

In questo articolo vedremo come integrare un interessante plugin maven per la compilazione automatica dei report realizzati con jasperReports.

Incominciamo con il creare il nostro progetto maven con la direttiva

```
mvn archetype:generate -DgroupId=com.mycompany.app -DartifactId=my-app  
-DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

La direttiva crea una cartella contenente il file pom.xml e la cartella src contenente i sorgenti e i file junit per l'esecuzione dei test.

Lanciamo la direttiva mvn eclipse:eclipse per creare i file necessari ad un progetto eclipse ed importiamo il progetto nel nostro workspace.

A questo punto agganciamo il plugin responsabile della compilazione dei sorgenti jrxml, tipici di JasperReports. Tale plugin prevede la presenza di una cartella jasperreports sotto src/main. Creiamo la cartella e modifichiamo il pom inserendo la dichiarazione del plugin.

```
<build> <plugins> <plugin> <groupId>org.codehaus.mojo</groupId>  
<artifactId>jasperreports-maven-plugin</artifactId> <executions> <execution>  
<goals> <goal>compile-reports</goal> </goals> </execution>  
</executions> </plugin> </plugins> </build>
```

A questo punto lanciamo la direttiva mvn compile per verificare la correttezza delle nostre operazioni. Se avete sbagliato a posizionare la cartella il plugin vi ricorderà il percorso esatto.

Restano da definire le dipendenze dalla versione di jasperReports da usare. Per farlo è sufficiente specificare la dipendenza a livello di progetto e a livello di plugin, nel nostro caso ho scelto la versione 4.0.2

```
<dependencies> <dependency> <groupId>net.sf.jasperreports</groupId>  
<artifactId>jasperreports</artifactId> <version>4.0.2</version> </dependency>
```

</dependencies>

Creiamo il nostro file jrxml nella cartella jasperreports e lanciamo la direttiva mcn compile. A questo sotto la cartella target avremo il nostro file compilato e pronto all'uso.

Il vantaggio di questa soluzione è di aver all'atto della compilazione tutti i report in linea con i file jrxml presenti nel progetto.

[Qui](#) trovate i sorgenti dell'articolo.