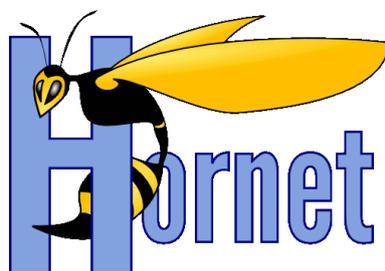




Débuter avec Hornet



Guide de création d'un projet Hornet

Cette création est mise à disposition selon le Contrat Paternité - Pas d'Utilisation Commerciale - Partage des Conditions Initiales à l'Identique disponible en ligne <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/> ou par courrier postal à Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA

SUIVI DES MODIFICATIONS

Version	Auteur	Description	Vérification	Date
1.0	D. Ferreira F. Bernier- Malcoiffe A. Frigout	Initialisation	S. Heurtematte	19/12/12
1.1	F. Bernier- Malcoiffe	Mise à jour du §1.2.3 Repository		04/03/12

DOCUMENTS DE REFERENCE

Version	Titre

SOMMAIRE

SUIVI DES MODIFICATIONS	2
DOCUMENTS DE REFERENCE	2
SOMMAIRE.....	3
TABLEAUX	3
FIGURES.....	3
1 INTRODUCTION ET PRE REQUIS	5
1.1 OBJET.....	5
1.2 PRE REQUIS	5
1.2.1 Logiciels.....	5
1.2.2 Encodage.....	5
1.2.3 Repository Ivy	6
2 CREATION D'UN PROJET HORNET	7
2.1 RECUPERATION DE HORNETTEMPLATE	7
2.2 INITIALISATION DU TEMPLATE	8
2.3 GENERATION D'UN PROJET.....	9
2.3.1 Types de projet.....	9
2.3.2 Projet basique	10
2.3.3 Projet avec la configuration ClamAV	15
2.3.4 Projet avec des pages d'exemples.....	17
2.3.5 Projet avec la configuration « Tests de Vie »	20
2.4 FINALISATION DU TEMPLATE.....	23
3 AJOUT DE NOUVEAU SERVICE.....	25
3.1 INTEGRATION DES TESTS DE VIE	25
3.1.1 Ajout des artefacts.....	25
3.1.2 Configuration de l'application.....	26
4 PROBLEMES CONNUS.....	28
4.1 1 ^{ERE} EXECUTION DES TACHES « SETPROJECTHORNET ».....	28
4.2 LES MODIFICATIONS DES PAGES NE SONT PAS PRISES EN COMPTE.....	28
5 ANNEXES.....	29
5.1 CONTENU INITIAL DE HORNETTEMPLATE	29
5.2 CONFIGURATION IVY	29
5.3 CONFIGURATION DES RESSOURCES STATIQUES	30
5.4 DETAIL DE LA CONSOLE LORS DE LA TACHE « IMPORTHORNETLIBSDEFAULT »	30

TABLEAUX

Aucune entrée de table d'illustration n'a été trouvée.

FIGURES

Figure 1 : vérification de l'encoding de la tâche Ant	6
Figure 2 : Récupération du projet sous SVN	7
Figure 3 : Tâche « initHornet » du « buildTemplate.xml »	8
Figure 4 : Saisie du répertoire Web (Par défaut : « WebContent »).....	8
Figure 5 : Résultat de la console après la tâche « initHornet »	9
Figure 6 : Vue du projet après initialisation du template	9
Figure 7 : Tâches du fichier « buildTmp.xml »	10
Figure 8 : Saisie du nom du package racine (par défaut : « hornettemplate »).....	11
Figure 9 : Résultat de la console après la tâche « setProjectHornetDefaut ».....	12

Figure 10 : Vue du projet après exécution de la tâche « setProjectHornet »	12
Figure 11 : Projet basique / Page « Accueil »	13
Figure 12 : Projet basique / Page « Plan du Site »	14
Figure 13 : Projet basique / Page « Accessibilité »	14
Figure 14 : Saisie du nom du package racine (par défaut : « hornettemplate »)	15
Figure 15 : Résultat de la console après la tâche « setProjectHornetWithClamAV »	16
Figure 16 : Saisie du nom du package racine (par défaut : « hornettemplate »)	17
Figure 17 : Résultat de la console après la tâche « setProjectHornetWithExample »	18
Figure 18 : Projet avec des pages d'exemples / Page « Exemple 1 »	18
Figure 19 : Projet avec des pages d'exemples / Page « Exemple 2 »	19
Figure 20 : Projet avec des pages d'exemples / Page « Plan du Site »	19
Figure 21 : Projet avec des pages d'exemples / Détail du Menu	19
Figure 22 : Saisie du nom du package racine (par défaut : « hornettemplate »)	20
Figure 23 : Résultat de la console après la tâche « setProjectHornetWithTestVie »	21
Figure 24 : Projet avec la configuration Test de Vie / Page « Info Serveur » (Test de Vie)	22
Figure 25 : Projet avec la configuration Test de Vie / Page « Système » (Test de Vie)	22
Figure 26 : Projet avec la configuration Test de Vie / Page « Test de Vie » (Test de Vie)	23
Figure 27 : Projet avec la configuration Test de Vie / Page « Détail » (Test de Vie)	23
Figure 28 : Résultat de la console après la tâche « cleanAllTemplate »	24
Figure 29 : Vue du projet après exécution de la tâche « cleanAllTemplate »	24
Figure 30 : Contenu initial du projet	29

1 Introduction et pré requis

1.1 *Objet*

Ce document présente les étapes de création d'un projet Hornet en utilisant « **hornettemplate** » en version 2.0 qui utilise la version 2.0 du Framework « **Hornet** ».

Une deuxième partie concerne l'intégration de nouveaux services au sein d'une application Hornet existante.

1.2 *Pré requis*

1.2.1 Logiciels

- Eclipse Galiléo SR1 (v 3.5.1) et supérieur.
- Ant 1.7.1 et supérieur

1.2.2 Encodage

1.2.2.1 Au niveau du projet

L'encodage des caractères doit être configuré en ISO-8859-1.

Clic droit sur le projet, « Properties », puis « Text file encoding »).

1.2.2.2 Tâches Ant

Les tâches Ant doivent hériter de l'encodage défini au niveau du projet.

Cependant, Eclipse montre un comportement instable à ce sujet. Aussi, il est recommandé de vérifier que l'encodage est correct avant d'exécuter une tâche. Pour ce faire, accéder aux propriétés de la tâche Ant :

Clic droit sur la tâche > « Run As » > « Ant build... » > onglet « Common » > Console encoding

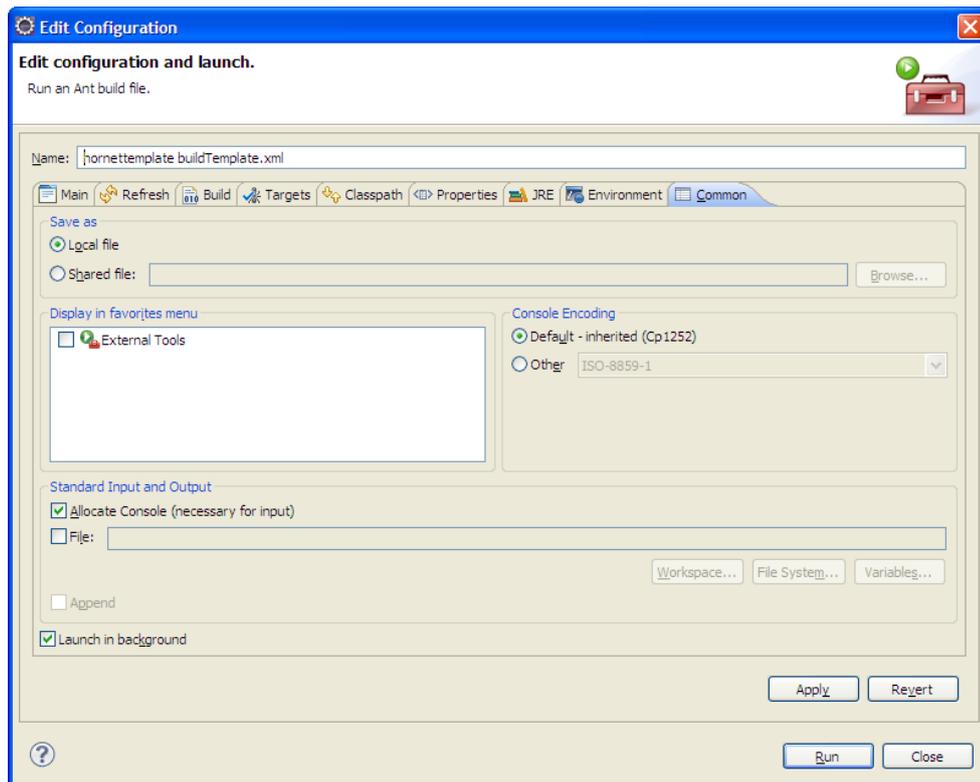


Figure 1 : vérification de l'encoding de la tâche Ant

Si la valeur de « Console encoding » est différente de celle spécifiée dans le fichier XML, il faut la modifier. Puis cliquer sur « Apply ».

1.2.3 Repository Ivy

Le repository Hornet le plus récent, utilisé par Ivy, doit être installé en local.

Créer un répertoire local qui sert de référence dans la suite du document pour la configuration des propriétés Ivy. Le chemin vers ce répertoire est représenté par la variable <REPertoire REPOSITORY>.

Récupérer l'archive contenant le repository (de la forme « Repository_AAAAMMDD.zip »). Cette archive contient un répertoire « Repository_AAAAMMDD », qui contient lui-même les sous-répertoires « technique », « cache », « snapshot » et « metier ».

Extraire l'archive et copier le contenu du répertoire « Repository_AAAAMMDD » dans <REPertoire REPOSITORY>.

L'arborescence créée est la suivante :

- <REPertoire REPOSITORY>/technique
- <REPertoire REPOSITORY>/cache
- <REPertoire REPOSITORY>/snapshot
- <REPertoire REPOSITORY>/metier

2 Création d'un projet Hornet

2.1 Récupération de *hornettemplate*

Récupérer le projet « **hornettemplate** » disponible sur le SVN de la forge Adullact.



Figure 2 : Récupération du projet sous SVN

Le nom donné au projet sous Eclipse n'interfère pas dans le déroulement de la génération du projet. Cependant, il est convenu que celui-ci ne doit pas contenir d'espace et doit être en minuscule.

Cf. 5.1 Contenu initial de *hornettemplate*.

2.2 Initialisation du template

Dans la vue Ant, lancer la tâche (cf. 1.2.2.2 Tâches Ant) par défaut « **initHornet** » du fichier XML « **buildTemplate.xml** ».

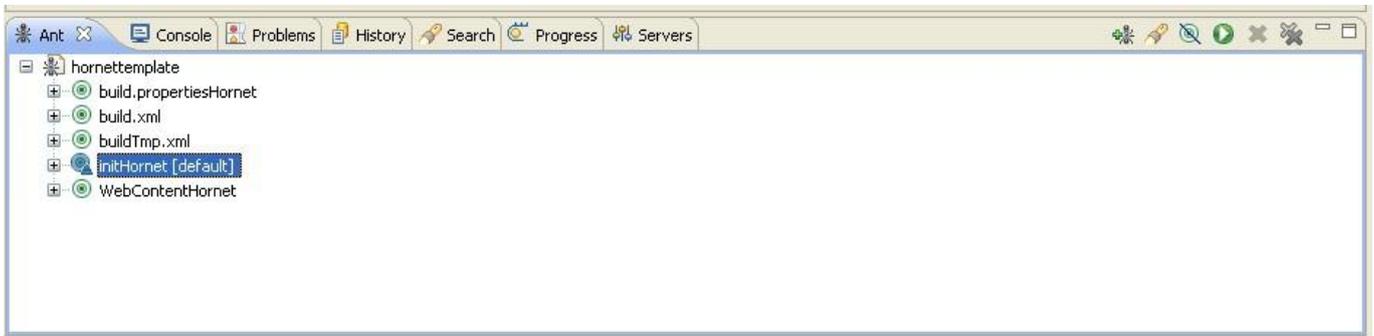


Figure 3 : Tâche « **initHornet** » du « **buildTemplate.xml** »

Un prompt demande la saisie du nom du répertoire Web (par défaut : « **WebContent** »).

