



# Une version compacte de serveurs locaux pour l'IDE AppInventor

publié le 14/09/2013

## Comment installer facilement l'IDE AppInventor.

### Descriptif :

L'utilisation de l'IDE AppInventor via l'emploi de serveurs locaux offre une ergonomie d'utilisation importante. Cela est lié à l'absence de la nécessité de connexion gmail mais également (et surtout) au gain de rapidité offert par l'emploi des réseaux internes. Cependant, l'implantation de ces serveurs était jusqu'alors assez fastidieuse, cela n'est plus le cas désormais. Une activité pédagogique est proposée à l'issue de l'article.

### Sommaire :

- 1. Prérequis
- 2. Téléchargement du fichier d'installation des serveurs
- 3. Une activité pédagogique en Première STI2D-SIN

**MIT App inventor**

MIT App Inventor Public Open Source

Depuis Janvier 2012, le MIT a repris le projet AppInventor initié par le GoogleLabs. Ce projet désormais Open Source peut largement être utilisé à des fins pédagogiques (algorithmie, sensibilisation à la programmation orientée objet...)



La création de serveurs d'application et de construction (design et programmation de l'application et compilation vers tablette ou smartphone) pouvait, dès 2012, s'effectuer via l'emploi de serveurs locaux en s'affranchissant d'une "connexion" gmail. Cependant l'installation de ces serveurs demeurait relativement complexe.

Voici une solution, fonctionnant sous Windows 7, qui a le mérite d'être très simple à installer. Cette version permet la création/développement d'applications pour smartphone et tablette Android, elle permet également l'implantation de l'application créée vers le système désiré (tablette, smartphone...).

Remerciements à Mr **Gary Fredricks** pour avoir pensé et réalisé cette intégration ainsi qu'à Mr **Alain Sudrat** pour avoir testé cette solution en "prime time".

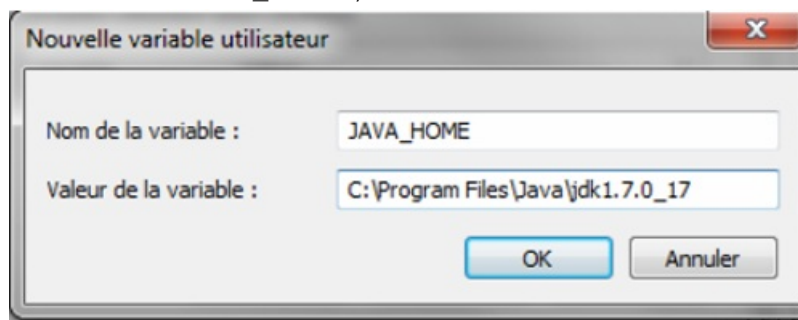
### ● 1. Prérequis

#### ○ 1.1 Télécharger Java JDK

Afin que l'IDE AppInventor puisse fonctionner il faut que la plateforme Java-sdk soit installée. Lien : [Java JDK](#) (ici une version 64 bits)

Java SE Development Kit 7u17		
You must accept the <a href="#">Oracle Binary Code License Agreement for Java SE</a> to download this software.		
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux x86	106.65 MB	<a href="#">jdk-7u17-linux-i586.rpm</a>
Linux x86	92.97 MB	<a href="#">jdk-7u17-linux-i586.tar.gz</a>
Linux x64	104.78 MB	<a href="#">jdk-7u17-linux-x64.rpm</a>
Linux x64	91.71 MB	<a href="#">jdk-7u17-linux-x64.tar.gz</a>
Mac OS X x64	143.78 MB	<a href="#">jdk-7u17-macosx-x64.dmg</a>
Solaris x86 (SVR4 package)	135.39 MB	<a href="#">jdk-7u17-solaris-i586.tar.Z</a>
Solaris x86	91.67 MB	<a href="#">jdk-7u17-solaris-i586.tar.gz</a>
Solaris SPARC (SVR4 package)	135.92 MB	<a href="#">jdk-7u17-solaris-sparc.tar.Z</a>
Solaris SPARC	95.32 MB	<a href="#">jdk-7u17-solaris-sparc.tar.gz</a>
Solaris SPARC 64-bit (SVR4 package)	22.97 MB	<a href="#">jdk-7u17-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	17.59 MB	<a href="#">jdk-7u17-solaris-sparcv9.tar.gz</a>
Solaris x64 (SVR4 package)	22.61 MB	<a href="#">jdk-7u17-solaris-x64.tar.Z</a>
Solaris x64	15.02 MB	<a href="#">jdk-7u17-solaris-x64.tar.gz</a>
Windows x86	88.75 MB	<a href="#">jdk-7u17-windows-i586.exe</a>
Windows x64	90.42 MB	<a href="#">jdk-7u17-windows-x64.exe</a>

