Lab 10

Hybrid Mobile Application Development

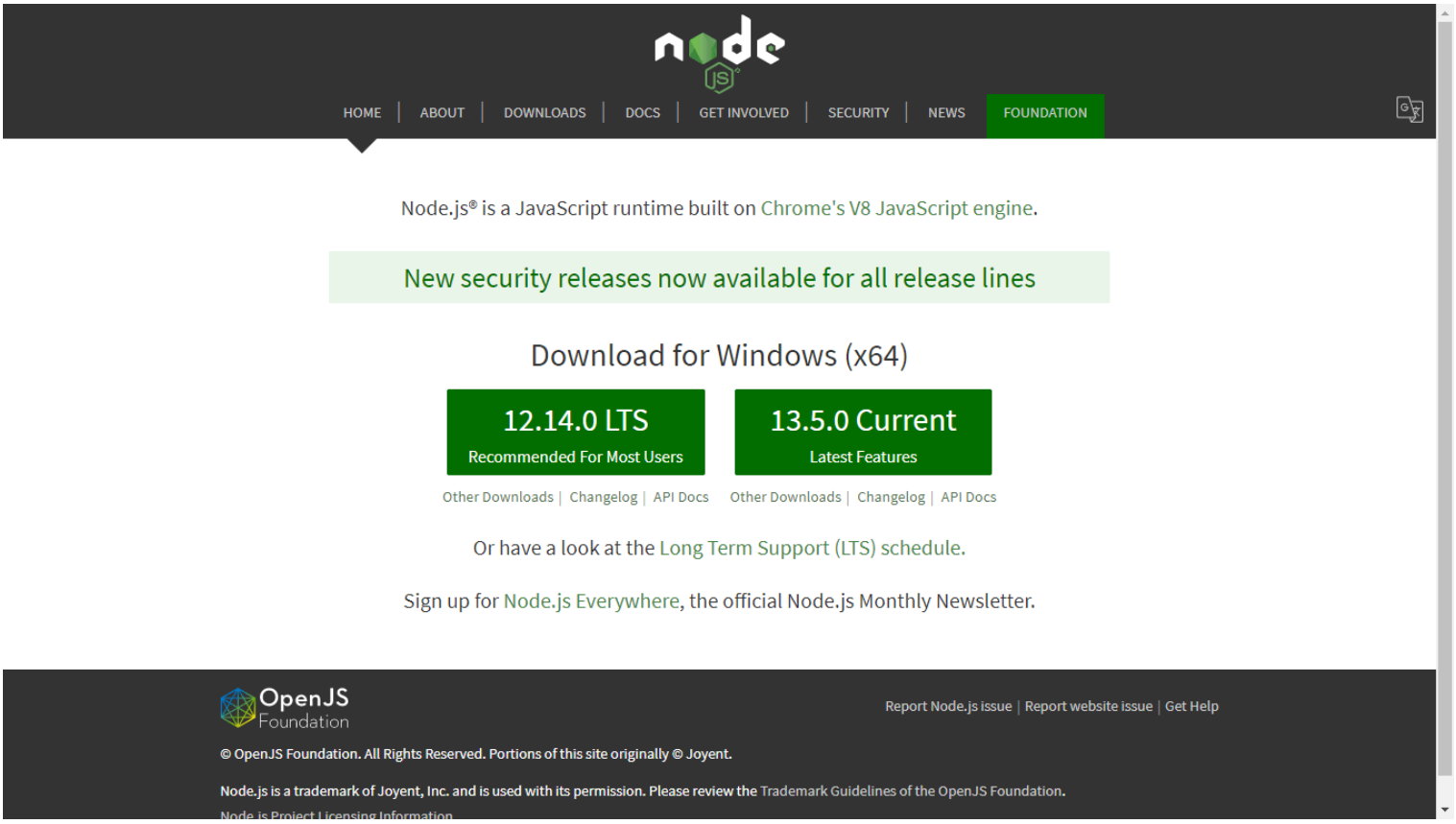
Cordova Installation

To install the Cordova command line tool, you must have to follow below steps:

Cordova basically runs on the **Node.js** platform. So, you have to install **Node.js** as the first step.