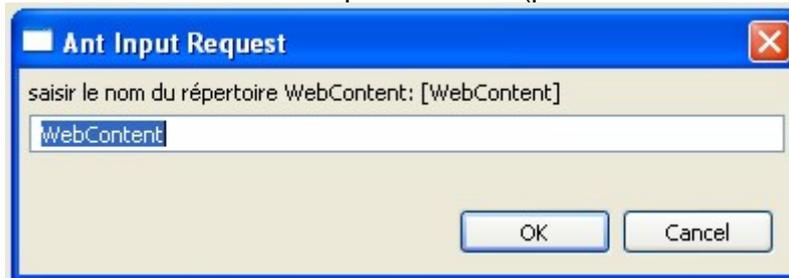


Figure 4 : Saisie du répertoire Web (Par défaut : « **WebContent** »)

La console affiche le résultat suivant :

```
Buildfile: D:\devjava\workspace_mae\hornettemplate\buildTemplate.xml
initHornet:
  [echo] initialisation du projet
  [mkdir] Created dir: D:\devjava\workspace_mae\hornettemplate\lib
  [mkdir] Created dir: D:\devjava\workspace_mae\hornettemplate\lib\dev
WebContentHornet:
  [echo] creation du repertoire WebContent WebContent
build.propertiesHornet:
  [echo] generation du fichier build.properties Hornet
  [xslt] Processing D:\devjava\workspace_mae\hornettemplate\.project to
D:\devjava\workspace_mae\hornettemplate\build.properties
  [xslt] Loading stylesheet D:\devjava\workspace_mae\hornettemplate\dev\style\propertiesHornet.xsl
build.xml:
  [echo] generation du fichier build.xml
  [xslt] Processing D:\devjava\workspace_mae\hornettemplate\dev\hornet\build\build.xml to
D:\devjava\workspace_mae\hornettemplate\build.xml
  [xslt] Loading stylesheet D:\devjava\workspace_mae\hornettemplate\dev\style\build.xsl
buildTmp.xml:
  [echo] generation du fichier buildTmp.xml
  [xslt] Processing D:\devjava\workspace_mae\hornettemplate\dev\hornet\build\buildTmp.xml to
D:\devjava\workspace_mae\hornettemplate\buildTmp.xml
  [xslt] Loading stylesheet D:\devjava\workspace_mae\hornettemplate\dev\style\buildTmp.xsl
[delete] Deleting: D:\devjava\workspace_mae\hornettemplate\buildTemplate.xml
BUILD SUCCESSFUL
Total time: 1 minute
```

Figure 5 : Résultat de la console après la tâche « **initHornet** »

Un rafraichissement du projet donne la vue suivante :

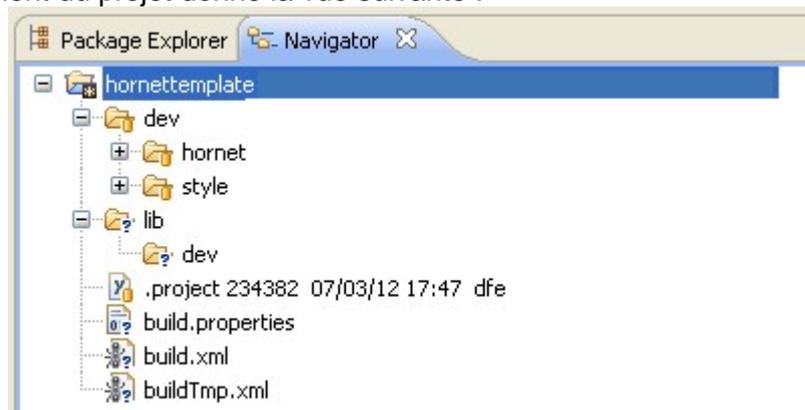


Figure 6 : Vue du projet après initialisation du template

Les changements sont :

- Le fichier « **buildTemplate.xml** » a été supprimé.
- Les répertoires « **lib/dev** » ont été créés.
- Le fichier « **buildTmp.xml** » a été créé.
- Les fichiers « **build.xml** » et « **build.properties** » ont été créés.

2.3 Génération d'un projet

2.3.1 Types de projet

Le fichier « **buildTmp.xml** » contient les tâches de génération du projet.

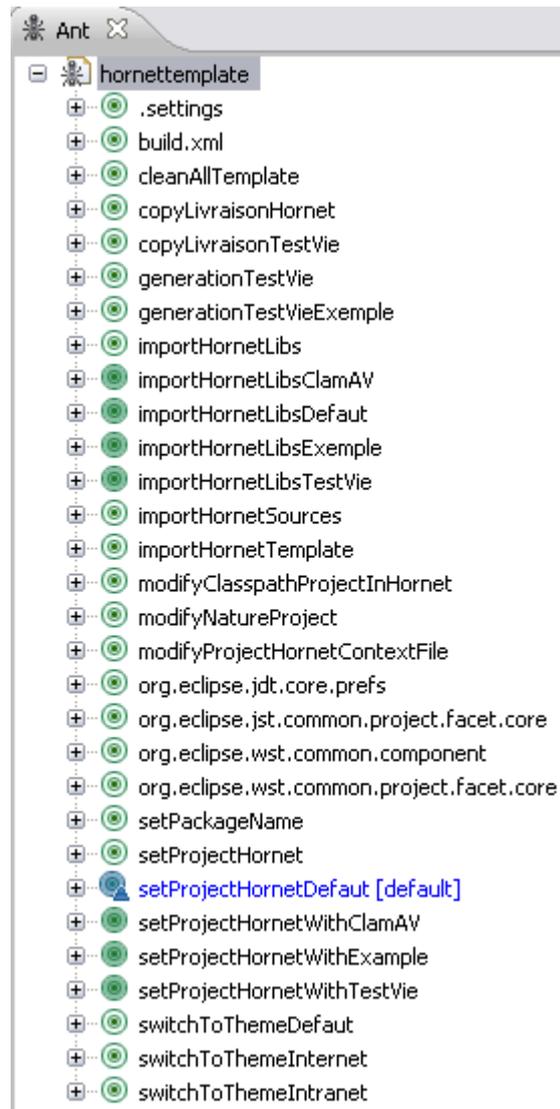


Figure 7 : Tâches du fichier « buildTmp.xml »

Afin de poursuivre la génération du projet, il faut ensuite **IMPERATIVEMENT** lancer une des tâches suivantes :

- « **setProjectHornetDefault** » : permet la génération d'un projet avec le thème par défaut.
- « **setProjectHornetWithClamAV** » : permet la génération d'un projet avec le thème par défaut et la configuration pour ClamAV.
- « **setProjectHornetWithExample** » : permet la génération d'un projet avec le thème par défaut et des pages d'exemples.
- « **setProjectHornetWithTestVie** » : permet la génération d'un projet avec le thème par défaut et la configuration pour les Tests de Vie.

2.3.2 Projet basique

1. Configurer le fichier **build.properties**, en particulier la propriété **build.projectName** qui définit le contexte de l'application
2. Lancer la tâche par défaut « **setProjectHornetDefault** » (cf. 1.2.2.2 Tâches Ant) du « **buildTmp.xml** ».

Un prompt demande la saisie du nom du package racine (par défaut : « **hornettemplate** »).



Figure 8 : Saisie du nom du package racine (par défaut : « **hornettemplate** »)

La console affiche le résultat suivant :

```
Buildfile: D:\devjava\workspace_mae\hornettemplate\buildTmp.xml
setPackageName:
setProjectHornetDefault:
    [echo] cr?ation du projet hornet (defaut) : hornettemplate
setProjectHornet:
.settings:
org.eclipse.wst.common.component:
    [xslt] Processing D:\devjava\workspace_mae\hornettemplate\dev\hornet\xml\blank.xml to
D:\devjava\workspace_mae\hornettemplate\.settings\org.eclipse.wst.common.component
    [xslt] Loading stylesheet D:\devjava\workspace_mae\hornettemplate\dev\style\org.eclipse.wst.common.xsl
org.eclipse.wst.common.project.facet.core:
    [xslt] Processing D:\devjava\workspace_mae\hornettemplate\dev\hornet\xml\blank.xml to
D:\devjava\workspace_mae\hornettemplate\.settings\org.eclipse.wst.common.project.facet.core.xml
    [xslt] Loading stylesheet
D:\devjava\workspace_mae\hornettemplate\dev\style\org.eclipse.wst.common.project.facet.core.xsl
org.eclipse.jst.common.project.facet.core:
    [xslt] Processing D:\devjava\workspace_mae\hornettemplate\dev\hornet\xml\blank.xml to
D:\devjava\workspace_mae\hornettemplate\.settings\org.eclipse.jst.common.project.facet.core.prefs
    [xslt] Loading stylesheet
D:\devjava\workspace_mae\hornettemplate\dev\style\org.eclipse.jst.common.project.facet.core.xsl
org.eclipse.jdt.core.prefs:
    [xslt] Processing D:\devjava\workspace_mae\hornettemplate\dev\hornet\xml\blank.xml to
D:\devjava\workspace_mae\hornettemplate\.settings\org.eclipse.jdt.core.prefs
    [xslt] Loading stylesheet D:\devjava\workspace_mae\hornettemplate\dev\style\org.eclipse.jdt.core.xsl
modifyNatureProject:
    [echo] modification de la nature du projet
[delete] Deleting: D:\devjava\workspace_mae\hornettemplate\.project
    [xslt] Processing D:\devjava\workspace_mae\hornettemplate\dev\hornet\ eclipse\core\.project to
D:\devjava\workspace_mae\hornettemplate\.project
    [xslt] Loading stylesheet D:\devjava\workspace_mae\hornettemplate\dev\style\project.xsl
importHornetTemplate:
    [echo] import du WebContent (Hornet)
[copy] Copying 23 files to D:\devjava\workspace_mae\hornettemplate\WebContent
    [echo] copie des images static
[copy] Copying 1 file to D:\devjava\workspace_mae\hornettemplate\WebContent\static\images
importHornetSources:
    [echo] import des sources du projet (Hornet)
[copy] Copying 7 files to D:\devjava\workspace_mae\hornettemplate\src\config
[copy] Copying 5 files to
D:\devjava\workspace_mae\hornettemplate\src\config\fr\gouv\diplomatie\hornettemplate
[copy] Copying 6 files to
D:\devjava\workspace_mae\hornettemplate\src\java\fr\gouv\diplomatie\hornettemplate
[copy] Copied 11 empty directories to 3 empty directories under
D:\devjava\workspace_mae\hornettemplate\src\java\fr\gouv\diplomatie\hornettemplate
    [echo] import des fichiers de test du projet (Hornet)
[copy] Copying 1 file to D:\devjava\workspace_mae\hornettemplate\tst\java
[copy] Copied 7 empty directories to 2 empty directories under
D:\devjava\workspace_mae\hornettemplate\tst\java
[copy] Copied 7 empty directories to 5 empty directories under
D:\devjava\workspace_mae\hornettemplate\tst\java\fr\gouv\diplomatie\hornettemplate
[copy] Copying 1 file to D:\devjava\workspace_mae\hornettemplate\tst\config
    [echo] import des librairies de test du projet (Hornet)
[copy] Copying 3 files to D:\devjava\workspace_mae\hornettemplate\lib
    [echo] import de la configuration ivy (Hornet)
[copy] Copying 2 files to D:\devjava\workspace_mae\hornettemplate
importHornetLibs:
    [echo] recuperation des librairies externes depuis une archive ZIP locale

Voir en Annexe le détail de la tâche « importHornetLibs »
modifyClasspathProjectInHornet:
    [echo] modification du classpath du projet (Hornet)
```

```
[copy] Copying 1 file to D:\devjava\workspace_mae\hornettemplate
modifyProjectHornetContextFile:
[echo] modification du fichier de contexte du projet (context.xml)
[xslt] Processing D:\devjava\workspace_mae\hornettemplate\dev\hornet\WebContent\META-INF\context.xml to
D:\devjava\workspace_mae\hornettemplate\WebContent\META-INF\context.xml
[xslt] Loading stylesheet D:\devjava\workspace_mae\hornettemplate\dev\style\contextHornet.xsl
copyLivraisonHornet:
[echo] copie de la structure de livraison
[copy] Copying 2 files to D:\devjava\workspace_mae\hornettemplate\livrable
[copy] Copying 4 files to D:\devjava\workspace_mae\hornettemplate\livraison
BUILD SUCCESSFUL
Total time: 12 seconds
```

Figure 9 : Résultat de la console après la tâche « **setProjectHornetDefault** »

Un rafraichissement du projet donne la vue suivante :

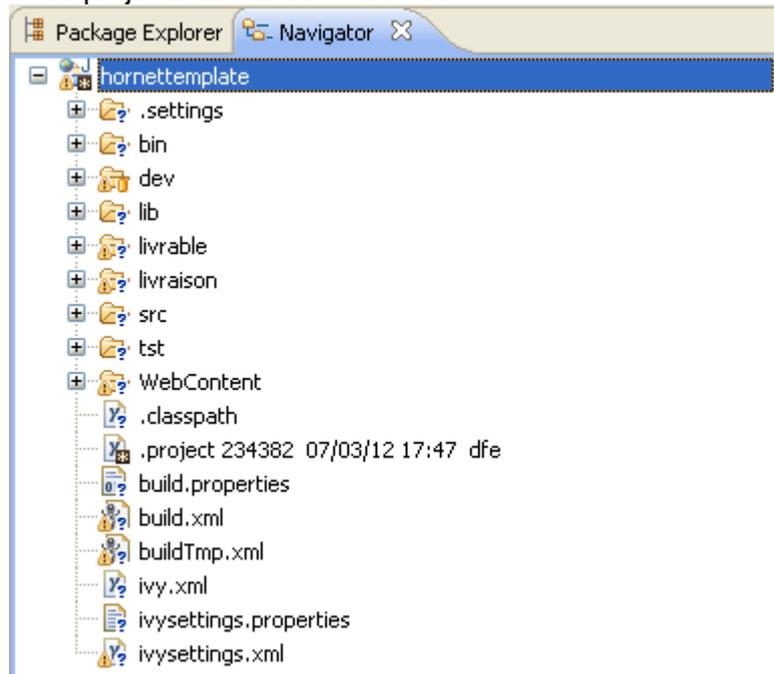


Figure 10 : Vue du projet après exécution de la tâche « **setProjectHornet** »

3. Vérification de la présence des librairies, cf. §5.2 Configuration Ivy.

Les changements sont :

- Création du répertoire « **.settings** » qui contient des fichiers de configurations propre à Eclipse. Cela permet de déclarer le projet comme un projet Web sous Eclipse qui utilise Apache Tomcat et qui a pour répertoire source « **src/config** », « **src/java** », et « **tst/java** ».
- Création du répertoire « **bin** », utilisé par Eclipse et qui contient les sources java compilées du projet
- Ajout de librairies dans le répertoire « **lib** » : le projet est déclaré sous Eclipse comme utilisant les librairies présentes dans « **lib/test** ». Ce sont les librairies utilisés pour les tests, la librairie contenue dans « **lib/dev** » étant utilisée pour la récupération des librairies Hornet par Ivy et le répertoire « **WebContent\WEB-INF\lib** ».
- Création des répertoires « **livrable** » et « **livraison** », utilisés lors de la création des livrables du projet par les tâches du fichier « **build.xml** ».
- Création du répertoire « **src/config** » contenant les fichiers properties, ex : « **log4j.properties** » ou « **mail.properties** », les fichiers de configuration Struts, Spring, Tiles et iBATIS et les fichiers « **package_fr.properties** » pour la gestion de l'internationalisation et de l'externalisation des libellés.

