

1. Goals

Learn how to work with properties and project configuration.

2. Lesson Notes

2.1. Define / Use an environment target

Start with a single *persistence.properties* file - responsible with holding all persistence related properties for the application

Once we need to **deploy on multiple environments**, we'll need to break that into multiple properties file - based on the environment:

- *persistence-local.properties*
- *persistence-cargo.properties*
- *persistence-dev.properties*
- *persistence-prod.properties*

To define this flexible properties files in the Spring context, we'll need to use:

```
@PropertySource({ "classpath:persistence-${envTarget:mysql}.properties" })
```

Notice how the *envTarget* property is general, defining the environment.

A step further - depending on how much configuration flexibility the project requires - would be to go more granular and define a *persistenceEnvTarget* property - specifically for persistence:

```
@PropertySource({ "classpath:persistence-${persistenceEnvTarget:mysql}.properties" })
```

2.2. Set the environment target

In Eclipse - steps

- in the Server Configuration -> Environment

In *cargo-maven2-plugin* - steps

```
<configuration>
  <container>
    ...
    <systemProperties>
      <envTarget>local</envTarget>
    </systemProperties>
  </container>
</configuration>
```

In a standalone Tomcat - steps

- in *conf/context.xml* inside of a *<Context>* element:

```
<Environment name="adminEnvTarget" value="nomock" type="java.lang.String"/>
```

3. Resources

- Cargo System Properties
- Tomcat - Environment Entries
- Spring - *Spring* - *@PropertySource annotation javadoc* annotation javadoc
- "Properties with Spring" on Baeldung