Using MidPoint with embedded Tomcat

This guide explains how to configure and run MidPoint with embedded Tomcat.

This feature is available since MidPoint version 3.7. No configuration changes needed in case you want to run MidPoint WAR inside standard Tomcat.

This feature is based on spring libraries, especially spring boot framework. Libraries used:

- boot - 1.5.8.RELEASE
- spring - 4.3.12.RELEASE
- security - 4.2.3.RELEASE

Configuration

MidPoint configuration is done as usual in config.xml file located in `midpoint.home`. Configuration for embedded tomcat can be done in two places. Default configuration file name `application.yml` is located on classpath (admin-gui/src/main/resources folder). Custom configuration file `application.yml` can be placed to `midpoint.home` folder. Following table shows list of available properties that can be used to cofnigure tomcat (e.g. http/https ports, session timeouts, logging, max-post-size, etc.).

Available properties

```
# EMBEDDED SERVER CONFIGURATION (ServerProperties)
server:
  address:   # Network address to which the server should bind.
  compression:
    enabled:  false # Whether response compression is enabled.
    excluded-user-agents:   # Comma-separated list of user agents for which responses should not be compressed.
    mime-types:  text/html,text/xml,text/plain,text/css,text/javascript,application/javascript,application/json,
application/xml # Comma-separated list of MIME types that should be compressed.
    min-response-size:  2KB # Minimum "Content-Length" value that is required for compression to be performed.
  connection-timeout:   # Time that connectors wait for another HTTP request before closing the connection.
When not set, the connector's container-specific default is used. Use a value of -1 to indicate no (that is, an
infinite) timeout.
  error:
    include-exception:  false # Include the "exception" attribute.
    include-stacktrace:  never # When to include a "stacktrace" attribute.
    path:  /error # Path of the error controller.
    whitelabel:
      enabled:  true # Whether to enable the default error page displayed in browsers in case of a server error.
  http2:
    enabled:  false # Whether to enable HTTP/2 support, if the current environment supports it.
  jetty:
    acceptors:  -1 # Number of acceptor threads to use. When the value is -1, the default, the number of
acceptors is derived from the operating environment.
    accesslog:
      append:  false # Append to log.
      date-format:  dd/MMM/yyyy:HH:mm:ss Z # Timestamp format of the request log.
      enabled:  false # Enable access log.
      extended-format:  false # Enable extended NCSA format.
      file-date-format:   # Date format to place in log file name.
      filename:   # Log filename. If not specified, logs redirect to "System.err".
      locale:   # Locale of the request log.
      log-cookies:  false # Enable logging of the request cookies.
      log-latency:  false # Enable logging of request processing time.
      log-server:  false # Enable logging of the request hostname.
      retention-period:  31 # Number of days before rotated log files are deleted.
      time-zone:  GMT # Timezone of the request log.
    max-http-post-size:  200000B # Maximum size of the HTTP post or put content.
    selectors:  -1 # Number of selector threads to use. When the value is -1, the default, the number of
selectors is derived from the operating environment.
  max-http-header-size:  8KB # Maximum size of the HTTP message header.
  port:  8080 # Server HTTP port.
  server-header:   # Value to use for the Server response header (if empty, no header is sent).
  use-forward-headers:   # Whether X-Forwarded-* headers should be applied to the HttpRequest.
  servlet:
    context-parameters.*:   # Servlet context init parameters.
    context-path:   # Context path of the application.
    application-display-name:  application # Display name of the application.
    jsp:
      class-name:  org.apache.jasper.servlet.JspServlet # Class name of the servlet to use for JSPs.
      init-parameters.*:   # Init parameters used to configure the JSP servlet.
      registered:  true # Whether the JSP servlet is registered.
    session:
      cookie:
        comment:   # Comment for the session cookie.
        domain:   # Domain for the session cookie.
        http-only:   # Whether to use "HttpOnly" cookies for session cookies.
        max-age:   # Maximum age of the session cookie. If a duration suffix is not specified, seconds will be
used.
        name:   # Session cookie name.
        path:   # Path of the session cookie.
        secure:   # Whether to always mark the session cookie as secure.
      persistent:  false # Whether to persist session data between restarts.
      store-dir:   # Directory used to store session data.
      timeout:  30m # Session timeout. If a duration suffix is not specified, seconds will be used.
      tracking-modes:   # Session tracking modes.
  ssl:
    ciphers:   # Supported SSL ciphers.
    client-auth:   # Whether client authentication is wanted ("want") or needed ("need"). Requires a trust
store.
    enabled:  true # Whether to enable SSL support.
    enabled-protocols:   # Enabled SSL protocols.
    key-alias:   # Alias that identifies the key in the key store.
    key-password:   # Password used to access the key in the key store.
    key-store:   # Path to the key store that holds the SSL certificate (typically a jks file).
    key-store-password:   # Password used to access the key store.
    key-store-provider:   # Provider for the key store.
    key-store-type:   # Type of the key store.
    protocol:  TLS # SSL protocol to use.
    trust-store:   # Trust store that holds SSL certificates.
    trust-store-password:   # Password used to access the trust store.
    trust-store-provider:   # Provider for the trust store.
    trust-store-type:   # Type of the trust store.
  tomcat:
    accept-count:  100 # Maximum queue length for incoming connection requests when all possible request
processing threads are in use.
      buffered:  true # Whether to buffer output such that it is flushed only periodically.
      directory:  logs # Directory in which log files are created. Can be absolute or relative to the Tomcat
base dir.
      enabled:  false # Enable access log.
      file-date-format:  .yyyy-MM-dd # Date format to place in the log file name.
      pattern:  common # Format pattern for access logs.
      prefix:  access_log # Log file name prefix.
      rename-on-rotate:  false # Whether to defer inclusion of the date stamp in the file name until rotate
time.
      request-attributes-enabled:  false # Set request attributes for the IP address, Hostname, protocol, and
port used for the request.
      rotate:  true # Whether to enable access log rotation.
      suffix:  .log # Log file name suffix.
    additional-tld-skip-patterns:   # Comma-separated list of additional patterns that match jars to ignore for
TLD scanning.
    background-processor-delay:  10s # Delay between the invocation of backgroundProcess methods. If a duration
suffix is not specified, seconds will be used.
    basedir:   # Tomcat base directory. If not specified, a temporary directory is used.
    internal-proxies:  10\\.\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}|\\
      192\\.168\\.\\d{1,3}\\.\\d{1,3}|\\
      169\\.254\\.\\d{1,3}\\.\\d{1,3}|\\
      127\\.\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}|\\
      172\\.1[6-9]{1}\\.\\d{1,3}\\.\\d{1,3}|\\
      172\\.2[0-9]{1}\\.\\d{1,3}\\.\\d{1,3}|\\
      172\\.3[0-1]{1}\\.\\d{1,3}\\.\\d{1,3}\\
      0:0:0:0:0:0:0:1\\
      ::1 # Regular expression that matches proxies that are to be trusted.
    max-connections:  10000 # Maximum number of connections that the server accepts and processes at any given
time.
    max-http-post-size:  2MB # Maximum size of the HTTP post content.
    max-swallow-size:  2MB # Maximum amount of request body to swallow.
    max-threads:  200 # Maximum amount of worker threads.
    min-spare-threads:  10 # Minimum amount of worker threads.
    port-header:  X-Forwarded-Port # Name of the HTTP header used to override the original port value.
    protocol-header:   # Header that holds the incoming protocol, usually named "X-Forwarded-Proto".
    protocol-header-https-value:  https # Value of the protocol header indicating whether the incoming request uses SSL.
    redirect-context-root:  true # Whether requests to the context root should be redirected by appending a / to the path.
    remote-ip-header:   # Name of the HTTP header from which the remote IP is extracted. For instance, `X-FORWARDED-FOR`.
    resource.allow-caching:  true # Whether static resource caching is permitted for this web application.
    resource.cache-ttl:   # Time-to-live of the static resource cache.
    uri-encoding:  UTF-8 # Character encoding to use to decode the URI.
    use-relative-redirects:   # Whether HTTP 1.1 and later location headers generated by a call to sendRedirect will use relative or absolute redirects.
```

