




Jenkins


Création d'un projet Maven [Version PDF]


- Installer les plugins suivants :


1. Maven integration
2. Parameterized Trigger
3. Delivery Pipeline Plugin


**Jenkins**


Jenkins >


 Nouveau Item


 Utilisateurs


 Historique des constructions


 Relations entre les builds


 Vérifier les empreintes numériques


 **Administrer Jenkins**

 Mes vues


 Open Blue Ocean


 Lockable Resources


 Identifiants


 New View


Administrer Jenkins


**Configurer le système**
Configurer les paramètres généraux et les chemins de fichiers.


**Configurer la sécurité globale**
Sécuriser Jenkins; définir qui est autorisé à accéder au système.

**Configure Credentials**
Configure the credential providers and types

**Configuration globale des outils**
Configurer les outils, leur localisation et les installeurs automatiques.

**Recharger la configuration à partir du disque**
Supprimer toutes les données en mémoire et recharger tout à partir du système de fichiers. Utile quand vous modifiez l

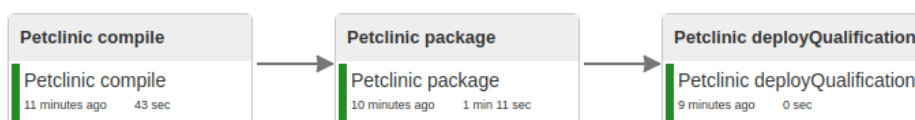
**Gestion des plugins**
Ajouter, supprimer, activer ou désactiver des plugins qui peuvent étendre les fonctionnalités de Jenkins.
🔴 mises à jour disponibles

**Informations sur le système**
Affiche diverses informations relatives au système pour aider à la résolution de problèmes.

File d'attente des constructions —
File d'attente des constructions vide

État du lanceur de compilations —
1 Au repos
2 Au repos

- Cible à atteindre :




Création de l'item "Petclinic_compile"


- Description : cet item se chargera de récupérer les sources du projet sur GitHub et lancera la compilation sous Maven.
- Nom : Petclinic_compile
- Type : Projet Maven

Saisissez un nom

Petclinic_compile

» Champ obligatoire

**Construire un projet free-style**
Ceci est la fonction principale de Jenkins qui sert à builder (construire) votre projet. Vous pouvez intégrer tous les outils de gestion de version avec pour tout autre chose qu'un build logiciel.

**Construire un projet maven**
Construit un projet avec maven. Jenkins utilise directement vos fichiers POM et diminue radicalement l'effort de configuration. Cette fonctionnalité e

- Actions :

1. Récupère les sources sur GitHub

Gestion de code source

☐ Aucune
 ☒ Git

Repositories

Repository URL
 Credentials

Branches to build

Branch Specifier (blank for 'any')

Navigateur de la base de code

Additional Behaviours

☐ Mercurial
 ☐ Subversion

2. Lance la tâche maven "clean compile"

Build

POM Racine

Goals et options

Création d'une vue de type "Delivery Pipeline View"

- Description : cette vue permettra d'afficher les enchaînements entre les items.
- Nom de la vue : Petclinic pipeline

Jenkins

Jenkins

Nom de la vue

☒ **Delivery Pipeline View**

Continuous Delivery pipelines, perfect for visualization on information radiators. Shows one or mc

☐ **Delivery Pipeline View for Jenkins Pipelines**

Continuous Delivery pipelines, perfect for visualization on information radiators. Shows one or mc

☐ **Ma vue**

Cette vue affiche automatiquement tous les jobs auxquels l'utilisateur a accès.

☐ **Vue liste**

Montre les jobs dans une simple liste. Vous pouvez choisir les jobs à afficher dans chaque vue.

- Dans la zone "**Pipelines**", ajouter un component avec :
- Name : Petclinic pipeline
- Initial Job : Petclinic_compile

Pipelines

Components

Component Name	<input type="text" value="Petclinic pipeline"/>
Initial Job	<input type="text" value="Petclinic compile"/>
Final Job (optional)	<input type="text"/>
Show upstream	<input checked="" type="checkbox"/>

- Afficher la nouvelle vue

Jenkins

[Jenkins](#) > [Petclinic pipeline](#)

[Nouveau Item](#)
[Utilisateurs](#)
[Historique des constructions](#)

Petclinic pipeline

Tous +

Petclinic pipeline

Création de l'item Petclinic_package

- Description : cet item doit construire le package WAR du projet. Il doit pouvoir utiliser le workspace de l'item précédent.
- Nom : Petclinic_package
- Type : Projet Maven

Jenkins

[Jenkins](#) > [Tous](#)

Saisissez un nom

» Champ obligatoire

Construire un projet free-style

Ceci est la fonction principale de Jenkins qui sert à builder (construire) votre projet. Vous pouvez utiliser tous les systèmes de build. Il est même possible d'utiliser Jenkins pour tout autre chose que...

Construire un projet maven

Construit un projet avec maven. Jenkins utilise directement vos fichiers POM et minimise les dépendances. Cette fonction est encore en bêta mais elle est disponible afin d'obtenir vos retours.

