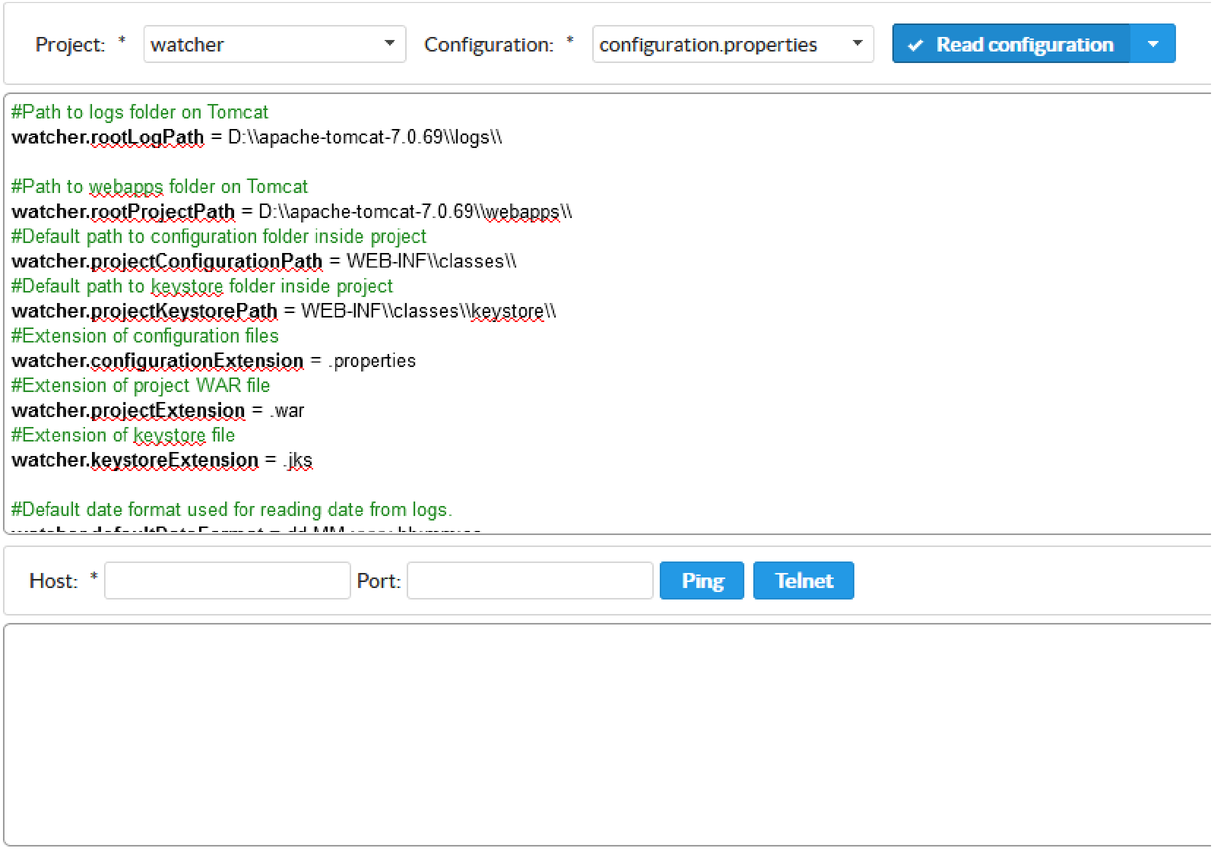
Project: * watcher

Configuration: * configuration.properties

```
#Path to logs folder on Tomcat
watcher.rootLogPath = D:\\apache-tomcat-7.0.69\\logs\\

#Path to webapps folder on Tomcat
watcher.rootProjectPath = D:\\apache-tomcat-7.0.69\\webapps\\
#Default path to configuration folder inside project
watcher.projectConfigurationPath = WEB-INF\\classes\\
#Default path to keystore folder inside project
watcher.projectKeystorePath = WEB-INF\\classes\\keystore\\
#Extension of configuration files
watcher.configurationExtension = .properties
#Extension of project WAR file
watcher.projectExtension = .war
#Extension of keystore file
watcher.keystoreExtension = .jks

#Default date format used for reading date from logs.
watcher.defaultDateFormat = dd.MM.yyyy hh:mm:ss
```

Check availability

Since URL to remote services are mostly defined in configuration file, access check for URL is included in this option.



