

Aplicació Canigó 3.1 sense frontend

A qui va dirigit

Aquest how-to va dirigit als perfils tècnics (desenvolupadors i arquitectes) que vulguin desenvolupar una aplicació Canigó sense frontend.

Versió de Canigó

Els passos descrits en aquest document apliquen a la versió del framework Canigó 3.1.x.

Introducció

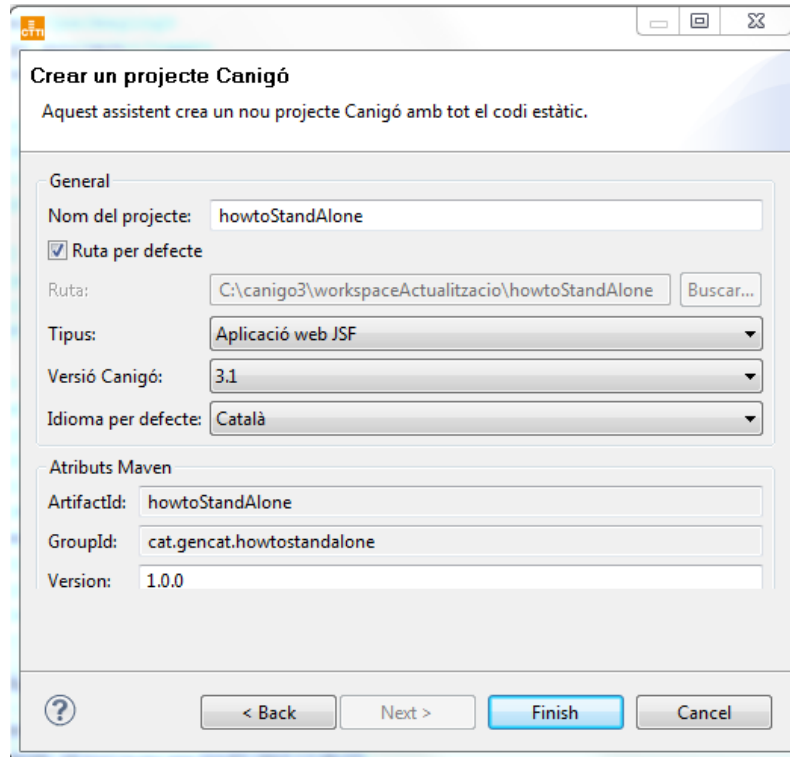
Canigó està preparat per al desenvolupament d'aplicacions webs, però potser per necessitat és té el requeriment de crear un backend sense interfície d'usuari, ja sigui web o d'escriptori. Per exemple, un backend que requereixi integracions amb sistemes Gencat com la PICA, Antivirus, etc, i/o un planificador de tasques que executi de forma periòdica certes funcionalitats. Tot i que Canigó no ha estat concebut per aquest tipus d'aplicacions, és possible fer-ho.

En aquest how-to expliquem com generar una aplicació d'aquest tipus a partir de l'aplicació base generada pel plugin de Canigó per a Eclipse. En aquesta aplicació d'exemple integrarem una tasca planificada cada 5 segons.

Aplicació Canigó 3.1 sense frontend

Crear el Projecte

Amb el plugin d'Eclipse es crea una aplicació Canigó:



Eliminem les parts que no es necessiten:

- src/main/webapp
- src/main/java/cat/gencat/howtostandalone/bean/LogBean.java
- src/main/resources/assemble
- src/main/resources/urlrewrite
- src/main/resources/canigo.skin.properties

Al pom.xml eliminem les referències a la part web:

- canigo.web.jsf
- jaxb-impl
- jaxb-api
- spring-web
- primefaces
- javax.servlet-api

Aplicació Canigó 3.1 sense frontend

pom.xml

```
...
<properties>
  <junit.version>4.11</junit.version>
  <canigo.core.version>[3.1.0,3.2.0)</canigo.core.version>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <org.springframework.version>4.1.0.RELEASE</org.springframework.version>
</properties>
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>${junit.version}</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-test</artifactId>
    <version>${org.springframework.version}</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>cat.gencat.ctti</groupId>
    <artifactId>canigo.core</artifactId>
    <version>${canigo.core.version}</version>
  </dependency>
</dependencies>
...
```

Aplicació Canigó 3.1 sense frontend

Scheduler

Per utilitzar el scheduler hem d'afegir al pom les següents dependències:

```
<!-- SERVEI DE TASQUES -->

<dependency>
  <groupId>org.quartz-scheduler</groupId>
  <artifactId>quartz</artifactId>
  <version>2.2.1</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context-support</artifactId>
  <version>${org.springframework.version}</version>
</dependency>
<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-collections4</artifactId>
  <version>4.0</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-tx</artifactId>
  <version>${org.springframework.version}</version>
</dependency>
<!-- FI SERVEI DE TASQUES -->
```

