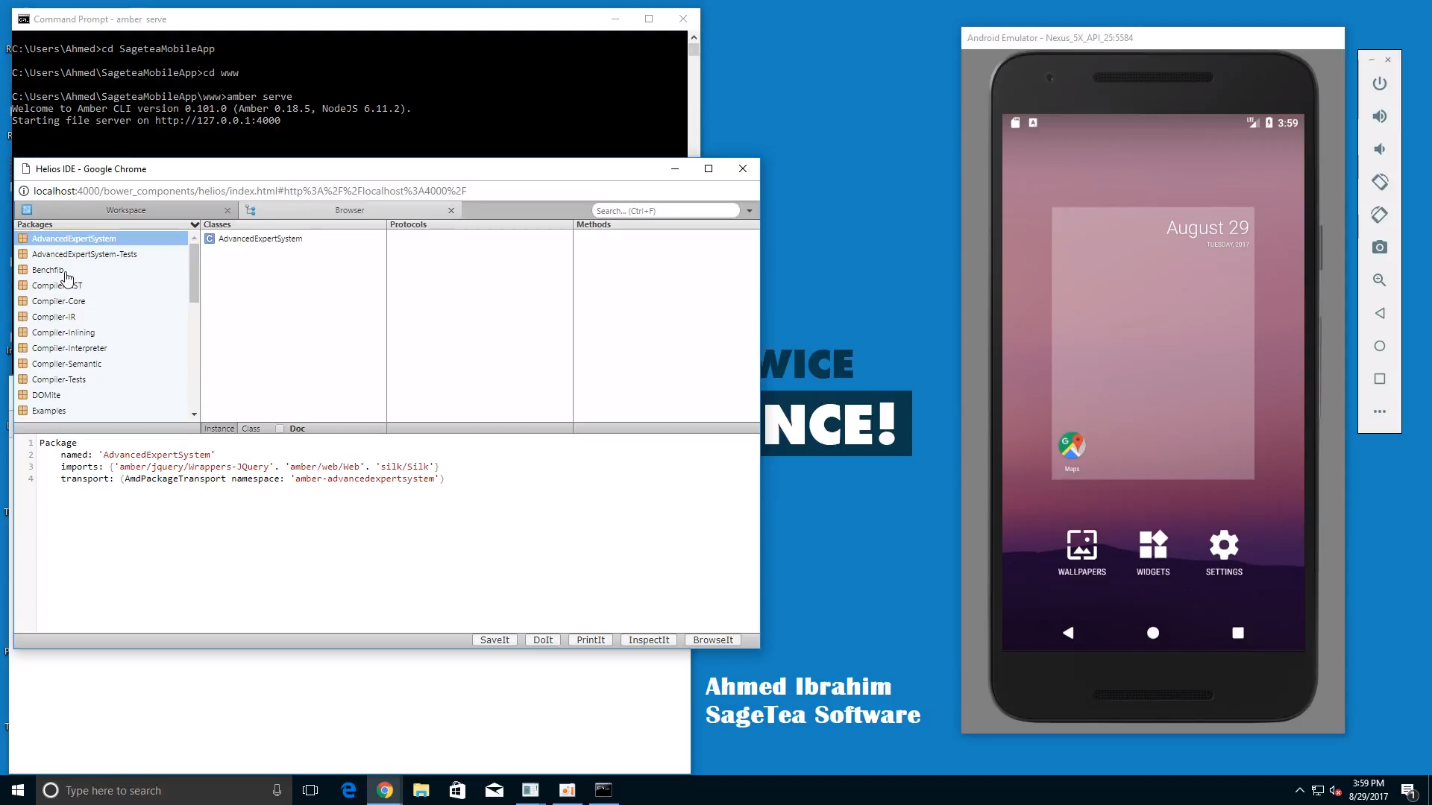
**Smalltalk Mobile Application**

****

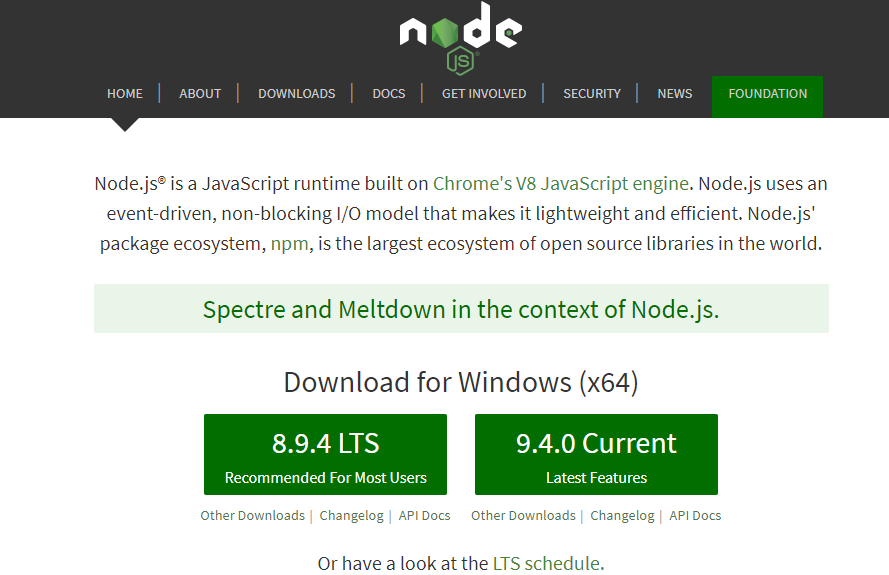
Before we start with the project, we must install several **tools** in the machine to help building the mobile application. On the other hand,the following YouTube link is a **demo** of the final step of launching the **Cordova** mobile application through the command prompt:

[**https://www.youtube.com/watch?v=bTzM0uOLFMs**](https://www.youtube.com/watch?v=bTzM0uOLFMs)

The goal of this project is to code in **Smalltalk** programming language and make the code works as an **Android, IOS, Blackberry**...etc. mobile application. For this to make it work, I had to use two cross compilers, the first cross compiler is called **Helios**. Helios takes **Smalltalk** code and coverts into **JavaScript** language that works as webpage application, from that point the second cross compiler converts webpages applications into mobile applications by translating **JavaScript** to **Android Java** that is understood by **Android Studio, Eclipse** and other IDEs which is called **Cordova**.

To install the initial tools to make **Helios, Cordova, Android (Emulator)** work, please follow these instructions/steps:

1. First, we need to install Amber that runs the **cross-compiler** Helios. Go ahead and open the command prompt and type **npm install -g amber-cli**
2. Open the browser and Install **Node**.**js**. Cordova runs on Node.js platform, which needs to be installed the first step.

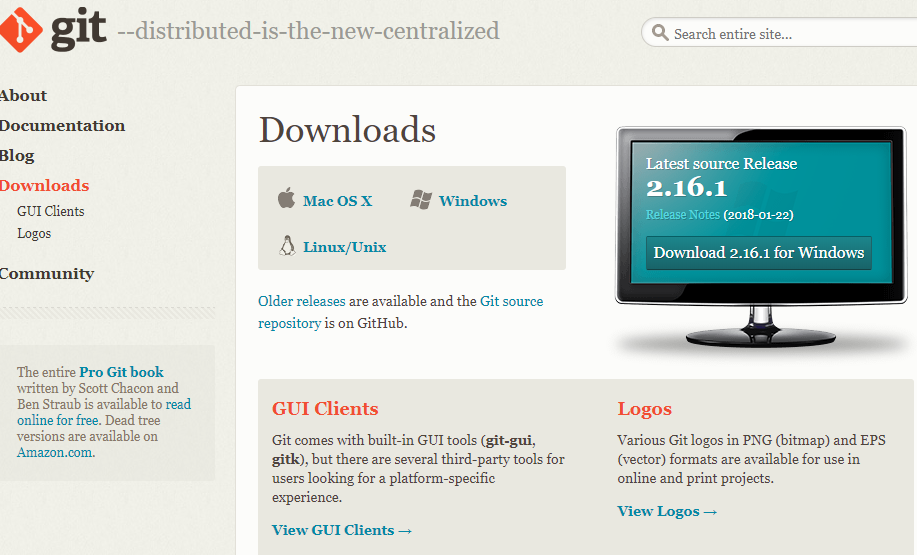


1. Run **Node.js** and It is recommended to use the default settings. Node.js needs to be added to the **PATH** environment variable, which is done by default.
2. To test the installation, open a command window (make sure you open a new command window to get the updated path settings made by the Node.js installation), and type:

**node –version**

if the version number is displayed, **Node.js** is installed and working!

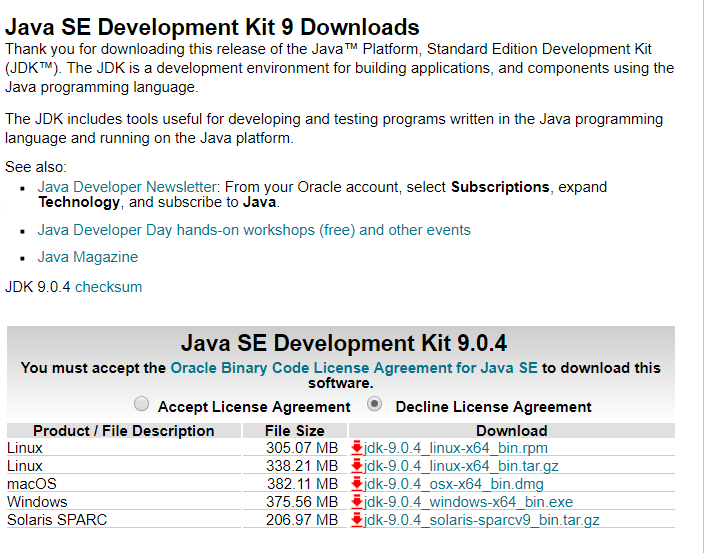
1. Install **Git**. Git is a version control system, which is used by Cordova behind-the-scenes. Download and install from**: http://git-scm.com**. Default settings are recommended.



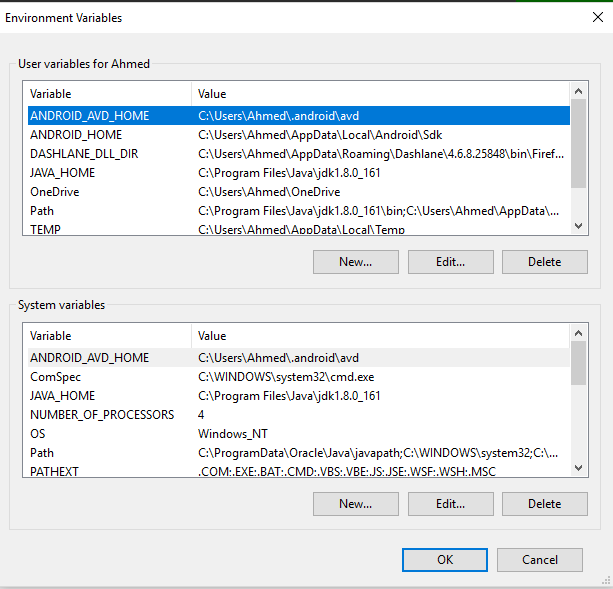
1. It’s also recommended to install apache ant, here are how to do it. <https://www.mkyong.com/ant/how-to-install-apache-ant-on-windows/>



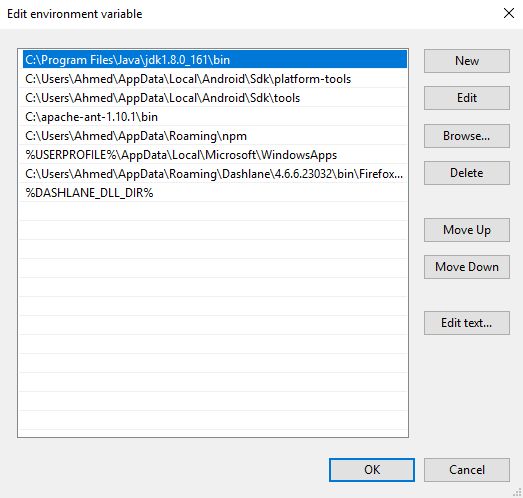
1. Install **Java**, The **Android** **SDK** needs the Java Development Kit (JDK) to be installed, version 1.7 or later. Note that the Java Runtime Environment (JRE) is not sufficient, you will need the JDK. To check if you have the JDK installed already, type this on the command line:  
   **javac -version**
2. Download the recent version of **Java SE JDK** (SE = Standard Edition) from Oracle: www.oracle.com/technetwork/java/javase/downloads/. Click the Java SE Download to see the list of downloads. Get the "Windows x86" download if you have 32-bit Windows, and "Windows x64" if you have 64-bit Windows.



1. Go along and run the downloaded installer file. Using the default selections should be fine, but take a note of the directory in which you install the **JDK**. You will need to add this to the **PATH** in a later step below.
2. Next, update your **path** to include the **JDK**. Open the **Control Panel**, click System and Security, click System, click Change settings, which will open the System Properties window. Select the Advanced tab, then click the **Environment Variables button.**

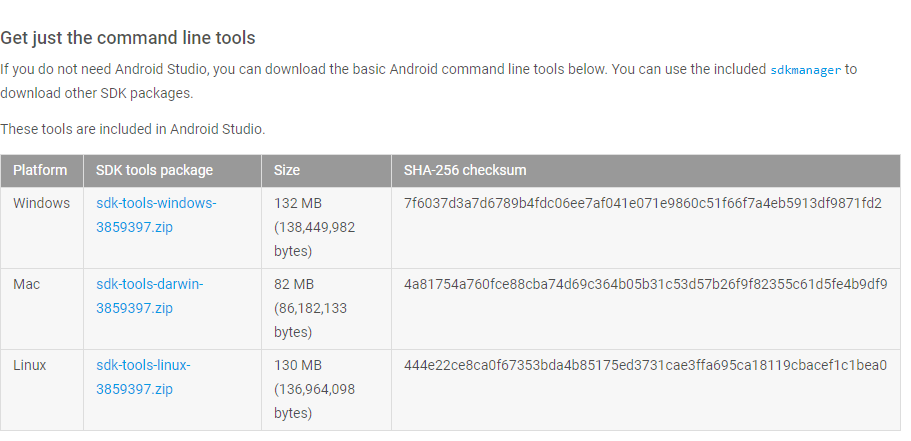


1. In the list User variables select **PATH** and click the Edit button. (If there is no PATH entry in the list, click the New button to create one.)
2. At the end of the field Variable value, add a semicolon followed by the path to the bin directory of the **JDK** install. Here is an example (note that this must be the actual path used for the install on your machine):  
   **C:\Program Files\Java\jdk1.8.0\_11\bin**

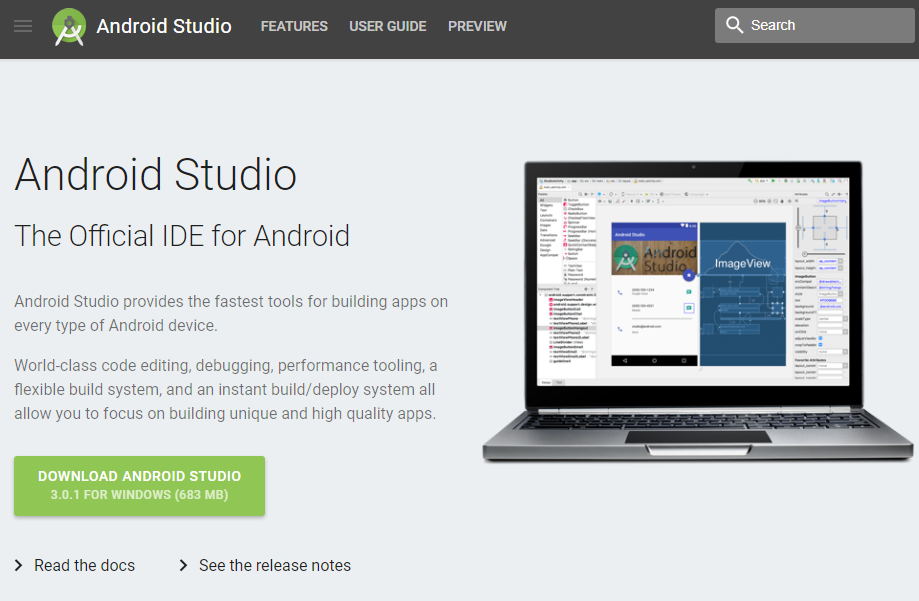
****

1. Next add the **JAVA\_HOME** variable if it is not present (and if it is in the list, you may need to update its value using the Edit button). Click the New button. In the field Variable name type:  
   **JAVA\_HOME**
2. In the field Variable value enter the path to the directory where the JDK is installed, without the semicolon and the /bin subdirectory, for example:  
   **C:\Program Files\Java\jdk1.8.0\_11**
3. Now you are ready to test the install. Close any open command windows, and open a new command window and type:  
   **javac -version**If you see a version number you are done with the JDK install!

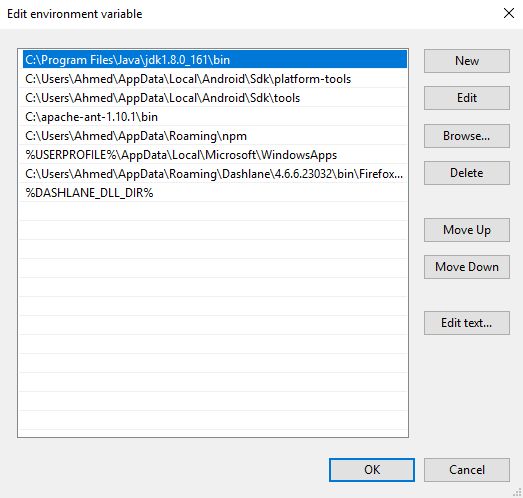
* Install the **Android Tools**, To install the tools needed to **build Android app** using **Cordova** you need the **Android SDK Tools**. The easiest way to install these tools is to install **Android Studio**. Optionally you can install the command line tools only. Follow these steps to install Android Studio:



* Go to the **Android** **Studio** download page and download and install Android Studio for your platform. (If you would wish to go for only the **SDK Tools**, you can find download links to the command line tools at the end of the Android Studio download page.)

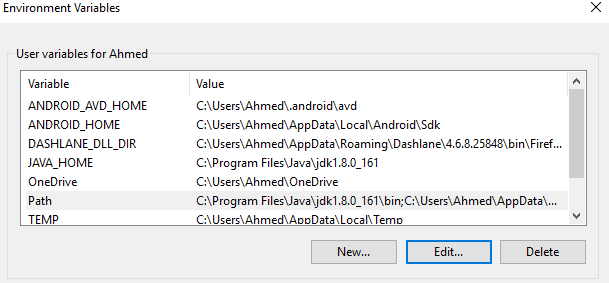


* Find the **path** of the Android SDK tools by consulting the **sdkmanager** documentation page **(Google sdkmanager**). Check in your system that the files are actually there.
* Add the path of the SDK Tools (directories tools and platform-tools to the system **PATH** variable. Open the **Control Panel**, click **System and Security**, click **System**, click **Change settings**, click the **Advanced tab**, then click the **Environment Variables button.**
* In the list User variables select **PATH** and click the Edit button.
* At the end of the field Variable value, add a semicolon followed by the path to the tools and platform-tools directores of the Android SDK install.
* Here is an example of what to add (note that there are two paths in one line, separated by a semicolon):  
  **C:\Users\”Name of the user”\AppData\Local\Android\android-sdk\tools  
  C:\Users\”Name of the user”\AppData\Local\Android\android-sdk\platform-tools**

****

* You can prepare the path in a text editor, copy it and paste at the end of the input field. Click the OK button when done.

1. Next add the **ANDROID\_HOME** environment variable in system settings in the same way that the **JAVA\_HOME** variable was added above when installing Java. Set **ANDROID\_HOME** to point to the root of the Android SDK folder, for example:  
   **C:\Users\”Name of the user”\AppData\Local\Android\android-sdk**



1. Click the OK button again to close **the Environment Variables** window.
2. Now test the install. Close any open command windows, open a new **command window** and type:  
   **adb version**
3. As now we have installed the necessary tools, let’s go ahead and install Cordova. Open the command prompt and type **npm install -g cordova@7.1**
4. Very important hint: don’t install the latest version of Cordova which is 8.0 because it will not work with the Android SDK. Install the version 7.1.
5. You create your Cordova application in the projects folder of your choice with command line: **cordova create YourAppFolder com.YourNameOrCompany.YourApp "Your App Title"**
6. **CD** to **YourAppFolder** and add a device platform. My first platform will be Android: **cordova platform add android** “and that’s where you choose which platform or operating system you want to run the application on”.
7. Add some basic plugins to your project:   
   **cordova plugin add cordova-plugin-device  
   cordova plugin add cordova-plugin-media  
   cordova plugin add cordova-plugin-geolocation**
8. An **Amber** application must live within the **Cordova** program structure, i.e., it replaces the existing contents of the **‘www’** folder of the Cordova application. This is because Cordova (**cordova**.**js**) is not a proper JavaScript library that can be imported into Amber; it has a great many dependencies on other parts of Cordova. So after you’ve created your Cordova application, erase the ‘**www’** contents, **CD** into ‘**www’** and run **‘amber init’.**
9. Building for Android, here is how you build your new Cordova project for Android. In the **command line** type:   
   **cordova build android.**
10. This command will launch the Android emulator depending on what kind of device you set up to. For example, it could be nexus, galaxy… etc. The Smalltalk application should run without errors.
11. If you would like to test the application on a real Android smartphone. After you the emulator runs, go to

**C:\Users\”Name of the user”\YourAppFolder\platforms\android\CordovaLib\build\outputs**

1. Copy the **.apk** file and paste it in the **File** **Manager** in your device that’s linked to the computer through the **USB** cable after enabling the feature of installing “**untrusted**” apps from external sources (which is not directly from the **PlayStore**)
2. Restart your smartphone, and you will see the application in your phone ready to test it.