



eBook Gratuit

APPRENEZ intellij-idea

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#intellij-idea

Table des matières

À propos.....	1
Chapitre 1: Démarrer avec intellij-idea.....	2
Remarques.....	2
Exemples.....	2
Installation ou configuration.....	2
Windows et Linux.....	2
OS X / MacOS.....	2
Arch Linux.....	3
Ubuntu.....	3
Autre.....	4
Bonjour le monde!.....	4
Migration d'Eclipse.....	14
Chapitre 2: Caractéristiques peu connues.....	20
Exemples.....	20
Injection de langue.....	20
Aperçu rapide.....	20
Vérifiez la vidéo screencast sur YouTube à propos de cette fonctionnalité.....	21
Chapitre 3: Comment installer des plugins.....	25
Introduction.....	25
Exemples.....	25
Pour télécharger et installer un plugin de référentiel.....	25
Pour installer un plugin depuis le disque.....	25
Chapitre 4: Exportation.....	28
Exemples.....	28
Construire un .jar.....	28
Chapitre 5: Hibernate HQL console et inspections.....	32
Introduction.....	32
Exemples.....	32
Configuration des inspections HQL.....	32

Chapitre 6: IdeaVim	33
Remarques.....	33
Exemples.....	33
Affichage des numéros de ligne.....	33
Allouer des frappes au clavier contradictoires à IdeaVim.....	33
Chapitre 7: Inspections	35
Introduction.....	35
Exemples.....	35
@NotNull / @Nullable inspections.....	35
Chapitre 8: Modèles en direct	37
Exemples.....	37
Ajouter facilement une méthode de test.....	37
Insérer le nom de la classe en cours.....	38
Chapitre 9: Optimisation	40
Exemples.....	40
Personnalisation des options de la machine virtuelle.....	40
Chapitre 10: Outils de base de données	41
Exemples.....	41
Créer une nouvelle source de données.....	41
Chapitre 11: Raccourcis utiles	47
Exemples.....	47
Compiler et exécuter.....	47
Faire un projet (compiler modifié et dépendant)	47
Windows: Ctrl + F9.....	47
OS X / macOS: Cmd + F9.....	47
Compiler le fichier, le package ou le module sélectionné	47
Windows: Ctrl + Maj + F9.....	47
OS X / macOS: Cmd + Shift + F9.....	47
Sélectionnez la configuration et exécutez	47
Windows: Alt + Maj + F10.....	47
OS X / macOS: Option + Maj + F10.....	47

Sélectionnez la configuration et le débogage	47
Windows: Alt + Maj + F9.....	47
OS X / macOS: Option + Maj + F9.....	47
Courir	47
Déboguer	48
Exécuter la configuration du contexte à partir de l'éditeur	48
Windows: Ctrl + Maj + F10.....	48
OS X / macOS: Cmd + Shift + F10.....	48
Achèvement du code.....	48
Achèvement du code de base (le nom de toute classe, méthode ou variable)	48
Windows: Ctrl + Espace.....	48
OS X / macOS: Cmd + Espace.....	48
Achèvement du code intelligent (filtre la liste des méthodes et des variables par type att	48
Windows: Ctrl + Maj + Espace.....	48
OS X / macOS: Cmd + Maj + Espace.....	48
Remplacer le code par une suggestion	48
Ajout de code à partir d'une suggestion d'achèvement	49
Rechercher / Remplacer.....	49
Rechercher partout	49
Trouver	49
Windows / Linux: Ctrl + F.....	49
OS X / macOS: Cmd + F.....	49
Trouver le prochain	49
Trouver précédent	49
Remplacer	49
Windows / Linux: Ctrl + R.....	49
OS X / macOS: Cmd + R.....	49
Trouver dans le chemin	49
Windows / Linux: Ctrl + Shift + F.....	49
OS X / macOS: Cmd + Shift + F.....	50

Remplacer dans le chemin	50
Windows / Linux: Ctrl + Maj + R.....	50
OS X / macOS: Cmd + Shift + R.....	50
Refactoring.....	50
Copie	50
Bouge toi	50
Supprimer en toute sécurité	50
Windows / Linux: Alt + Suppr.....	50
OS X / macOS: Cmd + Supprimer.....	50
Renommer	50
Méthode d'extraction	50
Windows / Linux: Ctrl + Alt + M.....	50
OS X / macOS: Cmd + Option + M.....	50
Extraire le champ	50
Windows / Linux: Ctrl + Alt + F.....	51
OS X / macOS: Cmd + Option + F.....	51
Extraire la variable	51
Windows / Linux: Ctrl + Alt + V.....	51
OS X / macOS: Cmd + Option + V.....	51
Extraire la constante	51
Windows / Linux: Ctrl + Alt + C.....	51
OS X / macOS: Cmd + Option + C.....	51
Extraire le paramètre	51
Windows / Linux: Ctrl + Alt + P.....	51
OS X / macOS: Cmd + Option + P.....	51
Autre.....	51
Surround avec	51
Windows / Linux: Ctrl + Alt + T.....	51
OS X / macOS: Cmd + Option + T.....	51
Navigation de base.....	51

Aller à l'éditeur (depuis la fenêtre de l'outil)	51
Mise au point sur la fenêtre d'outils correspondante	52
Windows: Alt + <numéro de fenêtre d'outil>.....	52
OS X / macOS: Cmd + <numéro de la fenêtre de l'outil>.....	52
Par exemple, basculer le focus sur la fenêtre du projet	52
Windows: Alt + 1.....	52
OS X / macOS: Cmd + 1.....	52
Fichiers récents popup	52
Windows: Ctrl + E.....	52
OS X / macOS: Cmd + E.....	52
Trouver une action	52
Windows: Ctrl + Maj + A.....	52
OS X / macOS: Cmd + Shift + A.....	52
Aller vers	52
Fichier :	52
Windows: Ctrl + Maj + N.....	53
OS X / macOS: Cmd + Shift + N.....	53
Classe:	53
Windows: Ctrl + N.....	53
OS X / macOS: Cmd + N.....	53
Symbole (classe / méthode / variable / nom de la constante):	53
Windows: Ctrl + Alt + Maj + N.....	53
OS X / macOS: Cmd + Option + Maj + N.....	53
Partout :	53
Windows: Maj + Maj.....	53
OS X / macOS: Maj + Maj.....	53
Aller au numéro de ligne	53
Windows: Ctrl + G.....	53
OS X / macOS: Cmd + L.....	54
Revenir au dernier emplacement d'édition	54

Windows: Ctrl + Maj + Retour arrière.....	54
OS X / macOS: Cmd + Maj + Retour arrière.....	54
Recherche d'utilisation.....	54
Rechercher des utilisations / Rechercher des utilisations dans un fichier.....	54
Windows / Linux: Alt + F7 / Ctrl + F7.....	54
OS X / macOS: Option + F7 / Ctrl + F7.....	54
Mettez en surbrillance les utilisations dans un fichier.....	54
Windows / Linux: Ctrl + Maj + F7.....	54
OS X / macOS: Cmd + Shift + F7.....	54
Afficher les usages.....	54
Windows / Linux: Ctrl + Alt + F7.....	54
OS X / macOS: Cmd + Option + F7.....	54
Afficher les paramètres de la méthode.....	54
Windows / Linux: Ctrl + P.....	54
OS X / macOS: Cmd + P.....	55
Sélection.....	55
Windows: Ctrl + W.....	55
OS X / macOS: Cmd + W.....	55
Windows: Ctrl + Maj + W.....	55
OS X / macOS: Cmd + Shift + W.....	55
Windows: Alt.....	55
OS X / macOS: Opt.....	55
Windows: Alt + Maj.....	56
OS X / macOS: Opt + Shift.....	56
Windows: Alt + J.....	56
OS X / macOS: ctrl + G.....	56
Crédits.....	58

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [intellij-idea](#)

It is an unofficial and free intellij-idea ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official intellij-idea.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapitre 1: Démarrer avec intellij-idea

Remarques

[intellij-idea](#) est un IDE conçu comme le successeur spirituel de l' [IDE Eclipse](#) largement utilisé pour le développement Java. Eclipse, bien que extrêmement puissant, est souvent critiqué pour son extrême lourdeur et son utilisation difficile.

IntelliJ IDEA tente de créer un IDE avec une puissance similaire à Eclipse, mais avec une finition soignée. Les développeurs auraient avantage à utiliser IDEA en raison des nombreux outils et crochets nécessaires pour gagner du temps sur tous les projets. La complétion de code intelligente, l'intégration de tests unitaires natifs et la gestion native de [Gradle](#) ne sont que quelques-uns des points forts de Java IDE de JetBrains

Exemples

Installation ou configuration

Il existe deux versions principales d'IntelliJ IDEA: l'édition communautaire et l'édition ultime. L'édition Community est gratuite et ne manque pas de fonctionnalités en termes de développement Java SE.

Windows et Linux

Téléchargez IntelliJ IDEA à partir [du site Web JetBrains](#) et suivez les procédures d'installation. Si le kit de développement Java (JDK) n'est pas installé, [téléchargez et installez le JDK](#) . Notez que vous avez besoin du JDK, seul l'environnement Java Runtime Environment (JRE) ne suffit pas.

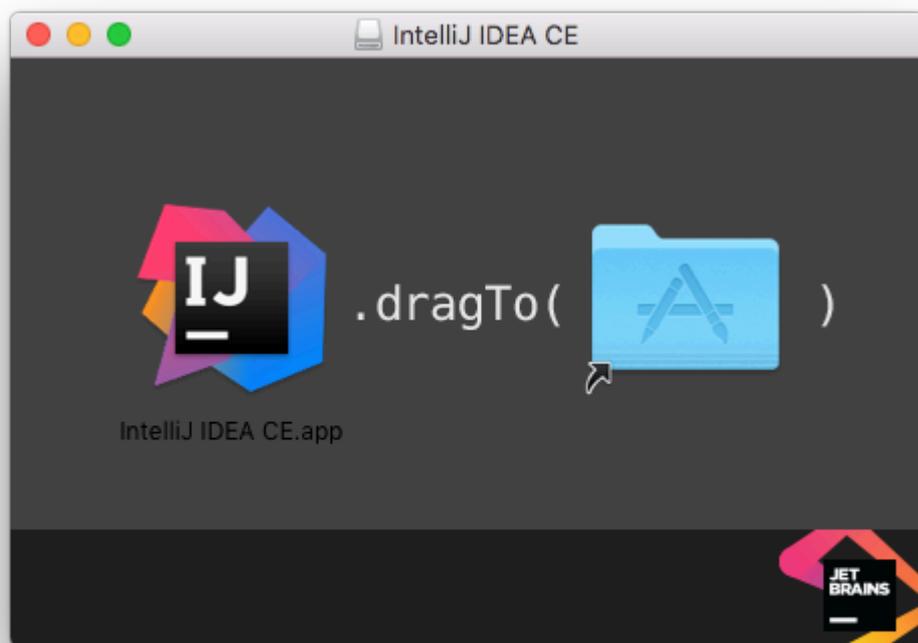
Une fois que IntelliJ IDEA a été téléchargé:

1. Exécutez le programme d'installation
2. Presse suivante
3. Choisissez un dossier dans lequel installer IntelliJ IDEA (dans la plupart des cas, laissez-le par défaut)
4. Choisissez un dossier de menu de démarrage pour créer des raccourcis IntelliJ IDEA (Dans la plupart des cas, laissez cela comme valeur par défaut)
5. Choisissez si vous souhaitez créer un raccourci sur le bureau et choisir d'associer différents fichiers Java à IntelliJ IDEA.
6. Appuyez sur suivant et attendez qu'il s'installe

OS X / MacOS

Téléchargez IntelliJ IDEA à partir [du site Web JetBrains](#) , ouvrez le fichier d'image disque (* .dmg)

téléchargé et faites glisser l'application dans l'alias de votre dossier `/Applications` .



Arch Linux

IntelliJ IDEA peut être installé sur Arch Linux en utilisant son gestionnaire de paquets, `pacman` . Ouvrez un terminal et entrez la commande suivante.

```
sudo pacman -S intellij-idea-community-edition
```

L'utilisation de `sudo` n'est pas requise si vous utilisez l'utilisateur `root`.

```
@ArchLinux ~$ sudo pacman -S intellij-idea-community-edition
[sudo] password for [redacted]:
resolving dependencies...
:: There are 2 providers available for java-environment:
:: Repository extra
   1) jdk7-openjdk  2) jdk8-openjdk

Enter a number (default=1): 2
looking for conflicting packages...

Packages (7) intellij-idea-libs-2:2016.2-1  java-environment-common-2-2  java-runtime-common-2-2  jdk8-openjdk-8
             intellij-idea-community-edition-2:2016.2-1

Total Download Size: 251.91 MiB
Total Installed Size: 593.61 MiB
```

Ubuntu

(1) Installez le paquet ubuntu-make.

Pour Ubuntu 16.04 et ultérieur,

```
sudo apt install ubuntu-make
```

Pour les versions précédentes d'ubuntu,

```
sudo add-apt-repository ppa:ubuntu-desktop/ubuntu-make  
sudo apt-get update  
sudo apt-get install ubuntu-make
```

(2) Après avoir installé Ubuntu Make, faites un

```
umake ide idea
```

Chemin d'installation par défaut: /home/current-user/.local/share/umake/ide/idea

Suivez le projet hello_world listé ci-dessus.

Suivez la [page ubuntu-make](#) pour modifier l'installation par défaut et installer d'autres IDE.

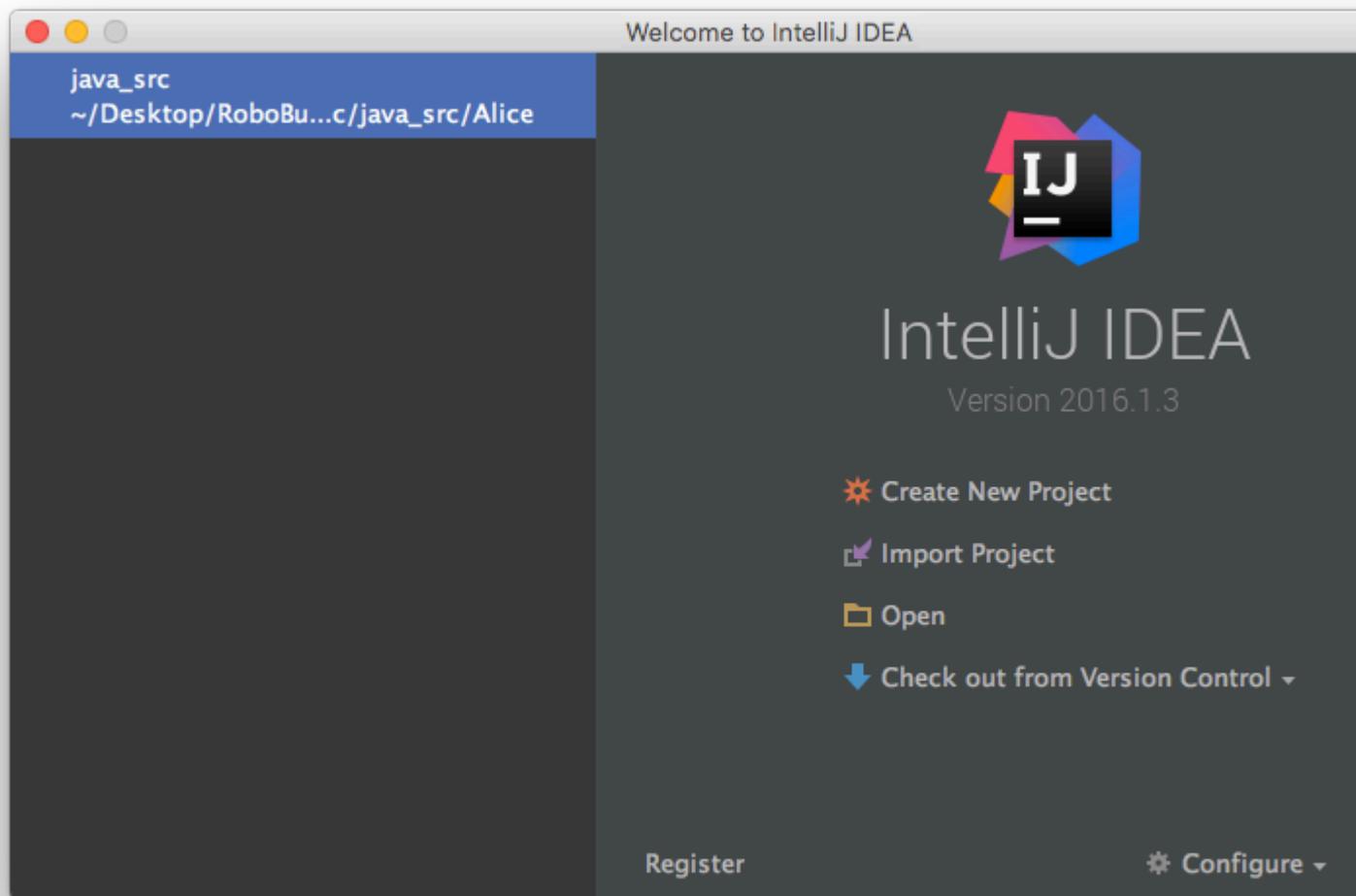
Autre

Vous trouverez plus de détails sur l'installation à l' [adresse suivante](https://www.jetbrains.com/help/idea/2016.1/installing-and-launching.html) :

