



Module 4

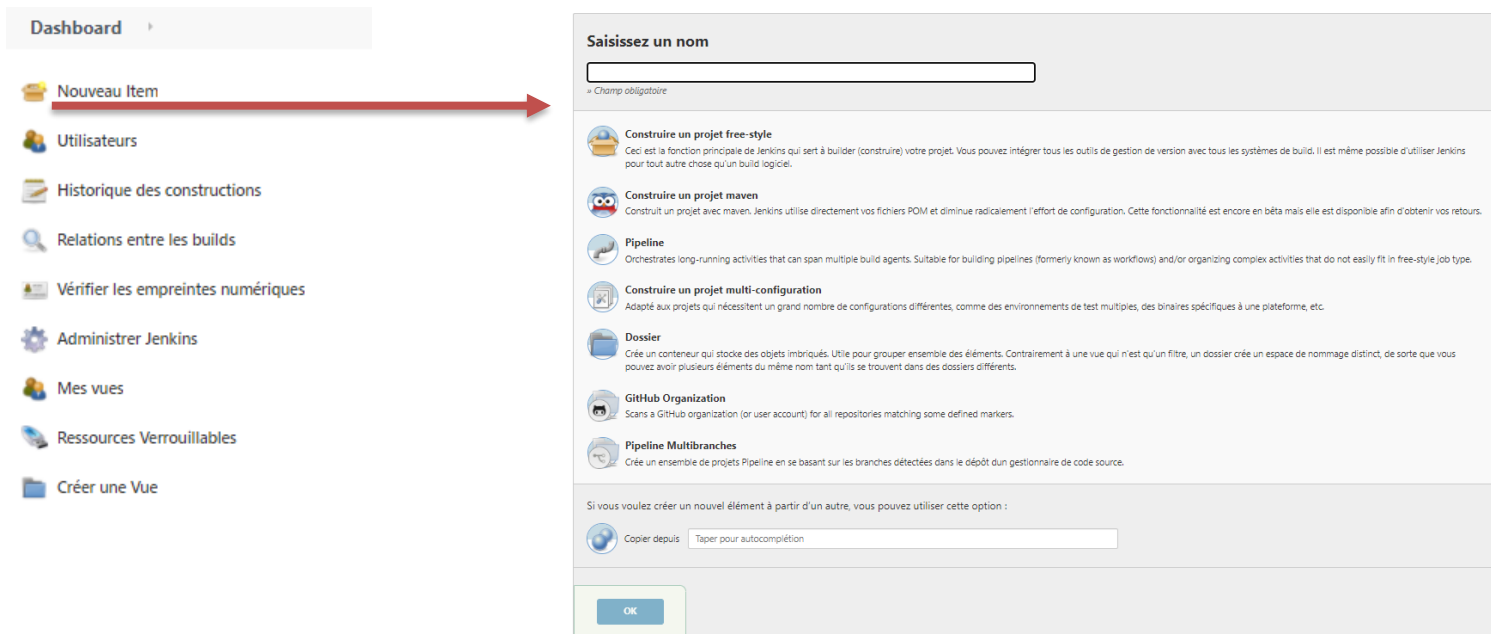
Les projets Jenkins: Les Jobs

Contenu du module

- Les différents types de jobs
- Création d'un job
 - Les déclencheurs
 - Les actions post-construction
- Visualisation des résultats produits par un job
- Automatisation des tests
 - Tests unitaires
 - Tests d'intégration et leurs contraintes
- Les alertes
 - Le système de notifications de Jenkins
 - Notifications par mail, ...

Ajouter un projet à Jenkins (job)

- Jenkins associe la notion de projet à un job
 - Par projet on peut comprendre livrable et non pas application (une application est constituée de plusieurs livrables)



The screenshot displays the Jenkins Dashboard on the left and the 'New Item' configuration page on the right. A red arrow points from the 'Nouveau Item' button in the dashboard to the 'Saisissez un nom' input field in the configuration page.