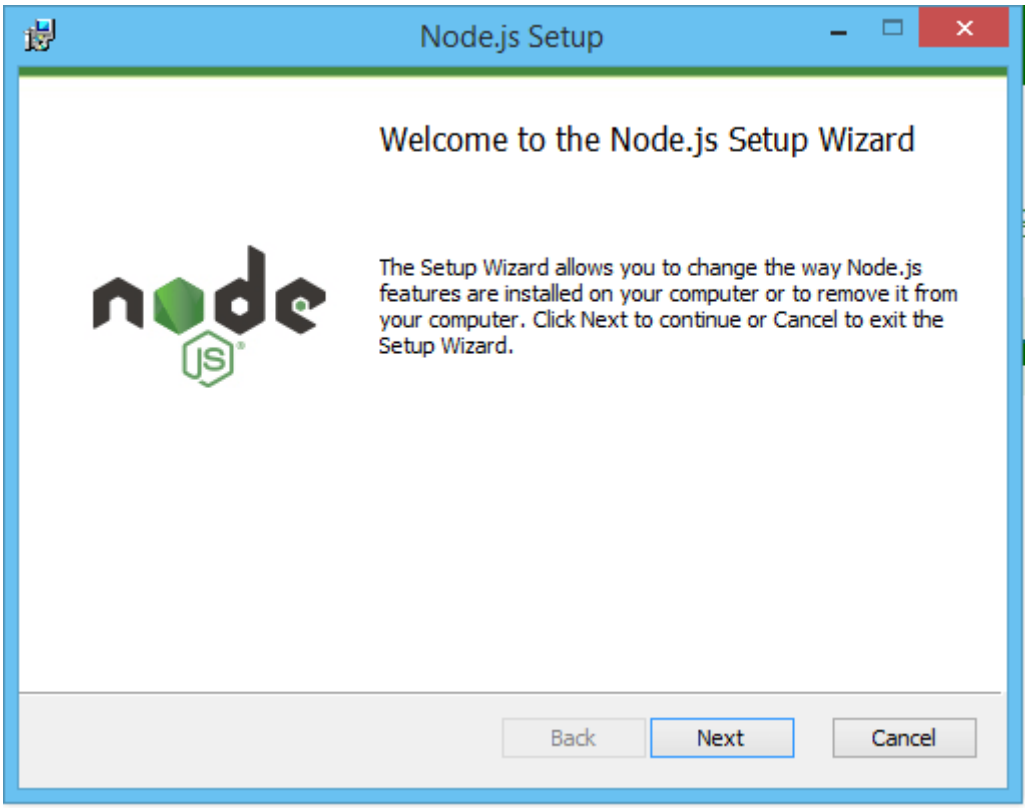
Installing Node.js

Step 1: Here is the link where you can easily install it -> http://nodejs.org The window that appears after clicking on this link will be shown as follows:



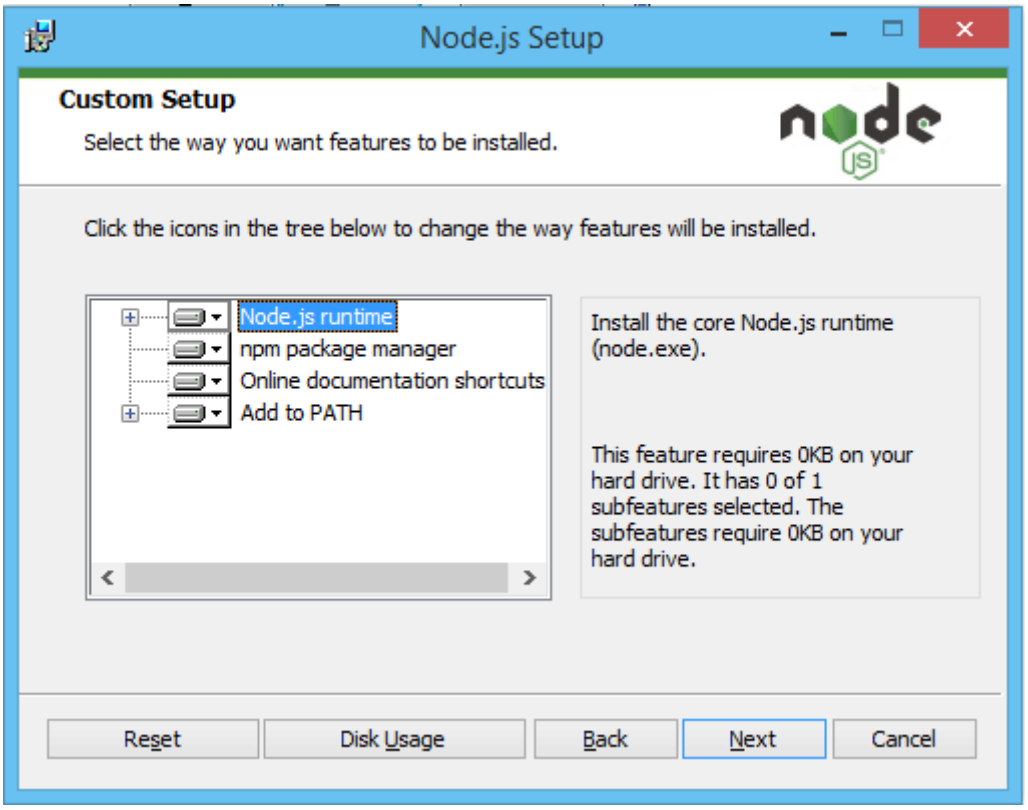
You can install **Node 12.14.0 LTS,** which is recommended for most users.

After completing the installation, a pop-up window will appear. Consider the below screenshot:



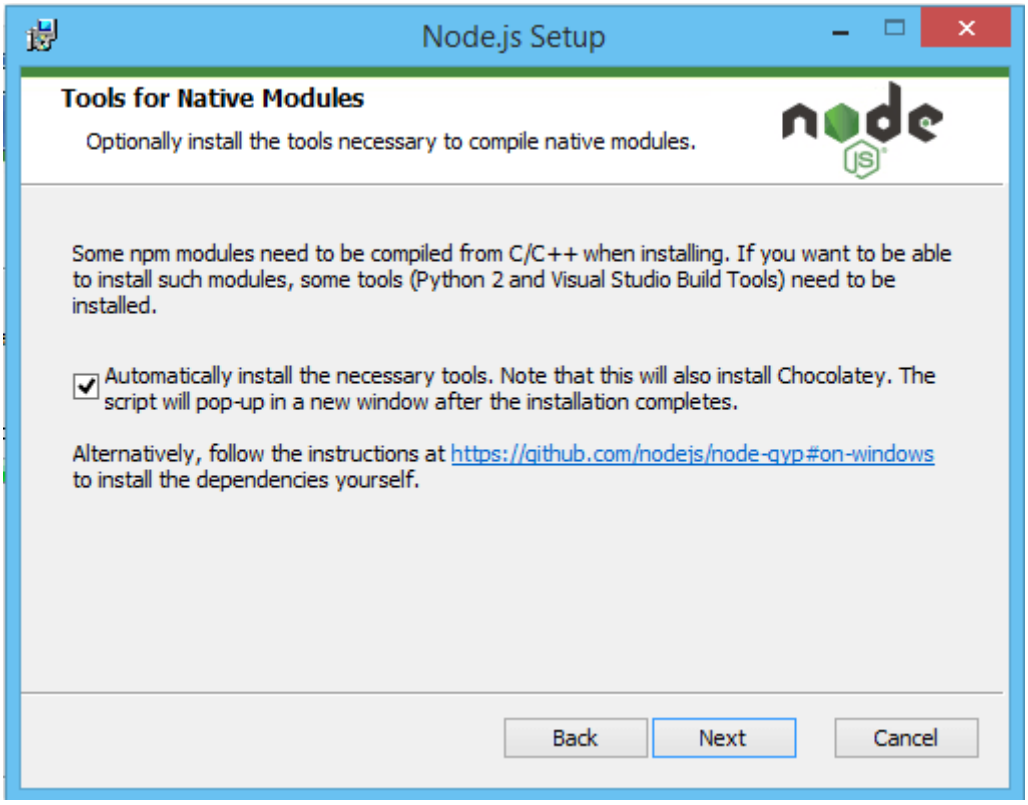
Click on the **Next** button to continue.

