

Meecrowave Jolokia

Coordinates:

```
<dependency>
  <groupId>org.apache.meecrowave</groupId>
  <artifactId>meecrowave-jolokia</artifactId>
  <version>${meecrowave.version}</version>
</dependency>
```

Jolokia only

If Hawt.io is not setup (not there or inactive) the module will set up jolokia alone.

It will deploy the jolokia servlet on `/jolokia/*` or the property `jolokia.web.mapping` set on the `Meecrowave.Builder`.



don't forget to secure this endpoint cause it let you access all the JMX MBeans. See Meecrowave core configuration for more details.

More on <https://jolokia.org/reference/html/>

Hawt.io

By default hawt.io will be deployed and activated on `/hawtio/` endpoint.

To activate it you need to add these dependencies:

```
<dependency>
  <groupId>io.hawt</groupId>
  <artifactId>hawtio-springboot</artifactId>
  <version>1.4.67</version>
  <optional>true</optional>
  <exclusions>
    <exclusion>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </exclusion>
  </exclusions>
</dependency>
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-slf4j-impl</artifactId>
  <version>${log4j2.version}</version>
  <optional>true</optional>
</dependency>
```



we don't need spring boot but we reuse hawt.io spring boot module which packages web resources in a jar.

By default it will setup the security to use the Servlet security which means you can use meecrowave options:

```
./meecrowave-runner --users=admin=admin --roles=admin=admin
```

And log with admin/admin user.

Running HawtIO war

To run <http://hawt.io>, the easiest is to setup jaas and run the hawt.io war:

```
java -Djava.security.auth.login.config=login.config \  
-jar meecrowave-runner.jar \  
-webapp hawtio-default-offline-1.4.67.war
```

This assumes you created a `login.config`:

```
karaf {  
  org.apache.catalina.realm.JAASMemoryLoginModule required  
  pathname="/path/to/tomcat-users.xml";  
};
```



the JAAS login module is named `karaf` because it is the default of hawt.io (check `io.hawt.web.AuthenticationFilter.init()`).

And if you use - as in previous snippet - `tomcat-users.xml` you need to create an admin (role) user:

```
<tomcat-users>  
  <user username="admin" password="admin" role="hawtio" roles="admin" />  
</tomcat-users>
```



don't try to use `io.hawt.web.tomcat.TomcatUserDatabaseLoginContext` of hawt.io which does the same as `JAASMemoryLoginModule` excepted it doesn't handle absolute paths which enforce you to use a `catalina.base`.

Using hawt.io war you don't need to run jolokia servlet by yourself, it is already done.