Full list of properties is available here.

Web descriptor - `web.xml` was removed, all servlets and filters are registered/defined using servlet api in `MidPointSpringApplication.java`.

Configuration example 1

Configuration example 1: application.yml

```
server.port: 8088
server.servlet.session.timeout: 60m
server.servlet.context-path: /idm
```

Configuration example 2

Configuration example 2: application.yml

```
server:
    port: 8088
    servlet:
        context-path: /idm
        session:
            timeout: 60m
```

Other configuration

- Web security configuration
    - defined in WebSecurityConfig.java
        - ctx-web-security-*.xml contexts are still available, but not used
    - cas and ldap configuration not available now (needs to be finished)
- Banner (midpoint logo in logs)
    - used spring boot standard banner.txt file
- static files moved to src/main/resources/static (default for spring boot)

JDBC Drivers

Currently midPoint bundles only PostgreSQL and H2 jdbc driver. If one wants to deploy standalone midPoint with different database, then jdbc driver must be copied to `midpoint.home/lib` .

Executable Jar Start/Stop

Example command with minimum options (memory and midpoint.home configuration) using `midpoint.war` from `dist/target` folder:

Start command example

```
java -Xms768m -Xmx2048m -Dmidpoint.home=/opt/midpoint-home -Dmidpoint.nodeId=node1 -jar midpoint.war
```

Other options can be added from list of properties (table above) using `-D` option, e.g. `-Dserver.port=12345`. Options explicitly stated in command will override defaults located in `midpoint.home`/application.yml.

Using midPoint with embedded Tomcat

Use the default URL (modify hostname and port as required): http://localhost:8080/

Autoconfiguration

MidPoint web applicaiton is autoconfigured by using the com.evolveum.midpoint.web.boot.MidPointSpringApplication class as a starting point. Spring boot will process all the annotated methods of this class in a "configuration code" approach. Additional autocofiguration classes are also used. Those are listed in the @ImportAutoConfiguration annotation.

Autoconfiguration is used as a replacement for JEE deployment descriptor (web.xml). E.g. servlets and servlet mappings are initialized in the MidPointSpringApplication class.

TODO

- Redirect from / to /midpoint
- Servelt for static content

See Also

- Authentication Configuration (Spring Security)