Maven + Struts 2

Installation

Aim: Install Maven version 3.6.3 on window 10

Requisites: JDK version 1.7+

Procedure: Guide

Basic about maven

- A **project management** and comprehension tool that provides developers a complete build lifecycle framework
- It uses a project object model (POM) file, `pom.xml` to manage project's build, dependencies, reporting and documentation

POM

An xml file named `pom.xml` resides in base directory.

It should be noted that there should be a single POM file for each project.

- All POM files require the **project** element and three mandatory fields: **groupId, artifactId, version**.
- Projects notation in repository is **groupId:artifactId:version**.

Example:

```xml
<project xmlns = "http://maven.apache.org/POM/4.0.0"
   xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation = "http://maven.apache.org/POM/4.0.0
   http://maven.apache.org/xsd/maven-4.0.0.xsd">
   <modelVersion>4.0.0</modelVersion>

   <groupId>com.companyname.project-group</groupId>
   <artifactId>project</artifactId>
   <version>1.0</version>
</project>
```

Super POM

Maven 's default POM, inherited by all others one by default. To view it, run command `mvn help:effective-pom`.

Add dependency

It 's easy when comes to manage dependencies, libraries in maven project. For instance, we want to add `jstl-1.2` lib, just add these lines in `pom.xml`

```xml
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
</dependency>
```

Build lifecycle

A well-defined sequences phases, which define the order in which the goal are to be executed.

Sequence of phases: `prepare-resource` -> `validate` all necessary info is available -> `compile` -> `test` compiled source -> `package` (create JAR/WAR based on pom.xml) -> `install` package in maven repo -> `deploy`: copy the final package to remote repo.

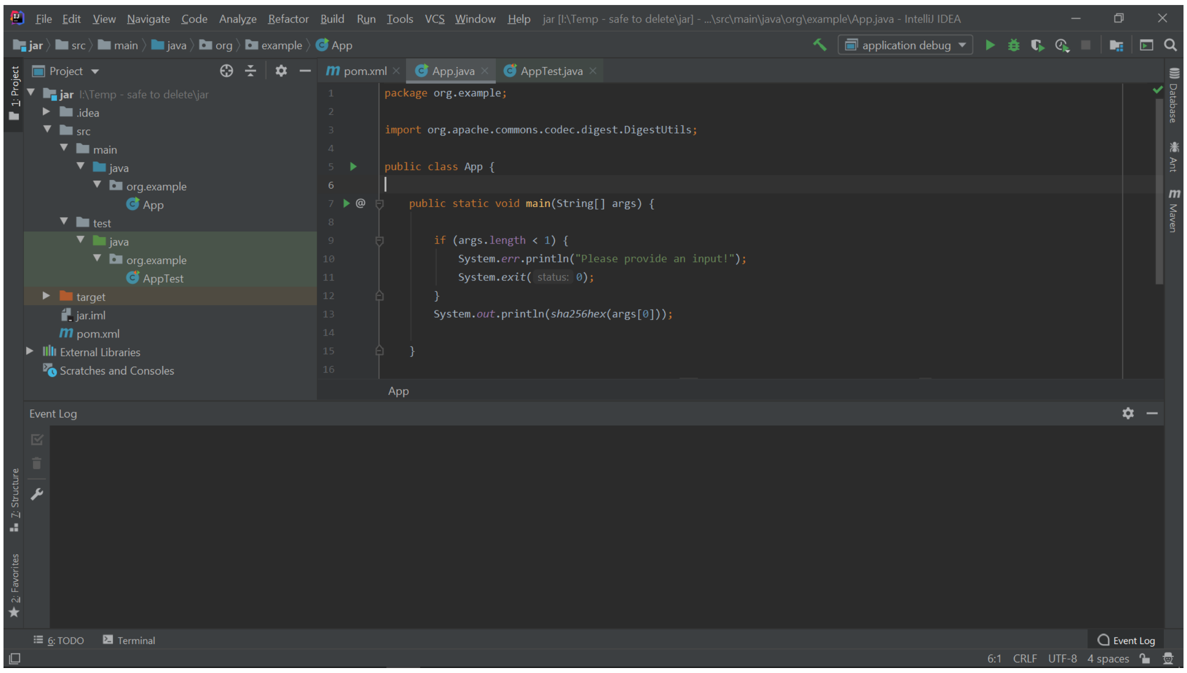
To call a phase: run command `mvn <phase>`. Only phases up to and including that phase will execute.

