

Multi Module Projects

Pom can have submodules...

There will be parent pom and other submodule provided as jar or pom with their own jar.

Module with least dependencies is build first.

Creating parent pom -> describing pom in packing tag.

Clicking on install will give you Reactor summary for the project.

The modules inherit the groupId as well as properties of the parent.

```
<xsd:annotation>
  <xsd:appinfo>
    <jaxb:schemaBindings>
      <jaxb:package name="guru.springframework.domain"/>
    </jaxb:schemaBindings>
  </xsd:appinfo>
</xsd:annotation>
```

We are saying we gonna generate the classes in this package.

Maven Dependencies between modules:

To create dependencies, maven look out for the modules as jar in local repo.

So first install the modules and then only you can use their properties.

Order of execution -> depends on module having other modules as dependencies then first least independent module will be build and then the dependent ones.

If a module depends on 2 modules and we want to include a parent module that depends on the same 2 modules then we can remove their dependency and keep the parent one.

To make code more simpler we can take up the version number of each module as they are same in revision tag of properties so that we can change from there. And we can replace version by \${revision}

This is possible from maven 3.5 and we can also change revision property from command line.

\${project.version} version is dependent on \${revision} version

One problem with above is that we get same hard code written in local repo. To overcome this we have flatten plugin. Helps in flattening the resources. On clicking on clean -> it removes the flattened-pom.xml..

Maven enforcer plugin -> will force to build only certain rules are met. Like our

project needs maven 3.5 or above and java version 11.. So we can setup that.