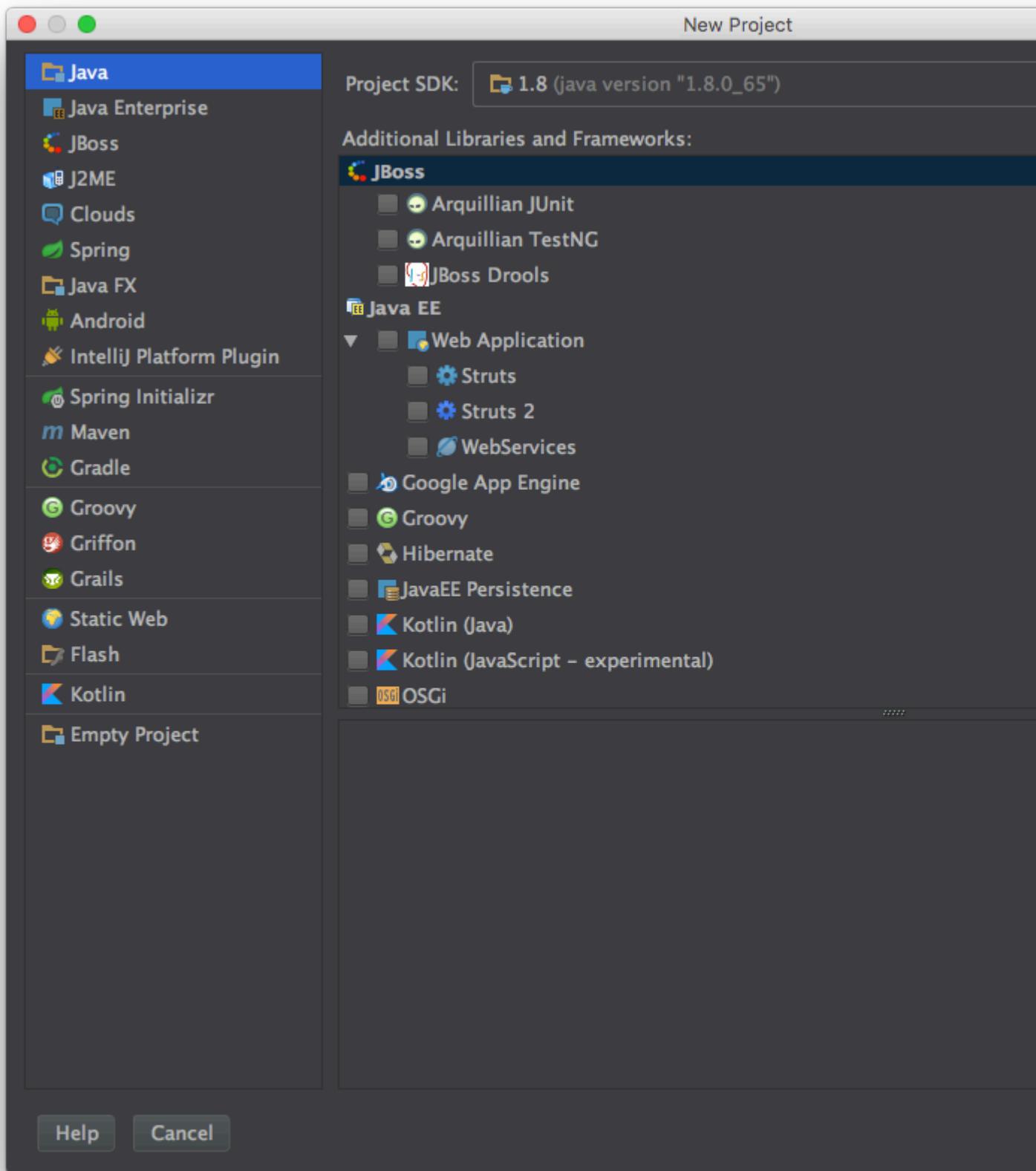
Bonjour le monde!

Cela vous apprendra comment réaliser votre premier projet en utilisant IDEA.

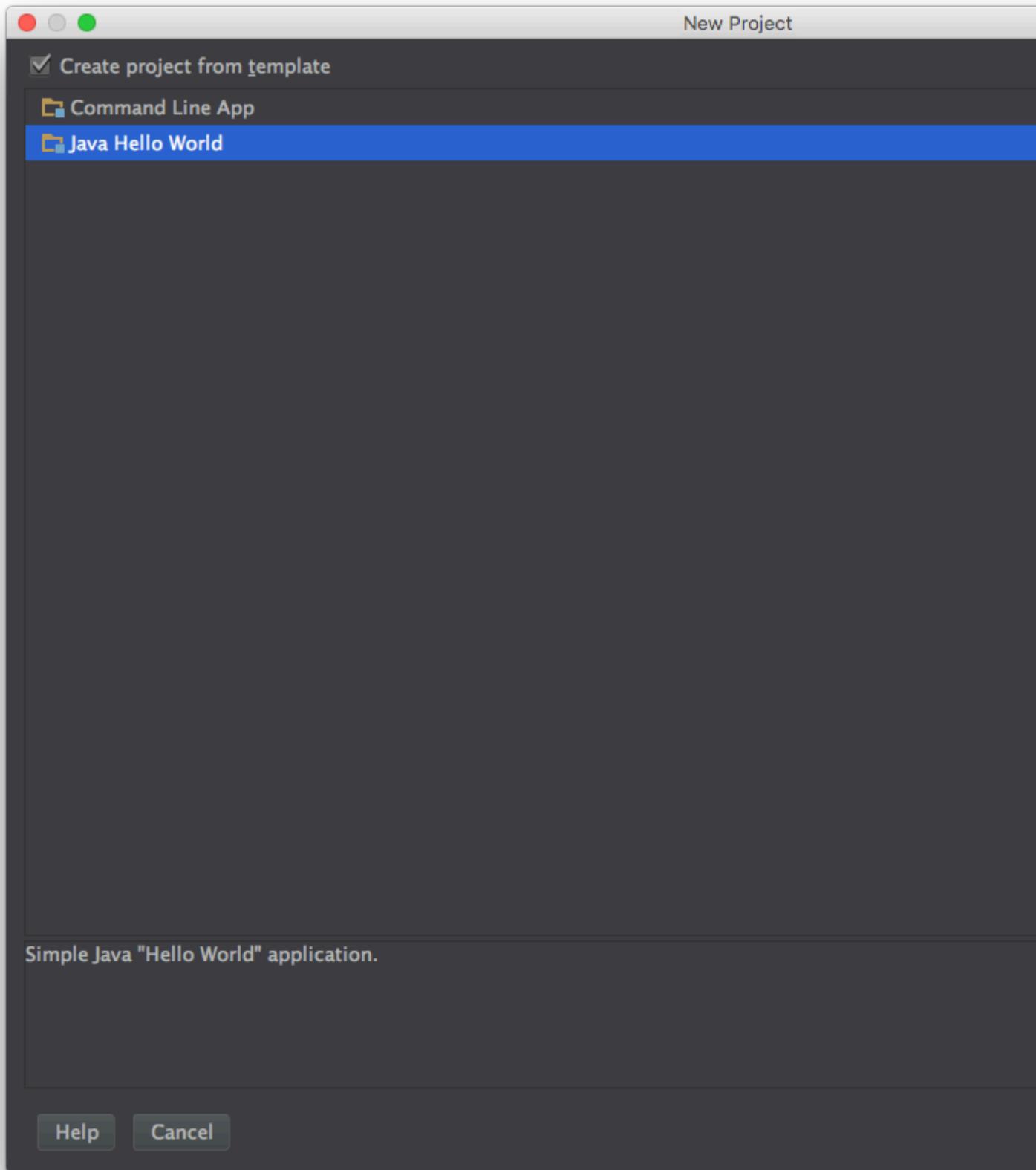
Lancez IDEA et cliquez sur `Create New Project` partir de l'écran de démarrage:



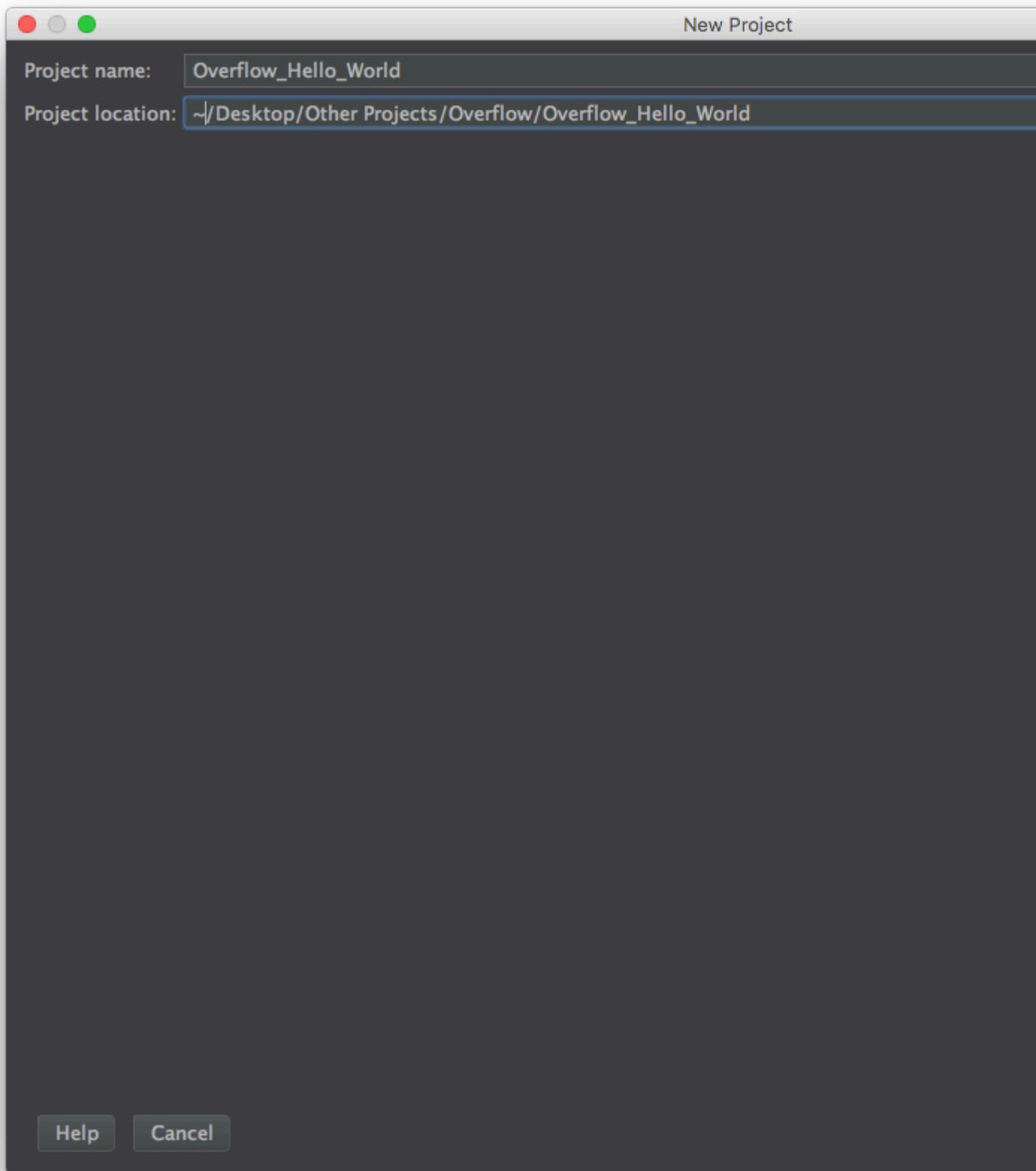
Cliquez sur `Next` sur l'écran suivant. Nous créons un projet Java simple, nous n'avons donc pas besoin d'ajouts ou d'extras à ce projet



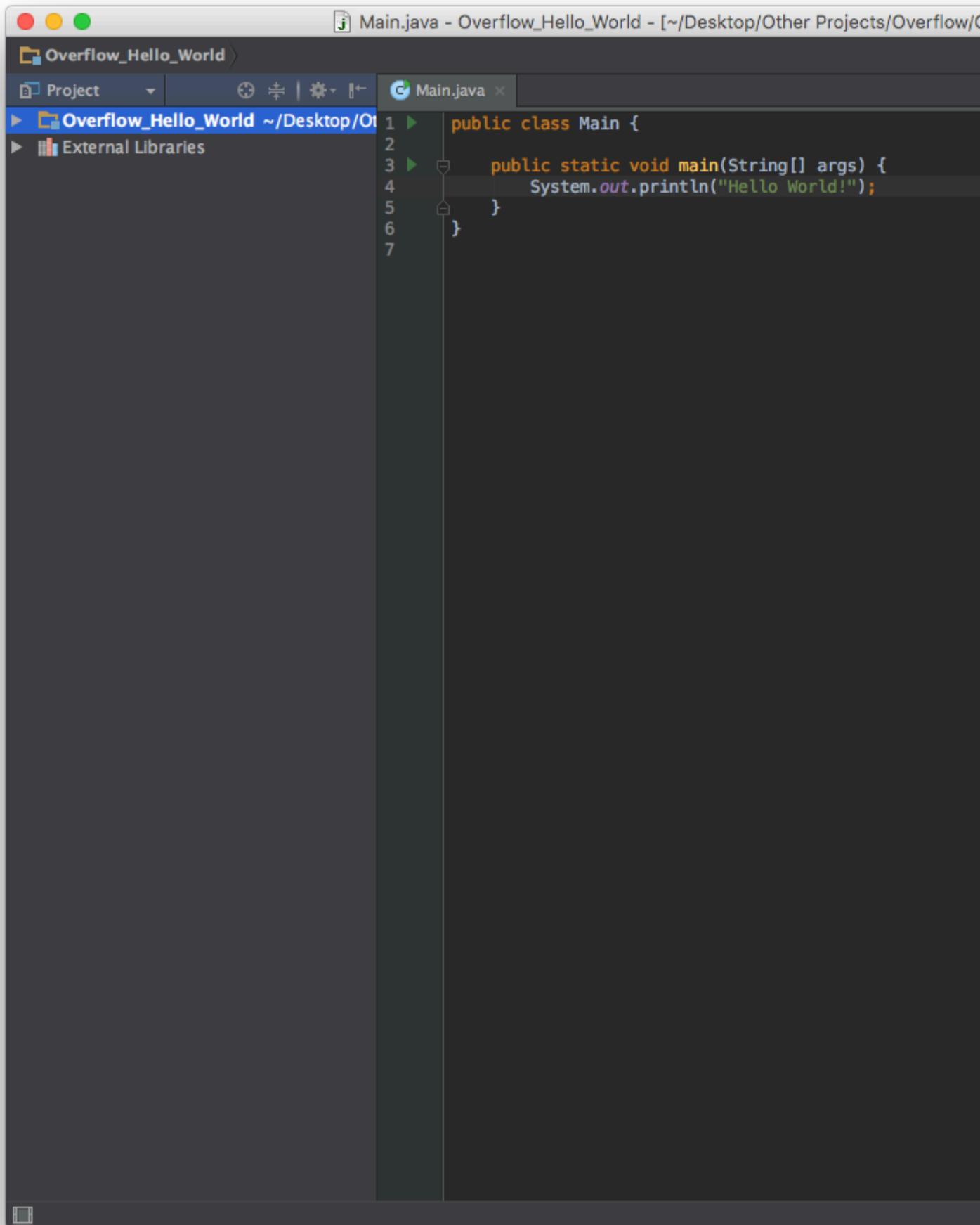
Utilisez l'écran suivant pour créer le projet de modèle `Java Hello World` :