A goal represents a specific task contributing to the building. It may be bound to 0+ phases. Syntax `<phase>:<goal>`, eg. 'dependency:copy-dependencies'.

Examples

Create Java project (JAR)

Guide: https://mkyong.com/maven/how-to-create-a-java-project-with-maven/



```
package org.example;

import org.apache.commons.codec.digest.DigestUtils;

public class App {

    public static void main(String[] args) {

        if (args.length < 1) {
            System.err.println("Please provide an input!");
            System.exit( status: 0);
        }
        System.out.println(sha256hex(args[0]));

    }
```

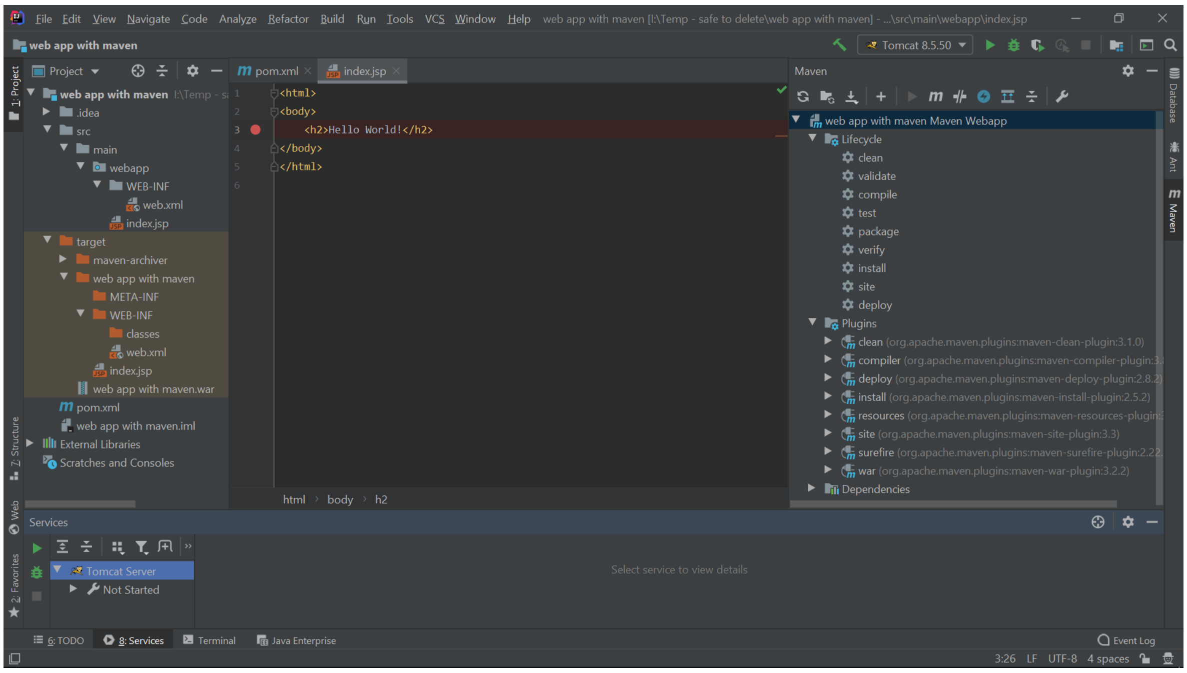
jar [I:\Temp - safe to delete\jar] - ...\src\main\java\org\example\App.java - IntelliJ IDEA
application debug
Event Log

Create Java web application (WAR)

Guide: https://mkyong.com/maven/how-to-create-a-web-application-project-with-maven/

Procedure:

1. Create sample application with `mvn archtype:<archtype>`
2. Config POM.xml:
   - Add dependency `javax servlet` for java web app.
   - Ensure these plugins: dependency, clean, compiler, surefire (for `surefire:test` goal), war, deploy (Optional if we config deployment in IDE).