Dashboard

- Nouveau Item
- Utilisateurs
- Historique des constructions
- Relations entre les builds
- Vérifier les empreintes numériques
- Administrer Jenkins
- Mes vues
- Ressources Verrouillables
- Créer une Vue

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 intégrer tous les outils de gestion de version avec tous les systèmes de build. Il est même possible d'utiliser Jenkins 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 est encore en bêta mais elle est disponible afin d'obtenir vos retours.
- Pipeline**
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Construire un projet multi-configuration**
Adapté aux projets qui nécessitent un grand nombre de configurations différentes, comme des environnements de test multiples, des binaires spécifiques à une plateforme, etc.
- Dossier**
Crée un conteneur qui stocke des objets imbriqués. Utile pour grouper ensemble des éléments. Contrairement à une vue qui n'est qu'un filtre, un dossier crée un espace de nommage distinct, de sorte que vous pouvez avoir plusieurs éléments du même nom tant qu'ils se trouvent dans des dossiers différents.
- GitHub Organization**
Scans a GitHub organization (or user account) for all repositories matching some defined markers.
- Pipeline Multibranches**
Crée un ensemble de projets Pipeline en se basant sur les branches détectées dans le dépôt d'un gestionnaire de code source.

Si vous voulez créer un nouvel élément à partir d'un autre, vous pouvez utiliser cette option :

Copier depuis Taper pour autocomplétion

OK

Les principaux types de projets

- Free-style
 - Un projet qui sera construit en exécutant des lignes de commandes ou bien des scripts shell/PowerShell
 - Certaines options apportées par des plugins sont également disponibles
 - C'est le type de projet le plus générique
- Maven
 - Intégration totale de Maven
 - Il utilise un fichier pom.xml et les cibles Maven
- Pipeline
 - Un pipeline est un type de projet qui permet de découper la construction en plusieurs étapes (séquentielles et/ou parallèles), idéal dans une approche CI/CD

Configuration du projet : Gestion du code source

- Essentiel !
 - L'intégration continue récupère les projets depuis un SCM
- Plusieurs SCM sont utilisables avec Jenkins, nativement ou via des plugins

Données
d'authentification
auprès du SCM

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

Configuration du projet : Ce qui déclenche le build

- En plus du déclenchement manuel, Jenkins va provoquer des constructions sur certains évènements

Ce qui déclenche le build

- ☐ Déclencher les builds à distance (Par exemple, à partir de scripts)
- ☐ Construire après le build sur d'autres projets
- ☐ Construire périodiquement
- ☐ GitHub hook trigger for GITScm polling
- ☐ Scrutation de l'outil de gestion de version



- « Construire périodiquement » et « Scrutation de l'outil de gestion de version » sont les options les plus couramment utilisées !
 - Construire un projet la nuit
 - Construire lorsque des modifications sont détectées dans le SCM

Les expressions CRON

- Les options « Construire périodiquement » et « Scruter... » sont liées à une programmation périodique
- Dans le cas d'un lien vers CVS, cette planification permet de définir à quelle fréquence Jenkins compare son workspace avec le contenu du référentiel.
- Dans le cas de Subversion ou Git, cela définit la fréquence à laquelle Jenkins consulte le dernier numéro de commit.

The screenshot displays two configuration sections in Jenkins, both with the checkbox 'Construire périodiquement' (Build periodically) checked.

Section 1: Construire périodiquement

- Planning:** A text box containing the CRON expression `H 2 * * *`.
- Execution Schedule:** A message indicating the build would have started on Thursday, January 7, 2021, at 02:07:06 (Central European Time), with the next execution on Friday, January 8, 2021, at 02:07:06 (Central European Time).

Section 2: Scrutation de l'outil de gestion de version

- Planning:** A text box containing the CRON expression `H/30 * * * *`.
- Execution Schedule:** A message indicating the scan would have started on Thursday, January 7, 2021, at 18:07:31 (Central European Time), with the next execution on Thursday, January 7, 2021, at 18:37:31 (Central European Time).