- Création du répertoire « **src/java** » contenant les sources java du projet
- Création du répertoire « **tst** » contenant les fichiers de test du projet.
- Création du répertoire Web « **WebContent** » par défaut, contenant les ressources web de l'application : les libraires Hornet, les pages JSP, les ressources statiques...
- Création du fichier « **.classpath** », fichier de configuration d'Eclipse.
- Création des fichiers « **ivy.xml** », « **ivysettings.properties** » et « **ivysettings.xml** » qui contiennent la configuration Ivy des librairies du projet.

4. Configurez les ressources statiques (cf. §5.3).

2.3.2.1 Vérification

Le projet contient 3 pages par défaut :

- La page « **Accueil** » : page d'accueil de l'application
- La page « **Plan du Site** » : page montrant le plan du site de l'application
- La page « **Accessibilité** » : page présentation un exemple de page sur l'accessibilité à implémenter pour l'application

Le déploiement du projet sous Tomcat nous donne le résultat suivant :

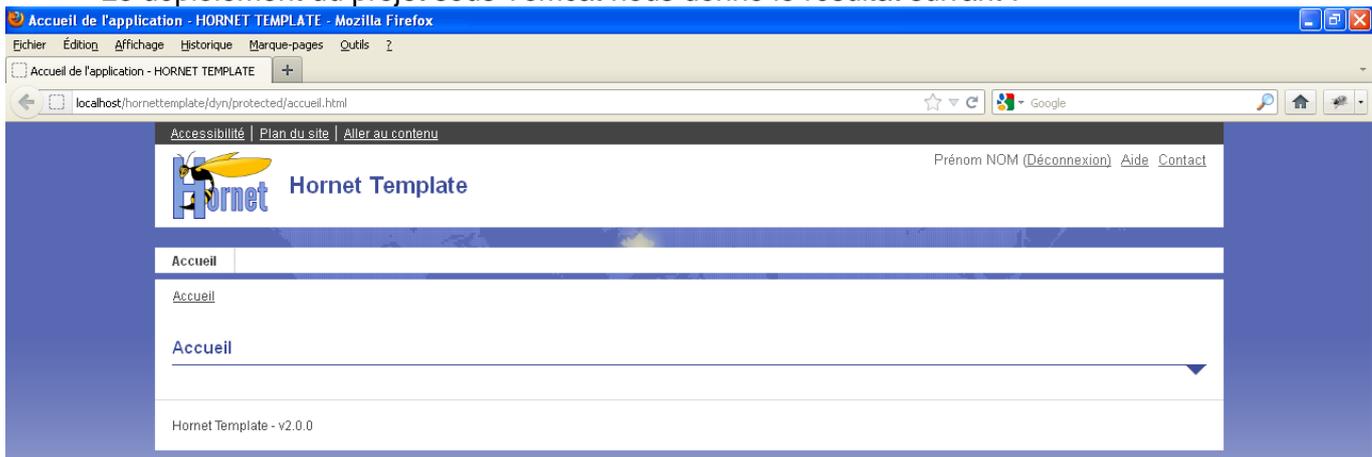


Figure 11 : Projet basique / Page « **Accueil** »

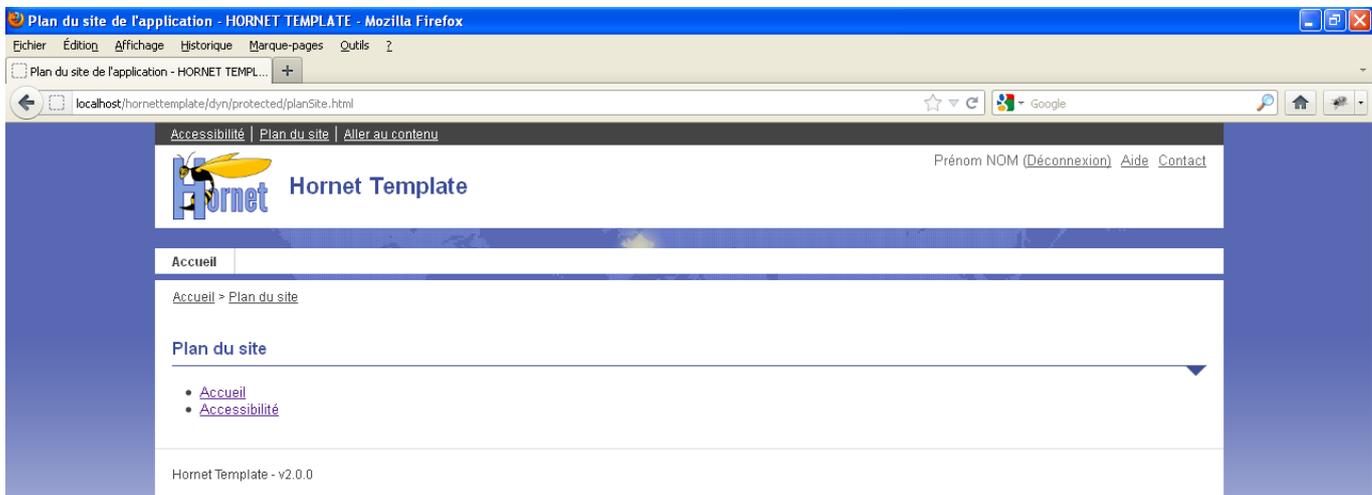
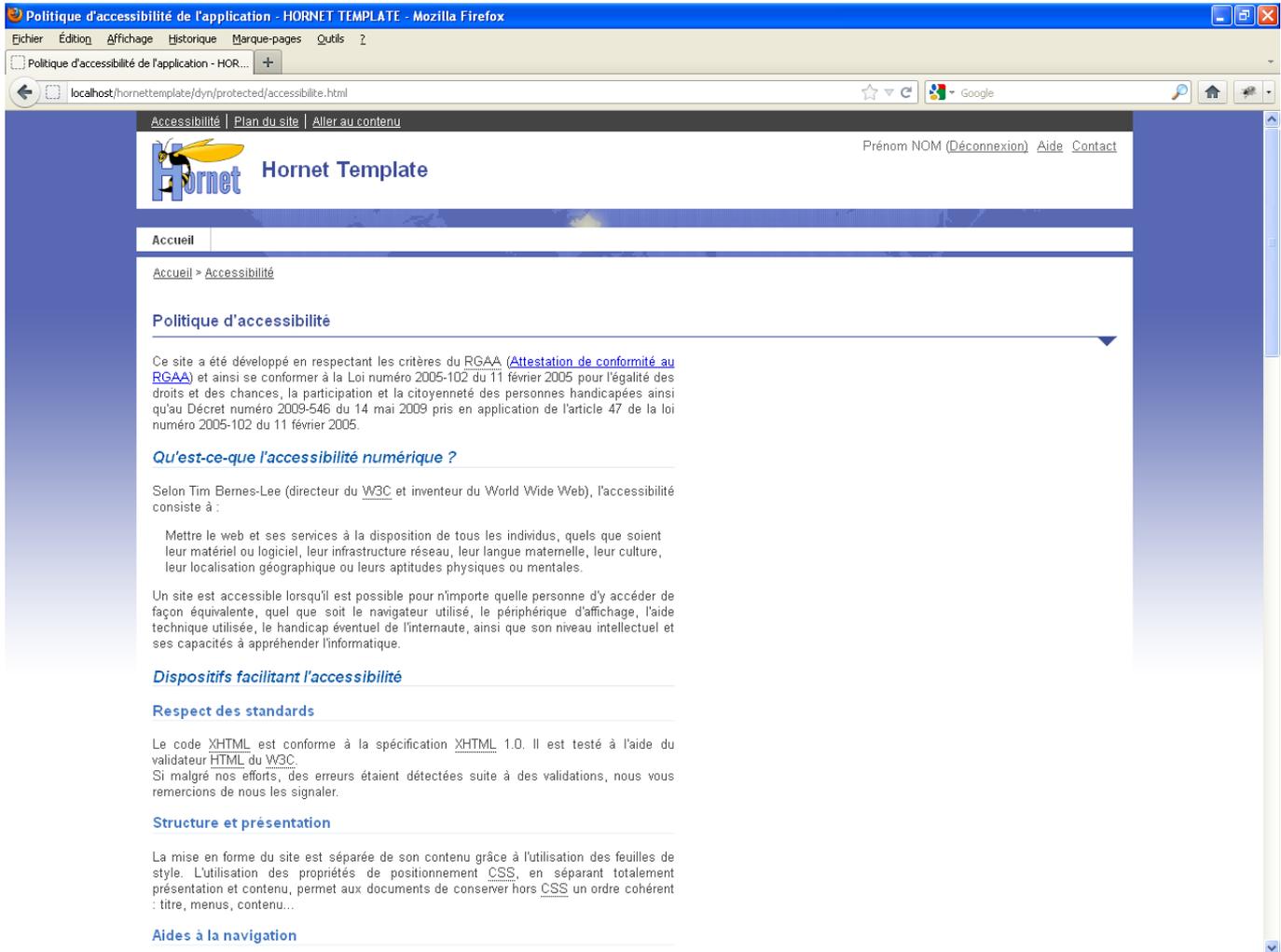


Figure 12 : Projet basique / Page « **Plan du Site** »Figure 13 : Projet basique / Page « **Accessibilité** »

2.3.3 Projet avec la configuration ClamAV

1. Configurer le fichier **build.properties**, en particulier la propriété **build.projectName** qui définit le contexte de l'application
2. Lancer la tâche Ant « **setProjectHornetWithClamAV** » (cf. 1.2.2.2 Tâches Ant) du « buildTmp.xml ».

Un prompt demande la saisie du nom du package racine, par défaut : « **hornettemplate** ».



Figure 14 : Saisie du nom du package racine (par défaut : « **hornettemplate** »)

La console affiche le résultat suivant :

```
Buildfile: D:\devjava\workspace_mae\hornettemplate\buildTmp.xml
setPackageName:
setProjectHornetWithClamAV:
  [echo] cr?ation du projet hornet (defaut) : hornettemplate
setProjectHornet:
.settings:
org.eclipse.wst.common.component:
org.eclipse.wst.common.project.facet.core:
org.eclipse.jst.common.project.facet.core:
org.eclipse.jdt.core.prefs:
modifyNatureProject:
  [echo] modification de la nature du projet
  [delete] Deleting: D:\devjava\workspace_mae\hornettemplate\.project
  [xslt] Processing D:\devjava\workspace_mae\hornettemplate\dev\hornet\eclipse\core\.project to
D:\devjava\workspace_mae\hornettemplate\.project
  [xslt] Loading stylesheet D:\devjava\workspace_mae\hornettemplate\dev\style\project.xsl
importHornetTemplate:
  [echo] import du WebContent (Hornet)
  [copy] Copying 23 files to D:\devjava\workspace_mae\hornettemplate\WebContent
  [echo] copie des images static
  [copy] Copying 1 file to D:\devjava\workspace_mae\hornettemplate\WebContent\static\images
importHornetSources:
  [echo] import des sources du projet (Hornet)
  [copy] Copying 7 files to D:\devjava\workspace_mae\hornettemplate\src\config
  [copy] Copying 5 files to
D:\devjava\workspace_mae\hornettemplate\src\config\fr\gouv\diplomatie\hornettemplate
  [copy] Copying 6 files to
D:\devjava\workspace_mae\hornettemplate\src\java\fr\gouv\diplomatie\hornettemplate
  [echo] import des fichiers de test du projet (Hornet)
  [copy] Copying 1 file to D:\devjava\workspace_mae\hornettemplate\tst\java
  [copy] Copying 1 file to D:\devjava\workspace_mae\hornettemplate\tst\config
  [echo] import des librairies de test du projet (Hornet)
  [copy] Copying 3 files to D:\devjava\workspace_mae\hornettemplate\lib
  [echo] import de la configuration ivy (Hornet)
  [copy] Copying 2 files to D:\devjava\workspace_mae\hornettemplate
importHornetLibs:
  [echo] recuperation des librairies externes depuis une archive ZIP locale

Voir en Annexe le détail de la tâche « importHornetLibs »
modifyClasspathProjectInHornet:
  [echo] modification du classpath du projet (Hornet)
  [copy] Copying 1 file to D:\devjava\workspace_mae\hornettemplate
modifyProjectHornetContextFile:
  [echo] modification du fichier de contexte du projet (context.xml)
copyLivraisonHornet:
  [echo] copie de la structure de livraison
  [copy] Copying 2 files to D:\devjava\workspace_mae\hornettemplate\livrable
```

```
[copy] Copying 4 files to D:\devjava\workspace_mae\hornettemplate\livraison  
BUILD SUCCESSFUL  
Total time: 11 seconds
```

Figure 15 : Résultat de la console après la tâche « **setProjectHornetWithClamAV** »

3. Vérification de la présence des librairies, cf. §5.2 Configuration Ivy.

Le rafraîchissement du projet nous donne la même vue que pour un projet basique.