Project: * watcher

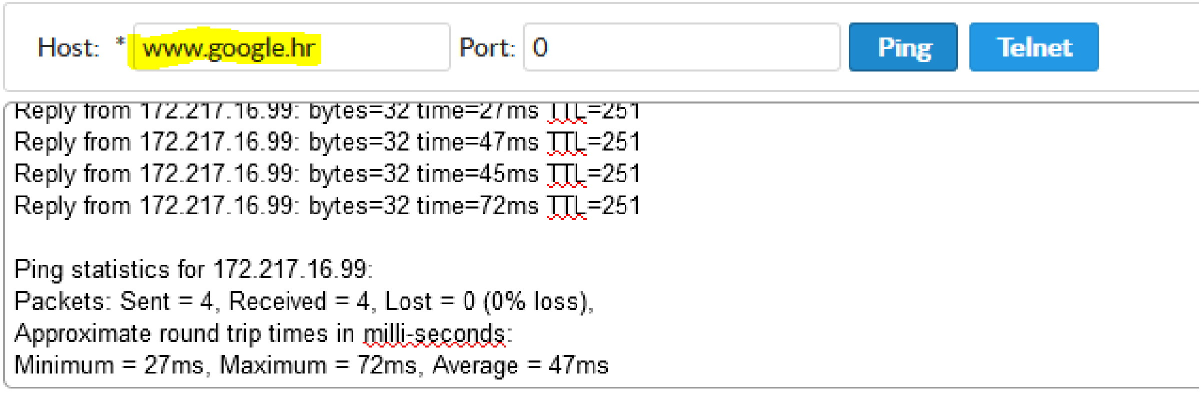
Configuration: * configuration.properties

```
#Path to logs folder on Tomcat
watcher.rootLogPath = D:\\apache-tomcat-7.0.69\\logs\\

#Path to webapps folder on Tomcat
watcher.rootProjectPath = D:\\apache-tomcat-7.0.69\\webapps\\
#Default path to configuration folder inside project
watcher.projectConfigurationPath = WEB-INF\\classes\\
#Default path to keystore folder inside project
watcher.projectKeystorePath = WEB-INF\\classes\\keystore\\
#Extension of configuration files
watcher.configurationExtension = .properties
#Extension of project WAR file
watcher.projectExtension = .war
#Extension of keystore file
watcher.keystoreExtension = .jks

#Default date format used for reading date from logs.
watcher.defaultDateFormat = dd.MM.yyyy hh:mm:ss
```

Host: *

Port:

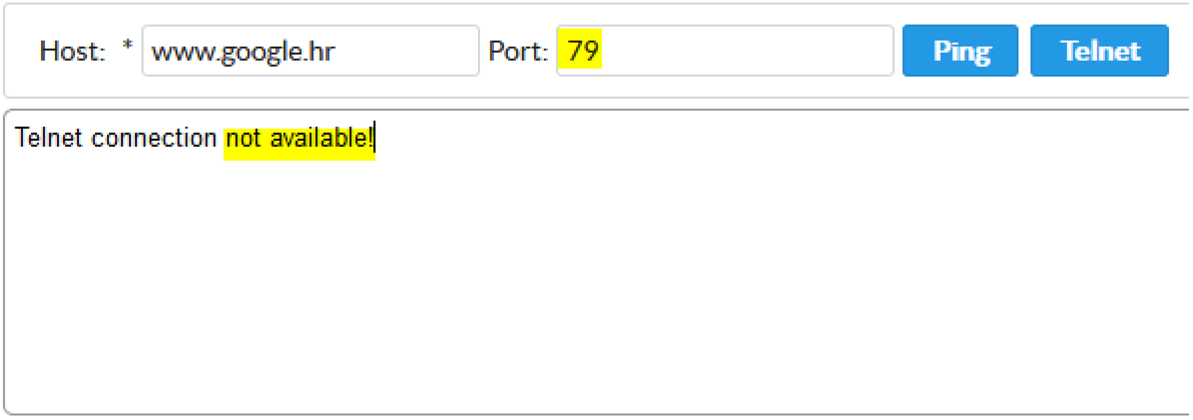
You have two options for checking access to URL. You can ping location by writing **host** in *Host* field and clicking *Ping* button. After that you need to wait for system to do ping command which may take a few seconds. After that full result is shown in window below.