Step2: Select the features that you want to add in your application.



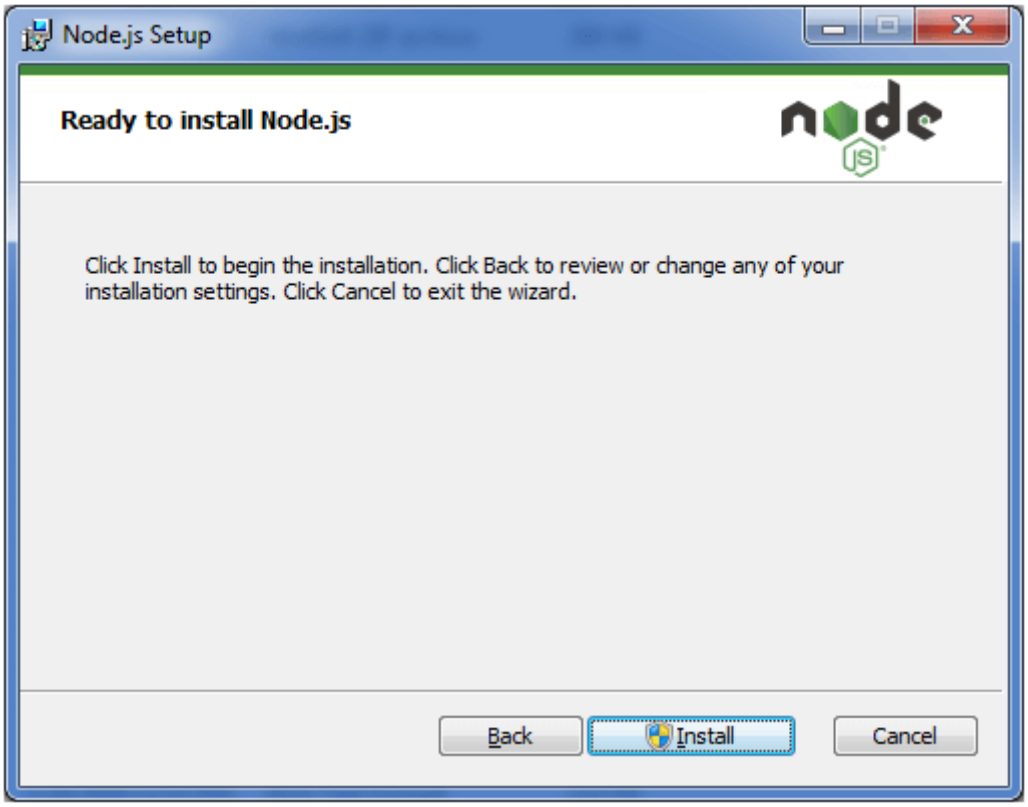
Click on the **Next** button to continue.