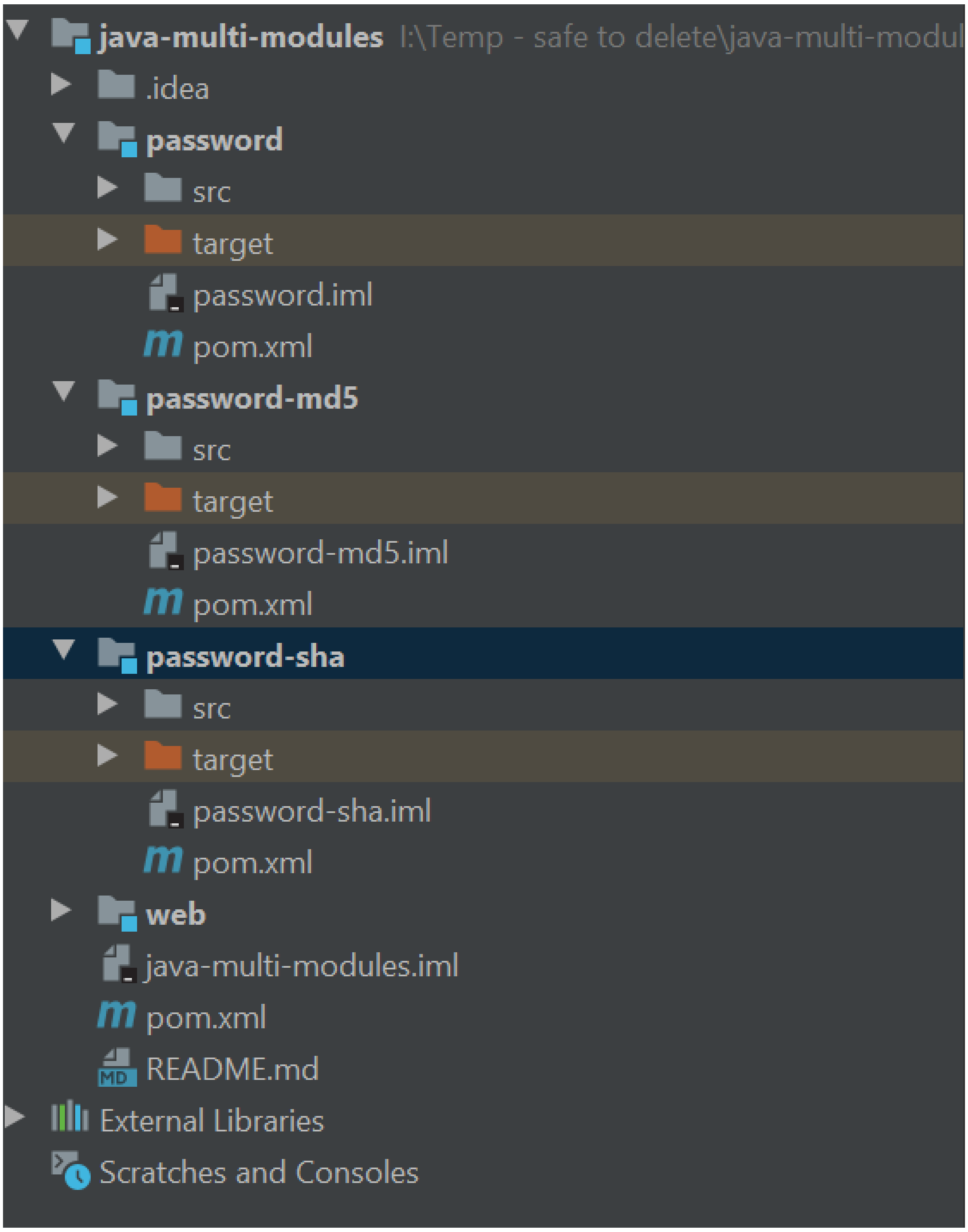


```
<html>
<body>
<h2>Hello World!</h2>
</body>
</html>
```

