

Generació aplicació Canigó 3.1. amb plugin d'Eclipse

A qui va dirigit

Aquest how-to va dirigit als perfils tècnics (desenvolupadors i arquitectes) que vulguin desenvolupar una aplicació Canigó 3.1.

Versió de Canigó

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

Introducció

En aquest how-to expliquem com crear l'esquelet d'una aplicació Canigó 3.1 utilitzant la última versió del plugin de Canigó(1.4.0) per a Eclipse. L'eclipse utilitzat és la versió Mars (4.5)

Generació aplicació Canigó 3.1. amb plugin d'Eclipse

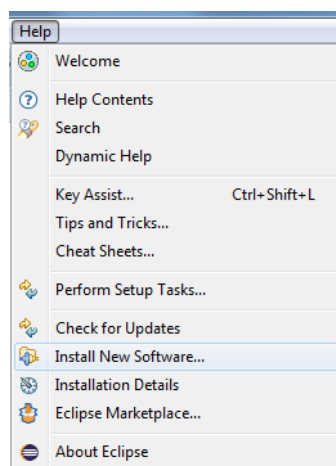
Obtenir Eclipse Mars

Si encara no teniu la versió Mars del IDE Eclipse es pot descarregar del següent enllaç:

<https://eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/marsr>

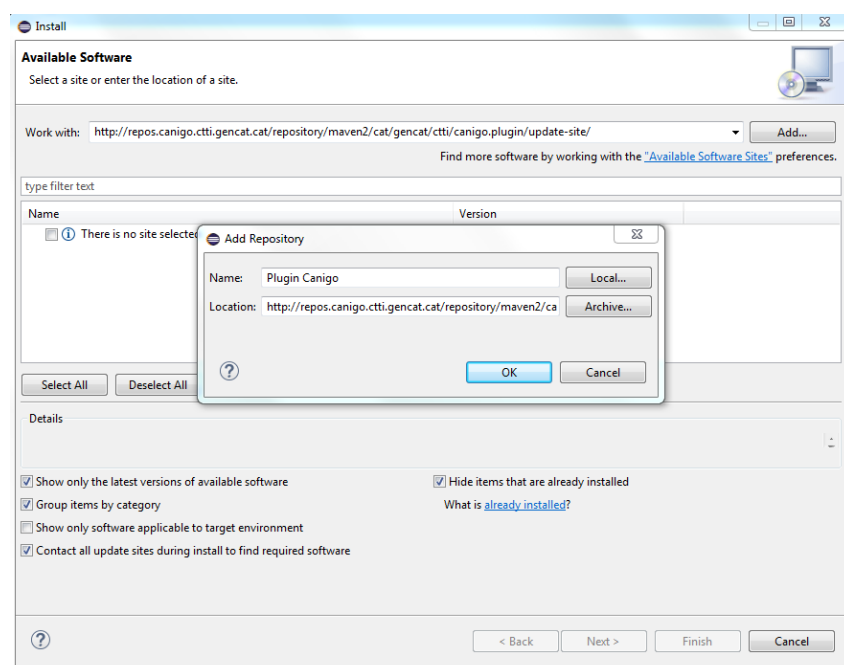
Instal·lar el Plugin

A la pestanya Help de l'Eclipse seleccionem "Install New Software..."

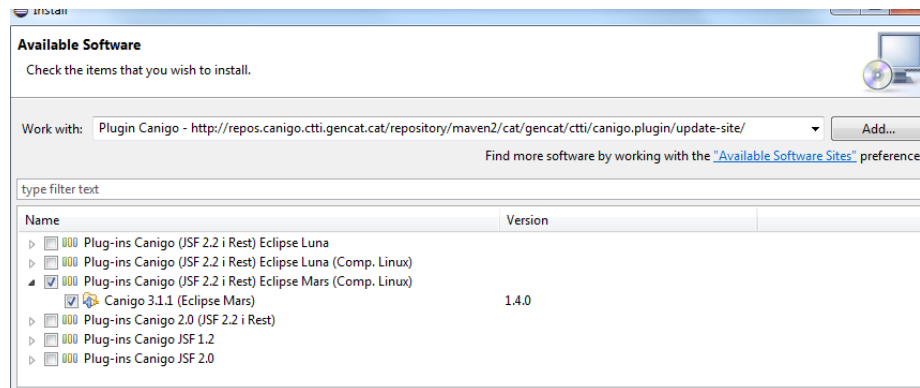


Premem sobre el botó Add i afegim el repository de Canigó.

