Installer

Definition

Installs a container

Explanation

An Installer is meant to install a container on your local machine. There is currently only a single Installer implementation: `ZipURLInstaller` which downloads a zipped container distribution from a URL and which installs it (i.e. unpacks it) in a specified directory. This is useful if you wish to fully automate a container installation without having to ask the user to manually install a container on their machine.

Of course you don't have to use an Installer if you want to use a container already installed on your machine.

Example using the Java API

```
Installer installer = new ZipURLInstaller(

"http://download.eclipse.org/jetty/7.2.2.v20
101205/dist/jetty-distribution-7.2.2.v201012
05.tar.gz",
    "target/downloads", "target/extracts");
installer.install();

InstalledLocalContainer container = new
Jetty7xInstalledLocalContainer(
    new
Jetty7xStandaloneConfiguration("target/jetty
7x"));
container.setHome(installer.getHome());
[...]
```

Example using the Ant API

```
<cargo containerId="jetty7x" [...]>
  <zipUrlInstaller

installUrl="http://download.eclipse.org/jett
y/7.2.2.v20101205/dist/jetty-distribution-7.
2.2.v20101205.tar.gz",
      downloadDir="target/downloads"
      extractDir="target/extracts"/>
  [...]
</cargo>
```

Example using the Maven2 plugin

```
<plugin>
    <groupId>org.codehaus.cargo</groupId>

<artifactId>cargo-maven2-plugin</artifactId>
    <configuration>
      <container>
        <containerId>jetty7x</containerId>
        <zipUrlInstaller>

<url>http://download.eclipse.org/jetty/7.2.2
.v20101205/dist/jetty-distribution-7.2.2.v20
101205.tar.gz</url>

<downloadDir>${project.build.directory}/down
loads</downloadDir>

<extractDir>${project.build.directory}/extra
cts</extractDir>
        </zipUrlInstaller>
      </container>
      [...]
    </configuration>
  </plugin>
```

Using Maven2-based distributions

New versions of most application servers have their distribution packages built by Maven2, hence distribute these as ZIP or TAR.GZ files on a Maven2 repository. It is also rather easy to upload servers' packages to an internal (for example, enterprise) Maven repository. In this case, CARGO lets you directly use such artifacts using an `Artifact Installer`. Here is the example for Jetty 7:

```
<properties>

<jetty.version>7.2.2.v20101205</jetty.versio
n>
  </properties>

  <build>
    <plugins>
      <plugin>

<groupId>org.codehaus.cargo</groupId>

<artifactId>cargo-maven2-plugin</artifactId>
        <version>${cargo.version}</version>
        <configuration>
          <container>

<containerId>jetty7x</containerId>
            <artifactInstaller>

<groupId>org.eclipse.jetty</groupId>

<artifactId>jetty-distribution</artifactId>

<version>${jetty.version}</version>
            </artifactInstaller>
          </container>
          [...]
        </configuration>
        [...]
      </plugin>
```

```
    </plugins>
  </build>
</project>
```