Les expressions CRON

- Ces périodicités sont définies avec des expressions CRON (du nom de l'outil Unix qui les utilise)
- On peut mettre une définition par ligne
- Les lignes commençant par # sont des commentaires
- Chaque définition est :
 - Soit un mot clé (@yearly, @annually, @monthly, @weekly, @daily, @midnight, @hourly)
 - Soit une liste de 5 contraintes sur le moment d'exécution, séparées par un espace, chaque contrainte portant dans l'ordre sur : MINUTES HEURES JOURMOIS MOIS JOURSEMAINE
 - * : Pas de contrainte
 - m-n : Une liste bornée de valeurs
 - */s ou m-n/s : Permet de spécifier un « saut » de valeurs dans l'étendue, par exemple : */3 → toutes les 3 valeurs ou 1-12/3 → 1,3,6,9,12
 - x,y,z : Une liste de valeurs
 - Si on utilise la contrainte sur le jour de la semaine, on ne doit pas utiliser celle sur le jour dans le mois. Les jours de la semaine sont numérotés de 0 à 7, du dimanche au dimanche
- Exemples
 - Toutes les minutes : * * * * *
 - A la 5^{ème} minute de chaque heure : 5 * * * *
 - Tous les 2 jours (impairs) : 0 0 */2 * *
 - Toutes les semaines (dimanche à 0h) : 0 0 * * 0 ou @weekly

Surveillance du SCM

- On l'a vu dans la configuration du projet, certains événements provoquent le déclenchement de la construction
- Parmi celles-ci, il y a la scrutation du SCM
- Il faut être vigilant lors de l'utilisation de cette option :
 - Elle est très coûteuse sur CVS : il faut trouver les évolutions des révisions sur chaque fichier
 - Elle est peu coûteuse sur SVN et Git : la révision concerne une transaction; la révision courante est stockée au niveau du SCM lui-même

Configuration du projet : Le build – Free-style

- Un projet free-style est constitué d'étapes séquentielles
- Les étapes proposées dans la liste déroulante peuvent être étendues avec des plugins !

Build

Ajouter une étape au build ▼

Ajouter une étape au build ▲

Appeler Ant

Exécuter un script shell

Exécuter une ligne de commande batch Windows

Invoke Gradle script

Invoquer les cibles Maven de haut niveau

Run with timeout

Set build status to "pending" on GitHub commit

Configuration du projet : Le build - Maven

- La configuration d'un projet Maven nécessite d'indiquer l'emplacement du fichier `pom.xml`
 - Par rapport à la racine de ce qui est ramené du SCM
- On peut également préciser la cible et les options Maven

Pre Steps

Ajouter une étape pré-build ▾

Build

POM Racine ?

Goals et options ?

Avancé...

Post Steps

☐ Run only if build succeeds ☐ Run only if build succeeds or is unstable ☒ Run regardless of build result

Should the post-build steps run only for successful builds, etc.

Ajouter une étape post-build ▾

Configuration du projet : Le build - dotnet

- Le plugin **.NET SDK Support** ajoute le support de la commande dotnet à Jenkins
 - Il ajoute des actions dans les étapes du build d'un projet free-style
- Pour configurer un projet utilisant dotnet, il faut donc simplement créer un projet free-style

Environnements de Build

- ☐ Delete workspace before build starts
- ☐ Use secret text(s) or file(s)
- ☐ Abort the build if it's stuck
- ☒ Add timestamps to the Console Output
- ☐ Inspect build log for published Gradle build scans
- ☐ Préparer l'environnement pour SonarQube Scanner
- ☒ Utiliser .NET