I a Spring hem de definir les tasques

src/main/resources/spring/quartz-config.xml

```
...
<bean id="SchedulerService" class="cat.gencat.howtostandalone.service.SchedulerService"
/>

    <bean id="FirstTaskJobDetail"
class="org.springframework.scheduling.quartz.MethodInvokingJobDetailFactoryBean">
    <property name="targetObject" ref="SchedulerService" />
    <property name="targetMethod" value="testScheduler" />
</bean>

    <bean id="FirstSimpleTrigger"
class="org.springframework.scheduling.quartz.SimpleTriggerFactoryBean">
    <property name="jobDetail" ref="FirstTaskJobDetail" />
    <property name="repeatInterval" value="5000" />
    <property name="startDelay" value="1000" />
</bean>

    <bean class="org.springframework.scheduling.quartz.SchedulerFactoryBean">
    <property name="jobDetails">
    <list>
    <ref bean="FirstTaskJobDetail" />
    </list>
    </property>
    <property name="triggers">
    <list>
    <ref bean="FirstSimpleTrigger" />
    </list>
    </property>
</bean>
...
```

Aplicació Canigó 3.1 sense frontend

Al fitxer quartz-config.xml es crea una tasca que s'executarà cada 5 segons. La tasca que s'executarà és crea al fitxer SchedulerService

src/main/java/cat/gencat/howtostandalone/service/SchedulerService.java

```
package cat.gencat.howtostandalone.service;

import java.util.Date;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;

public class SchedulerService {
    private static final Log log = LogFactory.getLog(SchedulerService.class);

    public void testScheduler() {
        System.out.println((new Date()) + " Executat metode de test de scheduler");
    }
}
```

En aquest exemple la tasca escriurà per pantalla cada cop que s'executi.

Aplicació Canigó 3.1 sense frontend

Main

Per a poder executar l'aplicació és necessari una classe Main. En aquesta classe carregarem el canigo-core i el servei de planificació de tasques que hem creat:

src/main/java/cat/gencat/howtostandalone/Main.java

```
package cat.gencat.howtostandalone;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Main {

    private static final Log log = LogFactory.getLog(Main.class);

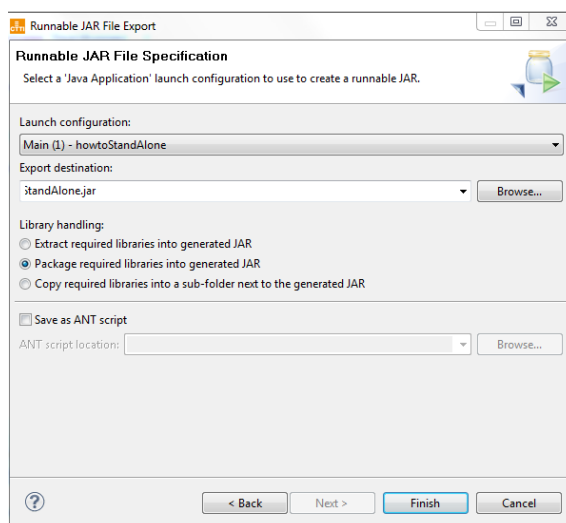
    /**
     * @param args
     */
    public static void main(String[] args) throws Exception{
        new
ClassPathXmlApplicationContext("cat/gencat/ctti/canigo/arch/core/config/canigo-
core.xml");
        new ClassPathXmlApplicationContext("spring/quartz-config.xml");
    }
}
```

Aplicació Canigó 3.1 sense frontend

Generar el Jar

Per a poder executar la nostra aplicació hem de generar el jar. Des d'Eclipse això es pot fer de la següent manera:

- Botó dret al projecte i prémer Exportar
- Seleccionar Java/Runnable Jar File
- Indicar la classe que s'executarà, el nom del fitxer de destí i com volem deixar les llibreries