- **Télécharger** et installer ;
- **Ajouter** une variable d'environnement **JAVA\_HOME** ;



(Vérifier que le répertoire correspond : ici `jdk1.7.0_017`, cela peut changer selon les versions...)

### ○ 1.2 Télécharger l'application AppInventor java

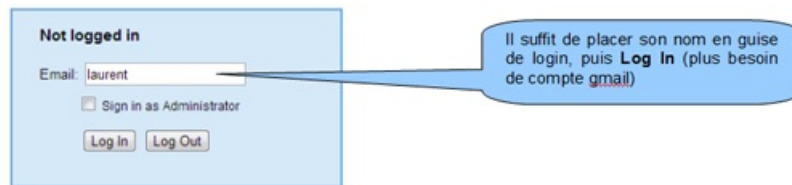
L'application [AppInventor](#) doit être installée (si cela n'a pas déjà été fait auparavant lors d'une précédente utilisation de cet IDE).

## ● 2. Téléchargement du fichier d'installation des serveurs

Lien : [ai4a MIT App inventor](#)

- Télécharger la ressource **V1.4.7.zip**, la décompresser dans le répertoire **c :\APPINVENTOR** ;
- Depuis le répertoire **C :\APPINVENTOR\AppEngine**, créer un raccourci du fichier **startAI.cmd** vers le bureau ;
- Modifier ce fichier afin de remplacer la ligne **start %AEJSDK%\bin\dev\_appserver.cmd —port=8888 —address=localhost war** par **start %AEJSDK%\bin\dev\_appserver.cmd —port=8888 —address=mon\_adresse\_ip war**
- Depuis le répertoire **C :\APPINVENTOR\BuildServer**, créer un raccourci du fichier **launch-buildserver.cmd** vers le bureau (pour une version 64bits) ;
- C'est terminé, il suffit de lancer le fichier **startAI.cmd** et d'attendre que le serveur d'application s'exécute. Il faut également faire de même avec le fichier **launch-buildserver.cmd**. Les deux serveurs sont désormais lancés.

Dès lors, depuis votre navigateur, il suffit d'entrer l'URL : `mon_adresse_ip:8888` afin d'avoir accès à l'environnement.





Les articles suivants informent sur les possibilités de ApplInventor :

- [Présentation de ApplInventor](#) ;
- [Communication Bluetooth](#) ;
- [Serveur local ApplInventor et Linux](#).

### ● 3. Une activité pédagogique en Première STI2D-SIN

A titre d'exemple, voici un document ressource , ainsi qu'une activité pédagogique portant sur l'algorithmie. Cette activité a été menée en STI2D-SIN. L'activité 1 porte particulièrement sur la structure alternative, l'écriture d'un algorithme, et l'exploitation des ressources d'une tablette (reconnaissance vocale et synthèse vocale). Le développement d'une application clôture l'activité.

 [Algorithmie et Programmation](#) (PDF de 256.7 ko)  
Document ressource.

 [Algorithmie et Programmation](#) (PDF de 300.3 ko)  
Activité 1 élève.

*Laurent Proust Septembre 2013*