SDK .NET .NET Core 3.1

Require Specific SDK Version ☐

☐ With Ant

Build


Ajouter une étape au build

- .NET: Exécuter des tests unitaires (test)
- .NET: Générer un projet (build)
- .NET: Nettoyer la sortie d'un projet (clean)
- .NET: Publier un package NuGet (nuget push)
- .NET: Publier un projet (publish)
- .NET: Restaurer les dépendances (restore)
- .NET: Restaurer les outils locales (tool restore)
- .NET: Supprimer/Délistier un package NuGet (nuget delete)
- Appeler Ant
- Construire un projet Visual Studio avec MSBuild
- Exécuter un script shell
- Exécuter une ligne de commande batch Windows

Configuration du projet : Notification

- Dans les actions exécutables à la suite du build, il est notamment possible d'envoyer des emails pour notifier des échecs de build

Actions à la suite du build

 **Notifier par email**

X

?

Destinataires

Liste des destinataires, séparés par un espace. Un email sera envoyé lors d'un échec d'un build.

☐ Envoyer un email pour chaque build instable

☐ Envoyer des emails séparés aux personnes qui ont cassé le build

?

Ajouter une action après le build ▼

Lancer une compilation manuelle

- Bien que l'outil serve essentiellement à automatiser les builds, il est possible de provoquer manuellement des constructions

Tous +

S	M	Nom du projet ↓	Demier succès	Demier échec	Dernière durée	
✓	⚙️	banque-csharp	7 mn 17 s #26	s. o.	1,4 s	▶

Tous +

S	M	Nom du projet ↓	Demier succès
✓	⚙️	banque-csharp	7 mn
✓	⚙️		12 mn
✓	⚙️		25 mn

⌄ Modifications

📁 Répertoire de travail

▶ Lancer un build

⚙️ Configurer

🗑 Supprimer Projet

✎ Renommer

Atom feed tout

Icône: S M

Tableau de bord > banque-csharp >

↑ Retour au tableau de bord

📄 État

⌄ Modifications

📁 Répertoire de travail

▶ Lancer un build




⚙️ Configurer

🗑 Supprimer Projet

✎ Renommer

Consulter les rapports de l'intégration continue

- Suite aux différentes exécutions des jobs, on peut consulter les résultats dans un tableau de bord récapitulatif

S	M	Nom du projet ↓	Dernier succès	Dernier échec	Dernière durée
		banque-csharp	6 mn 18 s #27	s. o.	3,3 s 

Résultat de la dernière construction :

- Vert = OK
- Jaune = Tests en Echec
- Rouge = Compilation en échec

Lien vers la dernière construction

La tendance : sous forme de pictogrammes météorologiques, permet de savoir dans quel sens va l'évolution des constructions

Consulter les rapports de l'intégration continue

■ Au sein du projet



Demiers résultats des tests (aucune erreur)

Liens permanents

- Demier build (#31), il y a 6 mn 35 s
- Demier build stable (#30), il y a 36 mn
- Demier build avec succès (#30), il y a 36 mn
- Demier build en échec (#22), il y a 2 h 4 mn
- Demier build instable (#25), il y a 1 h 6 mn
- Demier build non réussi (#25), il y a 1 h 6 mn
- Last completed build (#30), il y a 36 mn

■ On peut avoir pour chaque construction

- Les changements
- La sortie console
- Le résultat des tests

Résultats des tests

0 échecs (±0)

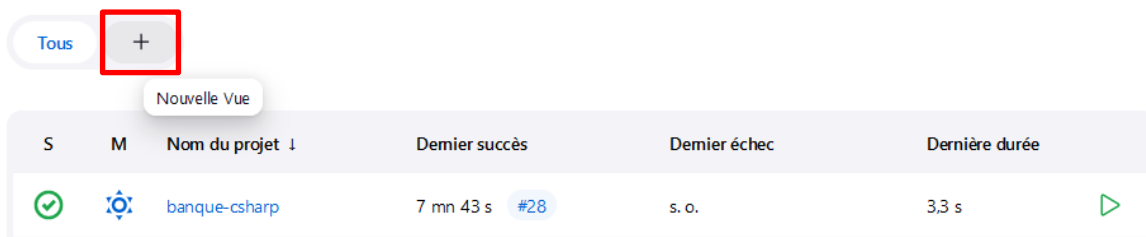


Tous les tests

Package	Durée	Échec (diff)	Sauté (diff)	Pass (diff)	Total (diff)
App.Tests.Unit.Backend	0,1 s	0	0	8	8
App.Tests.Unit.Business	0 ms	0	0	3	3
App.Tests.Unit.Entity	12 ms	0	0	20	20