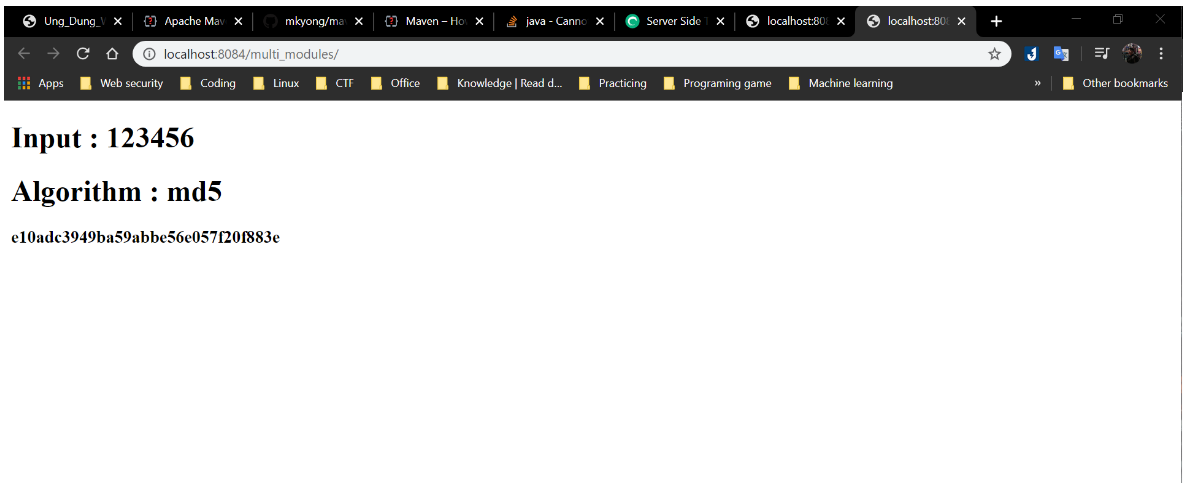
web app with maven [I:\Temp - safe to delete\web app with maven] - ...\src\main\webapp\index.jsp
Tomcat 8.5.50
Maven
web app with maven Maven Webapp
Lifecycle
clean
validate
compile
test
package
verify
install
site
deploy
Plugins
clean (org.apache.maven.plugins:maven-clean-plugin:3.1.0)
compiler (org.apache.maven.plugins:maven-compiler-plugin:3.
deploy (org.apache.maven.plugins:maven-deploy-plugin:2.8.2)
install (org.apache.maven.plugins:maven-install-plugin:2.5.2)
resources (org.apache.maven.plugins:maven-resources-plugin
site (org.apache.maven.plugins:maven-site-plugin:3.3)
surefire (org.apache.maven.plugins:maven-surefire-plugin:2.22
war (org.apache.maven.plugins:maven-war-plugin:3.2.2)
Dependencies
Services
Tomcat Server
Not Started
Select service to view details

Create Multi-modules web app

Guide: https://www.baeldung.com/maven-multi-module



```
java-multi-modules  I:\Temp - safe to delete\java-multi-modul
  .idea
  password
    src
    target
    password.iml
    pom.xml
  password-md5
    src
    target
    password-md5.iml
    pom.xml
  password-sha
    src
    target
    password-sha.iml
    pom.xml
  web
  java-multi-modules.iml
  pom.xml
  README.md
External Libraries
Scratches and Consoles
```