Host: * www.google.hr Port: 0

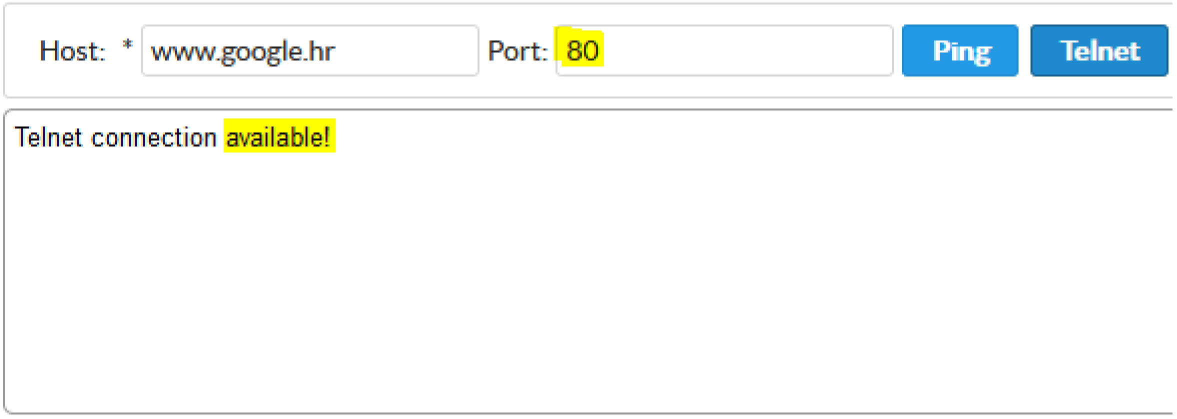
```
Reply from 172.217.16.99: bytes=32 time=27ms TTL=251
Reply from 172.217.16.99: bytes=32 time=47ms TTL=251
Reply from 172.217.16.99: bytes=32 time=45ms TTL=251
Reply from 172.217.16.99: bytes=32 time=72ms TTL=251

Ping statistics for 172.217.16.99:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 27ms, Maximum = 72ms, Average = 47ms
```

Other option is to try telnet connection to host and port. You need to fill *Host* and *Port* fields and click *Telnet* button. There are only two options for this connection so result shown is basic true/false.



Host: * www.google.hr Port: 79

```
Telnet connection not available!
```

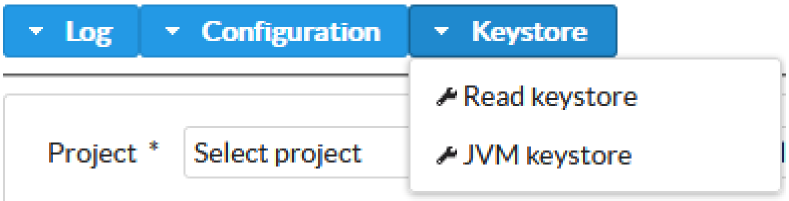


Host: * www.google.hr Port: 80

```
Telnet connection available!
```

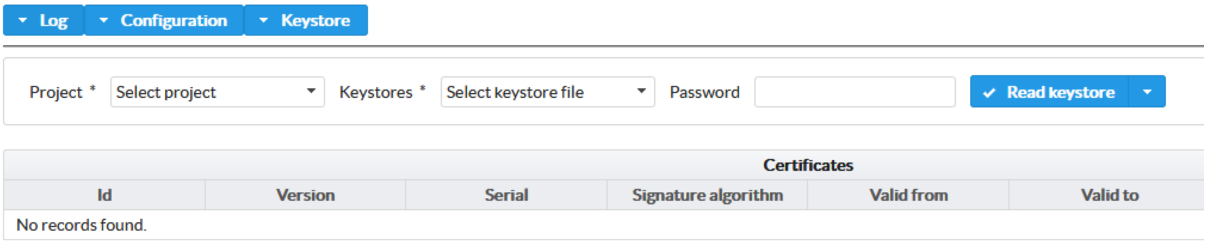
Keystore

This module is used for reading of keystore files from Tomcat server and to show their content.