Les vues

- Les projets peuvent être très nombreux et le récapitulatif peu lisible ou pertinent
- Les vues sont une solution qui permet de ne montrer que certains projets



The screenshot shows a user interface for managing projects. At the top, there is a filter bar with a 'Tous' button and a '+' button (highlighted with a red box). Below this is a 'Nouvelle Vue' button. The main part of the interface is a table with the following columns: 'S', 'M', 'Nom du projet ↓', 'Dernier succès', 'Dernier échec', and 'Dernière durée'. The table contains one row for a project named 'banque-csharp' with a status of '7 mn 43 s' and a value of '#28'. A green play button is visible at the end of the row.

S	M	Nom du projet ↓	Dernier succès	Dernier échec	Dernière durée
✓	⚙️	banque-csharp	7 mn 43 s #28	s. o.	3,3 s

Nom de la vue

Type

☐ 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.

Create

Les vues

- Les vues permettent non seulement de sélectionner les projets à afficher, mais aussi de choisir et d'ordonner les colonnes affichées

Filtres pour les tâches

☐ Rechercher récursivement dans les sous répertoires

Tâches

☐ banque-csharp
☐ banque-java
☐ banque-symfony

☐ Utilisez une expression régulière pour inclure les jobs dans la vue ?

Ajouter un filtre +

Colonnes

≡ Statut	×
≡ Météo	×
≡ Nom du projet	×
≡ Dernier succès	×
≡ Dernier échec	×
≡ Dernière durée	×
≡ Bouton de build	×

Ajouter une colonne +

Ok Apply

Les utilisateurs

- La gestion des utilisateurs se fait via « Configurer la sécurité globale » dans l'administration
- On peut y configurer :
 - Le référentiel de comptes utilisateurs
 - Les autorisations d'accès

Royaume pour la sécurité (Realm)

Base de données des utilisateurs de Jenkins

▼

?

Base de données des utilisateurs de Jenkins

Déléguer au conteneur de servlets

LDAP

None

Autorisations

Les utilisateurs connectés peuvent tout faire

▼

?

Les utilisateurs connectés peuvent tout faire

Mode legacy

Stratégie d'autorisation matricielle basée sur les projets

Sécurité basée sur une matrice

Tout le monde a accès à toutes les fonctionnalités

Ajout de plugins

- Jenkins est un produit extensible qui peut interagir avec de nombreux outils qui font souvent partie du quotidien de la gestion de projet
- L'installation de plugins est simple et se fait directement depuis l'interface de Jenkins

Plugin Manager

Mises à jour Disponibles Installés Avancé

🔍 Search

Install	Name ↕	Released
<input type="checkbox"/>	Git server 99.va_0826a_b_cdfa_d git Library plugins (for use by other plugins) Allows Jenkins to act as a Git server.	1 mo. 22 j ago
<input type="checkbox"/>	Pipeline: Deprecated Groovy Libraries 588.v576c103a_ff86 Miscellaneous Hosting of Pipeline Groovy libraries inside a Jenkins Git server. Deprecated. Use Pipeline: Groovy Libraries instead. If you see this plugin installed just because you upgraded, you can probably uninstall it now. This plugin should only be used if you have historically <i>pushed</i> libraries to a Git server inside Jenkins.	3 mo. 28 j ago

Travaux Pratiques



www.eni-service.fr