- Paramètre string : WORKSPACE_PARENT
- Actions :

1. Prends en paramètre le workspace du build parent (WORKSPACE_PARENT)

☒ Ce build a des paramètres

Paramètre String

Nom

Valeur par défaut

Description

[Plain text] [Prévisualisation](#)

☐ Trim the string

2. Lance la tâche maven "**package**"

Build

POM Racine	<input type="text" value="pom.xml"/>	?
Goals et options	<input type="text" value="package"/>	?
MAVEN_OPTS	<input type="text"/>	▼ ?

- Dans la section build, spécifier le répertoire de travail spécifique avec la valeur \$WORKSPACE_PARENT

☒ Utiliser un répertoire de travail spécifique ?

Répertoire	<input type="text" value="\$WORKSPACE_PARENT"/>
Niveau de validation du POM Maven	DEFAULT ▼
Settings file	Utiliser les réglages Maven par défaut ▼ ?
Global Settings file	Utiliser les réglages globaux Maven par défaut ▼ ?

- Modifier l'item "**Petclinic_compile**" pour ajouter l'appel à l'item "**Petclinic_package**" après le lancement de la tâche maven "**clean compile**". Initialiser la variable WORKSPACE_PARENT avec le workspace de l'item "**Petclinic_compile**" ("**Predefined parameters**" avec le paramètre : **WORKSPACE_PARENT=\$WORKSPACE**)

Astuce : Chaîner les items en utilisant une action "**Trigger parameterized build on other projects**"

Actions à la suite du build

Trigger parameterized build on other projects ?

Déclencheurs du build

Projects to build	<input type="text" value="Petclinic package"/>	?
Trigger when build is	Stable ▼	?
Trigger build without parameters	<input type="checkbox"/>	?
Trigger build from child projects	<input type="checkbox"/>	

Predefined parameters ?

Parameters	<input type="text" value="WORKSPACE_PARENT=\$WORKSPACE"/>	?
------------	---	---

Avancé...


Création de l'item "Petclinic_deployQualification"

- Description : cet item doit copier le WAR généré dans un répertoire sur le serveur
- Nom : Petclinic_deployQualification
- Type : Projet free-style

Saisissez un nom

Petclinic_deployQualification

» Champ obligatoire



Construire un projet free-style
Ceci est la fonction principale de Jenkins qui sert à builder (construire) votre projet. Vous pouvez intégrer tous les systèmes de build. Il est même possible d'utiliser Jenkins pour tout autre chose qu'un build logiciel.

- Paramètre string : WORKSPACE_PARENT
- Actions :

1. Prends en paramètre le workspace du build parent (WORKSPACE_PARENT)

☒ Ce build a des paramètres

Paramètre String

Nom

WORKSPACE_PARENT

Valeur par défaut

Description

[Plain text] [Prévisualisation](#)

☐ Trim the string

Ajouter un paramètre

2. Copie le fichier "\$WORKSPACE_PARENT/target/petclinic.war" du workspace dans le répertoire /var/tmp

Build

Exécuter un script shell

Commande

```
echo "deploy WAR";
cp "$WORKSPACE_PARENT"/target/*.war /var/tmp
```

Voir [la liste des variables d'environnement disponibles](#)

Avancé...

Ajouter une étape au build

- Modifier l'item "Petclinic_package" pour ajouter l'appel à l'item "Petclinic_deployQualification" après le lancement de la tâche maven "package". Initialiser la variable WORKSPACE_PARENT avec le workspace de l'item "Petclinic_package"

Actions à la suite du build

Trigger parameterized build on other projects

Déclencheurs du build

Projects to build

Petclinic_deployQualification

Trigger when build is

Stable

Trigger build without parameters

☐

Trigger build from child projects

☐

Predefined parameters

Parameters

WORKSPACE_PARENT=\$WORKSPACE

Avancé...

Lancement du build

- Lancer le build manuellement et vérifier que le WAR est bien déployé dans le répertoire **"/var/tmp"**
- Optimiser le démarrage du build en scrutant l'outil de gestion de version toutes les minutes

Astuce : Utiliser l'expression *** * * * *** pour exécuter la tâche toutes les minutes

Ce qui déclenche le build

☒ Lance un build à chaque fois qu'une dépendance SNAPSHOT est construite

☐ Schedule build when some upstream has no successful builds

☐ Déclencher les builds à distance (Par exemple, à partir de scripts)

☐ Construire après le build sur d'autres projets

☐ Construire périodiquement

☐ Build when another project is promoted

☐ GitHub hook trigger for GITScm polling

☒ Scrutation de l'outil de gestion de version

Planning

* * * * *

Voulez-vous vraiment dire "chaque minute" avec l'expression *** * * * ***? Peut-être voulez-vous dire **"H * * * * ***"?

Aurait été lancé à Thursday, December 12, 2019 8:38:58 PM GMT; prochaine exécution à Thursday, December 12, 2019 8:38:58 PM GMT.

☐ Ignore post-commit hooks

- Faire une modification sur le projet dans GitHub puis un commit
- Vérifier que le build se lance bien automatiquement