Reading keystore

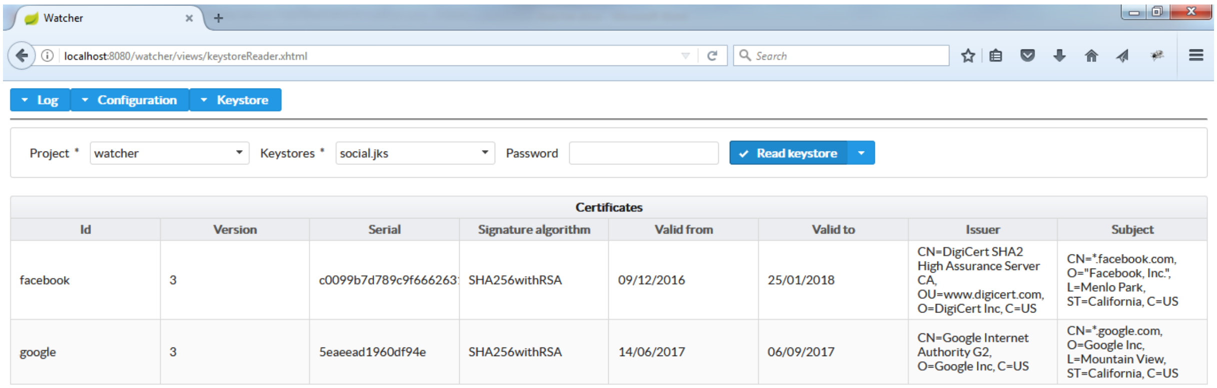
URL: http://localhost:8080/watcher/views/keystoreReader.xhtml



Certificates					
Id	Version	Serial	Signature algorithm	Valid from	Valid to
No records found.					

For reading keystores, you must first choose project from *Project* drowdown menu. After project is selected, list of keystore files from that project is updated. Now you can select keystore from *Keystore* dropdown menu. Specific thing here is to write password of keystore you want to see in a *Password* file. Since this is meant to be used just for testing environment, you can probably find keystore passwords without encryption in configuration files.

After you have selected everything and put in a password, click *Read keystore* button.

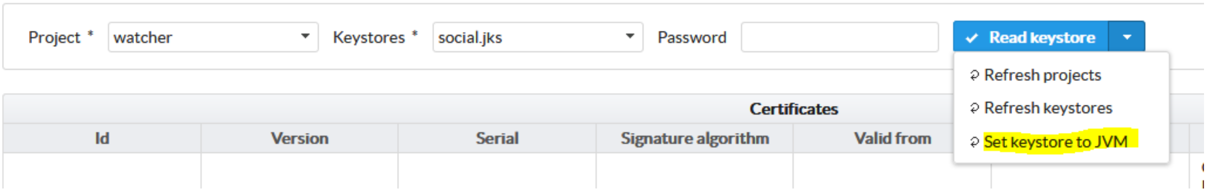


Project * watcher Keystores * social.jks Password

Certificates							
Id	Version	Serial	Signature algorithm	Valid from	Valid to	Issuer	Subject
facebook	3	c0099b7d789c9f666263	SHA256withRSA	09/12/2016	25/01/2018	CN=DigiCert SHA2 High Assurance Server CA, OU=www.digicert.com, O=DigiCert Inc, C=US	CN=*.facebook.com, O="Facebook, Inc.", L=Menlo Park, ST=California, C=US
google	3	5eaeead1960df94e	SHA256withRSA	14/06/2017	06/09/2017	CN=Google Internet Authority G2, O=Google Inc, C=US	CN=*.google.com, O=Google Inc, L=Mountain View, ST=California, C=US

Now you can see all certificates in selected keystore.