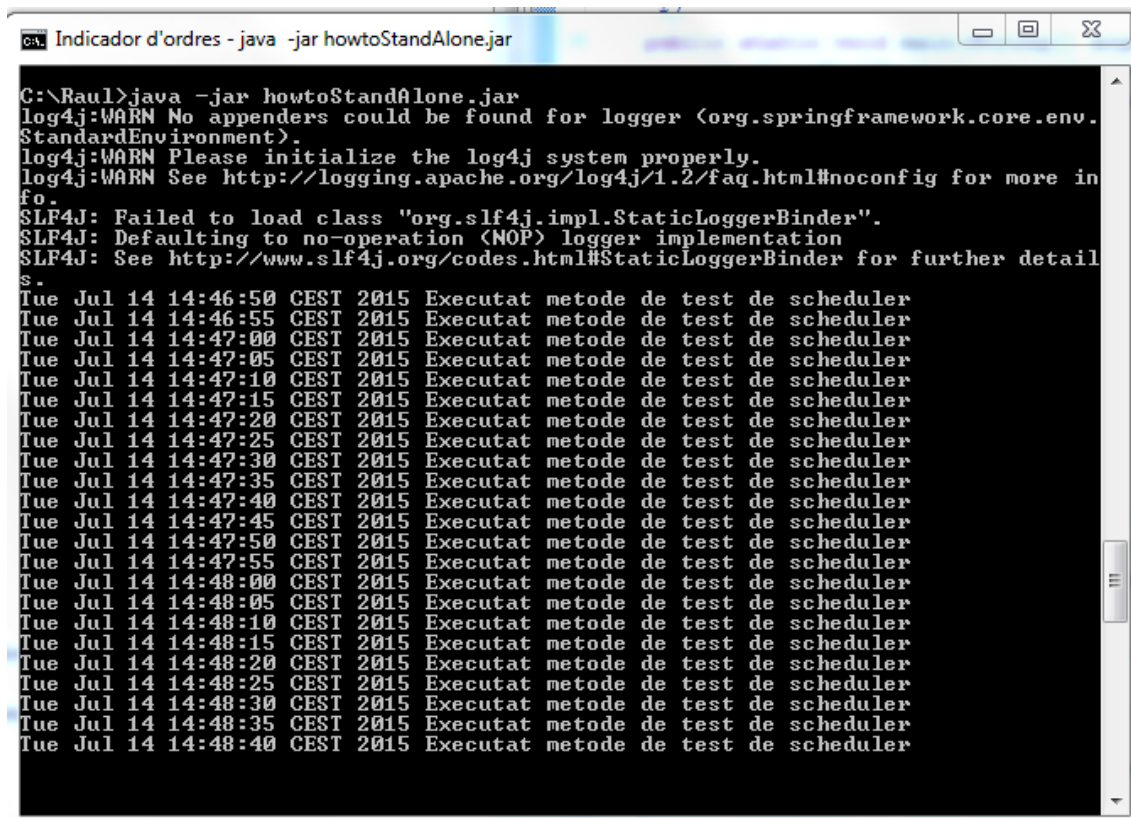


Aplicació Canigó 3.1 sense frontend

Resultat

Per a executar el backend Canigó només cal fer la següent crida:

```
java -jar howtoStandAlone.jar
```



```
C:\Raul>java -jar howtoStandAlone.jar
log4j:WARN No appenders could be found for logger <org.springframework.core.env.
StandardEnvironment>.
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more in
fo.
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further detail
s.
Tue Jul 14 14:46:50 CEST 2015 Executat metode de test de scheduler
Tue Jul 14 14:46:55 CEST 2015 Executat metode de test de scheduler
Tue Jul 14 14:47:00 CEST 2015 Executat metode de test de scheduler
Tue Jul 14 14:47:05 CEST 2015 Executat metode de test de scheduler
Tue Jul 14 14:47:10 CEST 2015 Executat metode de test de scheduler
Tue Jul 14 14:47:15 CEST 2015 Executat metode de test de scheduler
Tue Jul 14 14:47:20 CEST 2015 Executat metode de test de scheduler
Tue Jul 14 14:47:25 CEST 2015 Executat metode de test de scheduler
Tue Jul 14 14:47:30 CEST 2015 Executat metode de test de scheduler
Tue Jul 14 14:47:35 CEST 2015 Executat metode de test de scheduler
Tue Jul 14 14:47:40 CEST 2015 Executat metode de test de scheduler
Tue Jul 14 14:47:45 CEST 2015 Executat metode de test de scheduler
Tue Jul 14 14:47:50 CEST 2015 Executat metode de test de scheduler
Tue Jul 14 14:47:55 CEST 2015 Executat metode de test de scheduler
Tue Jul 14 14:48:00 CEST 2015 Executat metode de test de scheduler
Tue Jul 14 14:48:05 CEST 2015 Executat metode de test de scheduler
Tue Jul 14 14:48:10 CEST 2015 Executat metode de test de scheduler
Tue Jul 14 14:48:15 CEST 2015 Executat metode de test de scheduler
Tue Jul 14 14:48:20 CEST 2015 Executat metode de test de scheduler
Tue Jul 14 14:48:25 CEST 2015 Executat metode de test de scheduler
Tue Jul 14 14:48:30 CEST 2015 Executat metode de test de scheduler
Tue Jul 14 14:48:35 CEST 2015 Executat metode de test de scheduler
Tue Jul 14 14:48:40 CEST 2015 Executat metode de test de scheduler
```

Com es pot observar, per executar aquest tipus de backend, no cal cap servidor d'aplicacions ni contenidor de servlets.