L'ajout de la configuration ClamAV dans le projet ne rajoute que la librairie « hornetserver-clamav-2.0.0.jar » dans le répertoire « WEB-INF/lib » du projet.

4. Configurer les ressources statiques (cf. §5.3).

2.3.4 Projet avec des pages d'exemples

1. Configurer le fichier **build.properties**, en particulier la propriété **build.projectName** qui définit le contexte de l'application
2. Lancer la tâche « **setProjectHornetWithExample** » (cf. 1.2.2.2 Tâches Ant) du « buildTmp.xml ».

Un prompt demande la saisie du nom du package racine (par défaut : « **hornettemplate** »).



Figure 16 : Saisie du nom du package racine (par défaut : « **hornettemplate** »)

La console affiche le résultat suivant :

```
Buildfile: D:\devjava\workspace_mae\hornettemplate\buildTmp.xml
setPackageName:
setProjectHornetWithExample:
    [echo] cr?ation du projet hornet avec exemple : hornettemplate
setProjectHornet:
.settings:
org.eclipse.wst.common.component:
org.eclipse.wst.common.project.facet.core:
org.eclipse.jst.common.project.facet.core:
org.eclipse.jdt.core.prefs:
modifyNatureProject:
    [echo] modification de la nature du projet
[delete] Deleting: D:\devjava\workspace_mae\hornettemplate\.project
[xslt] Processing D:\devjava\workspace_mae\hornettemplate\dev\hornet\eclipse\core\.project to
D:\devjava\workspace_mae\hornettemplate\.project
[xslt] Loading stylesheet D:\devjava\workspace_mae\hornettemplate\dev\style\project.xsl
importHornetTemplate:
    [echo] import du WebContent (Hornet)
    [copy] Copying 23 files to D:\devjava\workspace_mae\hornettemplate\WebContent
    [echo] copie des images static
    [copy] Copying 1 file to D:\devjava\workspace_mae\hornettemplate\WebContent\static\images
importHornetSources:
    [echo] import des sources du projet (Hornet)
    [copy] Copying 7 files to D:\devjava\workspace_mae\hornettemplate\src\config
    [copy] Copying 5 files to
D:\devjava\workspace_mae\hornettemplate\src\config\fr\gouv\diplomatie\hornettemplate
    [copy] Copying 6 files to
D:\devjava\workspace_mae\hornettemplate\src\java\fr\gouv\diplomatie\hornettemplate
    [echo] import des fichiers de test du projet (Hornet)
    [copy] Copying 1 file to D:\devjava\workspace_mae\hornettemplate\tst\java
    [copy] Copying 1 file to D:\devjava\workspace_mae\hornettemplate\tst\config
    [echo] import des librairies de test du projet (Hornet)
    [copy] Copying 3 files to D:\devjava\workspace_mae\hornettemplate\lib
    [echo] import de la configuration ivy (Hornet)
    [copy] Copying 2 files to D:\devjava\workspace_mae\hornettemplate
importHornetLibs:
    [echo] recuperation des librairies externes depuis une archive ZIP locale

Voir en Annexe le détail de la tâche « importHornetLibs »
modifyClasspathProjectInHornet:
    [echo] modification du classpath du projet (Hornet)
    [copy] Copying 1 file to D:\devjava\workspace_mae\hornettemplate
modifyProjectHornetContextFile:
    [echo] modification du fichier de contexte du projet (context.xml)
copyLivraisonHornet:
    [echo] copie de la structure de livraison
    [copy] Copying 2 files to D:\devjava\workspace_mae\hornettemplate\livrable
```

```
[copy] Copying 4 files to D:\devjava\workspace_mae\hornettemplate\livraison
[echo] ajout de pages html Exemple
[copy] Copying 2 files to D:\devjava\workspace_mae\hornettemplate\src\config
[copy] Copying 1 file to
D:\devjava\workspace_mae\hornettemplate\src\config\fr\gouv\diplomatie\hornettemplate
[copy] Copying 3 files to D:\devjava\workspace_mae\hornettemplate\WebContent
BUILD SUCCESSFUL
Total time: 10 seconds
```

Figure 17 : Résultat de la console après la tâche « **setProjectHornetWithExample** »

3. Vérification de la présence des librairies, cf. §5.2 Configuration Ivy.

Le rafraichissement du projet nous donne la même vue que pour un projet basique. L'ajout des pages d'exemples dans le projet impacte les fichiers suivants :

- Les fichiers « **struts.xml** » et « **tiles.xml** » pour ajouter les chemins relatifs aux deux nouvelles pages d'exemple.
- Le fichier « **package_fr.properties** » contenant les libellés des deux nouvelles pages d'exemple.
- Le fichier « **menu.xml** » qui contient la définition des pages de l'application et qui embarque donc les deux nouvelles pages d'exemple. Ce fichier XML est utilisé pour la génération du menu, du plan du site et du fil d'Ariane.
- Les pages JSP « **page1.jsp** » et « **page2.jsp** ».

Les pages contiennent le texte latin « **Lorem ipsum** » (cf. [Wikipédia](#)).

A noter que les deux pages d'exemple sont affichées dans le menu mais seule la page 1 apparait dans le Plan du site (provient de la configuration dans le fichier « menu.xml »).

4. Configurez les ressources statiques (cf. §5.3).

2.3.4.1 Vérification

Le déploiement du projet sous Tomcat nous donne le résultat suivant :

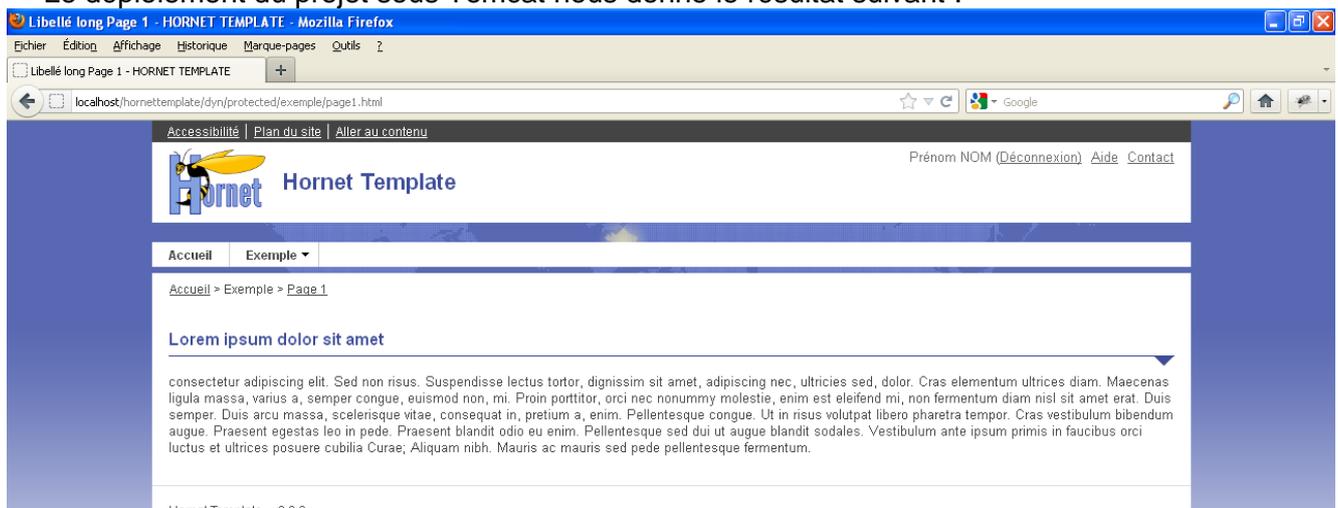


Figure 18 : Projet avec des pages d'exemples / Page « **Exemple 1** »

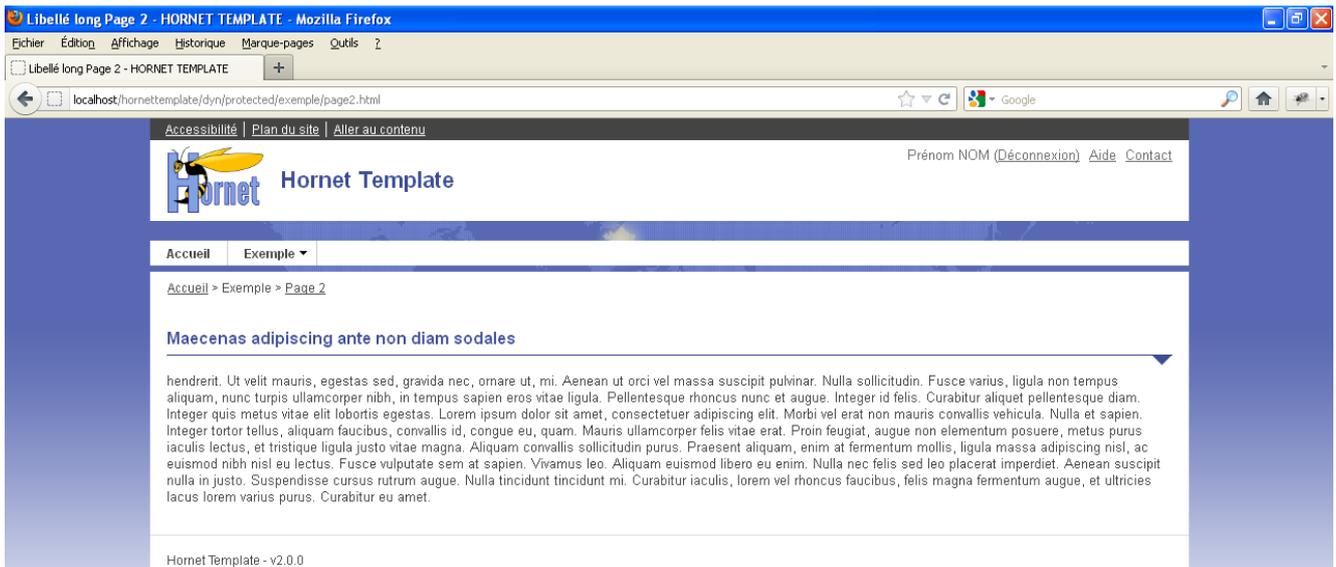


Figure 19 : Projet avec des pages d'exemples / Page « **Exemple 2** »

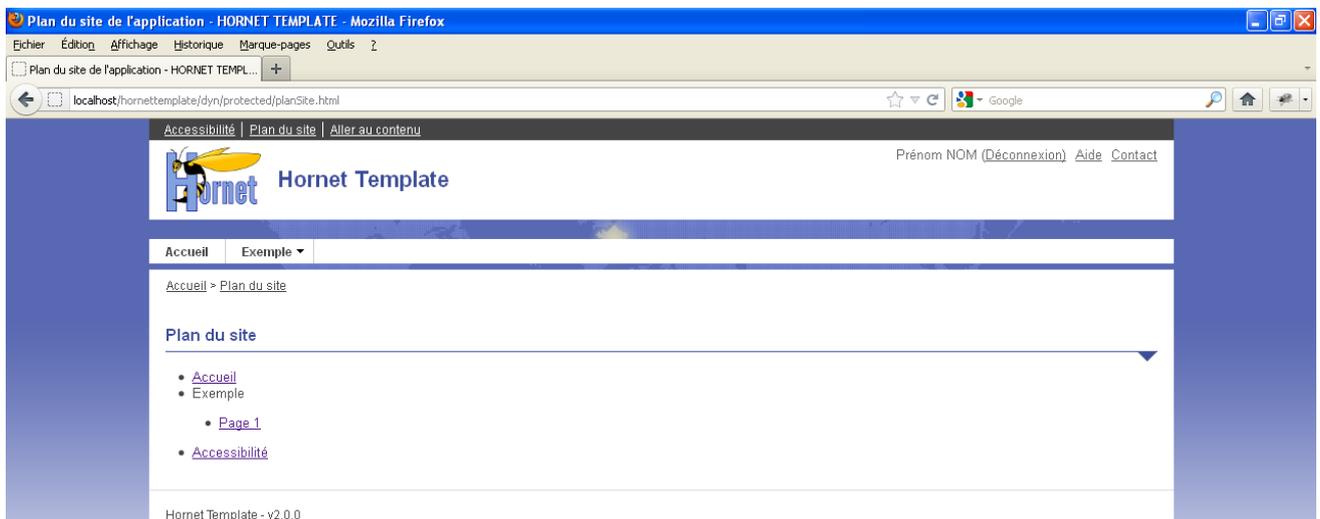


Figure 20 : Projet avec des pages d'exemples / Page « **Plan du Site** »

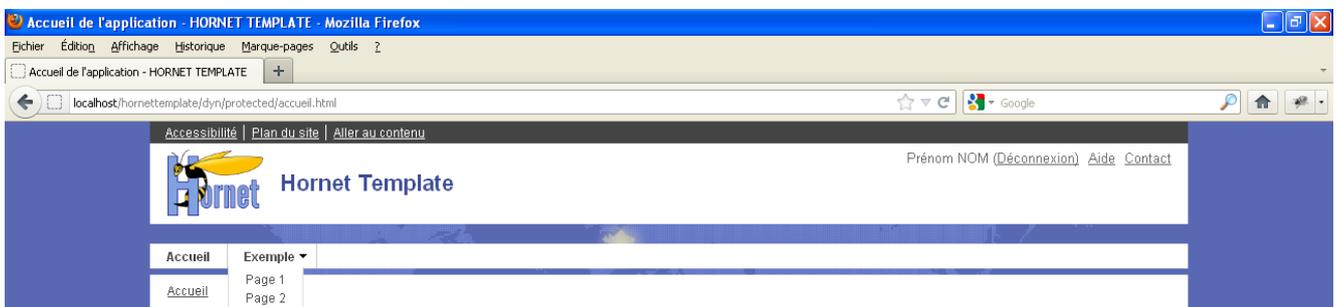


Figure 21 : Projet avec des pages d'exemples / **Détail du Menu**

2.3.5 Projet avec la configuration « Tests de Vie »

2.3.5.1 Initialisation du projet « Tests de vie »

1. Configurer le fichier **build.properties**, en particulier la propriété **build.projectName** qui définit le contexte de l'application
2. Lancer la tâche « **setProjectHornetWithTestVie** » (cf. 1.2.2.2 Tâches Ant) du « buildTmp.xml ».