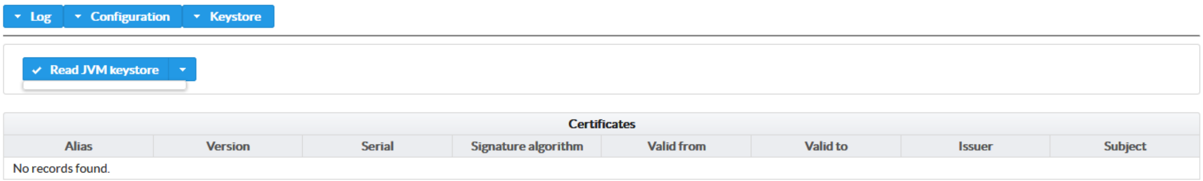
If you fill password again you can click dropdown on *Read keystore* button and select *Set keystore to JVM.* This option sets selected keystore to be default keystore in this JVM.



Project * watcher Keystores * social.jks Password

Reading JVM keystore

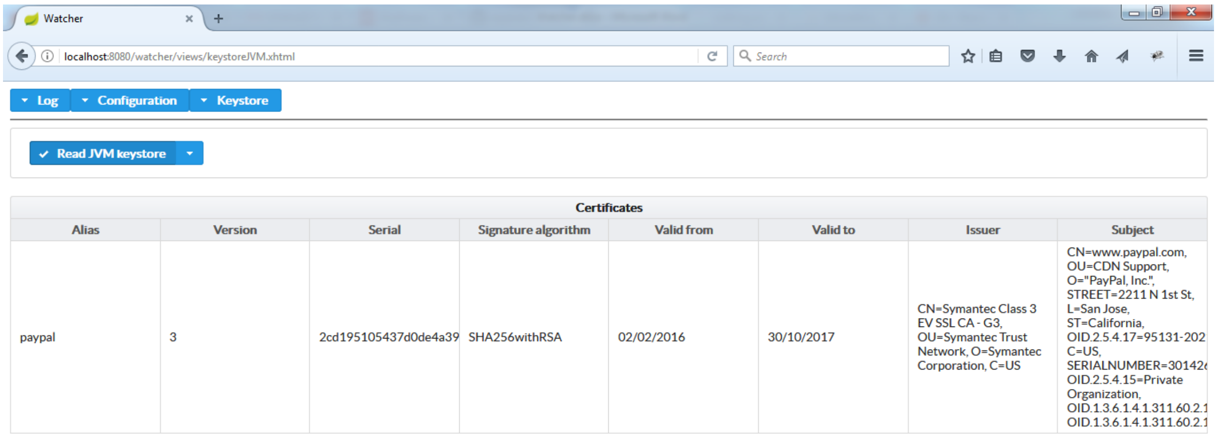
URL: http://localhost:8080/watcher/views/keystoreJVM.xhtml



Certificates

Alias	Version	Serial	Signature algorithm	Valid from	Valid to	Issuer	Subject
No records found.							

If you want to know which keystore is set as default in JVM, in this option you can just click *Read JVM keystore* (without password) and it will show all certificates available in default keystore.



Certificates

Alias	Version	Serial	Signature algorithm	Valid from	Valid to	Issuer	Subject
paypal	3	2cd195105437d0de4a39	SHA256withRSA	02/02/2016	30/10/2017	CN=Symantec Class 3 EV SSL CA - G3, OU=Symantec Trust Network, O=Symantec Corporation, C=US	CN=www.paypal.com, OU=CDN Support, O="PayPal, Inc.", STREET=2211 N 1st St, L=San Jose, ST=California, OID.2.5.4.17=95131-202 C=US, SERIALNUMBER=30142 OID.2.5.4.15=Private Organization, OID.1.3.6.1.4.1.311.60.2.1 OID.1.3.6.1.4.1.311.60.2.1