

Dispiegamento di un template su Artifactory

Ogni volta che si ha una struttura Maven che ci si ritrova ad utilizzare spesso, potrebbe essere una buona idea generare da essa un template su Artifactory.

Vantaggi Sviluppatori

- Poter tirare su un nuovo ambiente di lavoro, senza tirarsi giù l'intero codice sorgente di geoweb, ma semplicemente creando un nuovo progetto da template maven e configurandoselo. Il vantaggio è di lavorare con una black-box, avere un ambiente di sviluppo più snello, non oberato dai continui build del codice eseguiti da Eclipse
- Poter disporre rapidamente di una base funzionante per creare un qualsivoglia nuovo plugin

Vantaggi Configuratori

- avere la possibilità agli utenti configuratori di autoprodursi il war seguendo la procedura guidata automatizzata descritta qui, pensando solo alla parte di configurazione, alla versione di geoweb e da quali plugin utilizzare

Esempi Template

- **webclientTemplate**, rappresenta lo scheletro di una generica webapp di geoweb, dove sono presenti solo tutte le parti configurabili, ed il pom.xml dove si possono includere tutte le dipendenze necessarie (recuperate da repository pubblici o da artifactory). Questo progetto maven è già stato configurato in quest'ottica dentro il codice sorgente dalla versione 4.4.3. E' stato quindi reso disponibile questo template maven tramite Artifactory. Ora può essere usato da uno sviluppatore esterno a geoweb che, non volendo tirarsi giù l'intero codice, potrebbe far girare geoweb o prodursi un war semplicemente creando un nuovo progetto da template maven e configurandoselo. Il vantaggio è di lavorare con una black-box, avere un ambiente di sviluppo più snello, non oberato dai continui build del codice eseguiti da Eclipse. Il fatto che questo progetto sia anche un template maven ospitato in artifactory è anche basilare per dare la possibilità agli utenti configuratori di autoprodursi il war seguendo la procedura guidata automatizzata descritta qui.
- **gwPlugin**, rappresenta lo scheletro di un generico plugin di geoweb, già formato nella struttura dei package e dei file di configurazione. E' pensato per essere usato dagli sviluppatori, interni ed esterni, come una buona base di partenza per produrre velocemente un nuovo plugin per geoweb. Di norma va usato in congiunzione con webclientTemplate.

Procedura

- Come prima cosa, è necessario creare il file JAR del progetto utilizzando Eclipse. Dopo aver eseguito il *clean* del progetto Maven, cliccare con il tasto destro sulla cartella del progetto, selezionare **Run As** e poi **Maven install**; in questo modo verranno compilati tutti i file sorgente

e, nella sezione **Project explorer** del progetto, verrà creata la cartella *target*, contenente il file JAR .

- A questo punto, entrare nella cartella del proprio PC contenente il repository locale di Git dove si sta lavorando al nuovo plugin e, al suo interno, aprire il prompt dei comandi ed eseguire il seguente:

```
mvn archetype:create-from-project
```