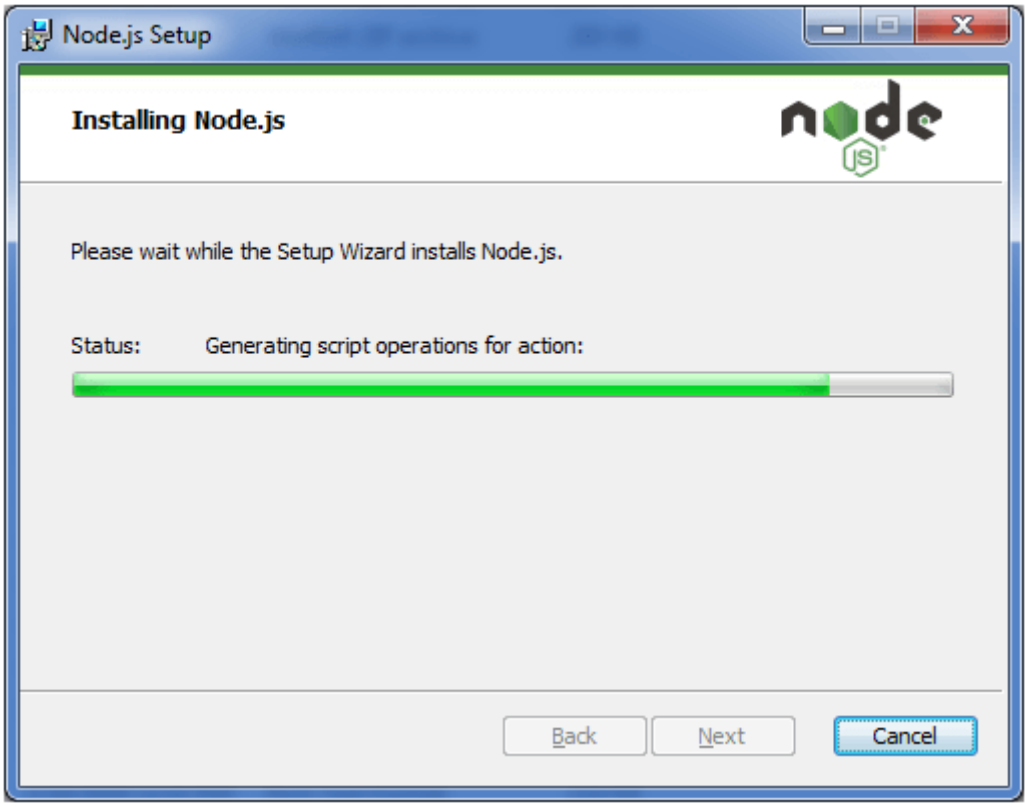
Step 3: You can install some additional tools that are necessary to compile native modules.

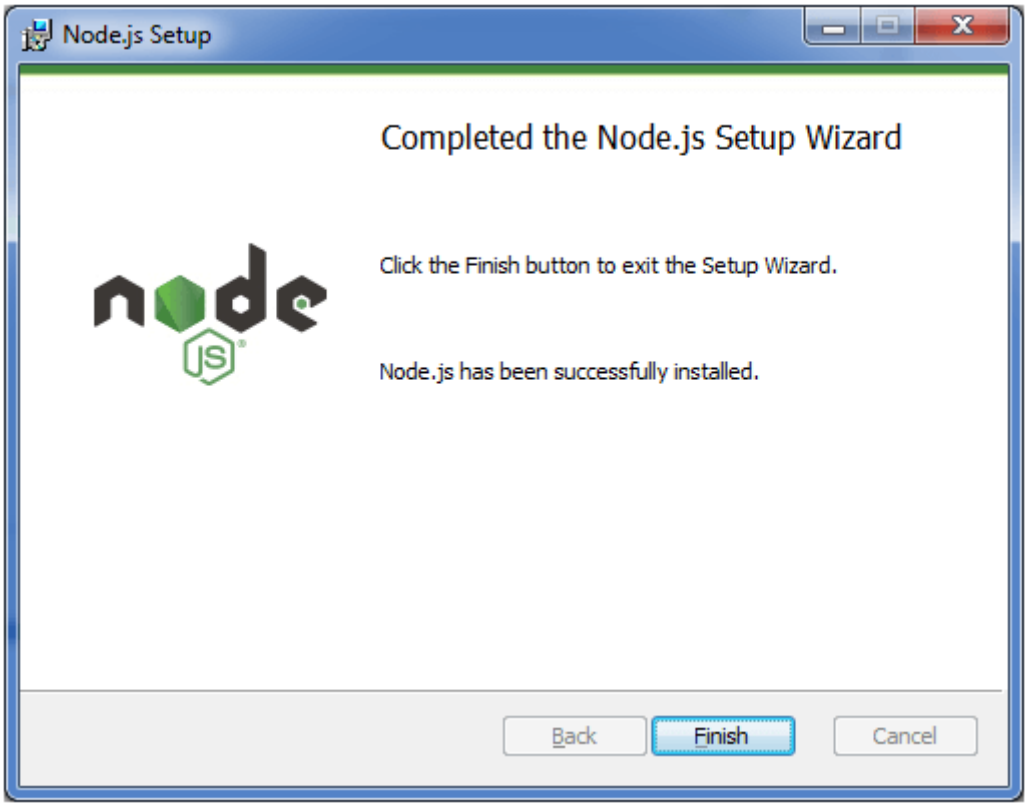


Click on the **Next** button to continue.