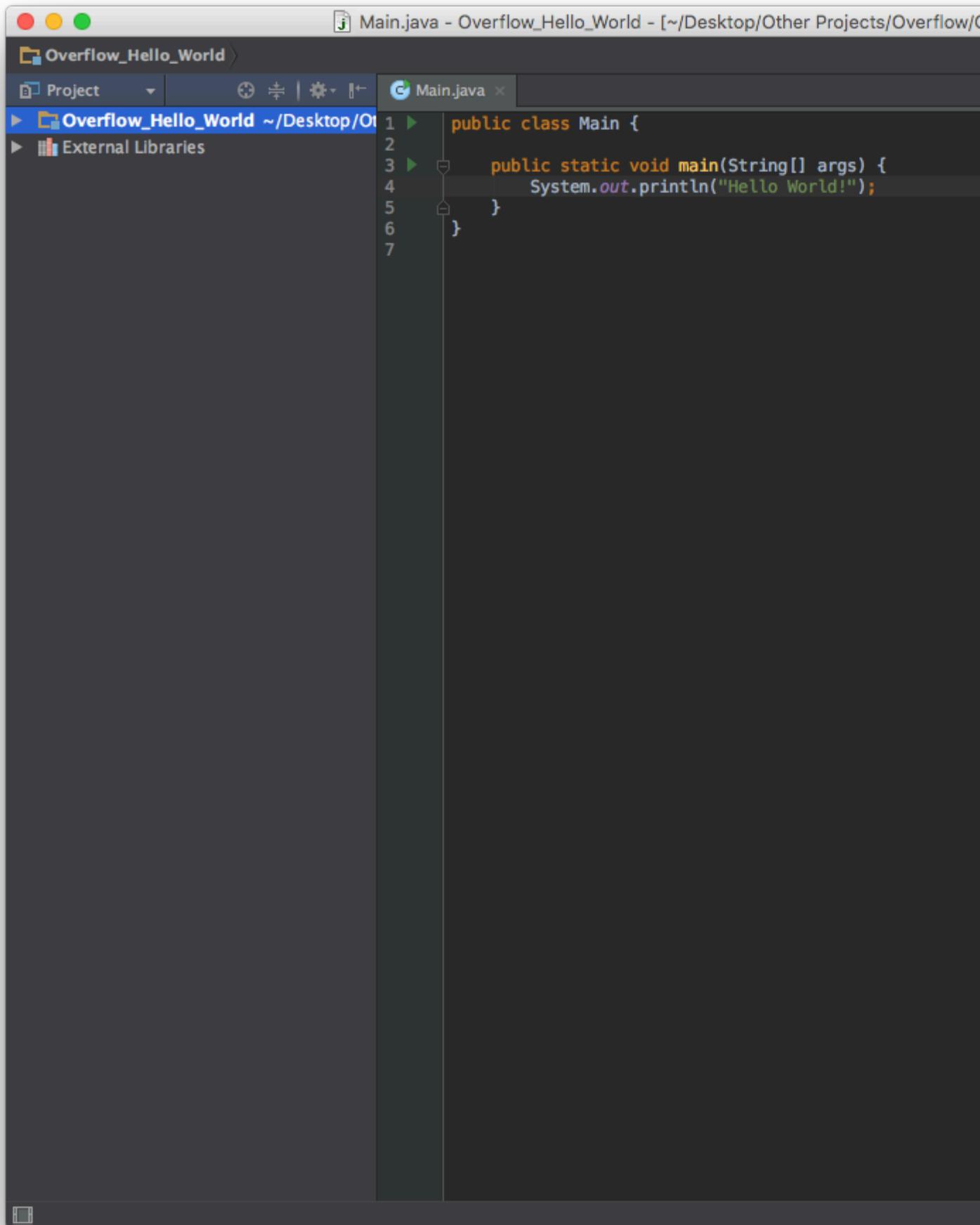
Enfin, nommez votre projet et sélectionnez un emplacement sur le disque, puis cliquez sur `Finish` :



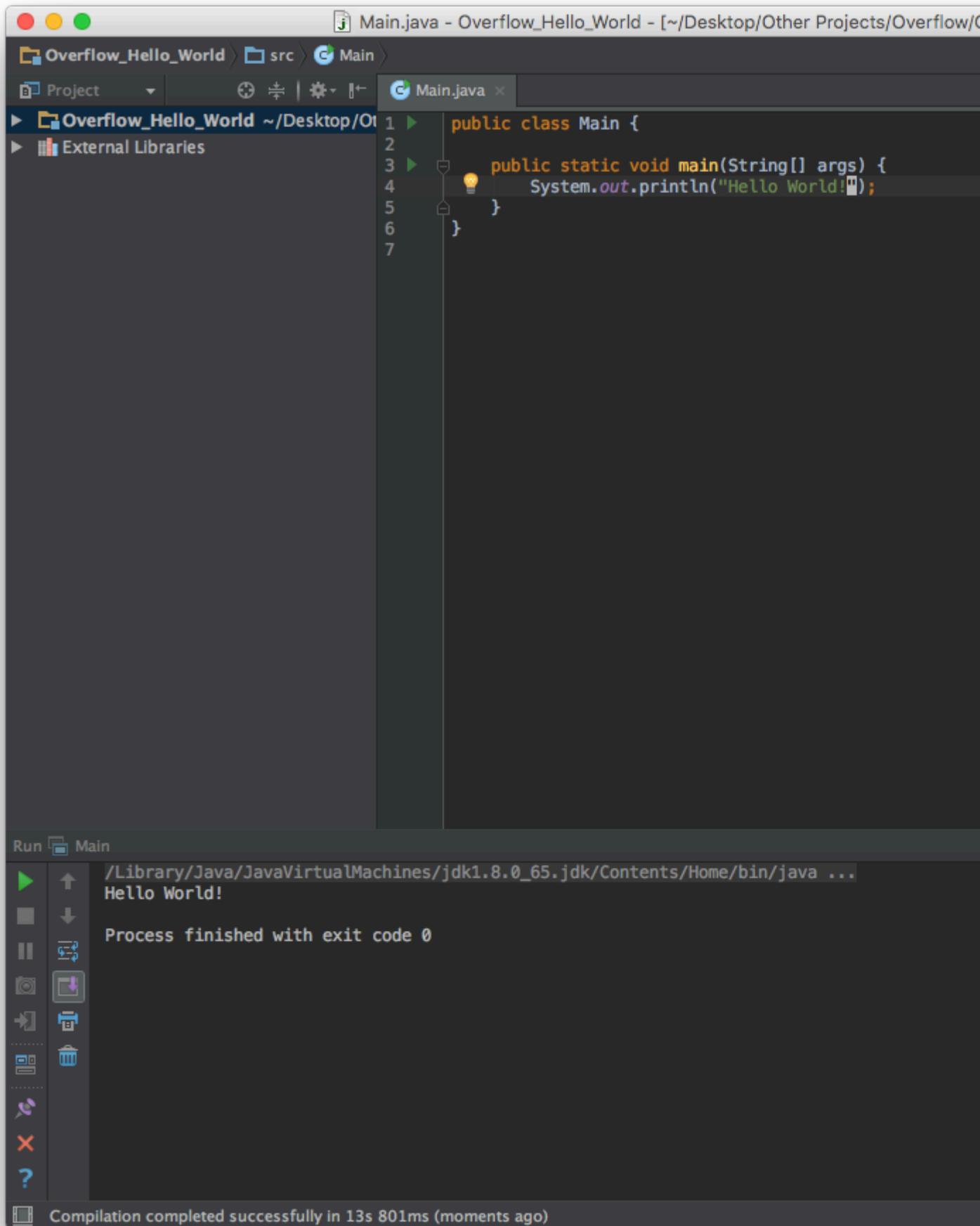
Vous devriez vous retrouver avec une fenêtre qui ressemble à ceci:



À ce stade, le projet est tout prêt, cliquez simplement sur le bouton `Run` ou sur `Run -> Run 'Main'`



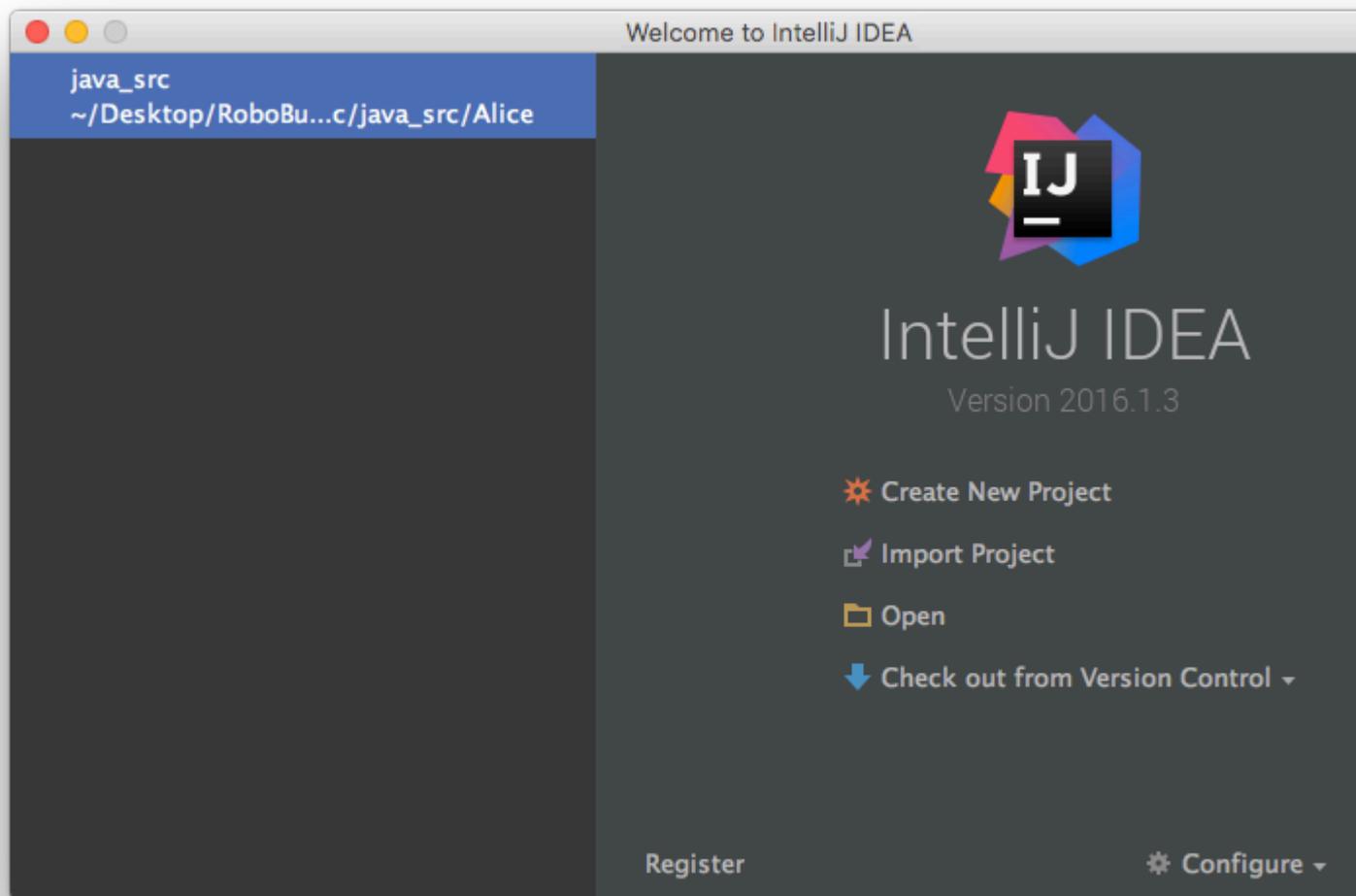
Et tu as fini! La console apparaîtra automatiquement, offrant ses salutations au monde entier!



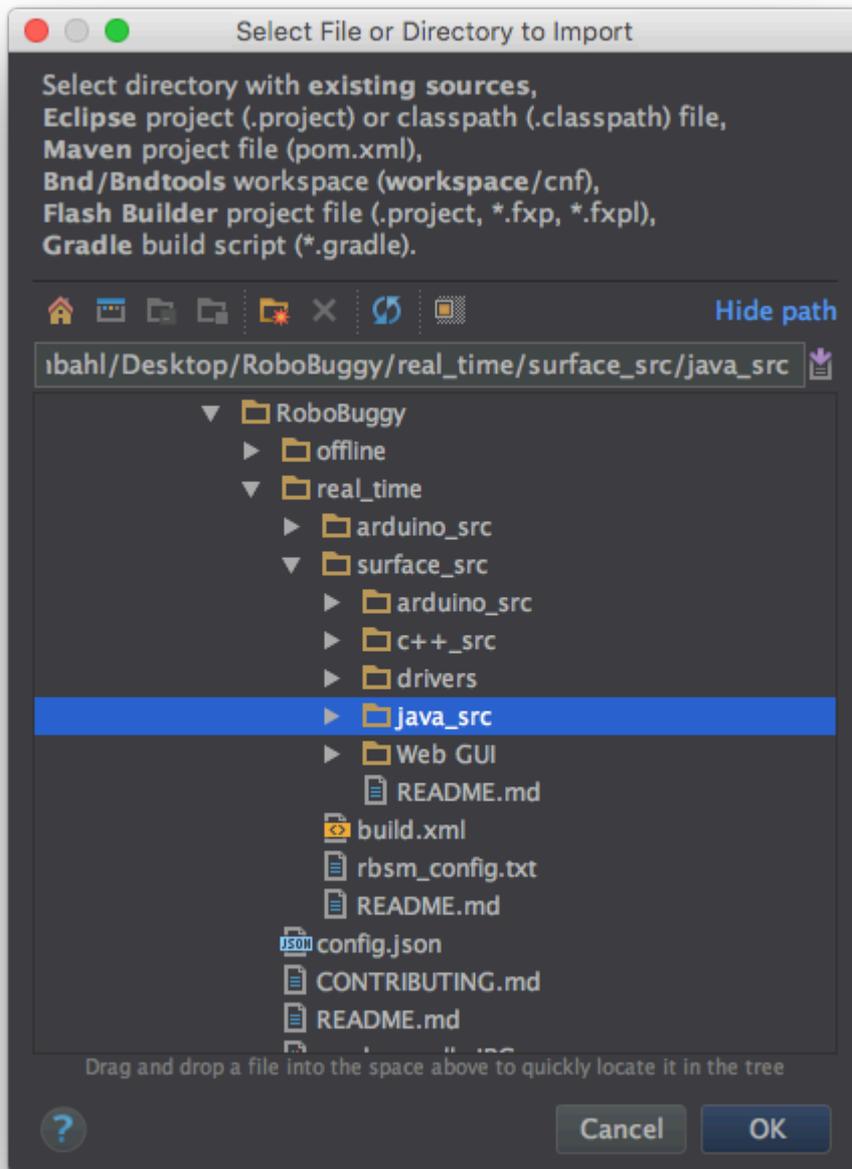
Migration d'Eclipse

IntelliJ IDEA tente de faire appel à la vaste base de données Java qui utilise Eclipse pour leur développement en permettant aux développeurs de migrer leurs projets Eclipse vers une structure IDEA en quelques clics!

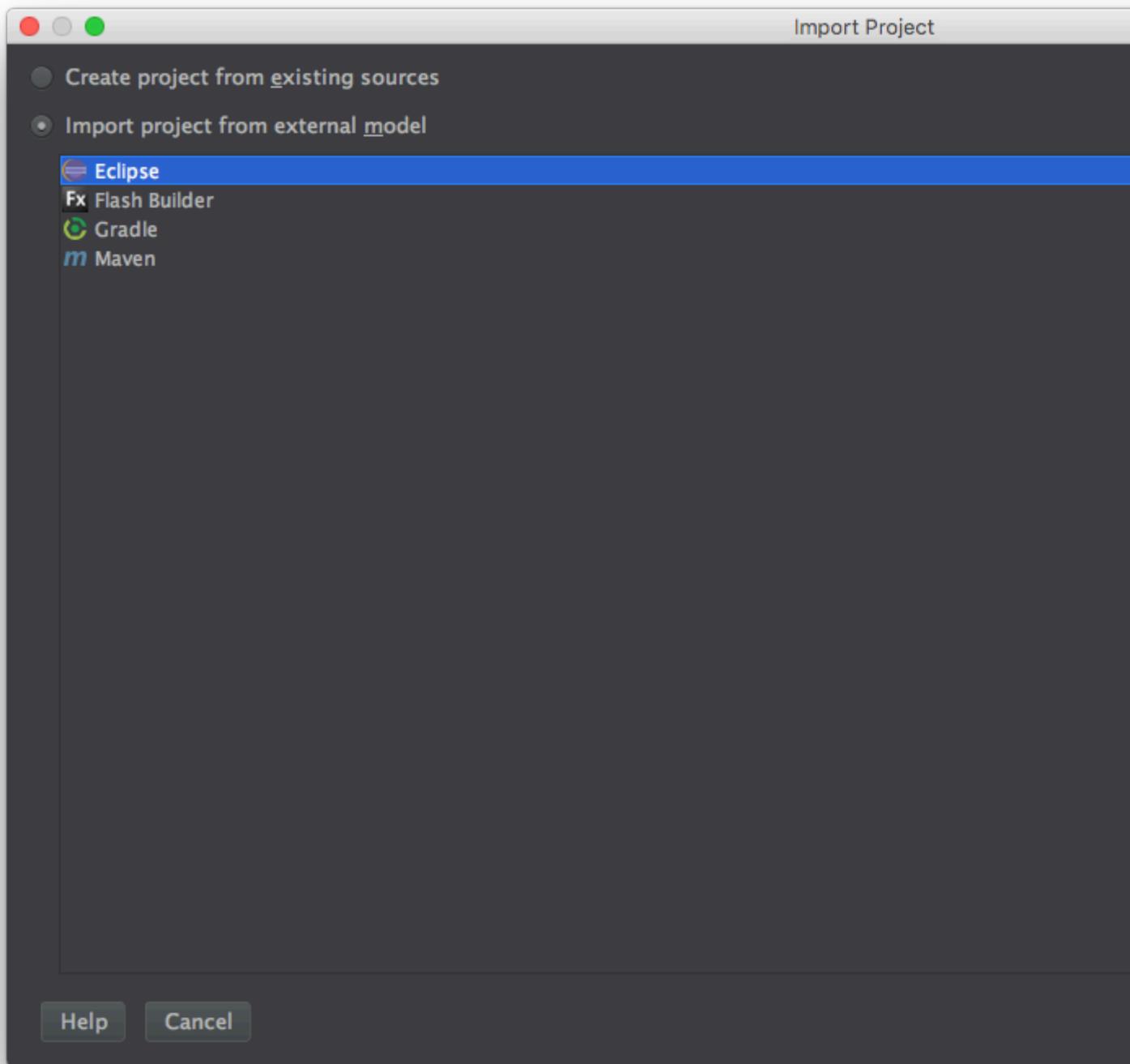
Commencez par démarrer IDEA et cliquez sur `Import Project` dans la fenêtre de démarrage:



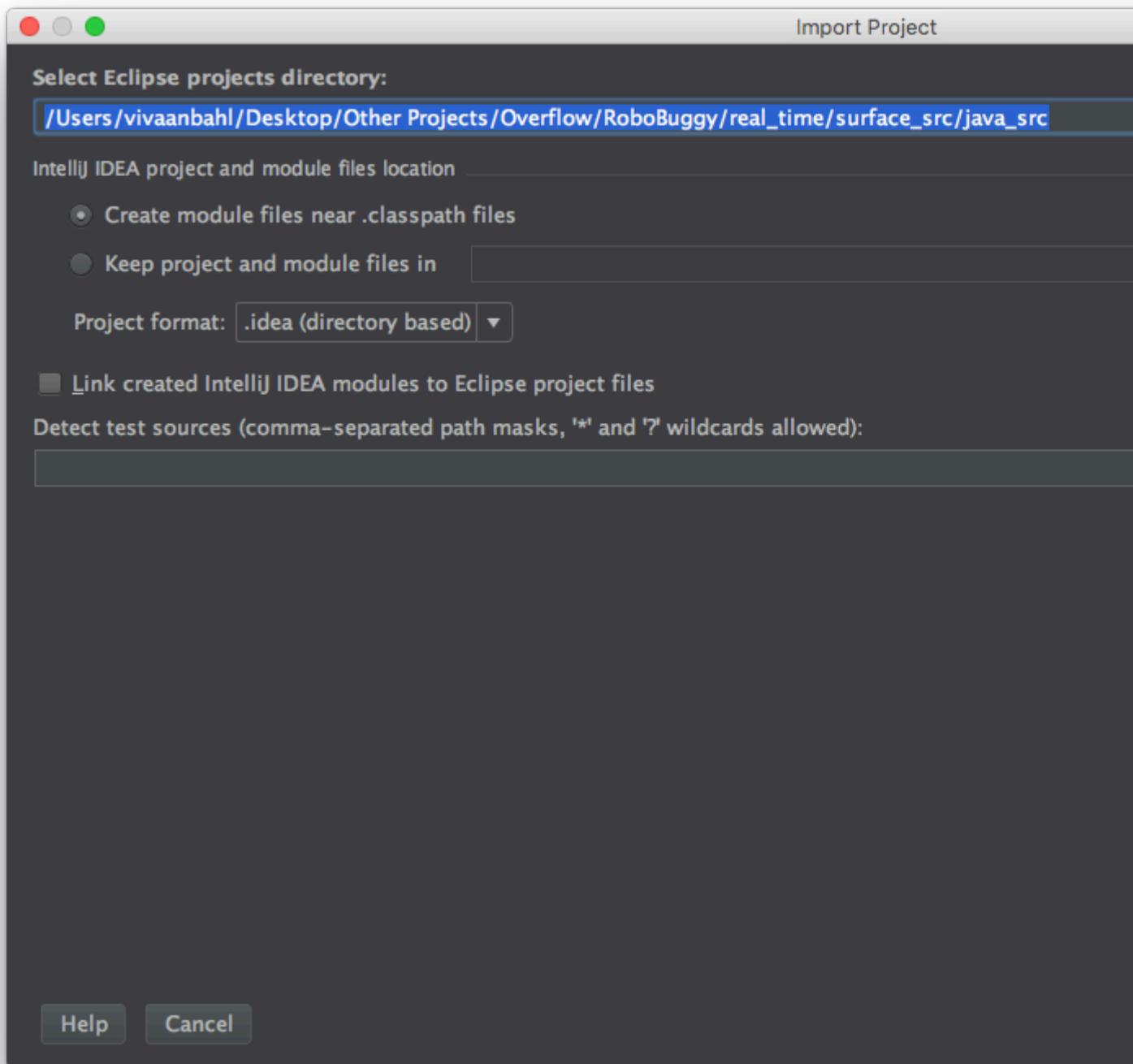
Ensuite, sélectionnez votre projet Eclipse en utilisant la fenêtre de l'explorateur



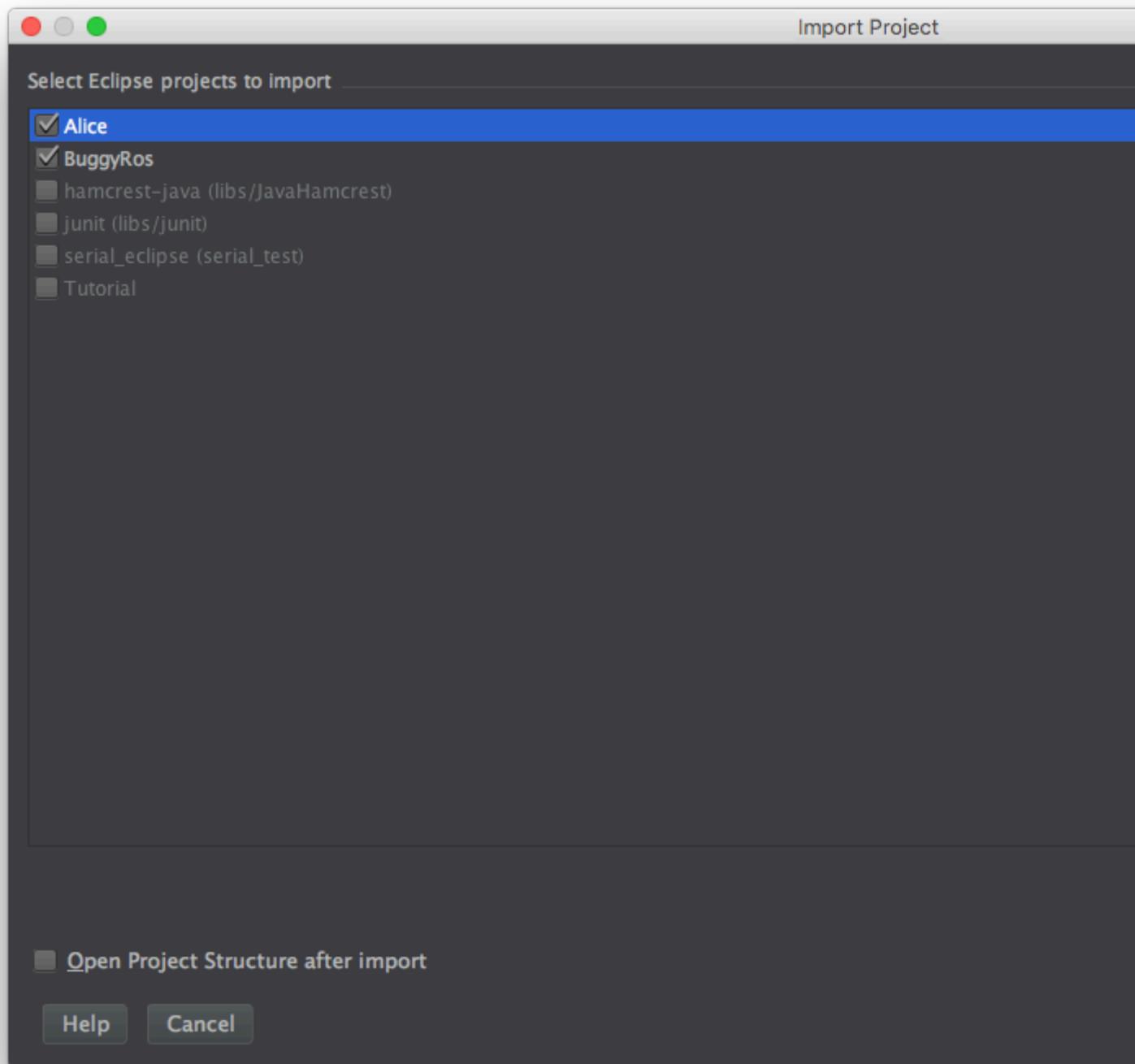
IntelliJ vous demandera le modèle à partir duquel vous importez, assurez-vous que `Eclipse` est sélectionné avant de cliquer sur `Next`



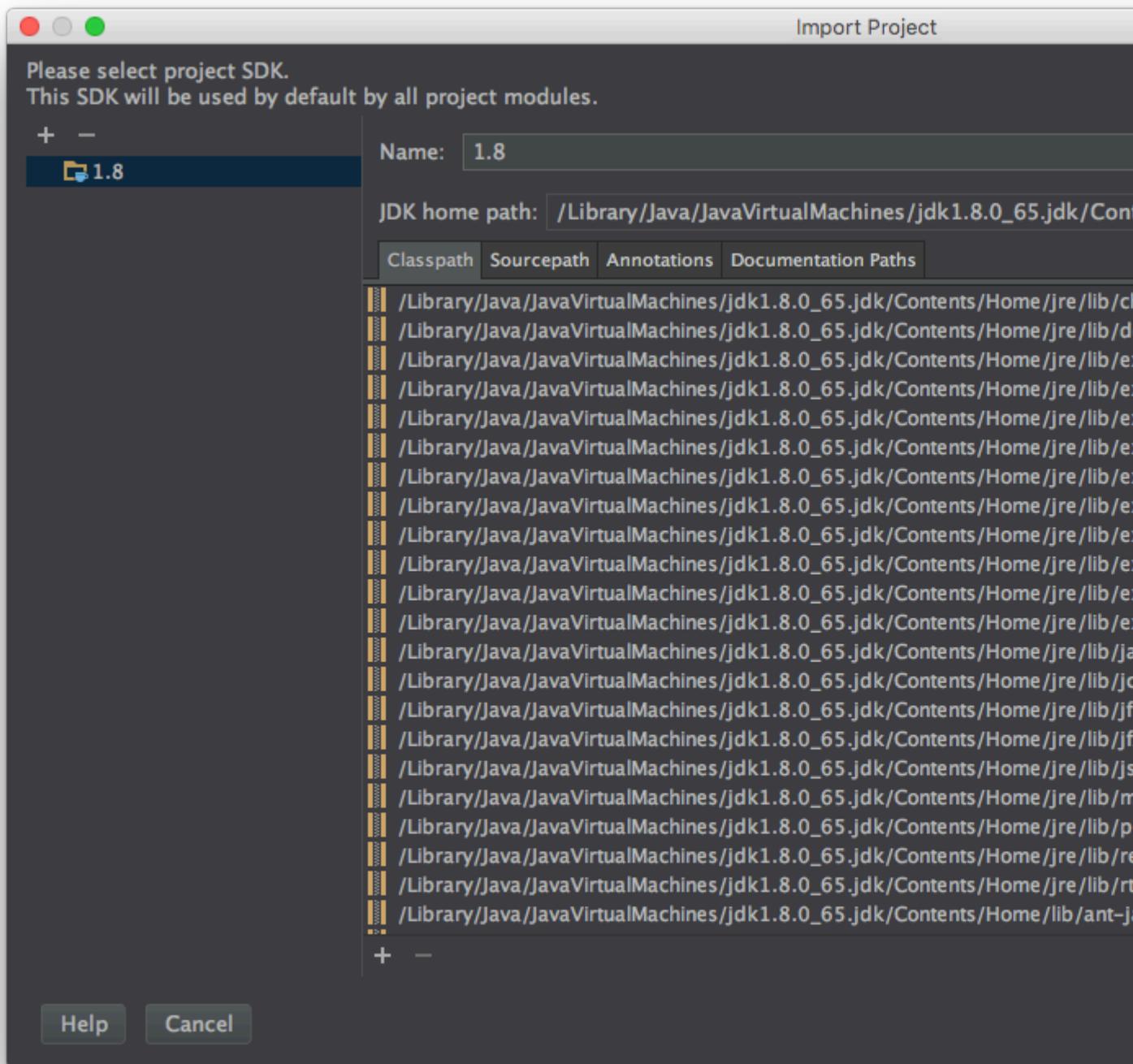
L'écran suivant affichera une confirmation du chemin que vous souhaitez importer, cliquez simplement sur `Next` :



Ensuite, sélectionnez les modules que vous souhaitez créer. Dans l'exemple de projet particulier, seuls les projets `Alice` et `BuggyRos` étaient importants pour Eclipse.



Enfin, assurez-vous que la version correcte du JDK est sélectionnée avant `Finish`



Et le projet Eclipse a été entièrement migré vers IntelliJ! Le projet sera toujours ouvert dans les deux IDE et sera entièrement fonctionnel à la fois

Lire Démarrer avec intellij-idea en ligne: <https://riptutorial.com/fr/intellij-idea/topic/1734/demarrer-avec-intellij-idea>

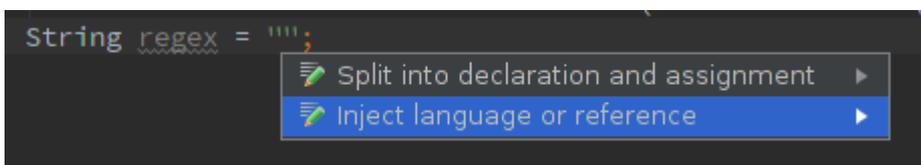
Chapitre 2: Caractéristiques peu connues

Exemples

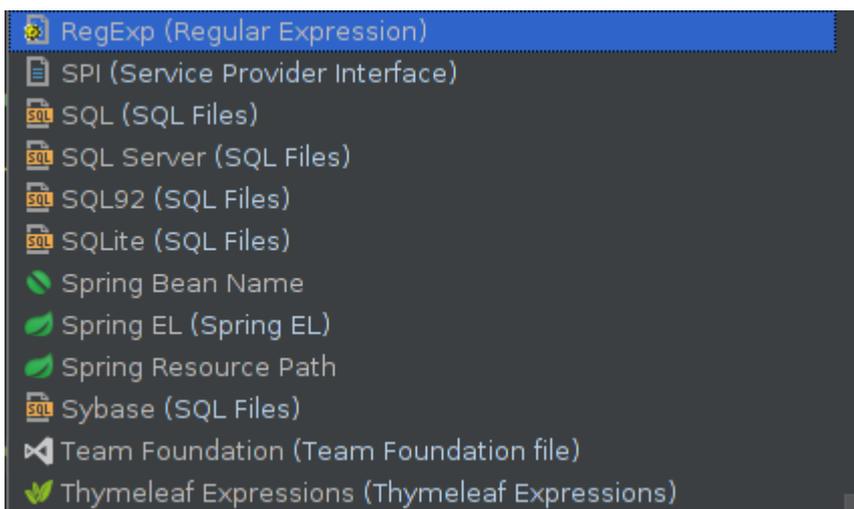
Injection de langue

Si vous voulez écrire des chaînes contenant d'autres langages (JSON, regexes), il est difficile de suivre les symboles qui s'échappent, et ce serait bien de recevoir de l'aide pour le code.

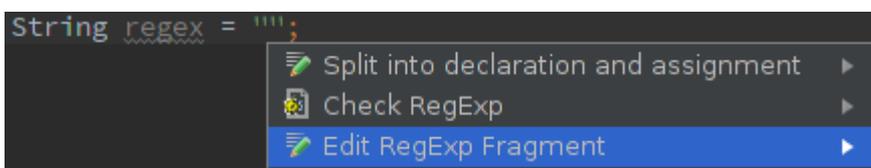
1. Placez votre curseur dans une chaîne vide
2. ALT + ENTER
3. Choisissez "langue ou référence"



4. Choisissez la langue souhaitée (`RegExp` dans mon cas) dans la fenêtre contextuelle



5. Encore une fois, utilisez ALT + ENTER et sélectionnez `Edit regex fragment`



6. Dans la nouvelle fenêtre d'outil, entrez la regex - notez comment elle est automatiquement mappée sur une chaîne Java correctement échappée. De même pour JSON, les indentations seront placées correctement.

Aperçu rapide

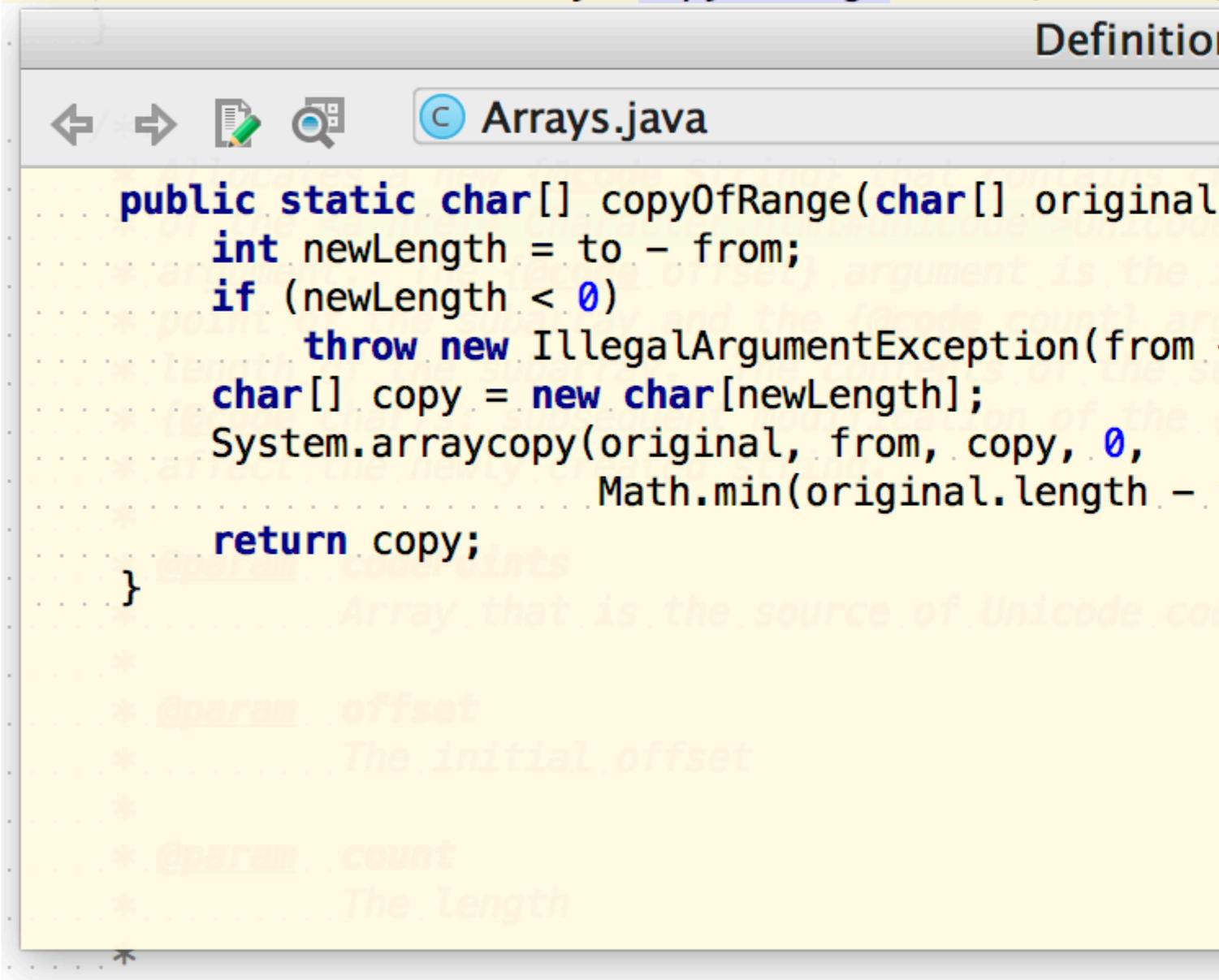
Vérifiez la vidéo screencast sur YouTube à propos de cette fonctionnalité

IntelliJ fournit une fonctionnalité de prévisualisation rapide appelée [Définition de visualisation](#). L'utilisation de cette fonctionnalité permet à un utilisateur de voir rapidement le contenu d'une méthode / classe sans naviguer dans la classe elle-même

- OS X - (⌘ + Y) ou (⌘ + Espace)
- Unix / Windows - Ctrl + Shift + I

Exemple: En regardant dans `Arrays.copyOfRange()` :

```
if (offset > value.length - count) {  
    throw new StringIndexOutOfBoundsException  
}  
this.value = Arrays.copyOfRange(value, offset,
```



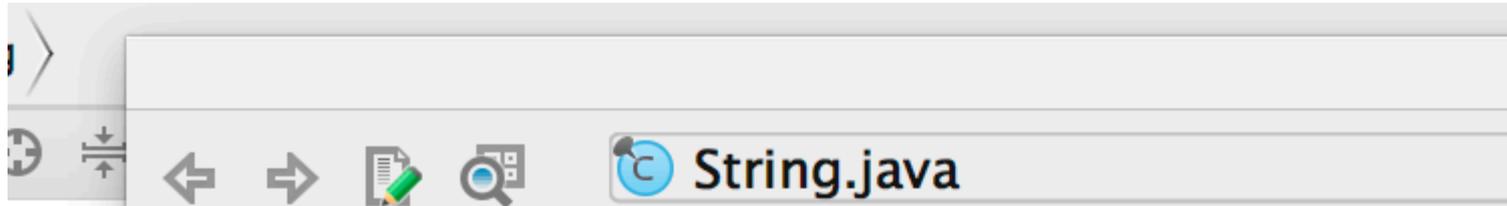
The screenshot shows the IntelliJ IDEA IDE with the 'Definition' window open for the `Arrays.copyOfRange()` method. The window title is 'Arrays.java'. The code in the window is as follows:

```
public static char[] copyOfRange(char[] original  
    int newLength = to - from;  
    if (newLength < 0)  
        throw new IllegalArgumentException(from  
    char[] copy = new char[newLength];  
    System.arraycopy(original, from, copy, 0,  
        Math.min(original.length -  
    return copy;  
}
```

Below the code, there are Javadoc comments for the parameters:

```
* @param offset  
*     The initial offset  
*  
* @param count  
*     The length
```

Vérifier si vous avez sélectionné la bonne classe dans `Search` :



```

public final class String
    implements java.io.Serializable, Comparable<String>
    /** The value is used for character storage. */
    private final char value[];

    /** Cache the hash code for the string */
    private int hash; // Default to 0

    /** use serialVersionUID from JDK 1.0.2 for interoperability */
    private static final long serialVersionUID = 0L;

    /**
     * Class String is special cased within the JVM
     */

```

ties

```

// Note: offset
if (offset < 0)
    throw new StringIndexOutOfBoundsException(offset);
}

```

Enter class name

- String (java.lang)
- String (com.sun.org.apache.xpath.internal.operations.String)
- STRING in DocFlavor (javax.print.attribute)
- String (scala.math.Ordering)
- String (scala.math.Ordering)
- StringAction in OperationFactory (com.sun.jdi.connect)
- StringAdd (scala.Predef)
- StringAdd (scala.runtime)
- StringAdd (scala.runtime)
- StringArgument in Connector (com.sun.jdi.connect)
- StringArgumentImpl in ConnectorImpl (com.sun.jdi.connect)

<https://riptutorial.com/fr/intellij-idea/topic/3755/caracteristiques-peu-connues>

Chapitre 3: Comment installer des plugins

Introduction

Les plugins nous aident à faire les choses plus facilement. IntelliJ propose une vaste gamme de plug-ins pour diverses technologies / langues. Il existe trois façons d'installer un plugin dans IntelliJ.

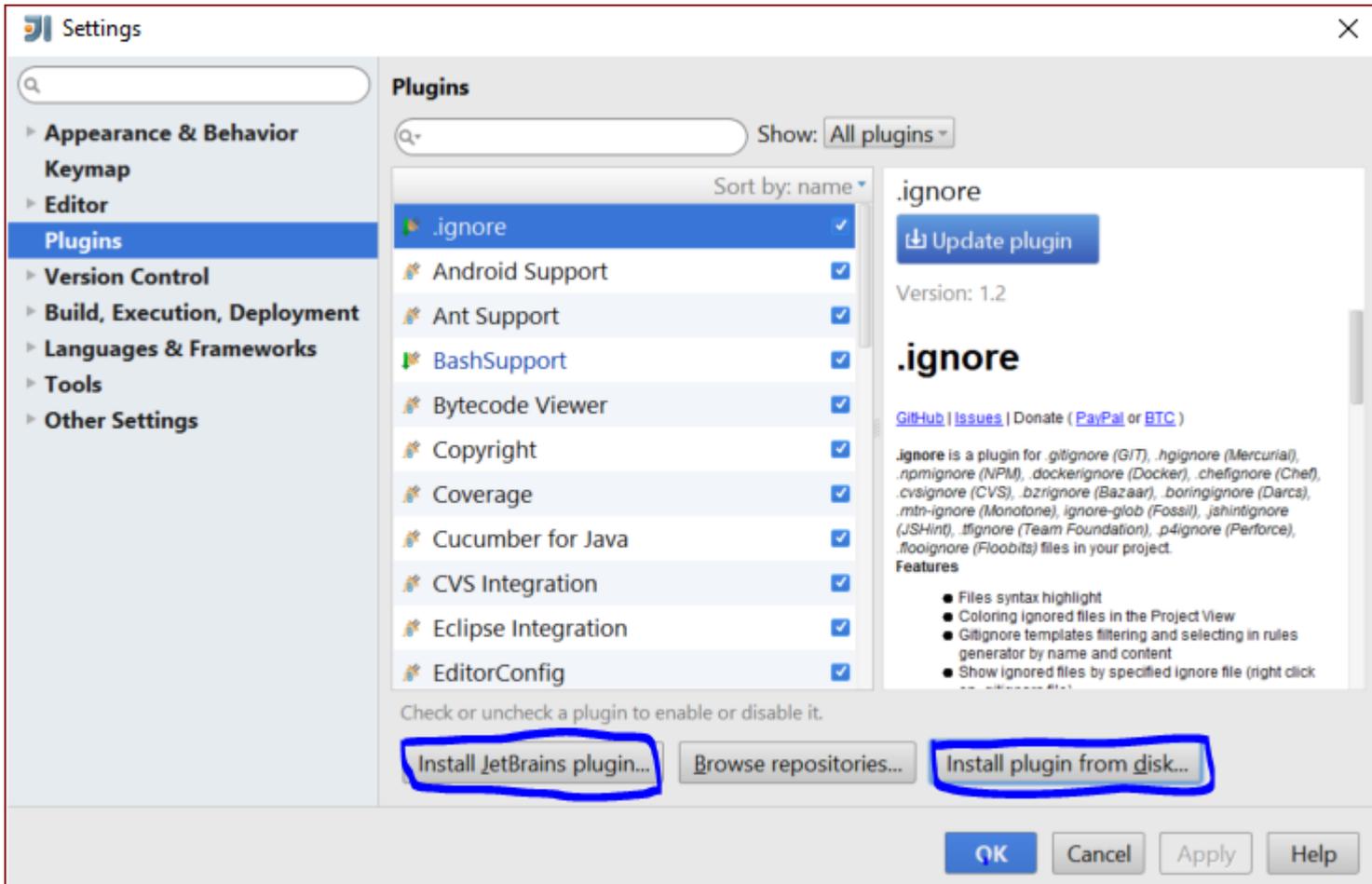
Exemples

Pour télécharger et installer un plugin de référentiel

1. Go to File --> Settings (e.g. Ctrl+Alt+S).
2. In the left-hand pane, select Plugins.
3. On the Plugins window, click "Install JetBrains plugin" or the "Browse repositories button".

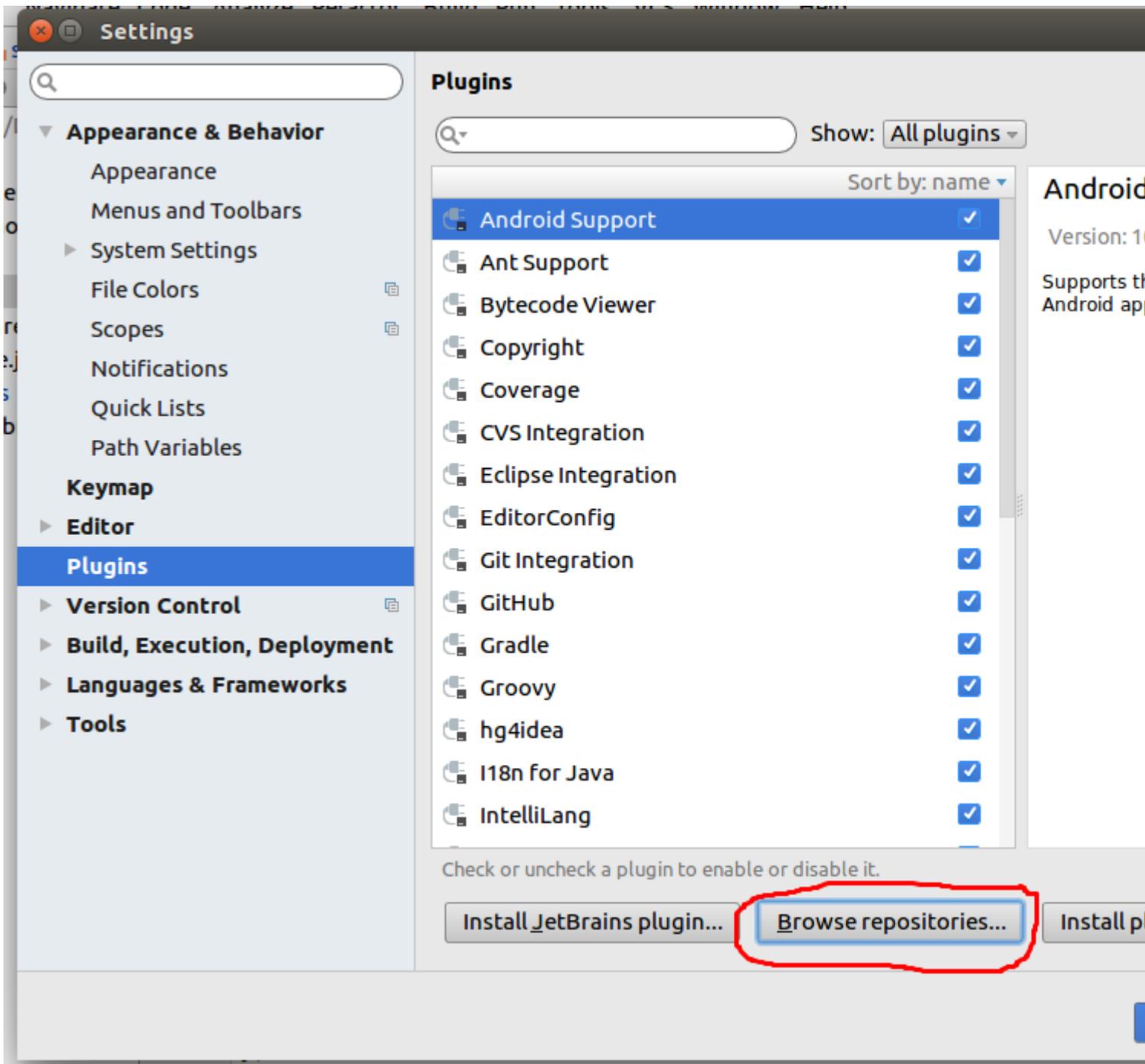
Pour installer un plugin depuis le disque

1. Go to File --> Settings (e.g. Ctrl+Alt+S).
 2. In the left-hand pane, select Plugins.
 3. On the Plugins window, click "Install plugin from disk button".
 4. Select the desired plugin from your local machine.
- Click Apply button of the Settings/Preferences dialog.



Ou vous pouvez directement ajouter des pugins des autres fournisseurs en recherchant directement les référentiels.

1. Go to "Browse Repositories"
2. Select the category (on the top of the window) that you need to search (or just search by the name if you know it).
3. Install it.



Lire Comment installer des plugins en ligne: <https://riptutorial.com/fr/intellij-idea/topic/8069/comment-installer-des-plugins>

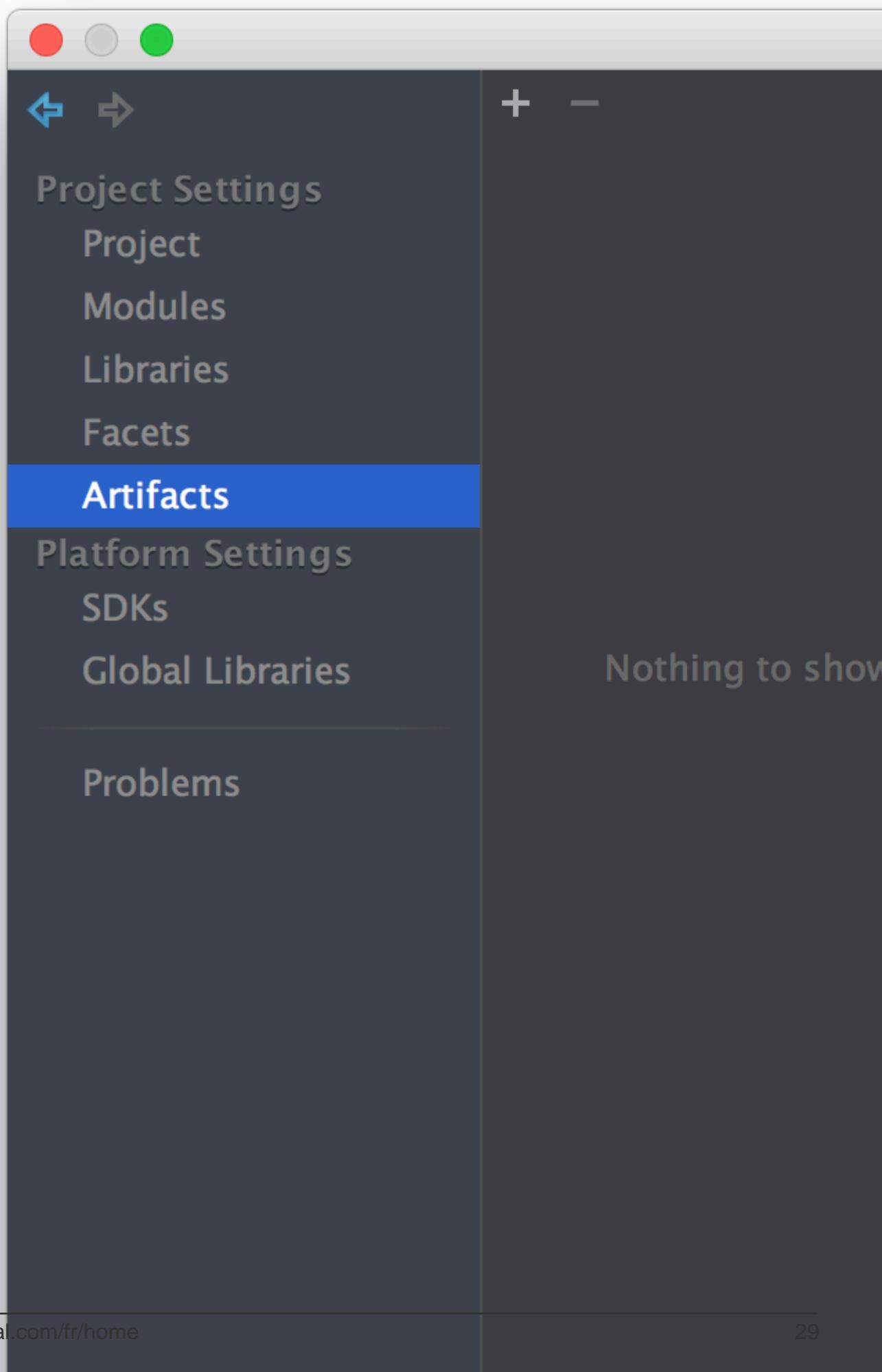
Chapitre 4: Exportation

Exemples

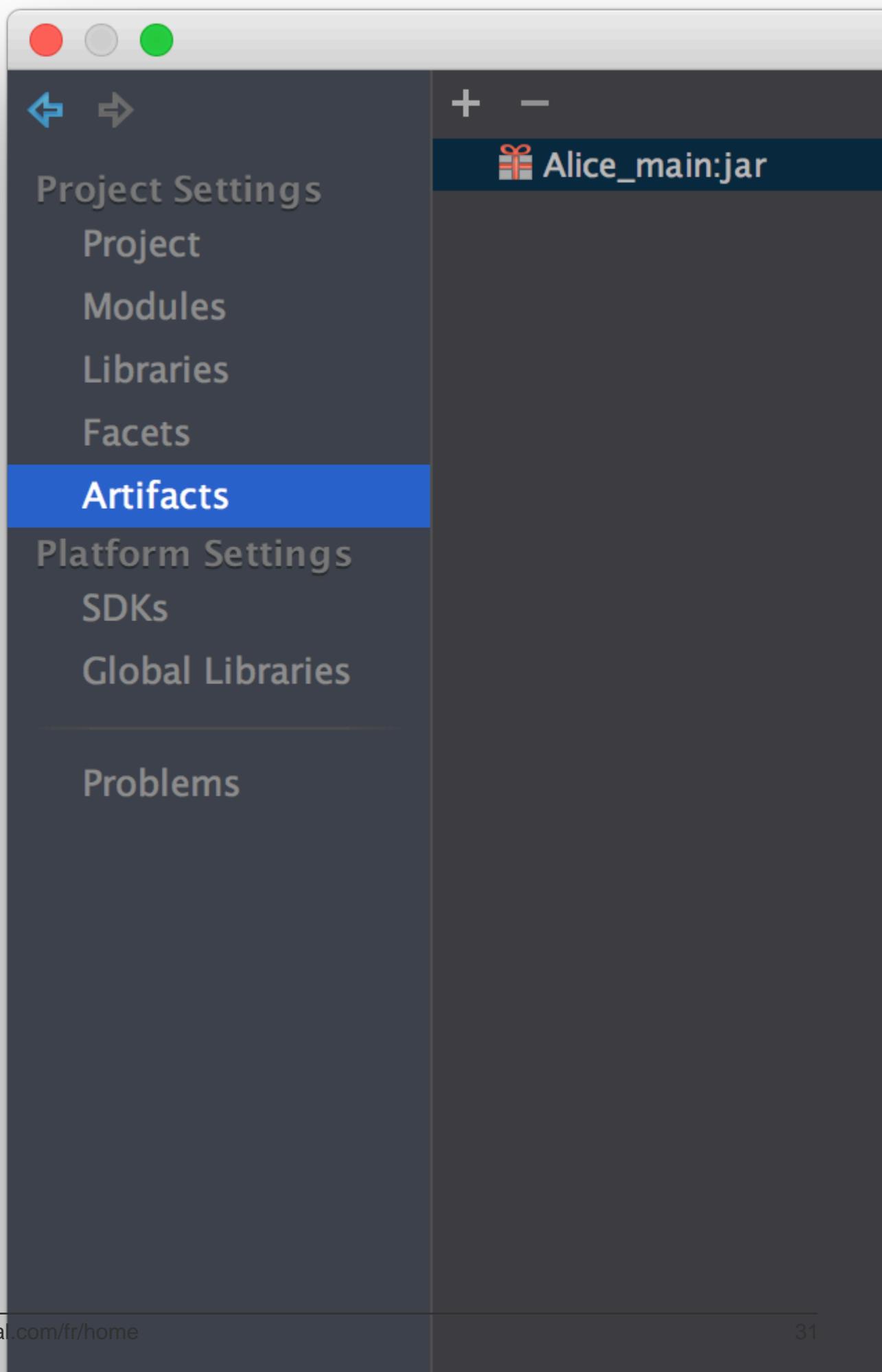
Construire un .jar

Finalement, lorsque vous êtes prêt à publier une version de votre code en production, vous aurez besoin d'un fichier `.jar` à distribuer. IntelliJ facilite la construction de fichiers JAR.

Tout d'abord, accédez à `File -> Project Structure` et cliquez sur `Artifacts` :



, vérifiez que toutes les informations relatives aux dépendances sont correctes, puis cliquez sur **OK** pour terminer la configuration de l'artefact.



Chapitre 5: Hibernate HQL console et inspections

Introduction

Intellij IDEA prend en charge l'achèvement automatique de HQL et l'exécution des requêtes HQL sur la console. Voici comment activer ce support.

Exemples

Configuration des inspections HQL

1. Allez dans Fichier -> Structure du projet -> Modules.
2. Ajouter un nouveau module Hibernate.
3. Faites un clic droit sur le module désiré -> Ajouter -> Hibernate.
4. Sélectionnez l'option de configuration Hibernate nouvellement créée et cliquez sur le signe (+) dans le volet droit pour créer le fichier hibernate.cfg.xml.
5. Allez dans Fichier -> Structure du projet -> Facettes et ajoutez un nouveau JPA.
6. Sélectionnez l'option de configuration JPA nouvellement créée, puis cliquez sur le signe (+) dans le volet droit pour lui attribuer votre fichier de configuration Hibernate.
7. Ouvrez la fenêtre Persistance, vous devriez voir la liste de vos modules de projet.
8. Développez le nom du module et attribuez votre source de données au fichier hibernate.cfg.xml.

Vous pouvez maintenant écrire des requêtes sur la console Hibernate et obtenir l'auto-complétion HQL.

Lire [Hibernate HQL console et inspections en ligne](https://riptutorial.com/fr/intellij-idea/topic/9617/hibernate-hql-console-et-inspections): <https://riptutorial.com/fr/intellij-idea/topic/9617/hibernate-hql-console-et-inspections>

Chapitre 6: IdeaVim

Remarques

IdeaVim est un [plugin pour les produits IDEA](#) qui vise à fournir des fonctionnalités Vim dans les vues de l'éditeur

Exemples

Affichage des numéros de ligne

À partir de la version 2016.2 d'IntelliJ IDEA et de la version 0.46 IdeaVim, l'option native d'IntelliJ pour afficher les numéros de ligne est inefficace. Lorsque vous cliquez sur *Afficher les numéros de ligne*, les numéros de ligne apparaissent et disparaissent immédiatement.

Ce problème est dû à un bogue dans le plug-in IdeaVim, qui peut être résolu en utilisant la commande Vim pour afficher les numéros de ligne:

```
:set number
```

et

```
:set nonumber
```

cachez.

Ces commandes peuvent également être utilisées comme raccourci `:set nu` et `:set nonu`.

Si vous souhaitez activer la fonction qui affiche les numéros de ligne relatifs, vous pouvez utiliser

```
:set relativenumber
```

ou un raccourci `:set rnu`. Rappelez-vous que vous pouvez mélanger `set relativenumber` avec le `set number`.

Allouer des frappes au clavier contradictoires à IdeaVim

Par défaut, certaines séquences de touches utiles dans Vim sont en contradiction avec les frappes de touches d'IntelliJ.

Par exemple, `^R` dans Vim est 'refaire', mais dans IntelliJ c'est le raccourci pour `Run`

Pour décider quel programme interprète la frappe, accédez à `Preferences -> Other Settings -> Vim Emulation` et choisissez les séquences de touches à utiliser avec IdeaVim et à utiliser avec IntelliJ:



- ▶ Appearance & Behavior
 - Keymap
- ▶ Editor
 - Plugins
- ▶ Version Control
- ▶ Build, Execution, Deployment
- ▶ Languages & Frameworks
- ▶ Tools
- ▼ Other Settings

Vim Emulation

Other Settings > Vi

Shortcut Conflicts for Vi

Shortcut
^2
^↑2
^↑6
^G
^H
^M
^N
^O
^V

Chapitre 7: Inspections

Introduction

IntelliJ IDEA fournit de nombreuses inspections de code, ce qui peut considérablement simplifier l'écriture de code.

Les paramètres d'inspection peuvent être trouvés dans `Preferences | Editor | Section des Inspections`. Par défaut, IDEA en a beaucoup activés. Et de nombreuses inspections prennent en charge les options de réparation automatique, que vous pouvez voir en appuyant sur `Alt + Entrée`.

Pour effectuer des inspections pour l'ensemble de votre projet (ou une étendue personnalisée), vous devez sélectionner `Analyze | Inspect code`.

Exemples

@NotNull / @Nullable inspections

Ces inspections sont extrêmement utiles pour empêcher les `NullPointerException`s. Par défaut, ils sont désactivés. Vous pouvez trouver ces inspections dans les préférences d' `Inspections : Java | Probable bugs | Constant conditions & exceptions` et `@NotNull/@Nullable problems`. Vous pouvez également configurer vos annotations. Vous pouvez utiliser [ce manuel](#) pour ajouter des annotations JetBrains à votre projet.

Par exemple, considérez cette méthode:

```
public int getLength() {
    return getString().length();
}

public String getString() {
    return "not null string";
}
```

Si `getString` ne peut pas retourner `null`, tout va bien. Mais si nous activons nos inspections et dans certains cas, elles peuvent retourner nulles, nous verrons immédiatement une inspection déclenchée:

```
public int getLength() {
    return getString().length();
}

public String getString() {
    if (condition()) {
        return null;
    }
    return "not null string";
}
```

qui dit 'null' is returned by the method which is not declared as `@Nullable`. Et si nous frappons `Alt + Entrée`, il y aura une option `Annotate method as '@Nullable'`. Si nous frappons à nouveau sur

Entrée, notre code ressemblera à ceci:

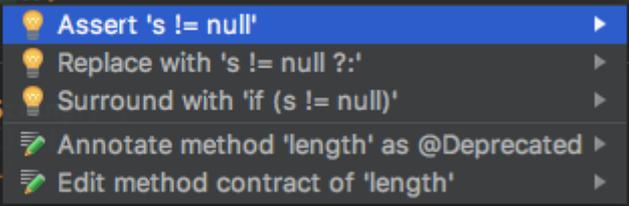
```
public int getLength() {
    return getString().length();
}

@Nullable
public String getString() {
    if (condition()) {
        return null;
    }
    return "not null string";
}
```

avec l'inspection déclenchée sur la méthode `length()` méthode `Method invocation 'length' may produce 'java.lang.NullPointerException'`. Et si nous allons plus loin et introduisons le résultat de la méthode `getString()` tant que variable, après avoir appuyé sur `Alt + Entrée`, IDEA vous proposera quelques méthodes pour corriger cette inspection:

```
public class MethodContainer {
    public int getLength() {
        final String s = getString();
        return s.length();
    }
}

@Nullable
public String getString() {
    if (condition()) {
        return null;
    }
    return "not null string";
}
```



De cette façon, vous pouvez inspecter votre code à la volée et corriger toutes les `NullPointerException`s potentielles. Si vous souhaitez vérifier l'ensemble de votre projet (ou une portée aléatoire), vous pouvez utiliser `Analyze | Inspect code`. Assurez-vous simplement que toutes les inspections nécessaires sont activées dans votre profil de contrôle sélectionné.

Lire Inspections en ligne: <https://riptutorial.com/fr/intellij-idea/topic/9361/inspections>

Chapitre 8: Modèles en direct

Exemples

Ajouter facilement une méthode de test

```
@org.junit.Test
public void should_$name$() {
    $END$
}
```

Assurez-vous de cocher la case **Noms FQ abrégés** lors de la création de ce modèle.

Abbreviation:

Descript

Template text:

```
@org.junit.Test
public void should_$name$() {
    $END$
}
```

Applicable in Java: declaration. [Change](#)

Lorsque vous tapez "should" (l'abréviation), cela ajoutera le `import org.junit.Test;` nécessaire `import org.junit.Test;` déclaration en haut du fichier, et ce code:

```
@Test
public void should_() {
}
```

C'est grâce à l'option des **noms Shorten FQ** que `@org.junit.Test` est réduit à simplement `@Test`.

La variable `$name$` n'est pas pertinente, elle pourrait être nommée autre chose. Le but de cette variable est que lorsque le modèle est inséré dans la classe, le curseur sera placé dans la position de `$name$`, vous demandant d'entrer quelque chose.

Après avoir entré une valeur pour `$name$` (en fait le nom de la méthode de test), le curseur passera enfin à `END`, une variable intégrée, pour pouvoir continuer à implémenter le scénario de test.

Insérer le nom de la classe en cours

Considérons le *modèle de classe utilitaire* : une classe avec uniquement `static` méthodes `static` et aucun champ. Il est recommandé d'empêcher l'instanciation de telles classes en ajoutant un constructeur privé.

Cet exemple de modèle live facilite l'ajout d'un constructeur privé à une classe existante, en utilisant le nom de la classe englobante.

```
private $className$() {  
    throw new AssertionError("utility class, forbidden constructor");  
}
```

Applicable en Java: portée de la déclaration.

Abbreviation:

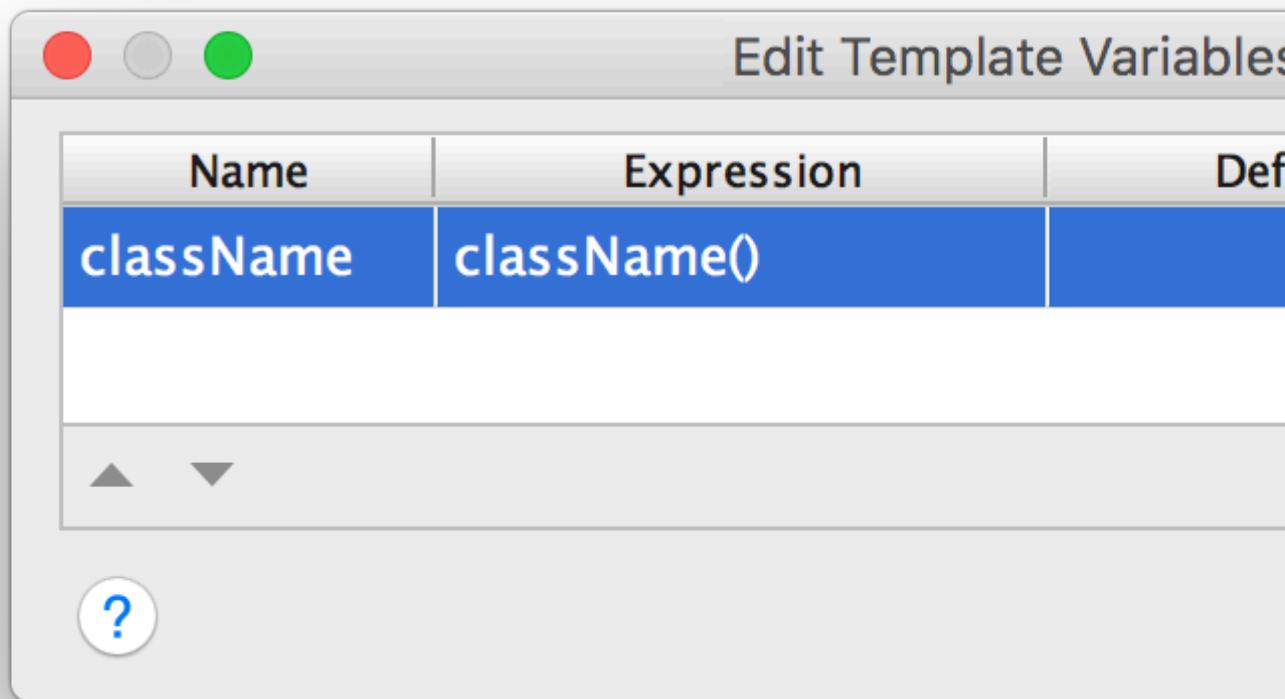
Desc

Template text:

```
private $className$() {  
    throw new AssertionError("utility class, forbidden constructor");  
}
```

Applicable in Java: declaration. [Change](#)

Cliquez sur **Modifier les variables** pour définir la variable `className` tant `className()`, puis cochez la case **Ignorer si définie** pour éviter de demander un nom personnalisé, inutile dans cet exemple.



Par exemple, dans une classe comme celle-ci:

```
class ListUtils {  
    // ...  
}
```

Lorsque vous tapez "utility_class" (l'abréviation), cela va insérer un constructeur comme ceci:

```
class ListUtils {  
    private ListUtils() {  
        throw new AssertionError("utility class, forbidden constructor");  
    }  
  
    // ...  
}
```

Lire Modèles en direct en ligne: <https://riptutorial.com/fr/intellij-idea/topic/2703/modeles-en-direct>

Chapitre 9: Optimisation

Exemples

Personnalisation des options de la machine virtuelle

Vous pouvez remplacer les `vmoptions` par défaut par vos propres paramètres personnels en choisissant **Aide > Modifier les options de machine virtuelle personnalisées** dans la barre d'outils. Cela créera une copie locale du fichier que vous êtes libre de modifier.

Par exemple, si vous doublez la valeur définie pour `xmx`, la taille maximale du pool d'allocation de mémoire sera doublée après le redémarrage de l'EDI. Sur de nombreuses machines, cela se traduira par des performances plus rapides.

Voir les options [ici](#) pour une description de chaque paramètre.

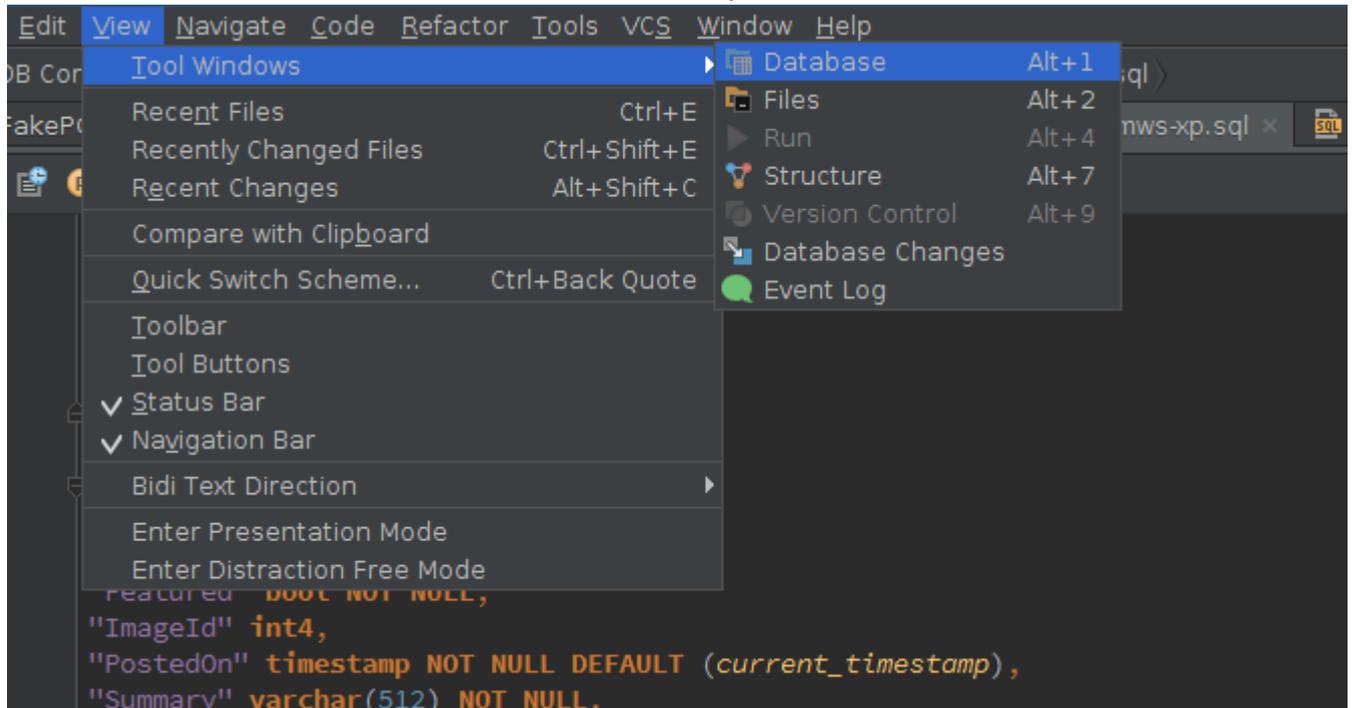
Lire Optimisation en ligne: <https://riptutorial.com/fr/intellij-idea/topic/5305/optimisation>

Chapitre 10: Outils de base de données

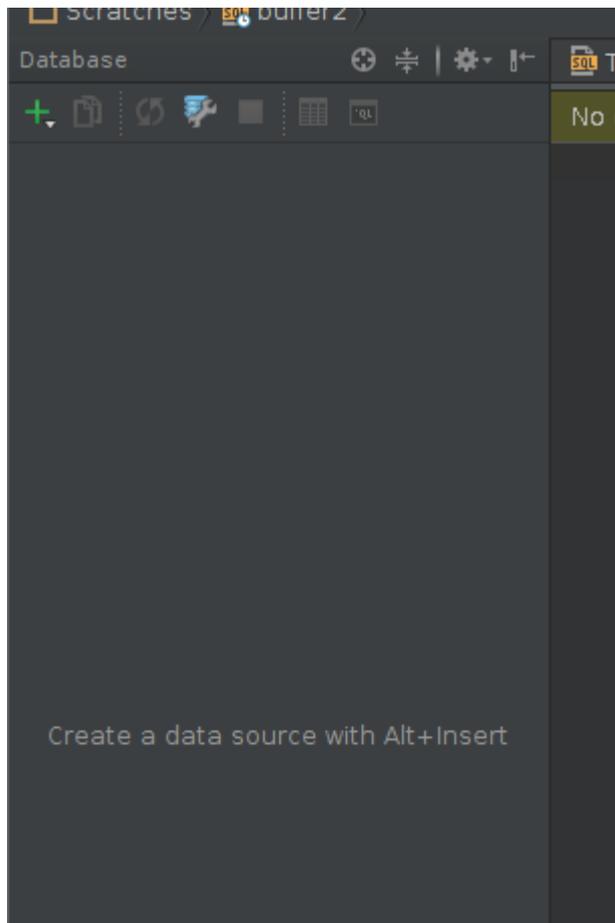
Exemples

Créer une nouvelle source de données

1. Ouvrez "Database Tool Window" si vous ne l'avez pas ouvert:

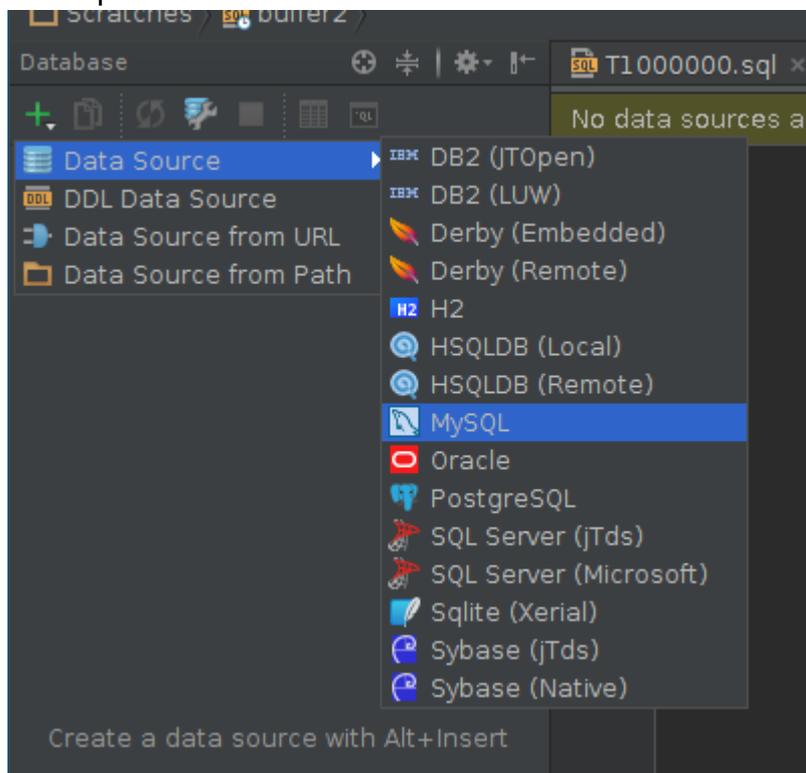


A quoi ça ressemble:

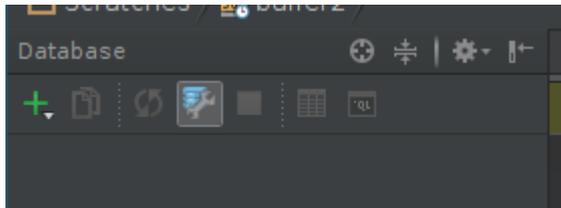


2. Créer une nouvelle source de données:

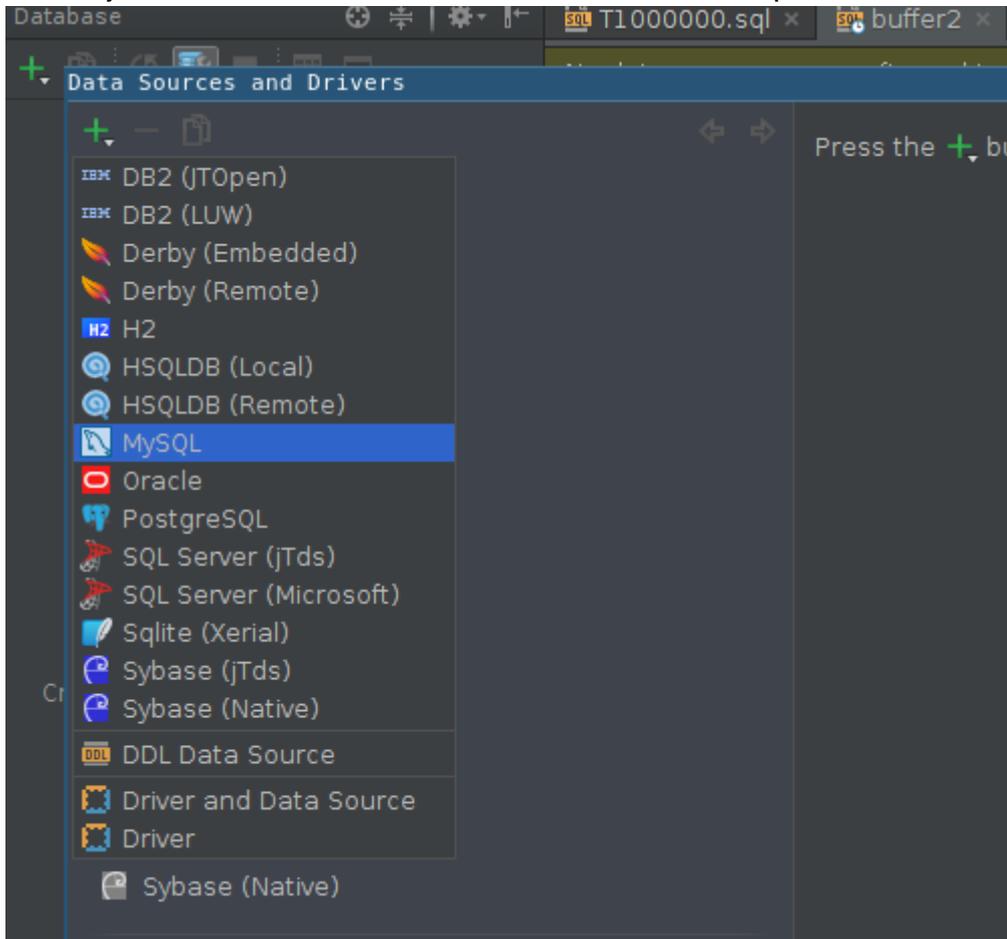
- En cliquant sur l'icône "+":



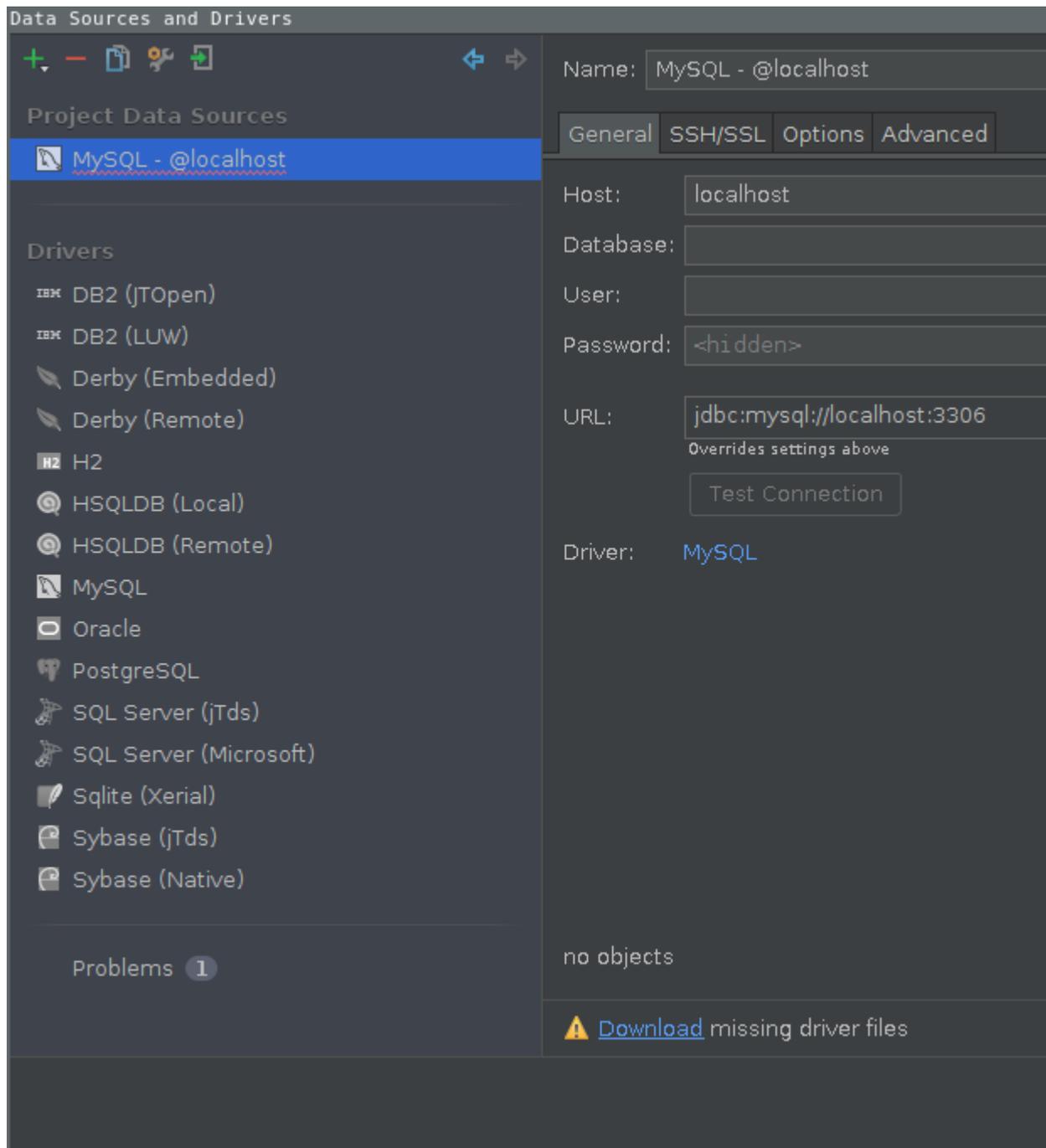
- Ou en ouvrant la boîte de dialogue "Sources de données et pilotes"



Et en ajoutant une nouvelle source de données en cliquant sur "+"



3. Si vous n'avez pas encore téléchargé le pilote JDBC, vous serez invité à le faire:



4. Entrez les paramètres de connexion:

Name: MySQL 5.7 - docker

General SSH/SSL Options Advanced

Host: localhost Port: 33057

Database: guest

User: guest

Password: Save on disk with master password protection

URL: jdbc:mysql://localhost:33057/guest default
Overrides settings above

Driver: MySQL

5. Configurez le tunnel SSH si vous en avez besoin:

Name: MySQL 5.7 - docker

General SSH/SSL Options Advanced

Use SSH tunnel [Copy from...](#)

Proxy host: 192.168.1.102 Port: 25

Proxy user: kassak

Auth type: Key pair (OpenSSH) ▼

Private key file: /home/user/.ssh/id_rsa

Passphrase: Save on disk with master password protection

Use SSL

6. Configurez SSL si vous en avez besoin:

Name:

General **SSH/SSL** Options Advanced

Use SSH tunnel Copy from...

Proxy host: Port:

Proxy user:

Auth type:

Private key file:

Password: Save on disk with master password protection

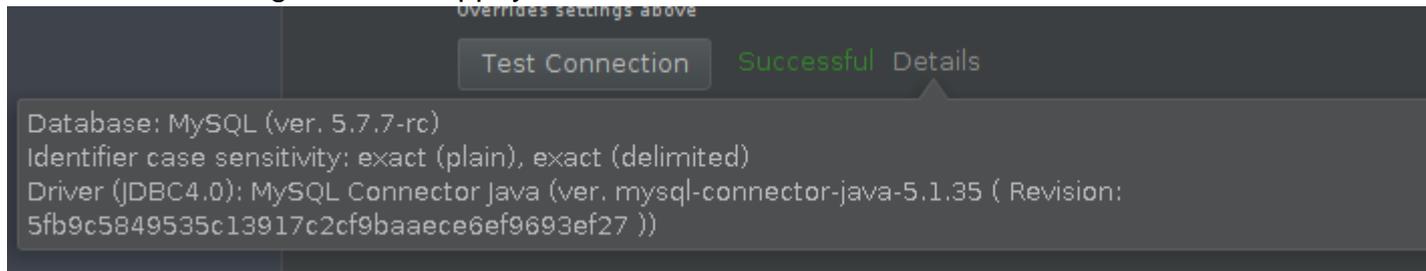
Use SSL Copy from...

CA file:

Client certificate file:

Client key file:

7. Vérifiez cette configuration en appuyant sur "Test Connection":



Lire Outils de base de données en ligne: <https://riptutorial.com/fr/intellij-idea/topic/6275/outils-de-base-de-donnees>

Chapitre 11: Raccourcis utiles

Exemples

Compiler et exécuter

Faire un projet (compiler modifié et dépendant)

Windows: `Ctrl + F9`

OS X / macOS: `Cmd + F9`

Compiler le fichier, le package ou le module sélectionné

C'est utile à savoir, car lors du débogage, ce raccourci peut être utilisé pour recharger rapidement les classes / hotswap.

Windows: `Ctrl + Maj + F9`

OS X / macOS: `Cmd + Shift + F9`

Sélectionnez la configuration et exécutez

Windows: `Alt + Maj + F10`

OS X / macOS: `Option + Maj + F10`

Sélectionnez la configuration et le débogage

Windows: `Alt + Maj + F9`

OS X / macOS: `Option + Maj + F9`

Courir

Maj + F10

Déboguer

Maj + F9

Exécuter la configuration du contexte à partir de l'éditeur

Windows: Ctrl + Maj + F10

OS X / macOS: Cmd + Shift + F10

Achèvement du code

Achèvement du code de base (le nom de toute classe, méthode ou variable)

Windows: Ctrl + Espace

OS X / macOS: Cmd + Espace

Achèvement du code intelligent (filtre la liste des méthodes et des variables par type attendu)

Windows: Ctrl + Maj + Espace

OS X / macOS: Cmd + Maj + Espace

Remplacer le code par une suggestion

Languette

Ajout de code à partir d'une suggestion d'achèvement

Entrer

Rechercher / Remplacer

Rechercher partout

Double décalage

Trouver

Windows / Linux: `Ctrl + F`

OS X / macOS: `Cmd + F`

Trouver le prochain

F3

Trouver précédent

Maj + F3

Remplacer

Windows / Linux: `Ctrl + R`

OS X / macOS: `Cmd + R`

Trouver dans le chemin

Windows / Linux: `Ctrl + Shift + F`

OS X / macOS: `Cmd + Shift + F`

Remplacer dans le chemin

Windows / Linux: `Ctrl + Maj + R`

OS X / macOS: `Cmd + Shift + R`

Refactoring

Copie

F5

Bouge toi

F6

Supprimer en toute sécurité

Windows / Linux: `Alt + Suppr`

OS X / macOS: `Cmd + Supprimer`

Notez que la clé de `suppression` sur OS X / macOS est l'équivalent de la touche `Retour arrière` sur les autres systèmes d'exploitation.

Renommer

Maj + F6

Méthode d'extraction

Windows / Linux: `Ctrl + Alt + M`

OS X / macOS: `Cmd + Option + M`

Extraire le champ

Windows / Linux: `Ctrl + Alt + F`

OS X / macOS: `Cmd + Option + F`

Extraire la variable

Windows / Linux: `Ctrl + Alt + v`

OS X / macOS: `Cmd + Option + v`

Extraire la constante

Windows / Linux: `Ctrl + Alt + c`

OS X / macOS: `Cmd + Option + c`

Extraire le paramètre

Windows / Linux: `Ctrl + Alt + p`

OS X / macOS: `Cmd + Option + p`

Autre

Surround avec

Entoure un bloc de code avec un `if`, `for`, `<editor-fold ...>` et plus.

Windows / Linux: `Ctrl + Alt + T`

OS X / macOS: `Cmd + Option + T`

Navigation de base

Aller à l'éditeur (depuis la fenêtre de l'outil)

Esc

Mise au point sur la fenêtre d'outils correspondante

Windows: **Alt + <numéro de fenêtre d'outil>**

OS X / macOS: **Cmd + <numéro de la fenêtre de l'outil>**

Par exemple, basculer le focus sur la fenêtre du projet

Windows: **Alt + 1**

OS X / macOS: **Cmd + 1**

Fichiers récents popup

Windows: **Ctrl + E**

OS X / macOS: **Cmd + E**

Trouver une action

Windows: **Ctrl + Maj + A**

OS X / macOS: **Cmd + Shift + A**

Aller vers

- **Fichier :**

- **Windows:** `Ctrl + Maj + N`
- **OS X / macOS:** `Cmd + Shift + N`

Classe:

- ◦ **Windows:** `Ctrl + N`
- **OS X / macOS:** `Cmd + N`

Symbole (classe / méthode / variable / nom de la constante):

- ◦ **Windows:** `Ctrl + Alt + Maj + N`
- **OS X / macOS:** `Cmd + Option + Maj + N`

Notez que vous pouvez utiliser le nom de classe pour restreindre la méthode / variable / recherche constante, par exemple pour trouver le symbole `usersCollection` dans la classe `type UserDao` :

```
UserDAO.usersCollection
```

Partout :

- ◦ **Windows:** `Maj + Maj`
- **OS X / macOS:** `Maj + Maj`

Pour rechercher quelque chose qui a plusieurs mots, par exemple, `InetAddressCachePolicy` vous pouvez simplement taper `InAddCacPo` ou quelque chose de similaire qui contient des parties de mots dans le nom entier.

Aller au numéro de ligne

Windows: `Ctrl + G`

OS X / macOS: `Cmd + L`

Revenir au dernier emplacement d'édition

Windows: `Ctrl + Maj + Retour arrière`

OS X / macOS: `Cmd + Maj + Retour arrière`

Recherche d'utilisation

Rechercher des utilisations / Rechercher des utilisations dans un fichier

Windows / Linux: `Alt + F7 / Ctrl + F7`

OS X / macOS: `Option + F7 / Ctrl + F7`

Mettez en surbrillance les utilisations dans un fichier

Windows / Linux: `Ctrl + Maj + F7`

OS X / macOS: `Cmd + Shift + F7`

Afficher les usages

Windows / Linux: `Ctrl + Alt + F7`

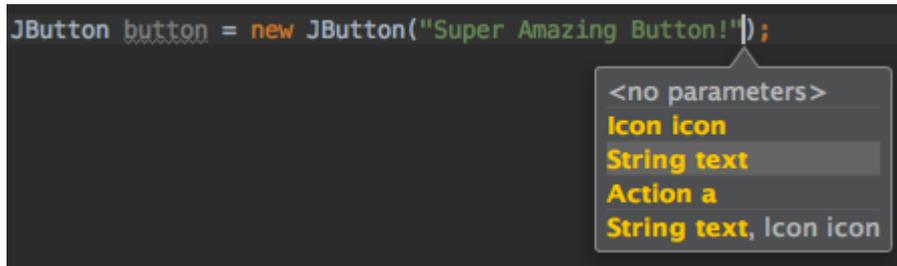
OS X / macOS: `Cmd + Option + F7`

Afficher les paramètres de la méthode

Windows / Linux: `Ctrl + P`

OS X / macOS: `Cmd + P`

Indique quels paramètres une méthode et toutes ses surcharges acceptent.



Sélection

Sélection avec portée croissante

Cela est pratique lorsque vous voulez sélectionner un bloc pour extraire une variable / méthode, pas besoin de faire un couplage précis, placez simplement le curseur quelque part dans la déclaration et continuez

Windows: `Ctrl + W`

OS X / macOS: `Cmd + W`

Sélection à portée décroissante

Windows: `Ctrl + Maj + W`

OS X / macOS: `Cmd + Shift + W`

Cette fonctionnalité est également très utile lors de l'édition / lecture avec des documents json dans votre IDE.

Sélection verticale

Appuyez et maintenez

Windows: `Alt`

OS X / macOS: `Opt`

et sélectionnez normalement en utilisant la souris / trackpad (la façon dont vous sélectionnez un mot dans une ligne, etc.)

Voilà comment cela devrait ressembler

```
...jsons": "~3",  
"angular": "1.5.7",  
"angular-animate": "1.5.7",  
"angular-route": "1.5.7",  
"angular-ui-router": "~0.2.11",  
"angular-resource": "1.5.7",  
"angular-translate": "~2",  
"angular-bootstrap": "~0.11",  
"angular-timer": "1.3.3",  
"angular-cookies": "~1.5.7",
```

Carets multiples

Appuyez et maintenez

Windows: `Alt + Maj`

OS X / macOS: `Opt + Shift`

et cliquez où tout ce que vous voulez mettre un caret. Vous pouvez choisir de placer plusieurs carets sur une seule ligne ou sur plusieurs lignes à différentes positions.

Vous pouvez maintenant effectuer toutes les opérations que vous auriez pu effectuer sur un seul mot sélectionné (maintenez la touche `Ctrl` (Windows) ou l' `option` (Mac OS) et utilisez les touches `Gauche` ou `Droite` pour passer d'un mot à l'autre). .

Vous pouvez même couper / coller plusieurs sélections d'un endroit à un autre.

```
"angular": "1.5.7",  
"angular-animate": "1.5.7",  
"angular-route": "1.5.7",  
"angular-ui-router": "~0.2.11",  
"angular-resource": "1.5.7",  
"angular-translate": "~2",  
"angular-bootstrap": "~0.11",  
"angular-timer": "1.3.3",
```

Avoir plusieurs carets est très utile lorsque vous voulez changer la structure du texte sur plusieurs lignes / plusieurs positions sur la même ligne.

Sélection des occurrences en double

Sélectionnez du texte et appuyez sur

Windows: `Alt + J`

OS X / macOS: `ctrl + G`

pour sélectionner la prochaine occurrence du même texte.

Vous obtenez un curseur à chaque occurrence sélectionnée qui pourrait être utilisée pour modifier chaque occurrence simultanément.

Par exemple, j'ai essayé de donner un exemple dans ce gif, j'espère que cela vous aidera

```
{  
  "keys": ["name1", "name2", "name3"]  
}
```

Lire Raccourcis utiles en ligne: <https://riptutorial.com/fr/intellij-idea/topic/3085/raccourcis-utiles>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec intellij-idea	Community , CraftedCart , foxtrot9 , Harmelodic , Jeeter
2	Caractéristiques peu connues	Anton Dozortsev , Jeeter , mszymborski , Prateek
3	Comment installer des plugins	Priya , RamenChef , Sudeepa Nadeeshan
4	Exportation	Jeeter
5	Hibernate HQL console et inspections	Shai Givati
6	IdeaVim	Jeeter , Mateusz Piotrowski , mnoronha , sevenforce
7	Inspections	esin88
8	Modèles en direct	janos
9	Optimisation	Kevin Cooper
10	Outils de base de données	kassak
11	Raccourcis utiles	CraftedCart , karel , mszymborski , Nadim Bahadoor , Prateek