- Il passo successivo è la sostituzione di alcuni file all'interno dell'archivio JAR che è stato generato nel percorso {git_plugin_directory}\target\generated-sources\archetype\target. In questo percorso è presente un file dal nome {nome_plugin}-archetype-{versione_plugin}.jar. Aprire l'archivio JAR con un'applicazione per la decompressione degli archivi (ad esempio 7zip) e spostarsi nel percorso META-INF\maven: qui dentro, sostituire il contenuto del file archetype-metadata.xml con ciò che segue:

```
<?xml version="1.0" encoding="UTF-8"?>
<archetype-descriptor
  xsi:schemaLocation="https://maven.apache.org/plugins/maven-archetype-pl
ugin/archetype-descriptor/1.1.0
  http://maven.apache.org/xsd/archetype-descriptor-1.1.0.xsd"
  name="webclientTemplate"
  xmlns="https://maven.apache.org/plugins/maven-archetype-plugin/archetyp
e-descriptor/1.1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <requiredProperties>
    <requiredProperty key="warName">
      <defaultValue>webclient</defaultValue>
    </requiredProperty>
    <requiredProperty key="gwVersion">
    </requiredProperty>
  </requiredProperties>
  <fileSets>
    <fileSet filtered="true" encoding="UTF-8">
      <directory>src/main/resources</directory>
      <includes>
        <include>**/*.txt</include>
        <include>**/*.xml</include>
        <include>**/*.properties</include>
      </includes>
    </fileSet>
    <fileSet filtered="true" encoding="UTF-8">
      <directory>src/main/webapp</directory>
      <includes>
        <include>**/*.xml</include>
      </includes>
    </fileSet>
    <fileSet encoding="UTF-8">
      <directory>src/main/resources</directory>
      <includes>
        <include>**/*.ini</include>
```

```

        </includes>
    </fileSet>
    <fileSet encoding="UTF-8">
        <directory>src/main/webapp</directory>
        <includes>
            <include>**/*.tld</include>
        </includes>
    </fileSet>
    <fileSet filtered="true" encoding="UTF-8">
        <directory>.settings</directory>
        <includes>
            <include>**/*.xml</include>
        </includes>
    </fileSet>
    <fileSet encoding="UTF-8">
        <directory>.settings</directory>
        <includes>
            <include>**/*.container</include>
            <include>**/*.component</include>
            <include>**/*.name</include>
            <include>**/*.jsdtscope</include>
            <include>**/*.prefs</include>
        </includes>
    </fileSet>
    <fileSet filtered="true" encoding="UTF-8">
        <directory></directory>
        <includes>
            <include>.classpath</include>
            <include>.project</include>
        </includes>
    </fileSet>
</fileSets>
</archetype-descriptor>

```

- Sempre all'interno dell'archivio {nome_plugin}-archetype-{versione_plugin}.jar, aprire il file pom.xml nella cartella archetype-resources e sostituire la riga

```
<finalName>nome_plugin</finalName>
```

con

```
<finalName>${warName}</finalName>
```

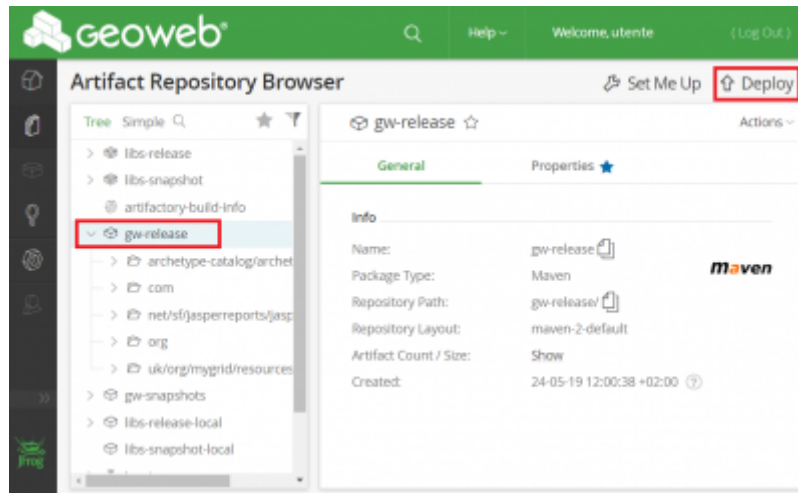
e la riga

```
<com.geowebframework.version>4.4.3</com.geowebframework.version>
```

con

```
<com.geowebframework.version>${gwVersion}</com.geowebframework.version>
```

- Infine, andare sul sito di Artifactory seguendo il link <http://artifactory.geowebframework.com/artifactory/webapp/#/artifacts/browse/tree/General/gw-release> e accedere con le proprie credenziali. Come mostrato in figura, sulla sinistra del sito si può trovare un albero di contenuti: aprire gw-release e cliccare sul pulsante **Deploy**, che si trova in alto a destra.



Dalla finestra che si apre, è possibile caricare il file, avendo cura di cambiare il groupId con "com.geowebframework" e di abilitare tutte le spunte disponibili; per terminare la procedura, cliccare infine **Deploy**. Si può, a questo punto, verificare il caricamento del progetto nel giusto percorso all'interno di gw-release.

From: <https://wiki.geowebframework.com/> - GeowebFramework

Permanent link: https://wiki.geowebframework.com/doku.php?id=custom:development_artifactory_deploy_template&rev=1669310839

Last update: 2022/11/24 18:27