Un prompt demande la saisie du nom du package racine, par défaut : « **hornettemplate** ».

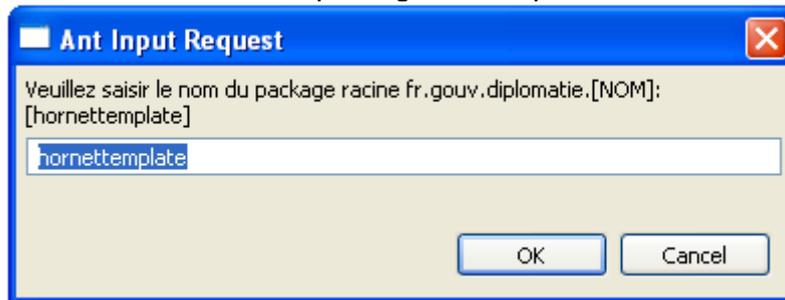


Figure 22 : Saisie du nom du package racine (par défaut : « **hornettemplate** »)

La console affiche le résultat suivant :

```
Buildfile: D:\devjava\workspace_mae\hornettemplate\buildTmp.xml
setPackageName:
setProjectHornetWithTestVie:
  [echo] cr?ation du projet hornet avec testVie : hornettemplate
setProjectHornet:
.settings:
org.eclipse.wst.common.component:
org.eclipse.wst.common.project.facet.core:
org.eclipse.jst.common.project.facet.core:
org.eclipse.jdt.core.prefs:
modifyNatureProject:
  [echo] modification de la nature du projet
  [delete] Deleting: D:\devjava\workspace_mae\hornettemplate\.project
  [xslt] Processing D:\devjava\workspace_mae\hornettemplate\dev\hornet\eclipse\core\.project to
D:\devjava\workspace_mae\hornettemplate\.project
  [xslt] Loading stylesheet D:\devjava\workspace_mae\hornettemplate\dev\style\project.xml
importHornetTemplate:
  [echo] import du WebContent (Hornet)
  [copy] Copying 23 files to D:\devjava\workspace_mae\hornettemplate\WebContent
  [echo] copie des images static
  [copy] Copying 1 file to D:\devjava\workspace_mae\hornettemplate\WebContent\static\images
importHornetSources:
  [echo] import des sources du projet (Hornet)
  [copy] Copying 7 files to D:\devjava\workspace_mae\hornettemplate\src\config
  [copy] Copying 5 files to
D:\devjava\workspace_mae\hornettemplate\src\config\fr\gouv\diplomatie\hornettemplate
  [copy] Copying 6 files to
D:\devjava\workspace_mae\hornettemplate\src\java\fr\gouv\diplomatie\hornettemplate
  [echo] import des fichiers de test du projet (Hornet)
  [copy] Copying 1 file to D:\devjava\workspace_mae\hornettemplate\tst\java
  [copy] Copying 1 file to D:\devjava\workspace_mae\hornettemplate\tst\config
  [echo] import des librairies de test du projet (Hornet)
  [copy] Copying 3 files to D:\devjava\workspace_mae\hornettemplate\lib
  [echo] import de la configuration ivy (Hornet)
  [copy] Copying 2 files to D:\devjava\workspace_mae\hornettemplate
importHornetLibs:
  [echo] recuperation des librairies externes depuis une archive ZIP locale

Voir en Annexe le détail de la tâche « importHornetLibs »
modifyClasspathProjectInHornet:
  [echo] modification du classpath du projet (Hornet)
  [copy] Copying 1 file to D:\devjava\workspace_mae\hornettemplate
modifyProjectHornetContextFile:
```

```
[echo] modification du fichier de contexte du projet (context.xml)
copyLivraisonHornet:
[echo] copie de la structure de livraison
[copy] Copying 2 files to D:\devjava\workspace mae\hornettemplate\livrable
[copy] Copying 4 files to D:\devjava\workspace mae\hornettemplate\livraison
generationTestVie:
generationTestVieExemple:
[echo] Ajout des pages testvie ? la webapp
[copy] Copying 1 file to
D:\devjava\workspace mae\hornettemplate\src\config\fr\gouv\diplomatie\hornettemplate
[copy] Copying 8 files to D:\devjava\workspace mae\hornettemplate\WebContent
[echo] Copie des nouvelles configuration pour test de vie
[copy] Copying 5 files to D:\devjava\workspace_mae\hornettemplate\src\config
[copy] Copying 1 file to D:\devjava\workspace mae\hornettemplate\src\config
[copy] Copying 1 file to D:\devjava\workspace mae\hornettemplate\src\config
[copy] Copying 1 file to D:\devjava\workspace mae\hornettemplate\WebContent\WEB-INF
[copy] Copying 2 files to D:\devjava\workspace_mae\hornettemplate\src\config
BUILD SUCCESSFUL
Total time: 10 seconds
```

Figure 23 : Résultat de la console après la tâche « **setProjectHornetWithTestVie** »

3. Vérification de la présence des librairies, cf. §5.2 Configuration Ivy.

Le rafraichissement du projet nous donne la même vue que pour un projet basique.

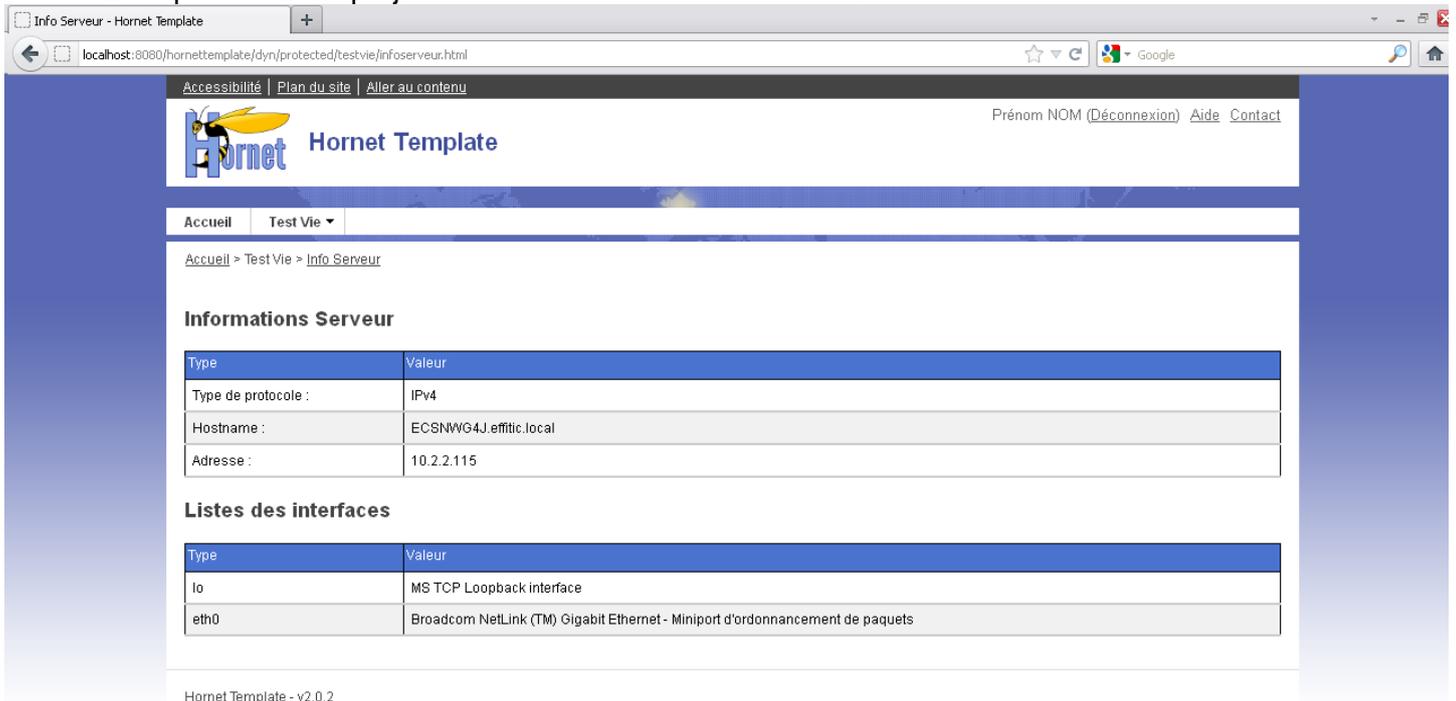
L'ajout des Tests de Vie dans le projet impacte les fichiers suivants :

- Le fichier de configuration Struts « **struts.xml** » pour inclure le fichier de configuration « **struts-hornet-testvie.xml** » propres aux Test de vie.
- Le fichier de configuration Tiles « **tiles.xml** » pour ajouter les nouvelles pages de Tests de Vie.
- Le fichier de configuration de l'application web « **web.xml** » pour faire référence aux fichiers de configurations Spring pour les Tests de Vie.
- Les fichiers de configurations Spring (« **spring-appContext-testvie-encode.xml** », « **spring-appContext-testvie.xml** ») propres aux Tests de vie et les fichiers de propriétés associés (« **testvie-service.properties** », « **testvie.properties** »).
- Le fichier « **package_fr.properties** » contenant les libellés des nouvelles pages de Tests de Vie.
- Le fichier « **menu.xml** » qui contient la définition des pages de l'application et qui embarque donc les nouvelles pages de Tests de Vie. Ce fichier XML est utilisé pour la génération du menu, du plan du site et du fil d'Ariane.
- Les pages JSP « **detail.jsp** », « **infoserveur.jsp** », « **system.jsp** » et « **testvie.jsp** » ainsi que les ressources statiques (« **tableau.css** » et « **tableau.js** » pour les Tests de Vie).
- Le fichier « **ivy.xml** » pour inclure les librairies nécessaires aux Tests de Vie.
- Les fichiers de construction du projet et du livrable applicatif (« **build.xml** », « **dynamique.xml** » et les fichiers de propriétés associés) qui contiennent des tâches Ant supplémentaires utiles aux Tests de Vie

4. Configurer les ressources statiques (cf. §5.3).

2.3.5.2 Vérification

Le déploiement du projet sous Tomcat nous donne le résultat suivant :



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/hornettemplate/dyn/protected/testvie/infoserveur.html'. The page title is 'Info Serveur - Hornet Template'. The browser's address bar also shows 'Google' and search icons. The page content includes a navigation menu with 'Accueil' and 'Test Vie' (selected). Below the navigation, there is a breadcrumb trail: 'Accueil > Test Vie > Info Serveur'. The main heading is 'Informations Serveur', followed by a table with server details:

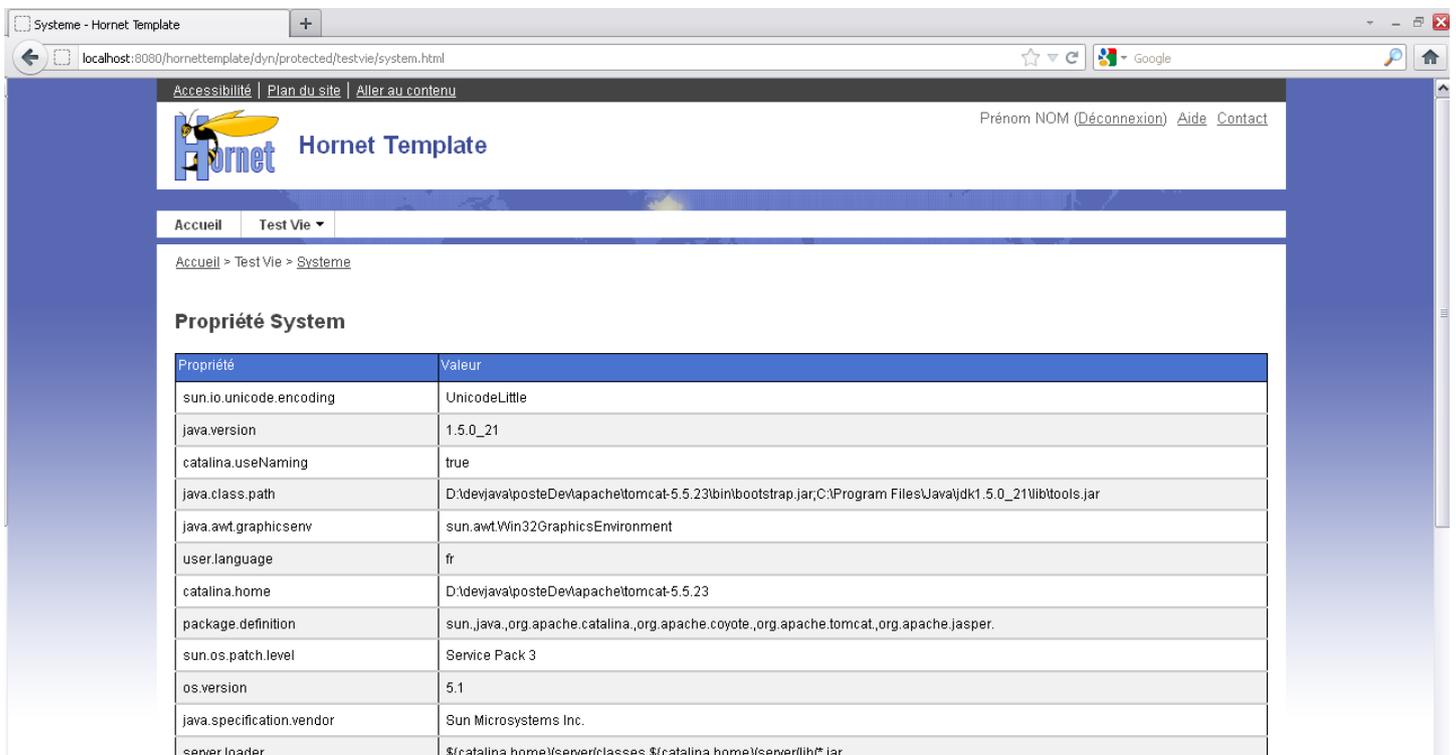
Type	Valeur
Type de protocole :	IPv4
Hostname :	ECSNMG4J.effitic.local
Adresse :	10.2.2.115

Below this table is the section 'Listes des interfaces', followed by another table:

Type	Valeur
lo	MS TCP Loopback interface
eth0	Broadcom NetLink (TM) Gigabit Ethernet - Miniport d'ordonnancement de paquets

At the bottom of the page, it says 'Hornet Template - v2.0.2'.

Figure 24 : Projet avec la configuration Test de Vie / Page « **Info Serveur** » (Test de Vie)



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/hornettemplate/dyn/protected/testvie/system.html'. The page title is 'Systeme - Hornet Template'. The browser's address bar also shows 'Google' and search icons. The page content includes a navigation menu with 'Accueil' and 'Test Vie' (selected). Below the navigation, there is a breadcrumb trail: 'Accueil > Test Vie > Systeme'. The main heading is 'Propriété System', followed by a table with system properties:

Propriété	Valeur
sun.io.unicode.encoding	UnicodeLittle
java.version	1.5.0_21
catalina.useNaming	true
java.class.path	D:\dev\java\posteDev\apache\tomcat-5.5.23\bin\bootstrap.jar;C:\Program Files\Java\jdk1.5.0_21\lib\tools.jar
java.awt.graphicsenv	sun.awt.Win32GraphicsEnvironment
user.language	fr
catalina.home	D:\dev\java\posteDev\apache\tomcat-5.5.23
package.definition	sun.java.,org.apache.catalina.,org.apache.coyote.,org.apache.tomcat.,org.apache.jasper.
sun.os.patch.level	Service Pack 3
os.version	5.1
java.specification.vendor	Sun Microsystems Inc.
server.loader	\${catalina.home}/server/classes,\${catalina.home}/server/lib/* jar

Figure 25 : Projet avec la configuration Test de Vie / Page « **Système** » (Test de Vie)

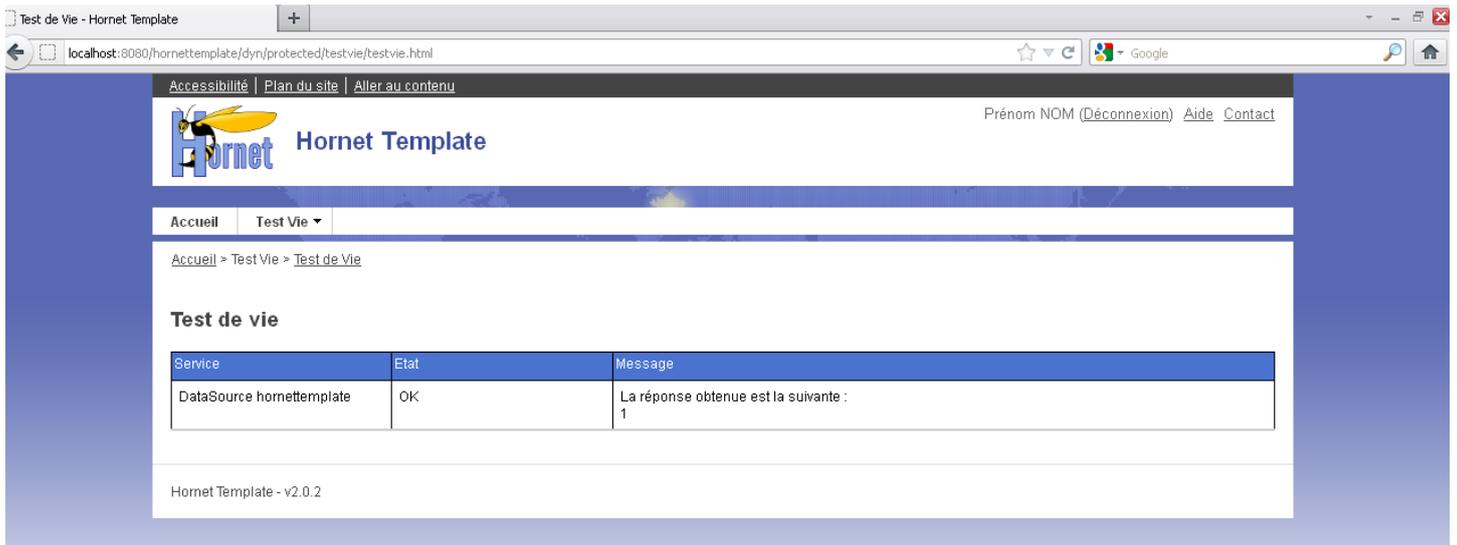


Figure 26 : Projet avec la configuration Test de Vie / Page « **Test de Vie** » (Test de Vie)

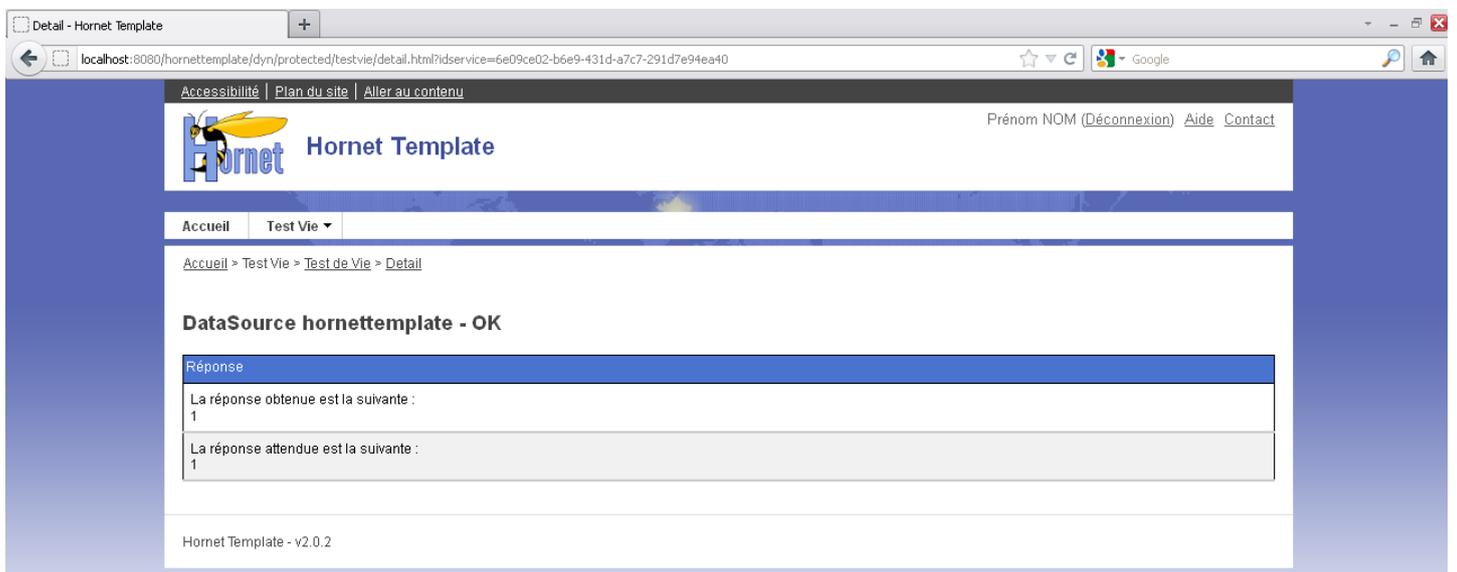


Figure 27 : Projet avec la configuration Test de Vie / Page « **Détail** » (Test de Vie)

2.4 Finalisation du template

Pour finaliser le projet, il est nécessaire de supprimer tout ce qui est lié au template.

Lancer la tâche « **cleanAllTemplate** » (cf. 1.2.2.2 Tâches Ant).

La console affiche le résultat suivant :

```
Buildfile: D:\devjava\workspace mae\hornettemplate\buildTmp.xml
cleanAllTemplate:
  [echo] suppression des elements du template
  [delete] Deleting directory D:\devjava\workspace_mae\hornettemplate\dev
  [delete] Deleting: D:\devjava\workspace mae\hornettemplate\buildTmp.xml
BUILD SUCCESSFUL
Total time: 33 seconds
```

Figure 28 : Résultat de la console après la tâche « **cleanAllTemplate** »

Un rafraichissement du projet donne la vue suivante :

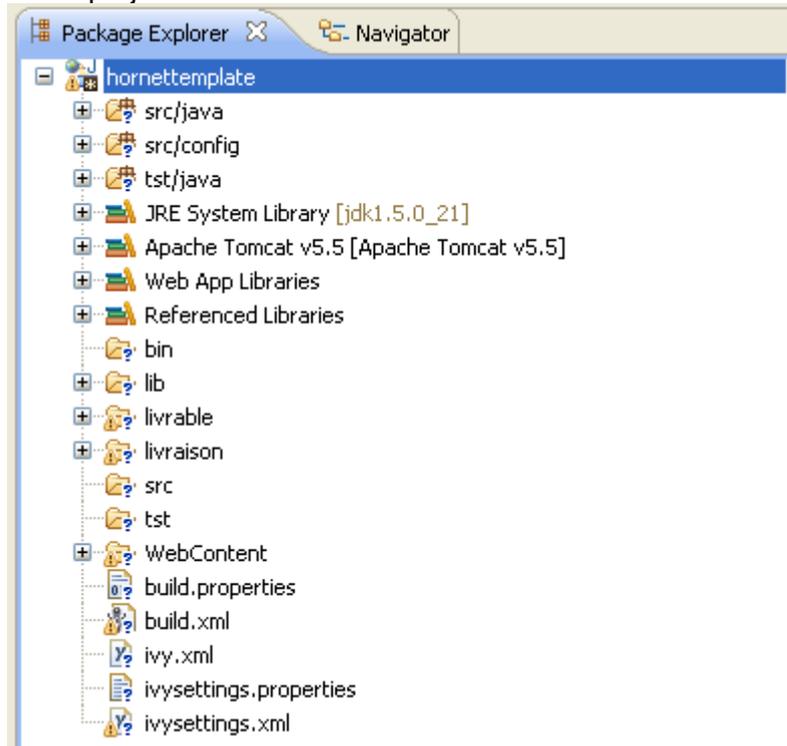


Figure 29 : Vue du projet après exécution de la tâche « **cleanAllTemplate** »

Les changements sont :

- Suppression du répertoire « **dev** ».
- Suppression du fichier « **buildTmp.xml** »

3 Ajout de nouveau service

3.1 Intégration des tests de vie

L'intégration des « **tests de vie** » à une application existante passe par quelques ajouts et configurations obligatoires :

- Récupération et intégration des artefacts :
 - Les pages JSP ;
 - Les ressources statiques ;
 - Les librairies pour les tests de vie ;
 - Les fichiers de configuration ;
 - Les fichiers de construction de l'application et du livrable applicatif.
- Configuration de l'application pour les « **tests de vie** » :
 - web.xml : déclaration des propriétés Spring et des rôles ;
 - Properties : propriétés utilisées par l'application ;
 - Struts : navigation et gestion des rôles dans les pages web ;
 - Spring : déclaration des services à tester et des rôles ;
 - Tiles : déclaration des pages web ;
 - menu.xml : arborescence des pages.

3.1.1 Ajout des artefacts

3.1.1.1 Récupération des éléments

Initialiser un nouveau projet Hornet avec la configuration « **Test de vie** ».

3.1.1.2 Copie dans l'application existante

Les éléments suivants sont nécessaires pour pouvoir utiliser les tests de vie dans une application.

- Les pages JSP et ressources statiques :
Recopier le contenu des dossiers suivants en gardant la même arborescence:
 - **WebContent/WEB-INF/static/testvie**
 - **WebContent/WEB-INF/tiles-jsp/testvie**
- Les librairies :
Recopier les dépendances nécessaires pour les tests de vie dans le fichier **ivy.xml**

```
<dependency org="fr.gouv.diplomatie.testvie" name="testvie"  
    rev="1.0" conf="compile-&gt;default" transitive="true">  
    <artifact name="testvie" type="jar" ext="jar" />  
</dependency>  
<dependency org="fr.gouv.diplomatie.testvie" name="testvie-encode"  
    rev="1.1" conf="compile-&gt;default" transitive="true">  
    <artifact name="testvie-encode" type="jar" ext="jar" />  
</dependency>
```

- Les fichiers de configuration Spring par défaut :
Recopier les fichiers suivant dans « src/config » :
 - **spring-appContext-testvie.xml**
 - **spring-appContext-testvie -encode.xml**
 - **testvie.properties** (préfixé éventuellement par un nom de package)
 - **testvie-service.properties**
- Le fichier de configuration Struts spécifique aux tests de vie :

Recopier les fichiers suivant dans « src/config » :

- **struts-hornet-testvie.xml**

- Les fichiers de construction contenant des taches supplémentaires pour les tests de vie :

Recopier les fichiers suivant :

- **build.xml**
- **livrable/CONFIG-WEBAPP/dynamique.xml**

3.1.2 Configuration de l'application

3.1.2.1 Configuration de la webapp

Ajouter au contexte de l'application les fichiers de configuration Spring à utiliser pour les tests de vie.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<web-app id="WebApp_ID" version="2.4"
  xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">

  <!-- Indique au ContextLoaderListener où sont les fichiers de configuration Spring -->
  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>
      classpath*:spring-appContext.xml
      classpath*:spring-appContext-dao.xml
      classpath*:spring-appContext-testvie.xml
    </param-value>
  </context-param>
```

3.1.2.2 Inclusion des pages web

L'application doit inclure dans ses fichiers de configuration un certain nombre d'éléments afin qu'elle puisse accéder aux pages de « **test de vie** ».

- Modifier le fichier de configuration Struts :
Inclure le fichier de configuration Struts spécifique aux tests de vie.

```
<struts>
  <include file="struts-hornet-testvie.xml" />
  ...
</struts>
```

- Modifier le fichier de configuration Tiles :
Ajouter les définitions des pages suivantes héritant du modèle de base.

```
<!-- Test de vie -->
<definition name="infoserveur" extends="baseLayout">
  <put-attribute name="filArianeKey" value="menu.testvie.infoserveur" />
  <put-attribute name="content" value="/WEB-INF/tiles-jsp/testvie/infoserveur.jsp" />
  <put-list-attribute name="appCssItems">
    <add-attribute value="/static/css/global.css" />
    <add-attribute value="/static/testvie/css/tableau.css" />
  </put-list-attribute>
</definition>
<definition name="system" extends="baseLayout">
  <put-attribute name="filArianeKey" value="menu.testvie.system" />
  <put-attribute name="content" value="/WEB-INF/tiles-jsp/testvie/system.jsp" />
  <put-list-attribute name="appCssItems">
    <add-attribute value="/static/css/global.css" />
    <add-attribute value="/static/testvie/css/tableau.css" />
  </put-list-attribute>
</definition>
<definition name="testvie" extends="baseLayout">
  <put-attribute name="filArianeKey" value="menu.testvie.testvie" />
  <put-attribute name="content" value="/WEB-INF/tiles-jsp/testvie/testvie.jsp" />
```

```
<put-list-attribute name="appCssItems">
  <add-attribute value="/static/css/global.css" />
  <add-attribute value="/static/testvie/css/tableau.css" />
</put-list-attribute>
</definition>
<definition name="detail" extends="baseLayout">
  <put-attribute name="filArianeKey" value="menu.testvie.detail" />
  <put-attribute name="content" value="/WEB-INF/tiles-jsp/testvie/detail.jsp" />
  <put-list-attribute name="appCssItems">
    <add-attribute value="/static/css/global.css" />
    <add-attribute value="/static/testvie/css/tableau.css" />
  </put-list-attribute>
</definition>
```

- Modifier le fichier contenant les libellés des pages de l'application:
Ajouter les libellés des pages dans le fichier package_fr.properties.

```
# Test Vie
menu.testvie.libelle=Test Vie
menu.testvie.libelleLong=Test Vie
menu.testvie.infoserveur.libelle=Info Serveur
menu.testvie.infoserveur.libelleLong=Test Vie - Info Serveur
menu.testvie.system.libelle=Systeme
menu.testvie.system.libelleLong=Test de Vie - Systeme
menu.testvie.testvie.libelle=Test de Vie
menu.testvie.testvie.libelleLong=Test de Vie - Test de Vie
menu.testvie.detail.libelle=Detail
menu.testvie.detail.libelleLong=Test de Vie - Detail
```

- Modifier le fichier de définition du menu :
Ajouter dans le fichier menu.xml l'arborescence pour les pages de test de vie.

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<pagelet>
  <root id="root" href="/dyn/protected/accueil.html"></root>
  <menu>
    <menu-item
      id="menu.testvie">
      <menu-item
        id="menu.testvie.infoserveur"
        href="/dyn/protected/testvie/infoserveur.html">
      </menu-item>
      <menu-item
        id="menu.testvie.system"
        href="/dyn/protected/testvie/system.html">
      </menu-item>
      <menu-item
        id="menu.testvie.testvie"
        href="/dyn/protected/testvie/testvie.html">
        <menu-item
          id="menu.testvie.detail"
          href="/dyn/protected/testvie/detail.html"
          visibleDansMenu="false" visibleDansPlan="false">
        </menu-item>
      </menu-item>
    </menu-item>
  </menu>
</pagelet>
```

Remarque : les attributs `visibleDansMenu` et `visibleDansPlan` sont optionnels et leur valeur par défaut est `true`. Pour ne pas afficher les pages dans le menu (ou le plan du site), valoriser ces attributs à « `false` ».

4 Problèmes connus

4.1 1^{ère} exécution des tâches « *setProjectHornet* »

Lors de la 1^{ère} exécution des tâches « **setProjectHornetXXX** », une erreur apparaît dans la console :

```
[taskdef] Could not load definitions from resource org/apache/ivy/ant/antlib.xml. It could not be found.
```

Cela provient du fait que la tâche de récupération des librairies utilise la librairie Ivy, définie en début de fichier XML mais qui n'est récupérée qu'au cours de l'exécution de la tâche.

Cet avertissement n'est pas bloquant et ne réapparaît pas lors d'autres exécutions de tâches « **setProjetHornetXXX** ».

4.2 Les modifications des pages ne sont pas prises en compte

Si après exécution de tâches les modifications des pages ne sont pas prises en compte, cela peut venir d'un problème de rafraîchissement des fichiers sous Eclipse.

Essayez de relancer le serveur Tomcat après avoir rafraîchi le projet (touche « F5 » sur le projet ou action « **Refresh** ») et effectué un « **Clean** » puis « **Publish** » sur le serveur.

Vider le cache navigateur et supprimer les cookies de session pour forcer le téléchargement du thème courant.

5 Annexes

5.1 Contenu initial de hornettemplate

Le projet se compose initialement des éléments suivants :

- Un répertoire « **dev** » : contient les éléments qui permettent la génération du template.
- Le répertoire est composé de deux sous-répertoires :
- « **dev/hornet** » : contient les sources du template permettant la génération du projet
- « **dev/style** » : contient l'ensemble des fichiers XSL qui seront utilisés pour générer des fichiers spécifiques
- Le fichier « **.project** » : fichier de configuration du projet sous Eclipse
- Le fichier XML « **buildTemplate.xml** » : contient les tâches qui permettent l'initialisation du template.

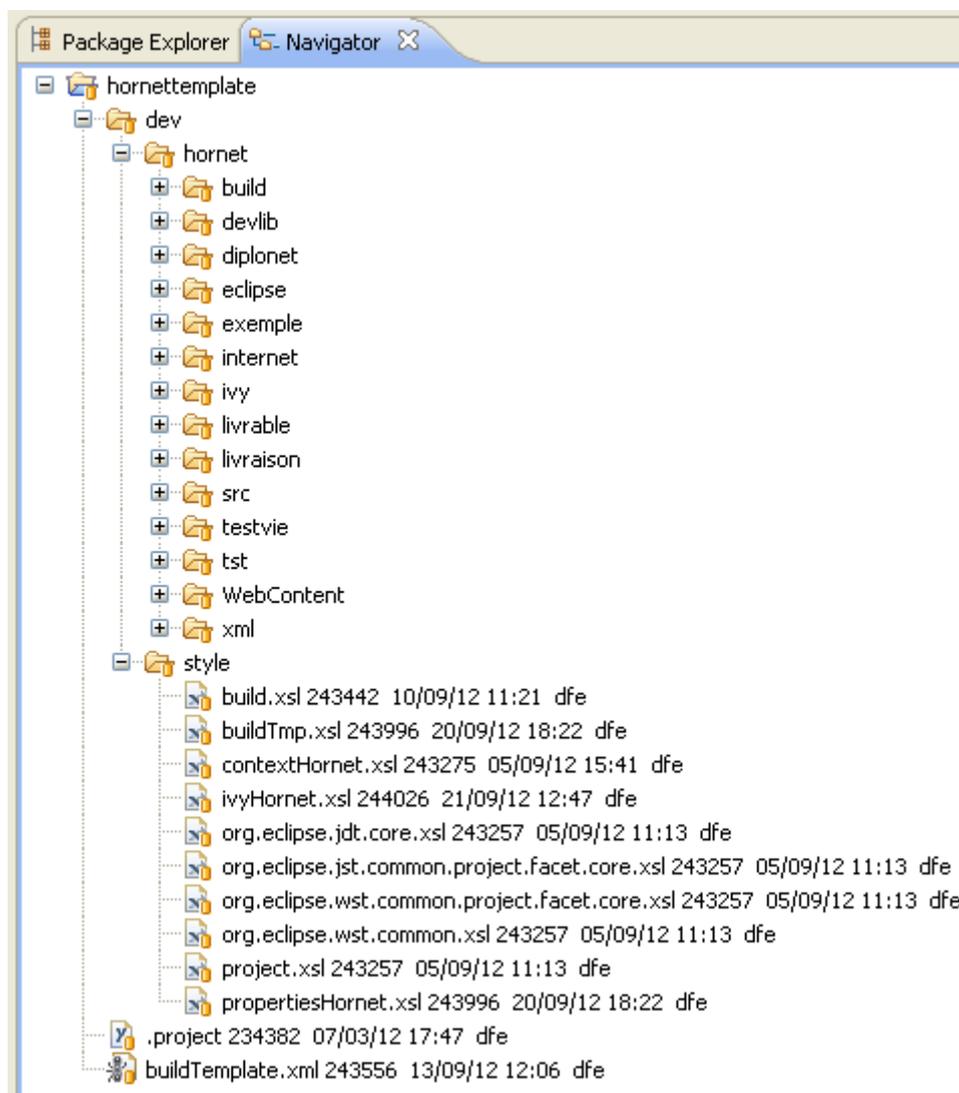


Figure 30 : Contenu initial du projet

5.2 Configuration Ivy

La procédure suivante s'applique si, suite à l'exécution d'une tâche « **setProjectHornetXXX** », les librairies ne sont pas présentes dans « **WebContent\WEB-INF\lib** ».

Par défaut, la configuration Ivy dans le template va interroger un repository local situé dans « **D:\devjava\workspace\Repository** ».

Si le repository n'est pas présent dans ce répertoire à l'exécution des tâches d'initialisation du template, les librairies ne seront pas rapatriées.

Ce qui peut être le cas, si le workspace des projets n'est pas celui spécifié par défaut (« **D:\devjava\workspace** »).

1. Configurez le fichier « **ivysettings.properties** ». Modifiez les trois lignes suivantes pour indiquer le repository :

```
repository.technique.url=<REPERTOIRE REPOSITORY>/technique
...
repository.cache.url=<REPERTOIRE REPOSITORY>/cache
...
repository.snapshot.url=<REPERTOIRE REPOSITORY>/snapshot
```

2. Lancez la tâche « **importHornetLibsXXX** » du buildTmp.xml.
 - a. Suivant le type de projet, il faut lancer la tâche associée
 - b. Ex : **setProjectHornetWithTestVie -> importHornetLibsTestVie**

5.3 Configuration des ressources statiques

- Ouvrir le fichier « **web.xml** » dans WebContent/WEB-INF et modifier les clés suivantes :
 - **fwkRoot** : chemin d'accès aux ressources statiques de hornet
 - ex : **http://130.177.222.247/hornet/hornetclient/2.0.2/fwk**
 - **yui3Root** : chemin d'accès aux ressources statiques de YahooUI
 - ex : **http://130.177.222.247/hornet/yui/yui/3.4.1**

5.4 Détail de la console lors de la tâche « **importHornetLibsDefault** »

```
importHornetLibs:
[echo] recuperation des librairies externes depuis une archive ZIP locale
[mkdir] Created dir: D:\devjava\workspace_mae\hornettemplate\WebContent\WEB-INF\lib
[xslt] Processing D:\devjava\workspace_mae\hornettemplate\dev\hornet\xml\blank.xml to
D:\devjava\workspace_mae\hornettemplate\ivy.xml
[xslt] Loading stylesheet D:\devjava\workspace_mae\hornettemplate\dev\style\ivyHornet.xsl
[echo] -----
[echo] Retrieve from repository ivy : compile@{qualifier}
[echo] -----
[ivy:resolve] :: Apache Ivy 2.2.x-local-20120102010511 - 20120102010511 :: http://ant.apache.org/ivy/ ::
[ivy:resolve] :: loading settings :: file = D:\devjava\workspace_mae\hornettemplate\ivysettings.xml
[ivy:resolve] :: resolving dependencies :: fr.gouv.diplomatie.hornettemplate#hornettemplate;1.0
[ivy:resolve] confs: [default, master, compile, provided, runtime, test, system, sources, javadoc, report,
optional]
[ivy:resolve] found fr.gouv.diplomatie.hornet#hornetserver-all;2.0.0 in repository-technique
[ivy:resolve] found fr.gouv.diplomatie.hornet#hornetserver-core;2.0.0 in repository-technique
[ivy:resolve] found javax.activation#activation;1.0.2 in repository-technique
[ivy:resolve] found org.apache.avalon#avalon-framework;4.2.0 in repository-technique
[ivy:resolve] found org.apache.xmlgraphics#batik-all;1.6 in repository-technique
[ivy:resolve] found net.sourceforge.cglib#cglib-nodep;2.1_3 in repository-technique
[ivy:resolve] found org.apache.commons#commons-beanutils;1.8.0 in repository-technique
[ivy:resolve] found org.apache.commons#commons-codec;1.3 in repository-technique
[ivy:resolve] found org.apache.commons#commons-collections;3.1 in repository-technique
[ivy:resolve] found org.apache.commons#commons-dbcp;1.3 in repository-technique
[ivy:resolve] found org.apache.commons#commons-digester;1.5 in repository-technique
[ivy:resolve] found org.apache.commons#commons-fileupload;1.1.1 in repository-technique
[ivy:resolve] found org.apache.commons#commons-httpclient;3.1 in repository-technique
[ivy:resolve] found org.apache.commons#commons-io;1.3.1 in repository-technique
[ivy:resolve] found org.apache.commons#commons-jxpath;1.1 in repository-technique
[ivy:resolve] found org.apache.commons#commons-lang;2.4 in repository-technique
[ivy:resolve] found org.apache.commons#commons-logging;1.0.4 in repository-technique
```

```
[ivy:resolve] found org.apache.commons#commons-pool;1.5.4 in repository-technique
[ivy:resolve] found org.apache.commons#commons-validator;1.3.0 in repository-technique
[ivy:resolve] found org.dom4j#dom4j;1.3 in repository-technique
[ivy:resolve] found org.apache.xmlgraphics#fop;0.95 in repository-technique
[ivy:resolve] found org.hsqldb#hsqldb;1.8.0 in repository-technique
[ivy:resolve] found com.lowagie#itext;2.1.7 in repository-technique
[ivy:resolve] found com.sun.media#jai_codec;1.1.2_01 in repository-technique
[ivy:resolve] found javax.media#jai core;1.1.2_01 in repository-technique
[ivy:resolve] found net.sourceforge.jasperreports#jasperreports;3.7.4 in repository-technique
[ivy:resolve] found com.octo.captcha#jcaptcha-all;1.0 in repository-technique
[ivy:resolve] found org.jdom#jdom;1.0 in repository-technique
[ivy:resolve] found javax.servlet#jstl;1.1.2 in repository-technique
[ivy:resolve] found javax.transaction#jta;1.0.1B in repository-technique
[ivy:resolve] found net.sourceforge.jtds#jtds;1.2.5 in repository-technique
[ivy:resolve] found org.apache.log4j#log4j;1.2.15 in repository-technique
[ivy:resolve] found javax.mail#mail;1.3.2 in repository-technique
[ivy:resolve] found org.mybatis#mybatis;2.3.5 in repository-technique
[ivy:resolve] found com.mysql#mysql-connector-java;5.0.7 in repository-technique
[ivy:resolve] found com.opensymphony#ognl;2.6.11 in repository-technique
[ivy:resolve] found org.apache.poi#poi;3.5-FINAL in repository-technique
[ivy:resolve] found com.opensymphony#quartz-all;1.6.0 in repository-technique
[ivy:resolve] found org.springframework#spring;2.5.1 in repository-technique
[ivy:resolve] found org.springframework#spring-security-core;2.0.3 in repository-technique
[ivy:resolve] found org.apache.jakarta.taglibs#standard;1.1.2 in repository-technique
[ivy:resolve] found org.apache.struts#struts2-core;2.0.11.2 in repository-technique
[ivy:resolve] found org.apache.struts#struts2-spring-plugin;2.0.11.2 in repository-technique
[ivy:resolve] found org.apache.struts#struts2-tiles-plugin;2.0.11.2 in repository-technique
[ivy:resolve] found org.apache.tiles#tiles-api;2.0.4 in repository-technique
[ivy:resolve] found org.apache.tiles#tiles-core;2.0.4 in repository-technique
[ivy:resolve] found org.apache.tiles#tiles-jsp;2.0.4 in repository-technique
[ivy:resolve] found org.apache.velocity#velocity;1.6.3 in repository-technique
[ivy:resolve] found org.apache.xmlgraphics#xmlgraphics-commons;1.2 in repository-technique
[ivy:resolve] found com.thoughtworks.xstream#xstream;1.2.2 in repository-technique
[ivy:resolve] found com.opensymphony#xwork;2.0.5 in repository-technique
[ivy:resolve] found net.sourceforge.freemarker#freemarker;2.3.8 in repository-technique
[ivy:resolve] found org.apache.tomcat#catalina;5.5.23 in repository-technique
[ivy:resolve] found org.apache.tomcat#tomcat-util;5.5.23 in repository-technique
[ivy:resolve] found org.apache.tomcat#naming-factory;5.5.23 in repository-technique
[ivy:resolve] found javax.servlet#servlet-api;2.4 in repository-technique
[ivy:resolve] found javax.servlet#jsp-api;2.0 in repository-technique
[ivy:resolve] found org.apache.xalan#xalan;2.7.0 in repository-technique
[ivy:resolve] found org.junit#junit;4.4 in repository-technique
[ivy:resolve] found org.springframework#spring-test;2.5 in repository-technique
[ivy:resolve] :: resolution report :: resolve 4297ms :: artifacts dl 125ms
-----
|               |             modules             |             artifacts             |
|               | number| search|dwnlded|evicted|| number|dwnlded|
|-----|-----|-----|-----|-----|-----|-----|
| default       | 58   | 0    | 0    | 0    || 57   | 0    |
| master        | 0    | 0    | 0    | 0    || 0    | 0    |
| compile       | 53   | 0    | 0    | 0    || 52   | 0    |
| provided      | 0    | 0    | 0    | 0    || 0    | 0    |
| runtime       | 58   | 0    | 0    | 0    || 57   | 0    |
| test          | 60   | 0    | 0    | 0    || 59   | 0    |
| system        | 0    | 0    | 0    | 0    || 0    | 0    |
| sources       | 0    | 0    | 0    | 0    || 0    | 0    |
| javadoc       | 0    | 0    | 0    | 0    || 0    | 0    |
| report        | 0    | 0    | 0    | 0    || 0    | 0    |
| optional      | 0    | 0    | 0    | 0    || 0    | 0    |
|-----|-----|-----|-----|-----|-----|
[ivy:retrieve] :: retrieving :: fr.gouv.diplomatie.hornettemplate#hornettemplate
[ivy:retrieve]   confs: [compile]
[ivy:retrieve]   52 artifacts copied, 0 already retrieved (30290kB/8422ms)
[echo] -----
[echo] Retrieve from repository ivy : test@{qualifier}
[echo] -----
[ivy:resolve] :: loading settings :: file = D:\devjava\workspace_mae\hornettemplate\ivysettings.xml
[ivy:resolve] :: resolving dependencies :: fr.gouv.diplomatie.hornettemplate#hornettemplate;1.0
[ivy:resolve]   confs: [default, master, compile, provided, runtime, test, system, sources, javadoc, report, optional]
[ivy:resolve] found fr.gouv.diplomatie.hornet#hornetserver-all;2.0.0 in repository-technique
[ivy:resolve] found fr.gouv.diplomatie.hornet#hornetserver-core;2.0.0 in repository-technique
[ivy:resolve] found javax.activation#activation;1.0.2 in repository-technique
[ivy:resolve] found org.apache.avalon#avalon-framework;4.2.0 in repository-technique
[ivy:resolve] found org.apache.xmlgraphics#batik-all;1.6 in repository-technique
[ivy:resolve] found net.sourceforge.cglib#cglib-nodep;2.1.3 in repository-technique
[ivy:resolve] found org.apache.commons#commons-beanutils;1.8.0 in repository-technique
[ivy:resolve] found org.apache.commons#commons-codec;1.3 in repository-technique
[ivy:resolve] found org.apache.commons#commons-collections;3.1 in repository-technique
[ivy:resolve] found org.apache.commons#commons-dbc;1.3 in repository-technique
```

```
[ivy:resolve] found org.apache.commons#commons-digester;1.5 in repository-technique
[ivy:resolve] found org.apache.commons#commons-fileupload;1.1.1 in repository-technique
[ivy:resolve] found org.apache.commons#commons-httpclient;3.1 in repository-technique
[ivy:resolve] found org.apache.commons#commons-io;1.3.1 in repository-technique
[ivy:resolve] found org.apache.commons#commons-jxpath;1.1 in repository-technique
[ivy:resolve] found org.apache.commons#commons-lang;2.4 in repository-technique
[ivy:resolve] found org.apache.commons#commons-logging;1.0.4 in repository-technique
[ivy:resolve] found org.apache.commons#commons-pool;1.5.4 in repository-technique
[ivy:resolve] found org.apache.commons#commons-validator;1.3.0 in repository-technique
[ivy:resolve] found org.dom4j#dom4j;1.3 in repository-technique
[ivy:resolve] found org.apache.xmlgraphics#fop;0.95 in repository-technique
[ivy:resolve] found org.hsqldb#hsqldb;1.8.0 in repository-technique
[ivy:resolve] found com.lowagie#itext;2.1.7 in repository-technique
[ivy:resolve] found com.sun.media#jai codec;1.1.2 01 in repository-technique
[ivy:resolve] found javax.media#jai core;1.1.2 01 in repository-technique
[ivy:resolve] found net.sourceforge.jasperreports#jasperreports;3.7.4 in repository-technique
[ivy:resolve] found com.octo.captcha#jcaptcha-all;1.0 in repository-technique
[ivy:resolve] found org.jdom#jdom;1.0 in repository-technique
[ivy:resolve] found javax.servlet#jstl;1.1.2 in repository-technique
[ivy:resolve] found javax.transaction#jta;1.0.1B in repository-technique
[ivy:resolve] found net.sourceforge.jtds#jtds;1.2.5 in repository-technique
[ivy:resolve] found org.apache.log4j#log4j;1.2.15 in repository-technique
[ivy:resolve] found javax.mail#mail;1.3.2 in repository-technique
[ivy:resolve] found org.mybatis#mybatis;2.3.5 in repository-technique
[ivy:resolve] found com.mysql#mysql-connector-java;5.0.7 in repository-technique
[ivy:resolve] found com.opensymphony#ognl;2.6.11 in repository-technique
[ivy:resolve] found org.apache.poi#poi;3.5-FINAL in repository-technique
[ivy:resolve] found com.opensymphony#quartz-all;1.6.0 in repository-technique
[ivy:resolve] found org.springframework#spring;2.5.1 in repository-technique
[ivy:resolve] found org.springframework#spring-security-core;2.0.3 in repository-technique
[ivy:resolve] found org.apache.jakarta.taglibs#standard;1.1.2 in repository-technique
[ivy:resolve] found org.apache.struts#struts2-core;2.0.11.2 in repository-technique
[ivy:resolve] found org.apache.struts#struts2-spring-plugin;2.0.11.2 in repository-technique
[ivy:resolve] found org.apache.struts#struts2-tiles-plugin;2.0.11.2 in repository-technique
[ivy:resolve] found org.apache.tiles#tiles-api;2.0.4 in repository-technique
[ivy:resolve] found org.apache.tiles#tiles-core;2.0.4 in repository-technique
[ivy:resolve] found org.apache.tiles#tiles-jsp;2.0.4 in repository-technique
[ivy:resolve] found org.apache.velocity#velocity;1.6.3 in repository-technique
[ivy:resolve] found org.apache.xmlgraphics#xmlgraphics-commons;1.2 in repository-technique
[ivy:resolve] found com.thoughtworks.xstream#xstream;1.2.2 in repository-technique
[ivy:resolve] found com.opensymphony#xwork;2.0.5 in repository-technique
[ivy:resolve] found net.sourceforge.freemarker#freemarker;2.3.8 in repository-technique
[ivy:resolve] found org.apache.tomcat#catalina;5.5.23 in repository-technique
[ivy:resolve] found org.apache.tomcat#tomcat-util;5.5.23 in repository-technique
[ivy:resolve] found org.apache.tomcat#naming-factory;5.5.23 in repository-technique
[ivy:resolve] found javax.servlet#servlet-api;2.4 in repository-technique
[ivy:resolve] found javax.servlet#jsp-api;2.0 in repository-technique
[ivy:resolve] found org.apache.xalan#xalan;2.7.0 in repository-technique
[ivy:resolve] found org.junit#junit;4.4 in repository-technique
[ivy:resolve] found org.springframework#spring-test;2.5 in repository-technique
[ivy:resolve] :: resolution report :: resolve 984ms :: artifacts dl 31ms
```

conf	modules					artifacts	
	number	search	downlded	evicted	number	downlded	
default	58	0	0	0	57	0	
master	0	0	0	0	0	0	
compile	53	0	0	0	52	0	
provided	0	0	0	0	0	0	
runtime	58	0	0	0	57	0	
test	60	0	0	0	59	0	
system	0	0	0	0	0	0	
sources	0	0	0	0	0	0	
javadoc	0	0	0	0	0	0	
report	0	0	0	0	0	0	
optional	0	0	0	0	0	0	

```
[ivy:retrieve] :: retrieving :: fr.gouv.diplomatie.hornettemplate#hornettemplate
[ivy:retrieve] confs: [test]
[ivy:retrieve] 57 artifacts copied, 2 already retrieved (31353kB/2938ms)
```