Input : 123456

Algorithm : md5

e10adc3949ba59abbe56e057f20f883e

Struts 2 - maven project

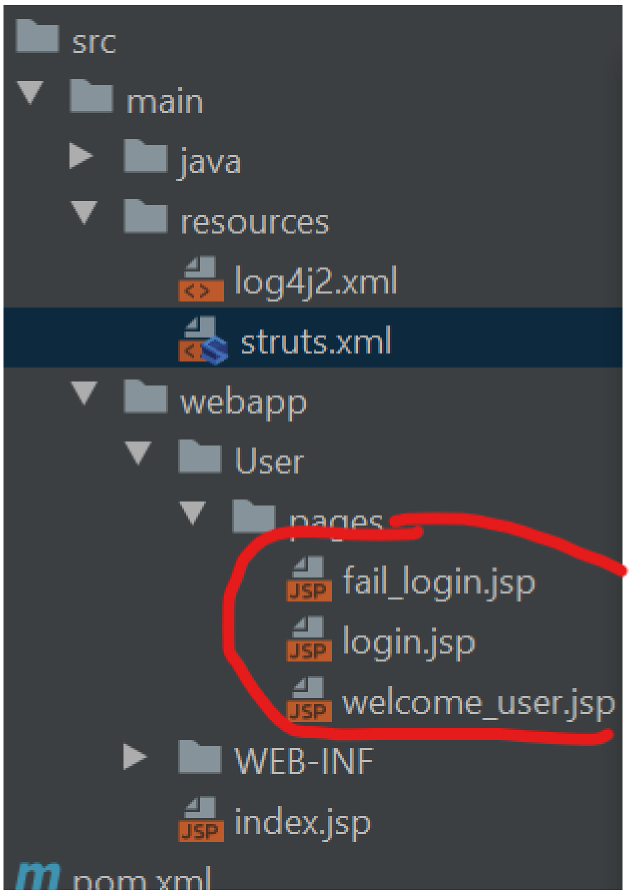
Common knowledge

1. Struts 2 will search for result page in `webapp/` folder which means under `ApplicationContext (Document root)`. For example:

   An action 's config:

   ```
   /* struts.xml */
   <action name="welcome" class="org.example.WelcomeUserAction">
       <result name="success">pages/welcome_user.jsp</result>
       <result name="fail">pages/fail_login.jsp</result>
   </action>
   ```

   Located .jsp page:



```
src
  main
    java
    resources
      log4j2.xml
      struts.xml
    webapp
      User
        pages
          fail_login.jsp
          login.jsp
          welcome_user.jsp
      WEB-INF
      index.jsp
pom.xml
```

Examples

Hello world (XML)

Hello world (Annotation)

1. `@ResultPath`

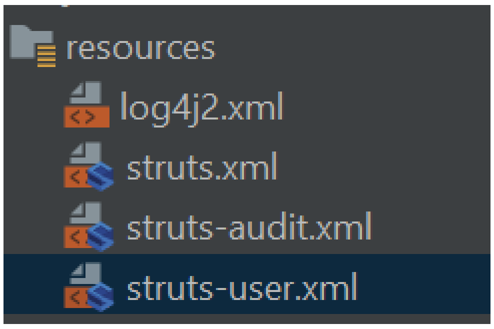
   If not specify this, web app will search for .jsp page in folder `webapp/WEB-INF/content/`

Now that the Convention plugin has been added to your application, let's start with a very simple example. This example will use an actionless result that is identified by the URL. By default, the Convention plugin assumes that all of the results are stored in **WEB-INF/content**. This can be changed by setting the property `struts.convention.result.path` in the Struts properties file to the new location. Don't worry about trailing slashes, the Convention plugin handles this for you. Here is our hello world JSP:

More information: documentation

2. Scanning procedure:
    1. Scan the annotated classes which located at the packaged named "**struts, struts2, action or actions**".
    2. Next, scan the file which match either of the following criteria :
        - Implemented the **com.opensymphony.xwork2.Action** interface.
        - Extends the **com.opensymphony.xwork2.ActionSupport** class.
        - File name ends with Action (e.g UserAction, LoginAction).

Multiple struts config files



Struts.xml

```xml
<struts>

    <package name="default" namespace="/" extends="struts-default">
    </package>

    <include file="struts-user.xml"></include>
    <include file="struts-audit.xml"></include>

</struts>
```

struts-user.xml (for namespace = '/User')

```xml
<struts>
    <package name="user" namespace="/User" extends="struts-default">
        <action name="Login">
            <result>/WEB-INF/User/pages/login.jsp</result>
        </action>
        <action name="Welcome" class="org.example.user.WelcomeUserAction">
            <result name="success">/WEB-
INF/User/pages/welcome_user.jsp</result>
            <result name="fail">/WEB-INF/User/pages/fail_login.jsp</result>
        </action>
    </package>
</struts>
```

struts-audit (for namespace = '/audit')

```xml
<struts>
    <package name="user" namespace="/User" extends="struts-default">
        <action name="Login">
            <result>/WEB-INF/User/pages/login.jsp</result>
        </action>
        <action name="Welcome" class="org.example.user.WelcomeUserAction">
            <result name="success">/WEB-
INF/User/pages/welcome_user.jsp</result>
            <result name="fail">/WEB-INF/User/pages/fail_login.jsp</result>
        </action>
    </package>
</struts>
```

Using interceptors

1. Interceptor work flow
2. default interceptors stack
3. create own interceptors stack
4. create own interceptor by implement: guide
5. provide parameters to interceptor