<http://repos.canigo.ctti.gencat.cat/repository/maven2/cat/gencat/ctti/canigo.plugin/update-site/>



Generació aplicació Canigó 3.1. amb plugin d'Eclipse



Seleccionem Plug-ins Canigo (JSF 2.2 i Rest) Eclipse Mars (Comp.Linux).

Una vegada instal·lat tanquem l'Eclipse i sobreescrivim la llibreria:

<eclipse_home>/plugins/org.eclipse.m2e.maven.runtime_1.6.0.20150526-2031/jars/maven-embedder-3.3.3.jar

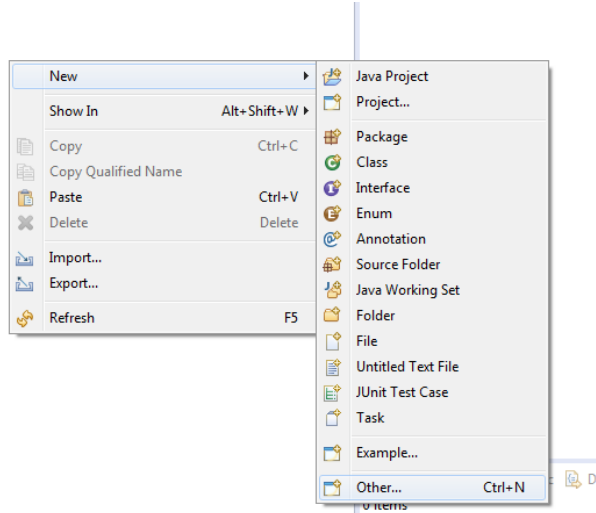
Amb la que podem descarregar a:

http://canigo.ctti.gencat.cat/devenv/patch_plugin_canigo/maven-embedder-3.3.3.jar

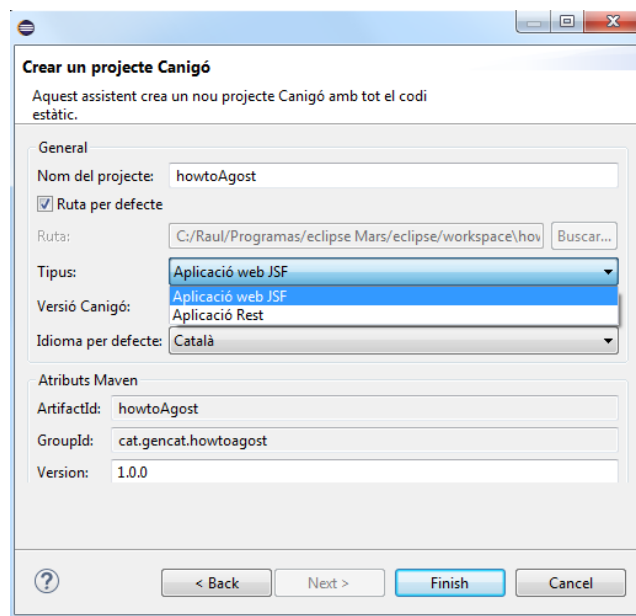
Generació aplicació Canigó 3.1. amb plugin d'Eclipse

Crear aplicació Canigó

A la vista Package Explorer de l'Eclipse fem botó dret->New->Other



I seleccionem Assistant Projectes Canigó -> Crear un Projecte Canigó



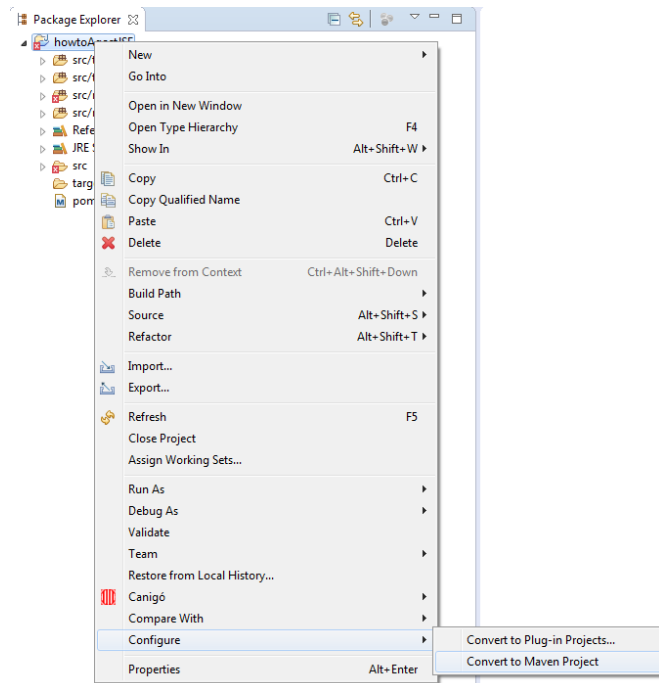
El plugin pot generar dos tipus d'aplicacions:

- Aplicació web JSF: Crea una aplicació demo amb el front-end utilitzant Primefaces 5.0
- Aplicació REST: Crea una aplicació demo amb un CRUD de demo implementat amb HTML5/Javascript. Es dóna llibertat per triar la tecnologia per a realitzar el front-end (AngularJS, Bootstrap, EmberJS...) sempre que es compleixi el PIV de Gencat.

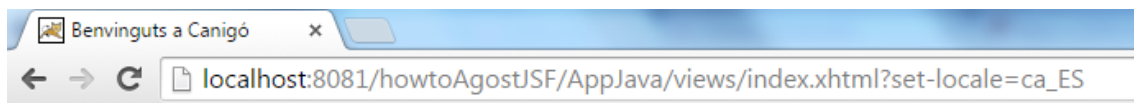
Generació aplicació Canigó 3.1. amb plugin d'Eclipse

Si seleccionem Aplicació web JSF:

Es genera un projecte a l'eclipse. Premem el botó dret sobre el nou projecte i seleccionem Configure -> Convert to maven project



Una vegada convertit a maven project els errors que tenia el projecte desapareixen al actualitzar-se les llibreries. Compilem el projecte i el despleguem i ja tenim en marxa l'aplicació Demo Canigó 3.1



CSCanigó

Plantilla Canigó

Apartat 1

Administració

[Apartat 1](#) > [Administració](#)

Benvinguts a Canigó

Generació aplicació Canigó 3.1. amb plugin d'Eclipse

Si seleccionem Aplicació REST:

A l'arrel del projecte existeix un fitxer README.txt amb algunes indicacions:

1-Susbtituir RUTA_A_SUSTITUIR per la ruta corresponent:

Es cerca al projecte la frase "RUTA_A_SUSTITUIR" i es substitueix per la correcta segons el nom que li haguem ficat al nostre projecte.

```

WEB-INF
├── dispatcher-servlet.xml
│   └── 10: <context:component-scan base-package="cat.gencat.RUTA_A_SUSTITUIR.endpoints" />
├── web.xml
│   └── 17: <filter-class>cat.gencat.RUTA_A_SUSTITUIR.filters.CorsFilter</filter-class>
└──

```

En el nostre cas que hem posat howtoagostrest com a nom de projecte quedaria:
cat.gencat.howtoagostrest...

2-A la carpeta client, els fitxers html funcionen desplegant l'aplicació a

http://localhost:8081/serveisrestSimple en cas de desplegar en una altre direcció, substituir.

```

'serveisrestSimple' - / matches in workspace
├── howtoAgostREST
│   └── client
│       ├── index.html
│       │   └── 17: <a href="http://localhost:8081/serveisrestSimple/services/equipaments/?rpp=100&page=1">Llistat equipament</a>
│       ├── test-CREATE.html (2 matches)
│       │   ├── 24: url: 'http://localhost:8081/serveisrestSimple/services/equipaments/'
│       │   └── 42: <form id="equipmentForm" name="equipmentForm" action="http://localhost:8081/serveisrestSimple/services/equipaments/" method="POST">
│       ├── test-DELETE.html
│       │   └── 17: url: "http://localhost:8081/serveisrestSimple/services/equipaments/" + $('#id').val(),
│       ├── test-READ.html
│       │   └── 20: url: "http://localhost:8081/serveisrestSimple/services/equipaments/" + $('#id').val(),
│       └── test-UPDATE.html
│           └── 24: url: 'http://localhost:8081/serveisrestSimple/services/equipaments/' + $('#id').val(),

```

Al nostre cas també ho substituïm per <http://localhost:8081/howtoAgostREST>

Premem el botó dret sobre el nou projecte i seleccionem Configure -> Convert to maven project i el compilem.

Despleguem el war generat a un servidor d'aplicacions i arrastrem el fitxer index.html fins a un navegador:

