

# Introduction SPRING

**esprit**   
Se former autrement

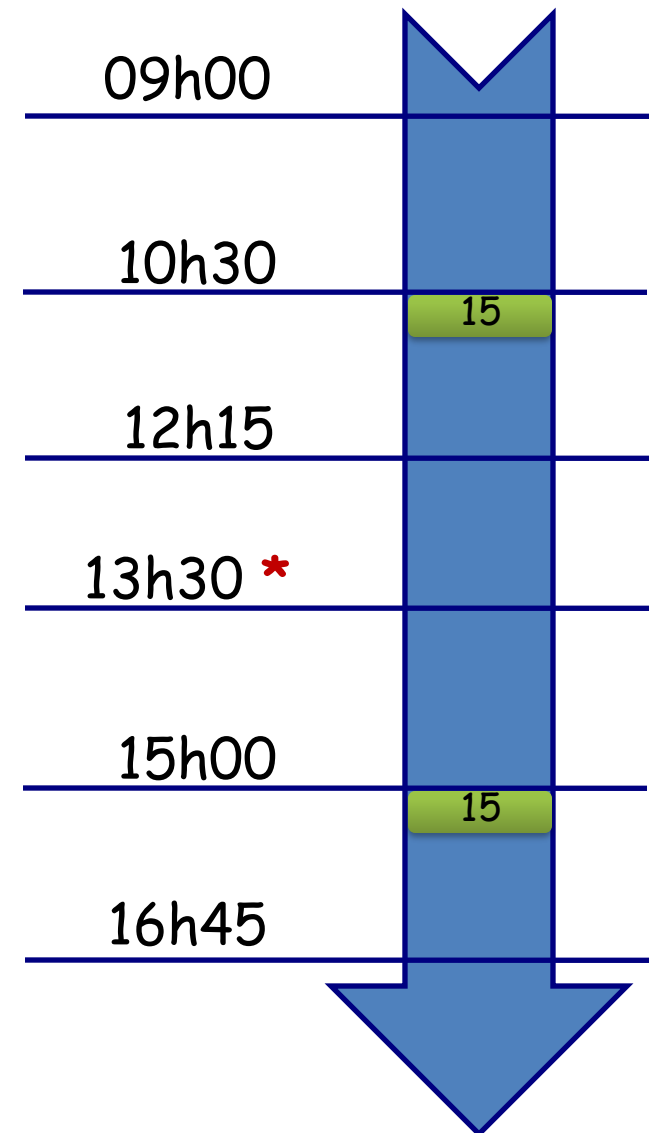


# Plan du Cours

- Introduction : Horaires, Evaluation
- Spring et le Marché de l'Emploi
- Contenu du Module **Spring**
- Historique Spring
- Concurrents Spring
- projets Spring
- Framework Spring
- Présentation et Installation des Outils : JDK, IntelliJ, MySQL

# Horaires

- Durée Totale : **42 heures**
- Séances : **14 séances**
- Cours : **12 heures**
- TP : **30 heures**
  
- Durée de chaque Séance : **3 heures**
- Durée de la Pause : **15 minutes**
  
- \* Vendredi : **13h45**



# Evaluation

- L'évaluation se fait tout au long du module, et non pas uniquement à la fin.
- La moyenne du module est calculée comme suit :  
**Moyenne = Note Contrôle Continu \* 40% + Note Examen \* 60%**
- Le **Contrôle Continu** prend en compte l'Assiduité, la Participation, les **TP** à faire en cours ou chez vous et aussi un **Examen Blanc Pratique**.
- Vous aurez le temps de faire tous vos **Homeworks en classe** (à la fin de chaque séance). Si vous êtes présents, vous n'aurez rien à faire chez vous.
- L'**Examen** sera pratique.

# Evaluation

- Exemple d'évaluation :

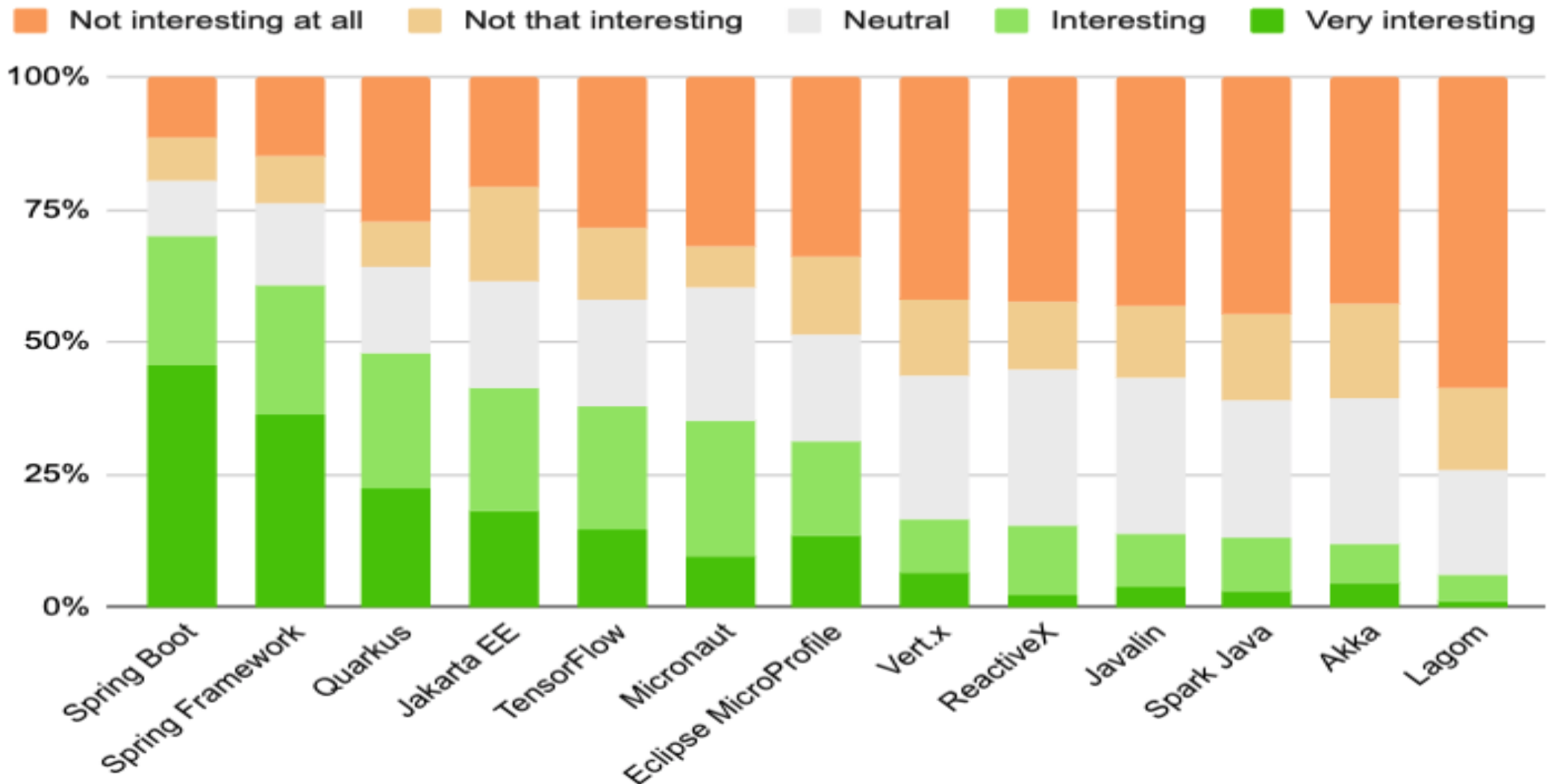
Note Participation (/6)	Homeworks (/12)	Bonus (/2)	Moy	Moy VF
4,222222222	11,5	2	17,72222222	18
6	11	2	19	19
5,111111111	10	2	17,11111111	17,5
0,888888889	6	1	7,88888889	8
1,777777778	7	1	9,77777778	10
0	10	0	10	10
6	12	2	20	20

# Spring et le Marché du Travail

- Les offres d'emploi « **Cherche Développeur Java** », en général cherchent un développeur dans une partie des technologies suivantes :
- Front-End : Angular, **Spring MVC**, VueJS, ReactJS ...
- Back-End : **Spring**, EJB, Hibernate, **Spring Data JPA**, Quarkus
- Outils : **Spring Boot**, **Maven**, GIT, Jenkins, Sonar, ...
- Base de Données : PostgreSQL, Oracle, **MySQL**, ...
- Méthode Agile : Scrum, XP (Extreme Programming)
- <https://www.optioncarriere.tn/jobad/tn21f84970d5c87b7eea3c329f6ea274d9>

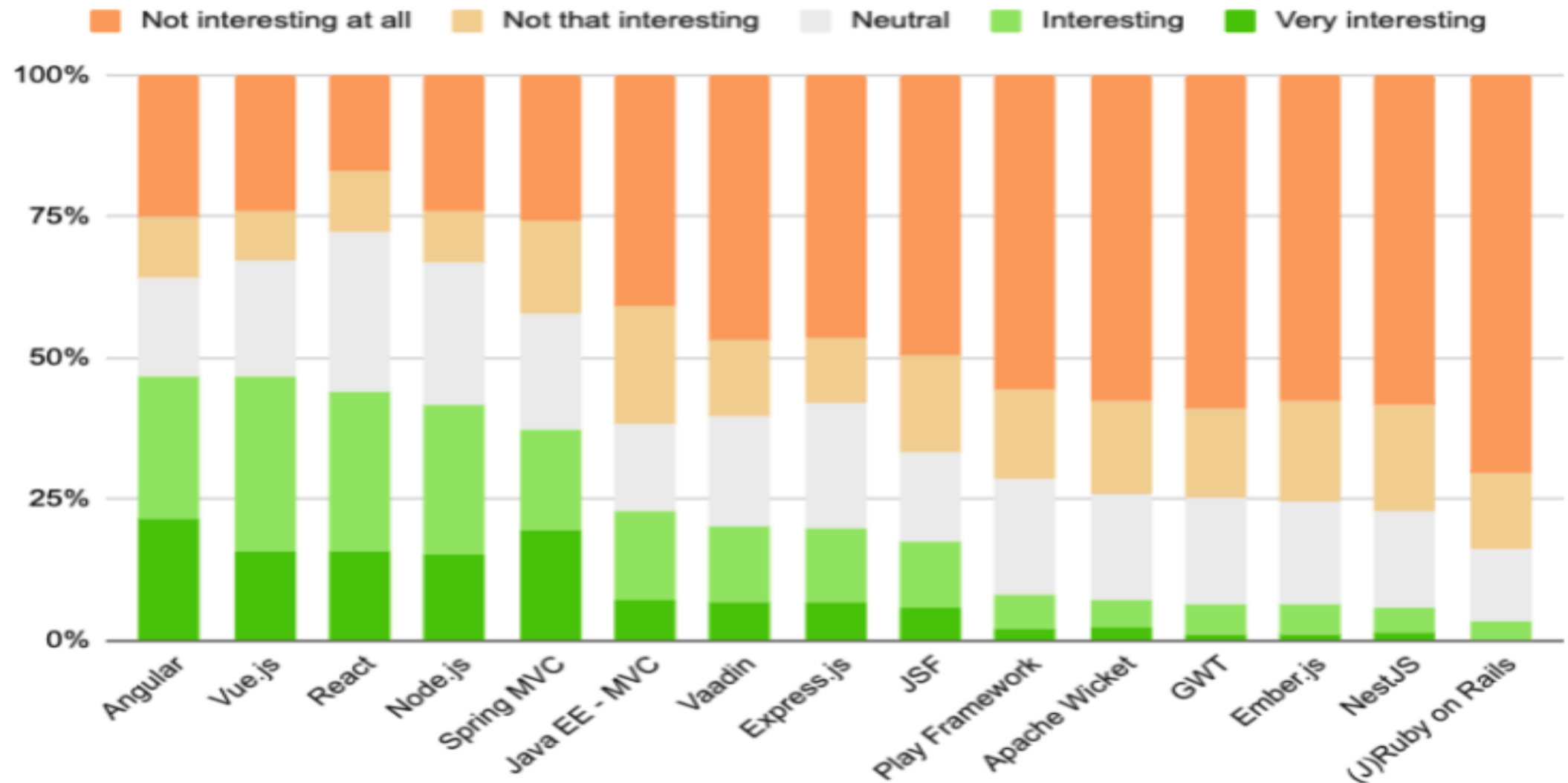
# Spring et le Marché de l'Emploi

## Application Frameworks



# Spring et le Marché de l'Emploi

## Web Frameworks





# Contenu du Module Spring

- Introduction à **Spring** et Mise en place de l'Environnement
- **Spring Boot /Maven**
- **Spring Data JPA** (Première Entité - Associations - CRUD / JPA Repository – JPQL)
- **Lombok**
- **IoC & DI** (Injection des dépendances)
- **Spring MVC REST ( Postman+ Spring Doc)**
- **TP étude de cas**
- **Spring Scheduler**
- **Spring Batch**
- **AOP** (Aspect Oriented Programmation)
- **Examen Blanc Pratique** (Spring Boot - REST - Spring Data – AOP - etc..)

# Bref Historique JavaEE

- **1995** : Java est apparu, développé par **James Gosling** chez **Sun Microsystems**.
- **1997** : Les servlets arrivent pour la création des pages web dynamiques.
- **1999** : Les JSP arrivent pour **faciliter** la création des pages web dynamiques et résoudre les problèmes de servlets.
- Les JSP séparent les contrôleurs de la partie présentation.
- La question qui se pose, comment les servlets/JSP vont assurer, la logique métier et la persistance ?
- **Toujours en 1999** : J2EE vient comme réponse à la question précédente (transactions, sécurité, messaging ...) en introduisant les **EJB**.  
EJB promettait sécurité et disponibilité mais la réalité était tout autre ( couplage fort avec le serveur d'application, temps de configuration considérable...).

# Bref Historique JavaEE

- 1999 : **J2EE** (Java 2 Enterprise Edition) depuis la version 1.**2** et jusqu'à la version 1.4 en 2003).
- 2006 : **JavaEE** 5 ou **JEE** (Java Enterprise Edition) depuis la version 5 (Ajout des annotations)
- 2010 : Sun a été rachetée par Oracle en 2010 (Java est à la base une propriété de **Sun** Microsystems )  
Oracle a cédé la plateforme JavaEE à **Eclipse Foundation**, mais en exigeant le changement de son nom
- 2017 : Java EE 8 (version obsolète)
- 2019 : Jakarta EE 8 (Nouveau nom JakartaEE à la place de JavaEE )

# Historique Spring

- **2002** : **Rod Johnson** publie son livre «Expert One-on-One J2EE Design and Development», dans lequel il propose du code, qui va devenir plus tard le Framework Spring
- **2004** : Rod Johnson publie son livre «**J2EE Development without EJB**».
- **2004** : Spring 1.0, licence Apache 2.0
- **2005** : Spring devient populaire, en particulier en réaction par rapport aux EJB 2.x très complexe
- **2006** : Spring 2.0 ( Introduction de l'injection de dépendances )
- **2007** : Spring 2.5, avec support des **annotations**
- **2009** : Spring 3.0
- **2013** : Spring 4.0
- **2017** : Spring 5.x,
- **2022** : Spring 5.3.22 (version courante )

# Concurrents Spring

**Jakarta EE**

**vs.**

**Spring**

**vs.**

**Quarkus**



# Concurrents Spring

## Jakarta EE

- Jakarta EE 8, comme son nom l'indique, est fonctionnellement identique à Java EE 8, publié par Oracle et possède les mêmes spécifications.
- Jakarta EE a été optimisé pour le **cloud computing**.
- Selon la fondation Eclipse, l'un des principaux objectifs de Jakarta EE est d'accélérer le développement d'applications d'entreprise pour le Cloud Computing (les applications cloud natives).



# Concurrents Spring

## Jakarta EE

- La plateforme **JakartaEE** est en pleine mutation et modification. Attendons qu'elle soit stable.
- La plateforme **JavaEE** est un ensemble de spécifications **JSF, EJB, JPA**. Elle n'est plus maintenue par Oracle, donc bientôt obsolète.
- Le cœur de cette plateforme est les EJB.
- Inconvénients des EJB :
  - Difficile à coder, il faut implémenter des interfaces spécifiques ...
  - Les tests unitaires sont difficiles à réaliser.
  - C'est une solution qui nécessite un serveur d'application ⇒ lourd et gourmand en ressources.

# Concurrents Spring

## Jakarta EE vs Spring

- Le choix d'une technologie dépend de plusieurs critères dont les fonctionnalités offertes et l'expérience du développeur (un développeur avec 10 ans d'expérience JavaEE te dira toujours que JakartaEE est le meilleur peut importe les concurrents (sentiment d'appartenance)).
- Les défenseurs de JakartaEE mentionneront que le développement de la plateforme n'est pas lié à une seule compagnie comme c'est le cas pour Spring. Il s'agit plutôt d'une communauté qui assure la longévité de leurs projets pour les années à venir.
- Les pro-Spring vous diront qu'au contraire Spring est plus fiable sur la durée et que le fait qu'il est très demandé sur le marché ( beaucoup plus que JakartaEE) est une raison évidente pour choisir ce Framework.



# Concurrents Spring

## Quarkus

Quarkus est un framework Java natif Kubernetes conçu pour les machines virtuelles Java (JVM) optimisant Java spécifiquement pour les conteneurs et lui permettant de devenir une plate-forme efficace pour les environnements sans serveur, cloud et Kubernetes.

Tout comme Spring, Quarkus a été conçu pour être facile à utiliser dès le départ, avec des fonctionnalités avec peu ou pas de configuration.



# Concurrents Spring

## Quarkus vs. Spring

Spring et Quarkus sont en concurrence sur les critères **du temps du lancement** et **l'occupation de la mémoire**.

**L'évolutivité** et l'architecture **sans serveur** sont aussi des critères pris en considération dans les dernières versions fournies par ces deux frameworks.

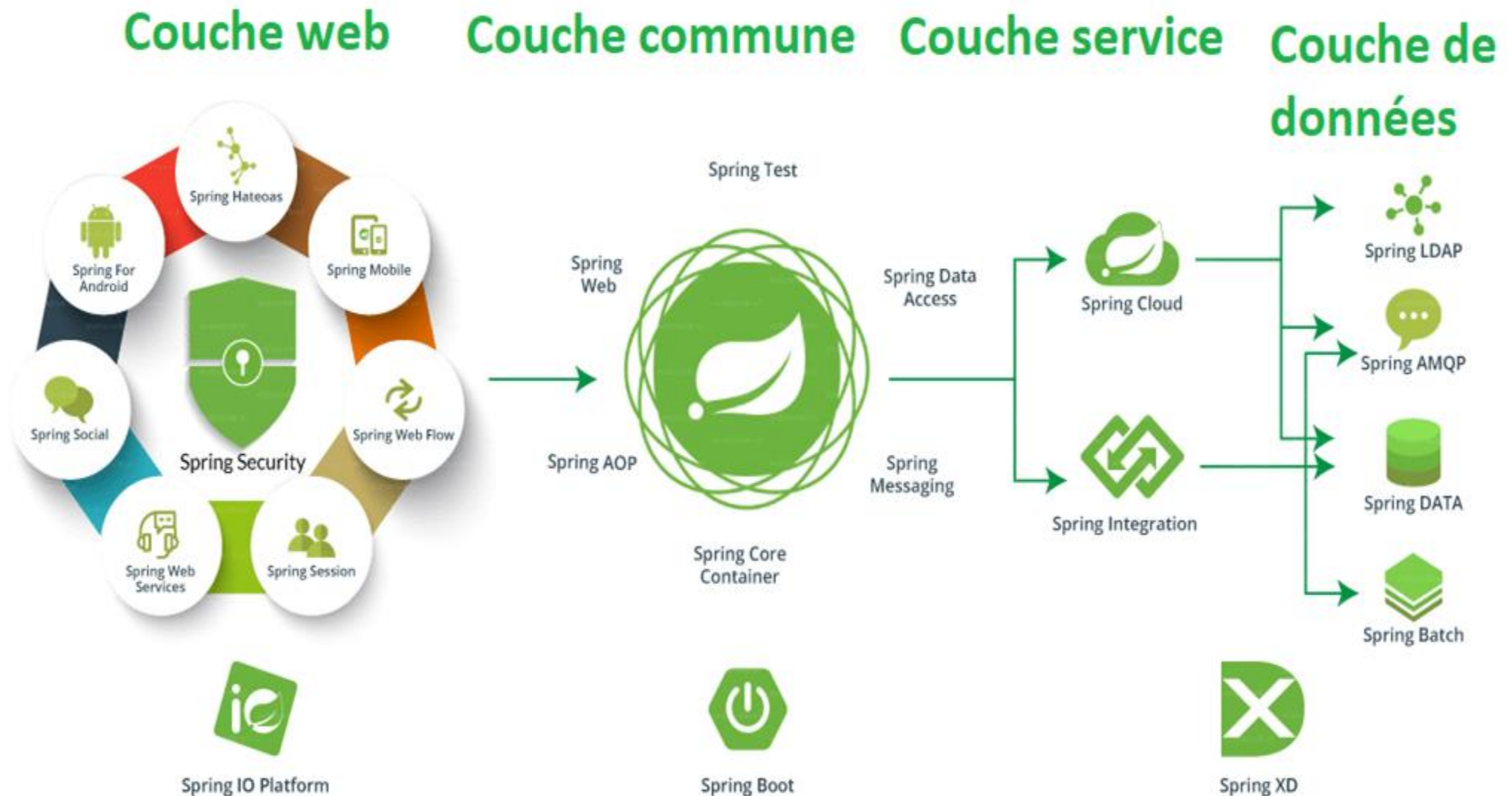
# Concurrents Spring

Basé sur la comparaison et la demande du marché, nous allons nous focaliser dans notre cours sur **Spring** et utiliser une implémentation de JPA fournie par Spring : **Spring Data JPA**.

# Les Projets Spring

- <https://spring.io/projects> :
- **Spring est un ensemble de projets.**
- Spring a une vaste communauté et une bonne documentation.
- Le code est sous license Apache 2, il est Open Source et disponible sur GitHub.

# Les Projets Spring



# Les Projets Spring

- **Spring Framework (Spring Core)**: C'est le cœur des projets Spring. Ce framework contient les fonctionnalités de base de Spring. Ce framework contient **Spring MVC**.
- **Spring Batch** : permet le développement des applications de type batch qui peuvent gérer de gros volumes de données
- **Spring Integration** : il s'agit d'un ESB (Enterprise Service Bus) pour interconnecter les applications d'une entreprise
- **Spring Android** : a pour but de faciliter le développement d'applications Android
- **Spring Boot** : C'est un outil Spring qui vous permet d'embarquer un serveur Tomcat dans votre livrable, et simplifier la livraison et le test de votre application.
- **Spring Data JPA** : fournit une implémentation de la couche d'accès aux données.
- **Spring Security** : permet de gérer l'authentification et les habilitations d'une application web.

# Framework Spring (Spring Core)

- C'est quoi un **Framework** :
  - C'est un **cadre de développement**
  - Contient des «bonnes pratiques»
  - Permet d'éviter de recoder des classes utilitaires
  - Permet de se focaliser sur le métier
- Spring fournit la «plomberie» : le socle technique
- Les développeurs peuvent se concentrer sur le code métier (le vrai travail)
- « Attention un **framework** ne doit pas être considéré comme une **plate-forme**, dans la mesure où il n'intègre pas d'environnement d'exécution système ou applicatif. »

# Framework Spring

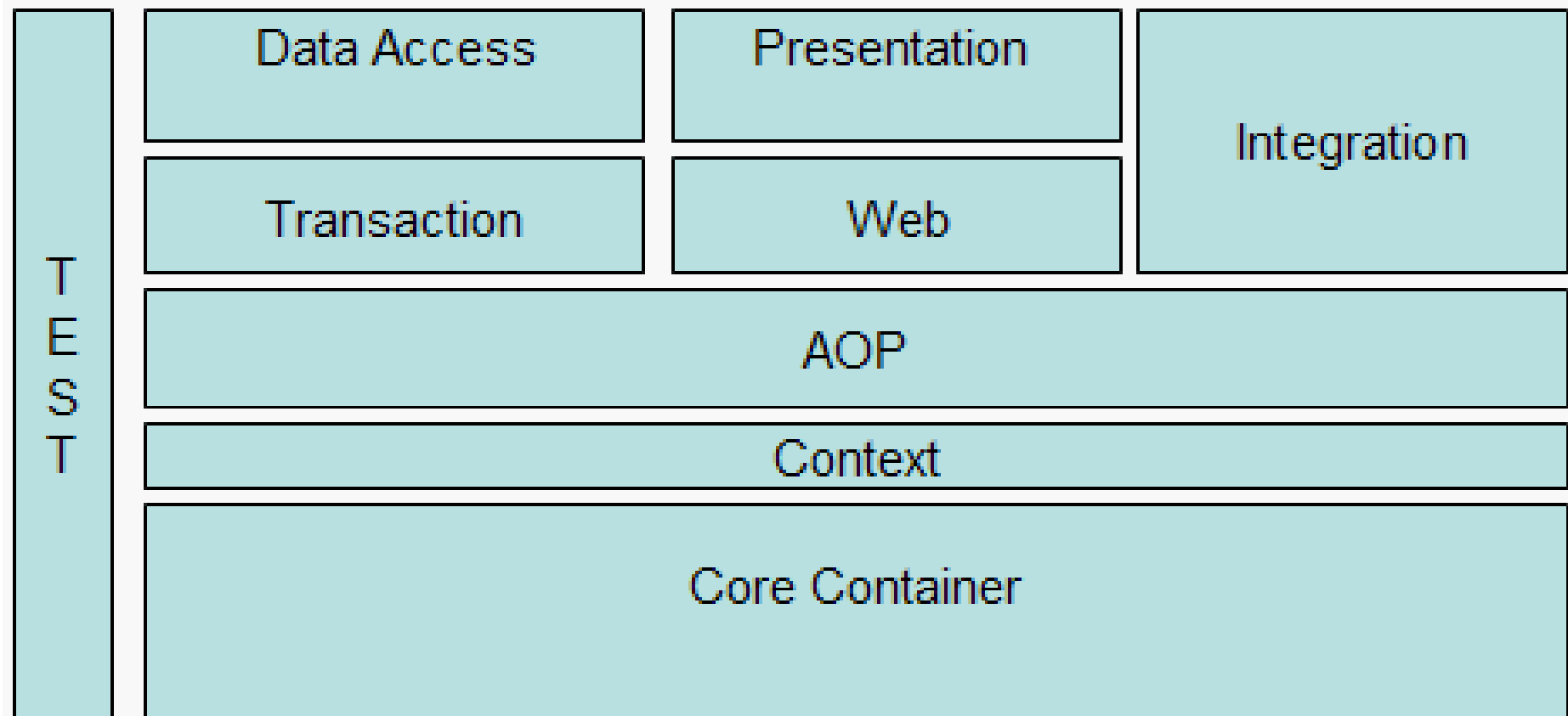
- Le Framework Spring est un socle pour le développement d'applications.
- Il fournit de nombreuses fonctionnalités.
- C'est l'un des Frameworks les plus répandus dans le monde Java.
- Framework Open Source
- **Framework Spring (Spring Core)** : <http://projects.spring.io/spring-framework>



# Framework Spring

- Le but de Spring est de faciliter et de rendre productif le développement d'applications.
- Il fournit beaucoup de fonctionnalités :
  - **Injection de Dépendances** (IoC : Inversion de Control).
  - **AOP** : Aspect Oriented Programming : Injection de Code en RunTime.
  - **Data Access** : Permet de simplifier l'accès aux données (DAO, ORM, Transaction, ...).
  - **Web** : Permet de développer des interfaces web évoluées (MVC).
  - **Testabilité** de l'application facilitée
  - **Intégration** : Spring offre un ESB, qui permet l'intégration et la communication entre les applications (EAI).

# Framework Spring

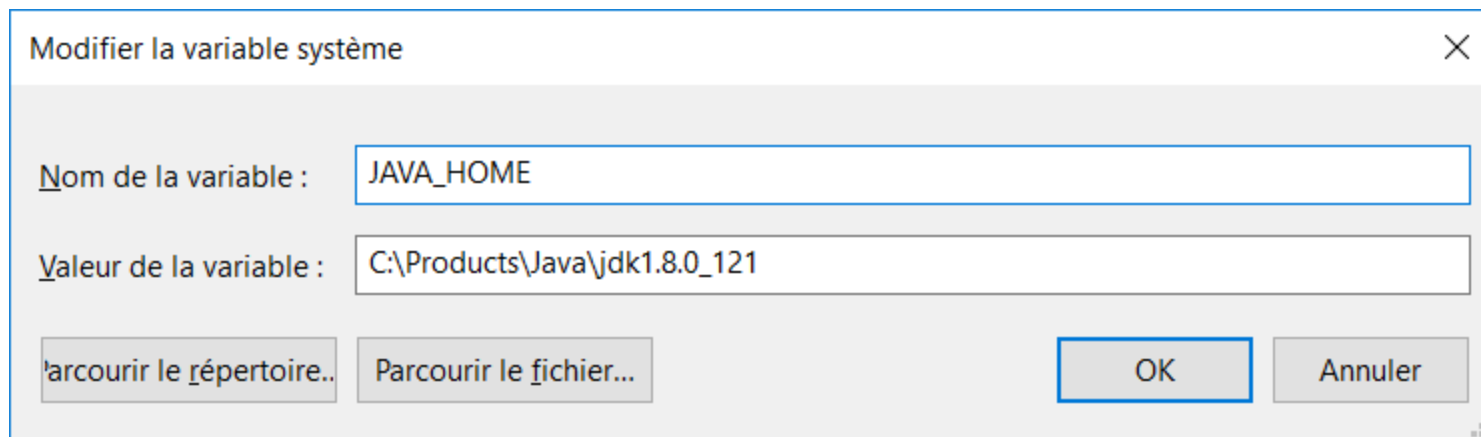


# Installation des Outils

- Les outils suivant vont nous permettre de développer des applications Web avec Spring.
- Avant de faire l'installation, vérifiez que vous n'avez pas ces outils déjà installés. Si c'est la cas, pas besoin de les réinstaller (vous pouvez utiliser d'autres versions) :
- **JDK 8** (1.8 - peu importe la version mineure)
- **IntelliJ**
- **WAMP ou XAMP**, pour avoir une base de données
- Par souci d'homogénéité, vous pouvez créer un dossier **C:\Products**, dans lequel, vous allez installer tous les outils que nous allons utiliser par la suite.

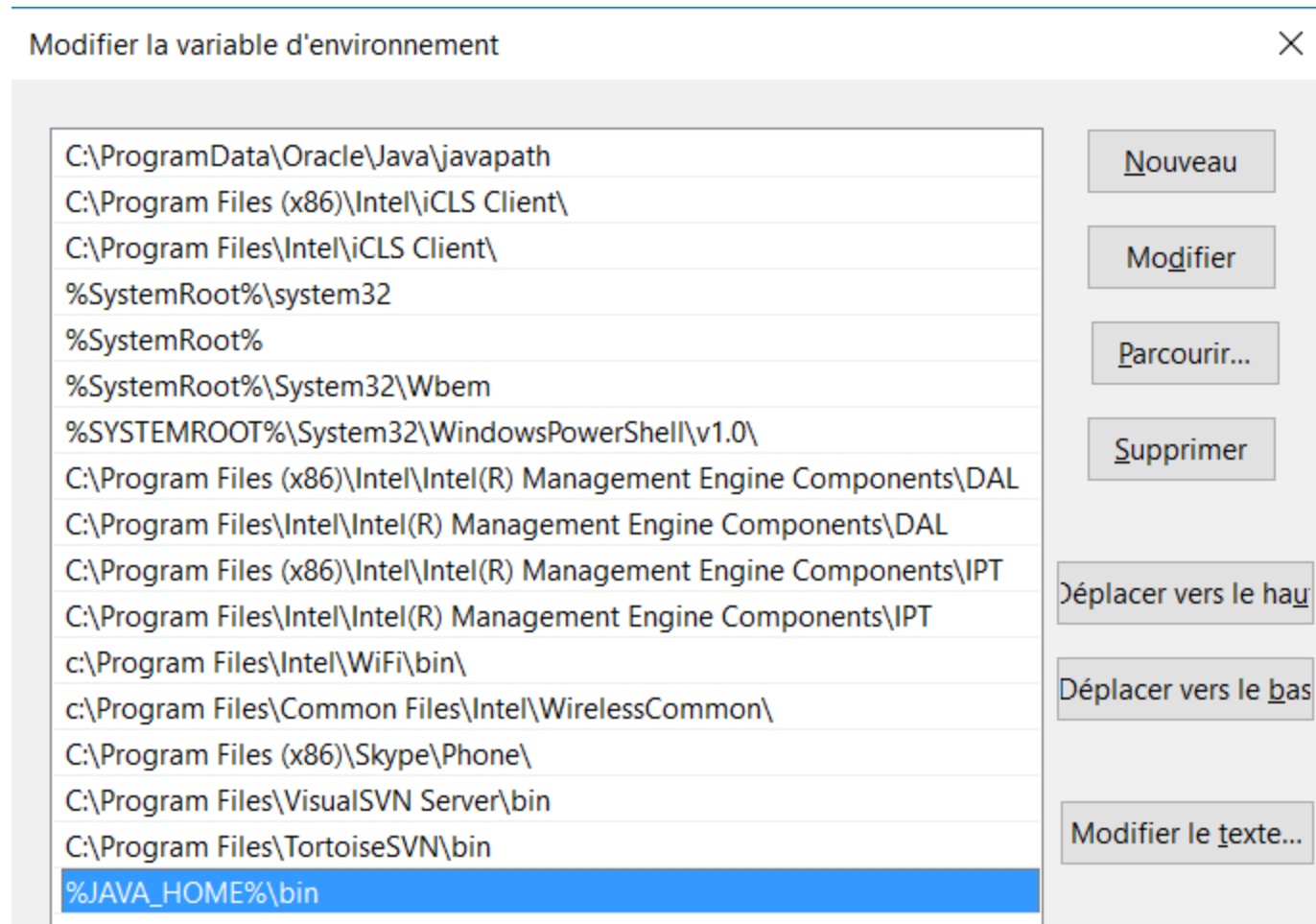
# Installation JDK 8

- Installer la JDK dans **C:\Products\java\jdk-1.8.0.60**
- Installer la JRE dans **C:\Products\java\jre-1.8.0.60**
- La JRE est installée en même temps que la JDK.
- Créer la variable d'environnement **système** : **JAVA\_HOME** qui contient le chemin de la JDK:



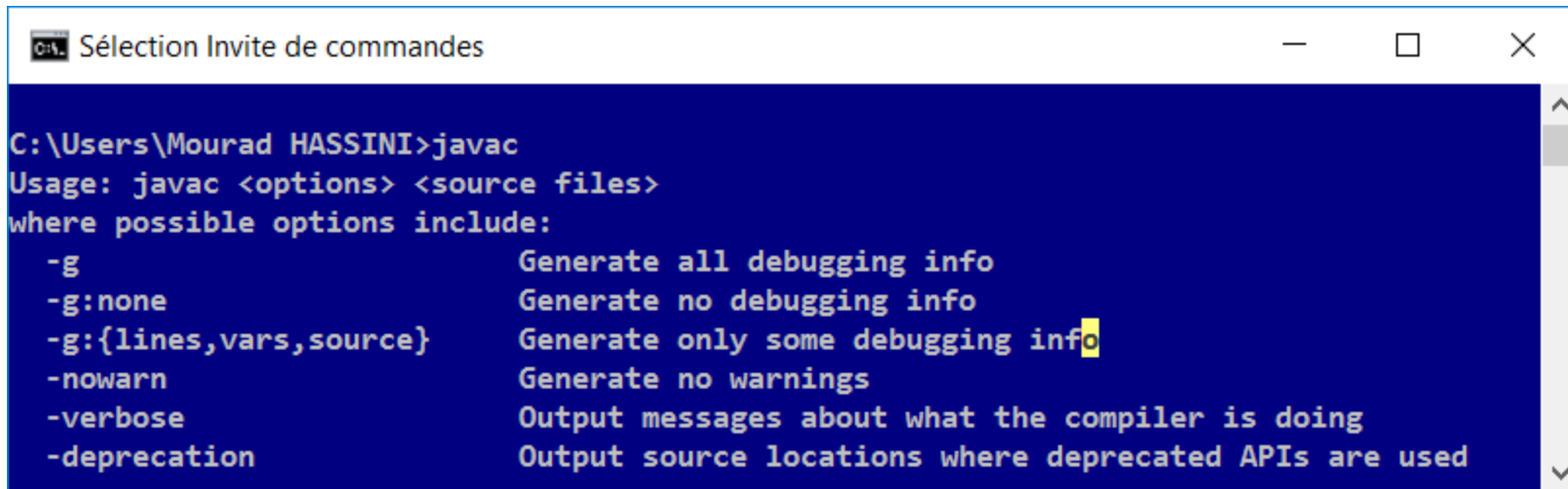
# Installation JDK 8

- Ajouter **%JAVA\_HOME%\bin** au Path :

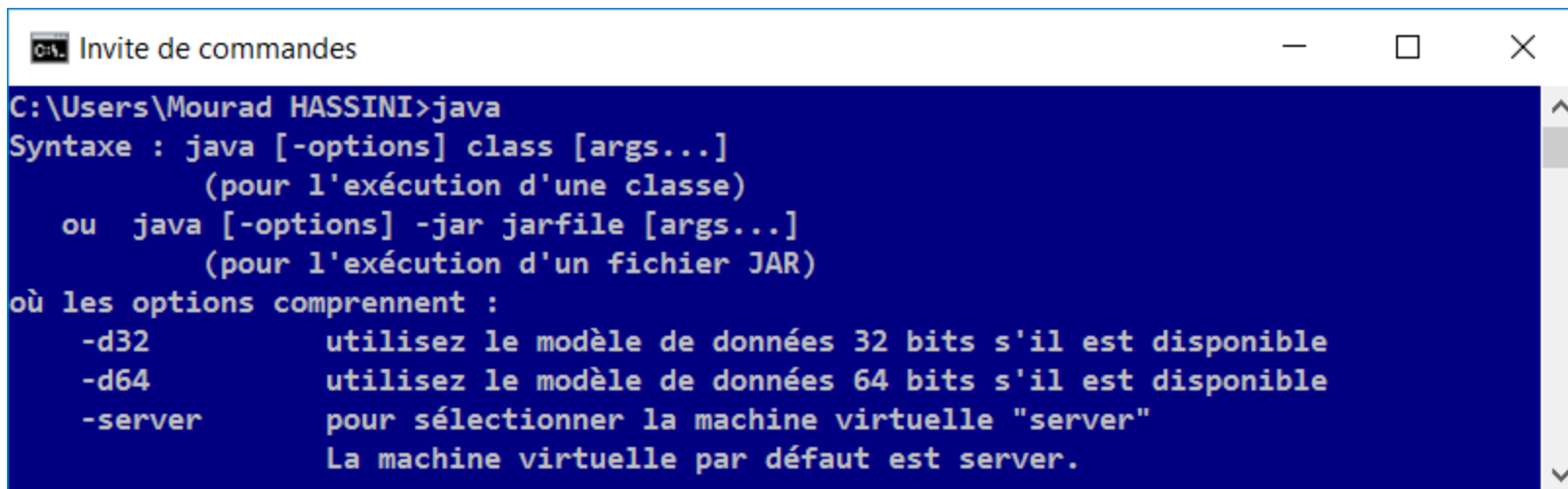


# Installation JDK 8

- Vérifier que Java est bien installé et que la variable d'environnement est bien positionnée (**javac** et **java**) :



```
C:\Users\Mourad HASSINI>javac
Usage: javac <options> <source files>
where possible options include:
    -g                Generate all debugging info
    -g:none           Generate no debugging info
    -g:{lines,vars,source}  Generate only some debugging info
    -nowarn           Generate no warnings
    -verbose          Output messages about what the compiler is doing
    -deprecation      Output source locations where deprecated APIs are used
```



```
C:\Users\Mourad HASSINI>java
Syntaxe : java [-options] class [args...]
           (pour l'exécution d'une classe)
ou java [-options] -jar jarfile [args...]
           (pour l'exécution d'un fichier JAR)
où les options comprennent :
    -d32             utilisez le modèle de données 32 bits s'il est disponible
    -d64             utilisez le modèle de données 64 bits s'il est disponible
    -server          pour sélectionner la machine virtuelle "server"
                    La machine virtuelle par défaut est server.
```

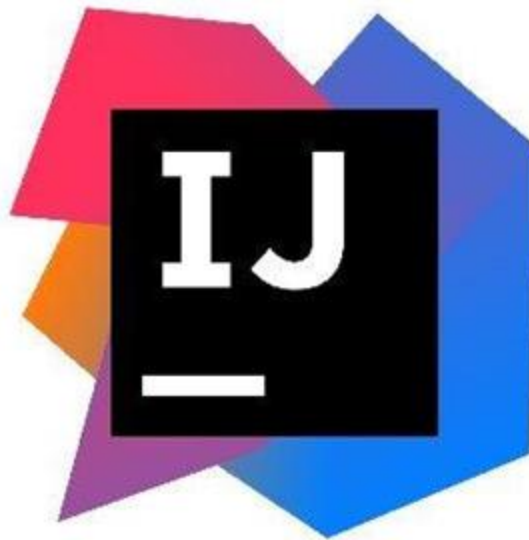
# Installation IntelliJ

- Suivre les instructions du workshop (voir [Drive](#) : 1- Configuration IntelliJ.pf) pour installer et configurer IntelliJ



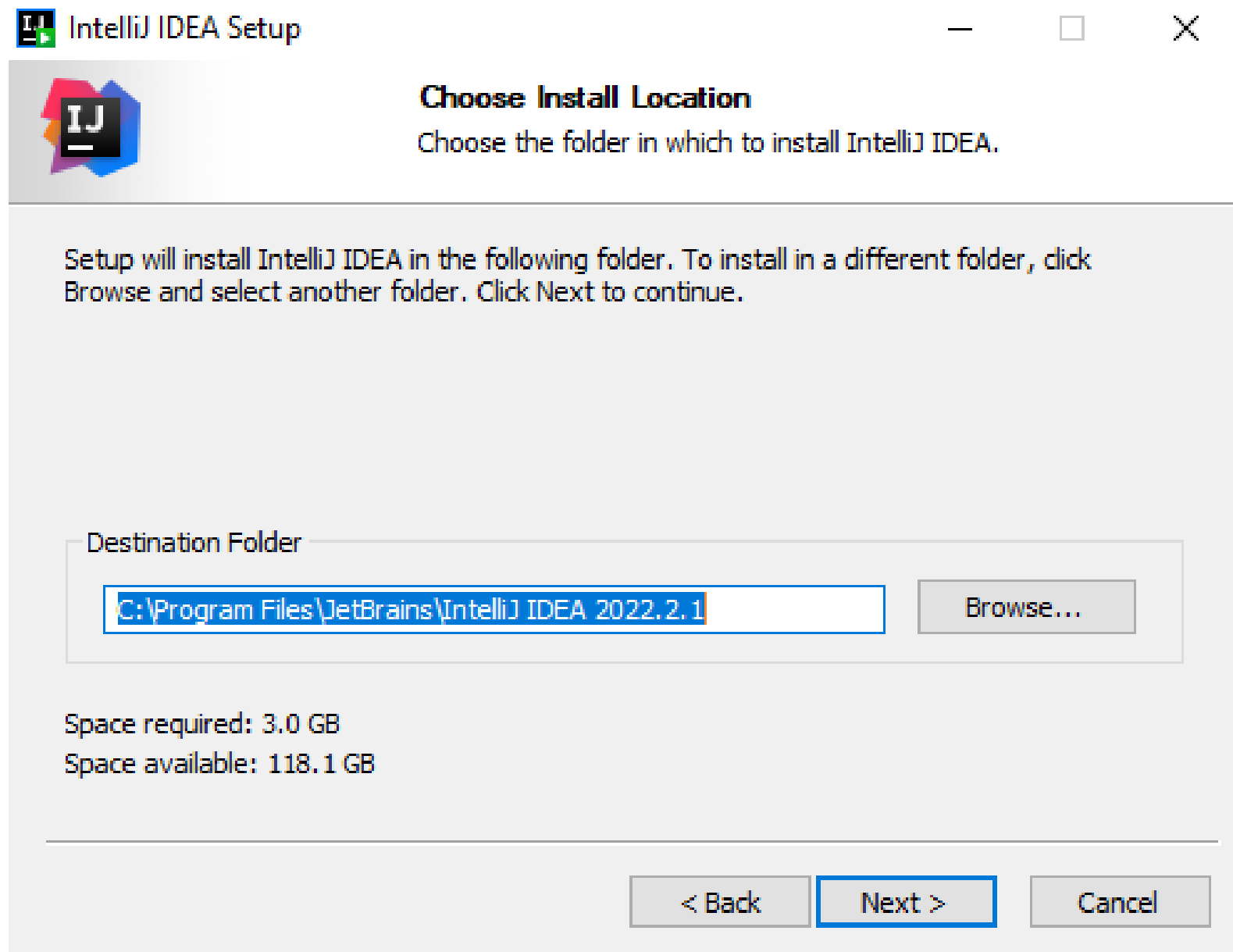
# Installation IntelliJ

- Il suffit de télécharger la dernière version d'IntelliJ et de l'activer avec votre compte Esprit
- Si vous n'avez pas le fichier vous pouvez le télécharger : <https://www.jetbrains.com/idea/>

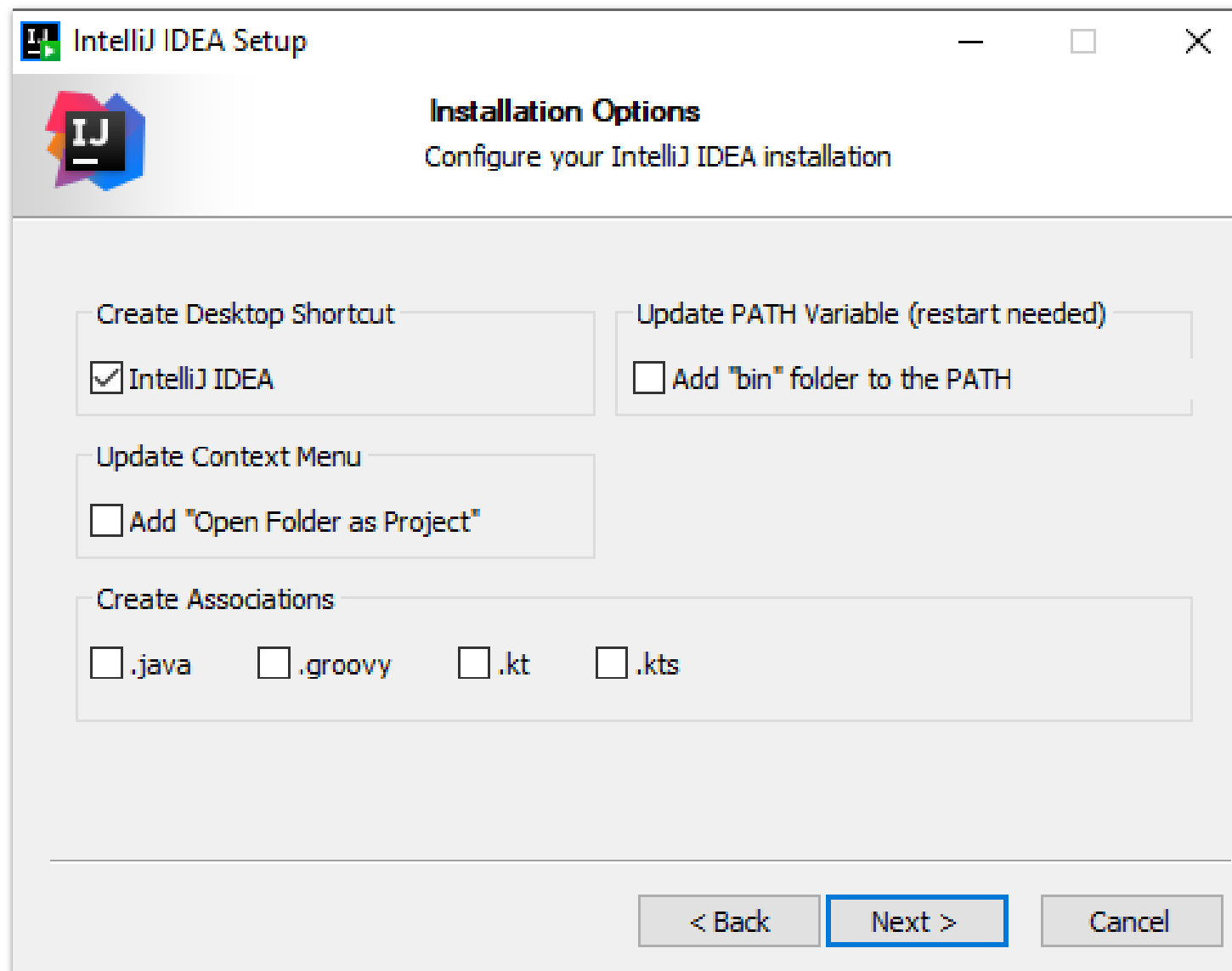




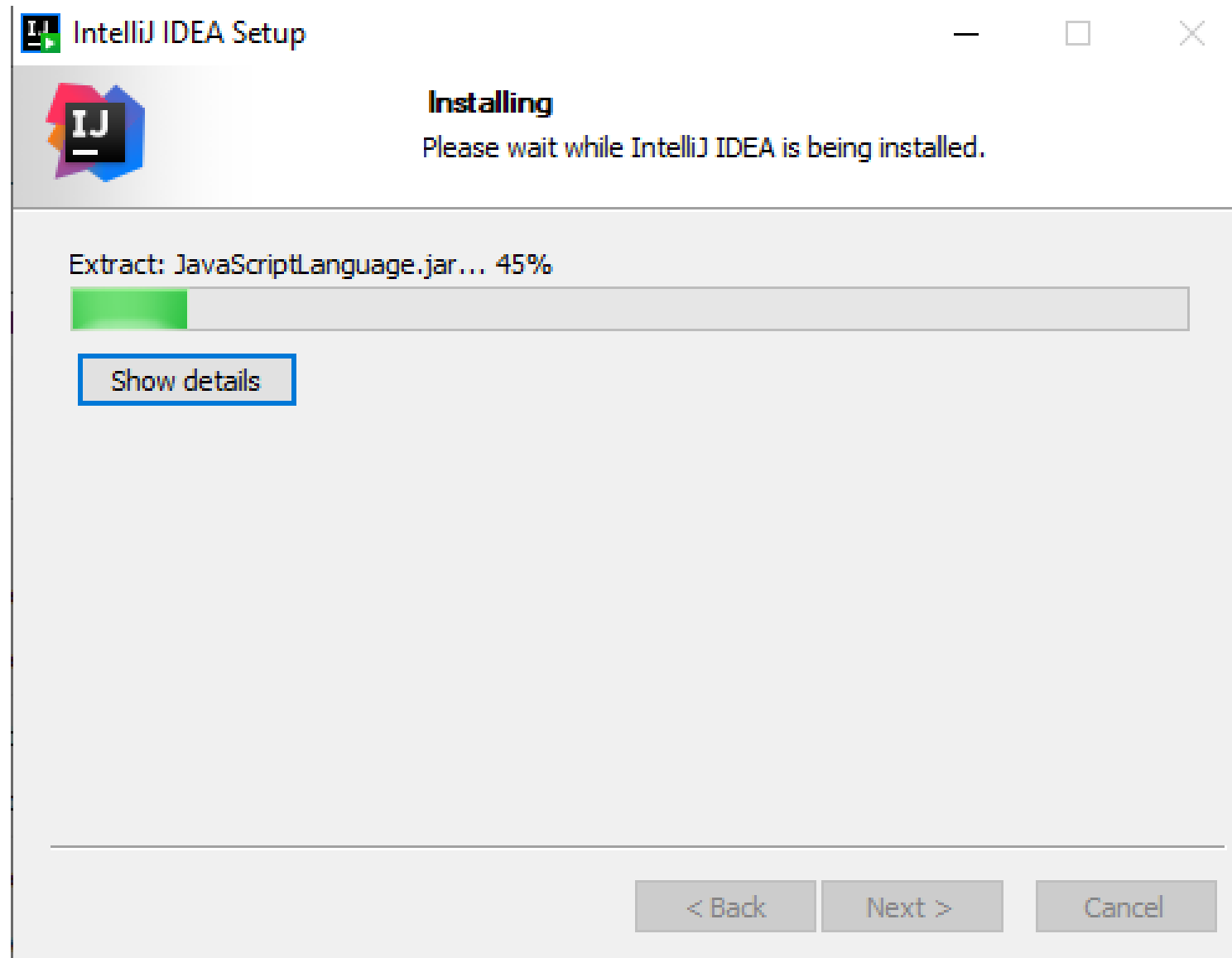
# Installation IntelliJ



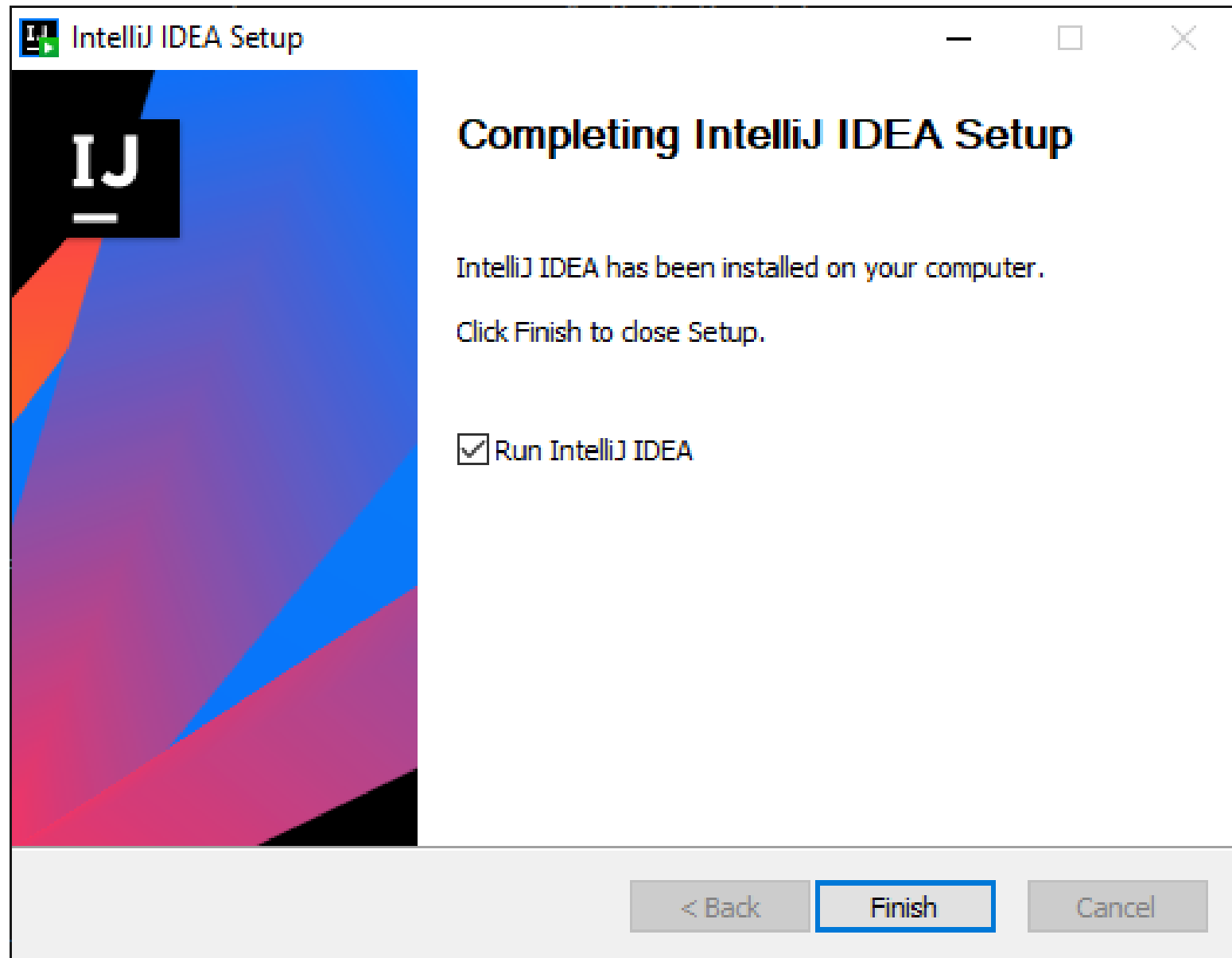
# Installation IntelliJ




# Installation IntelliJ



# Installation IntelliJ



# Installation IntelliJ

 IntelliJ IDEA User Agreement ×

## JETBRAINS USER AGREEMENT

Version 1.4, effective as of September 22, 2021

IMPORTANT! READ CAREFULLY:

THIS IS A LEGAL AGREEMENT. BY CLICKING ON THE "I AGREE" (OR SIMILAR) BUTTON THAT IS PRESENTED TO YOU AT THE TIME OF YOUR FIRST USE OF THE JETBRAINS SOFTWARE, SUPPORT, OR PRODUCTS, YOU BECOME A PARTY TO THIS AGREEMENT, YOU DECLARE YOU HAVE THE LEGAL CAPACITY TO ENTER INTO SUCH AGREEMENT, AND YOU CONSENT TO BE BOUND BY ALL THE TERMS AND CONDITIONS SET FORTH BELOW.

### 1. PARTIES

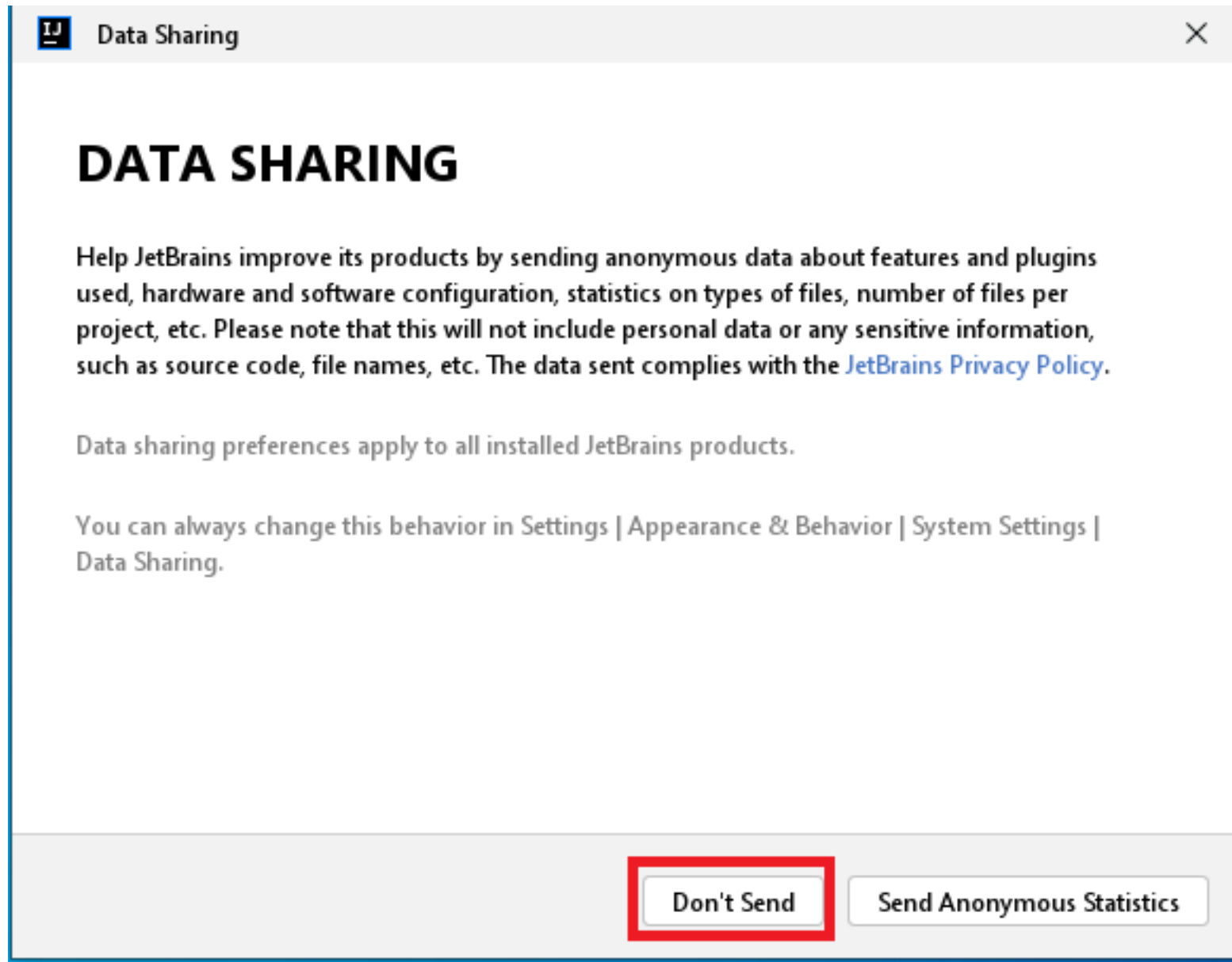
1.1. "JetBrains" or "we" means JetBrains s.r.o., having its principal place of business at Na Hřebenech II 1718/10, Prague, 14000, Czech Republic, registered in the Commercial Register maintained by the Municipal Court of Prague, Section C, File 86211, ID No.: 265 02 275.

☒ I confirm that I have read and accept the terms of this User Agreement

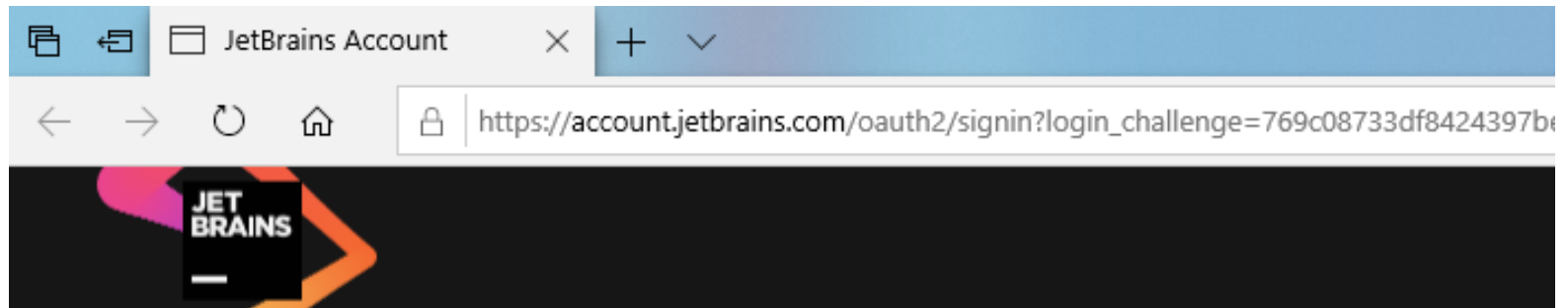
Exit

Continue

# Installation IntelliJ



# Installation IntelliJ



Sign In

Create Account

Username

Password

Sign In

[Forgot password?](#)

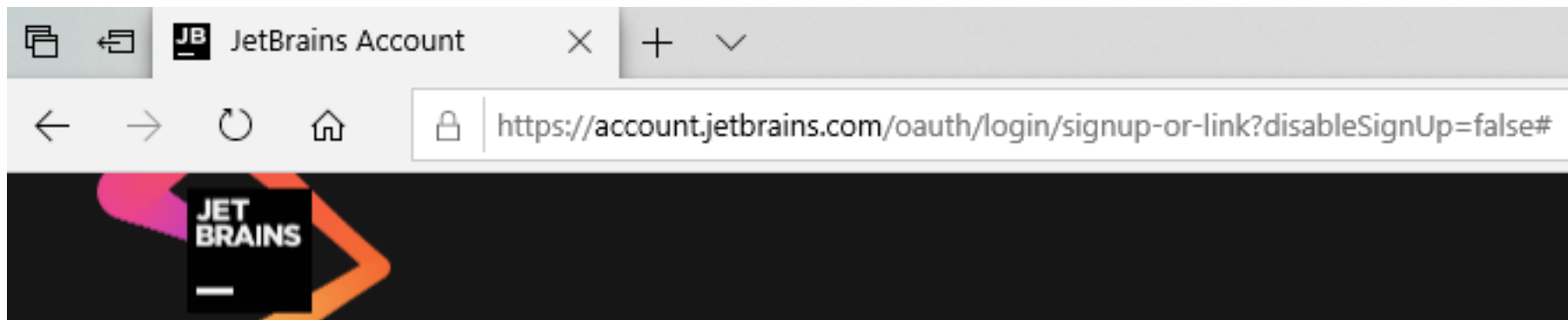
Or sign in with:



Google



# Installation IntelliJ



## JetBrains Account Clarification

It looks like you already have a JetBrains Account.

Would you like to link your Google Account (ghassen.ammouda@esprit.tn) to it?

- ☒ Link to JetBrains Account **ghassen ammouda (ghassen.ammouda@esprit.tn)**
- ☐ Link to another JetBrains Account

[Link Accounts](#)



# Installation IntelliJ

## 1 Student Pack License

[Buy new license](#)

### JetBrains Product Pack for Students

License ID:  
JQWTLAERJZ

Licensed to: **ghassen hammouda**  
License restriction: For educational use only

Buy with 25% OFF ▼

Renew my Education Pack

Valid through: **March 14, 2022**

Following products included:

- [AppCode](#)
- [CLion](#)
- [DataGrip](#)
- [DataSpell](#)
- [dotCover](#)
- [dotMemory](#)
- [dotTrace](#)
- [GoLand](#)
- [IntelliJ IDEA Ultimate](#)
- [PhpStorm](#)
- [PyCharm](#)
- [ReSharper](#)
- [ReSharper C++](#)
- [Rider](#)
- [RubyMine](#)
- [WebStorm](#)

After downloading and installing the software, simply run it and follow the on-screen prompts to sign in with your JetBrains Account.

# Installation IntelliJ

## 1 Student Pack License

[Buy new license](#)

### JetBrains Product Pack for Students

License ID:  
JQWTLAERJZ

Licensed to: **ghassen hammouda**  
License restriction: For educational use only

Buy with 25% OFF

Renew my Education Pack

Valid through: **March 14, 2022**

Following products included:

- [AppCode](#)
- [CLion](#)
- [DataGrip](#)
- [DataSpell](#)
- [dotCover](#)
- [dotMemory](#)
- [dotTrace](#)
- [GoLand](#)
- [IntelliJ IDEA Ultimate](#)
- [PhpStorm](#)
- [PyCharm](#)
- [ReSharper](#)
- [ReSharper C++](#)
- [Rider](#)
- [RubyMine](#)
- [WebStorm](#)

After downloading and installing the software, simply run it and follow the on-screen prompts to sign in with your JetBrains Account.

Apply with:

University email address

ISIC/ITIC membership

Official document

GitHub

Status:

☐ I'm a student

☒ I'm a teacher

Email address:

ghassen.hamhouda@esprit.tn

I certify that the university email address provided above is valid and belongs to me.

Name:

Your **real name** as it appears in your passport, driver's license, or other legal documents.

ghassen

hammouda

# Installation IntelliJ



**JetBrains Account** <no\_reply@jetbrains.com>

À moi ▾

🌐 anglais ▾ > français ▾ [Traduire le message](#)

Hi,

You're receiving this email because your email address was used to register or update a JetBrains Educational Pack.

Please click this link to proceed:

<https://www.jetbrains.com/shop/eform/students/request?code=39afjnxmndkukwsl5bjyn6d5k>

After accepting the License Agreement, you will be asked to sign up for a JetBrains Account.

You will need to use this account whenever you want to access JetBrains tools.

If you have any questions, please email us – we're always happy to help.

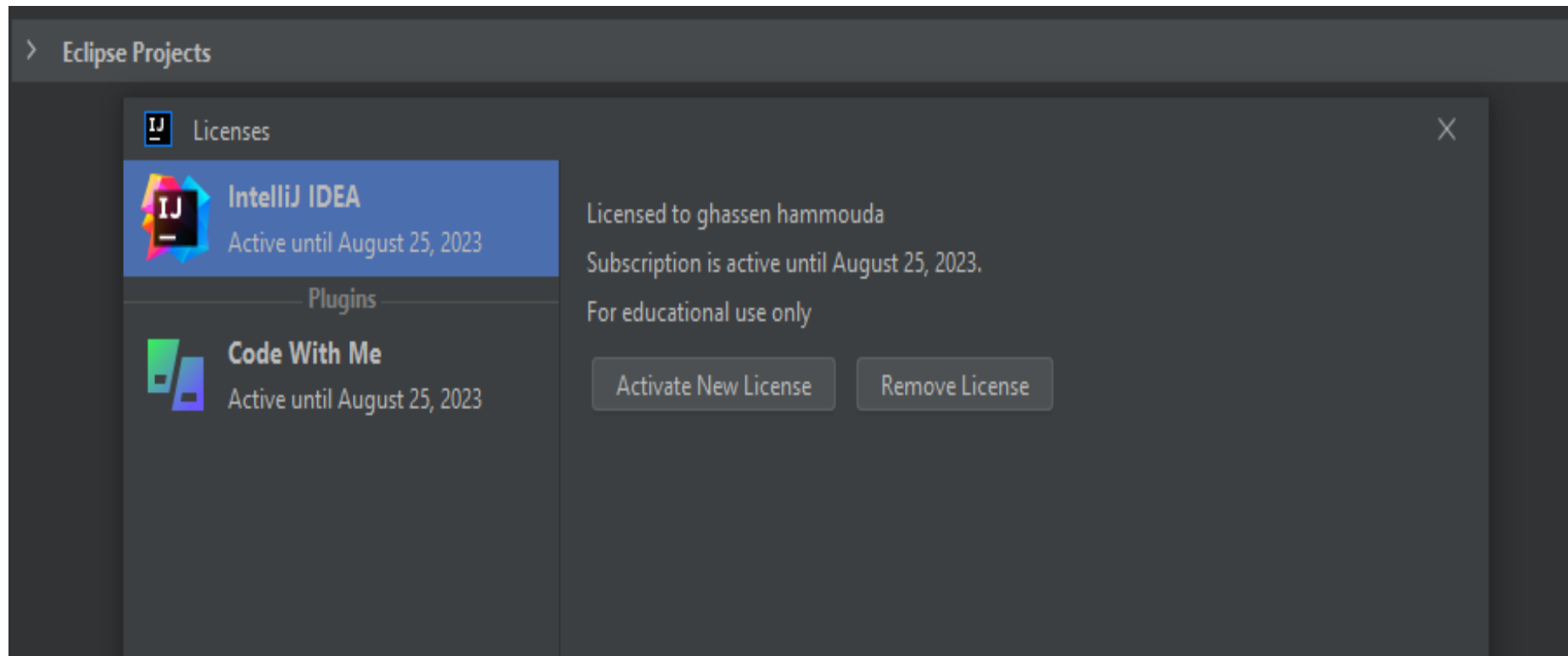
Kind Regards,

The JetBrains team

[www.jetbrains.com](https://www.jetbrains.com)

The Drive to Develop

# Installation IntelliJ



# Prochain Cours

La prochaine séance nous allons :

- Finaliser l'Installation de l'Environnement de travail (JDK, INTELLIJ, WAMP ou XAMP)
- Créer notre premier projet **Spring Boot**
- Manipuler ce projet Spring Boot avec **Maven**

# Introduction SPRING