Step 4: Click on the **Install** button to begin the installation.







If you want to test the installation, you can write following command on the command prompt:

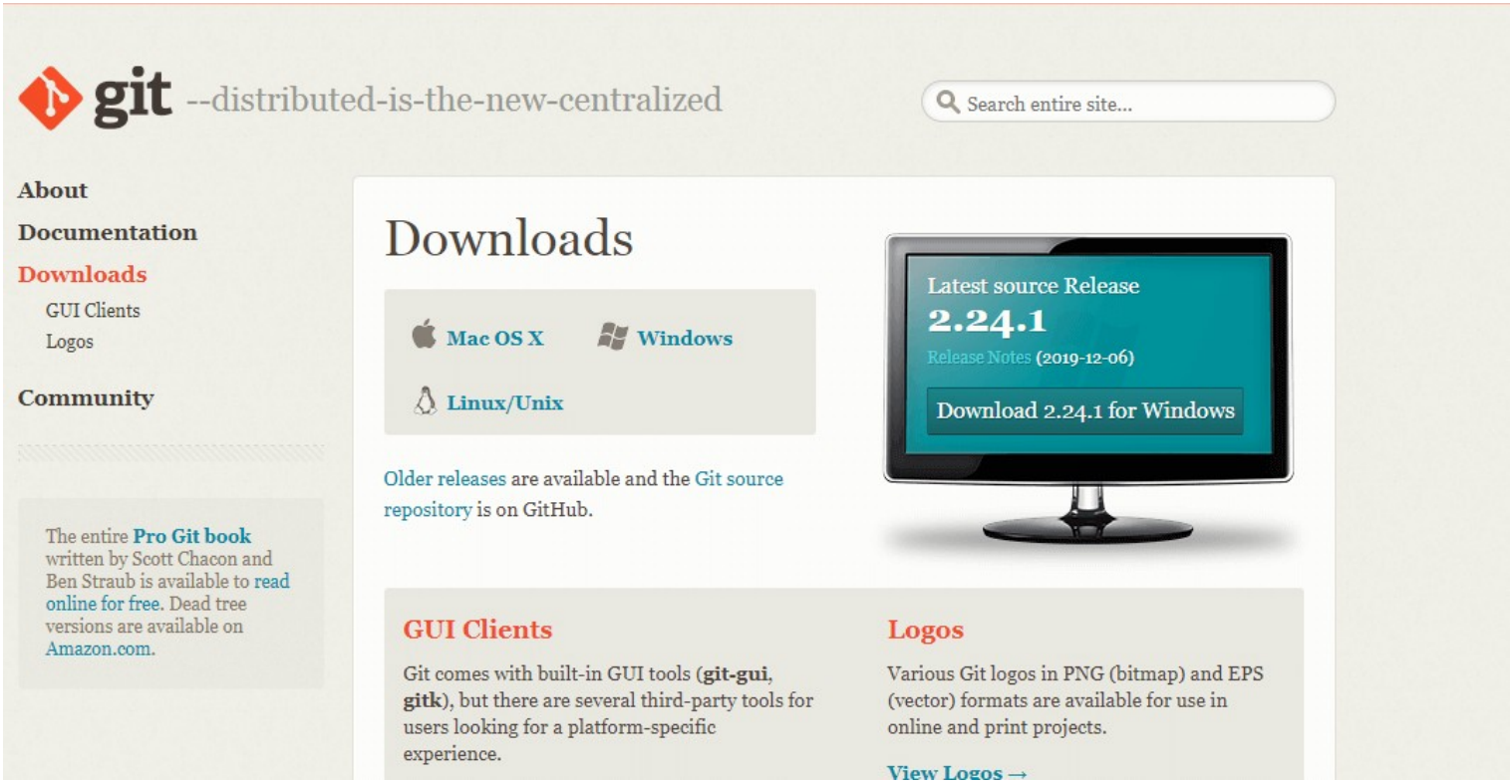
1. node --version

If the version number is displayed on screen, it means that **Node.js** is properly installed on your system.

Installing Git:

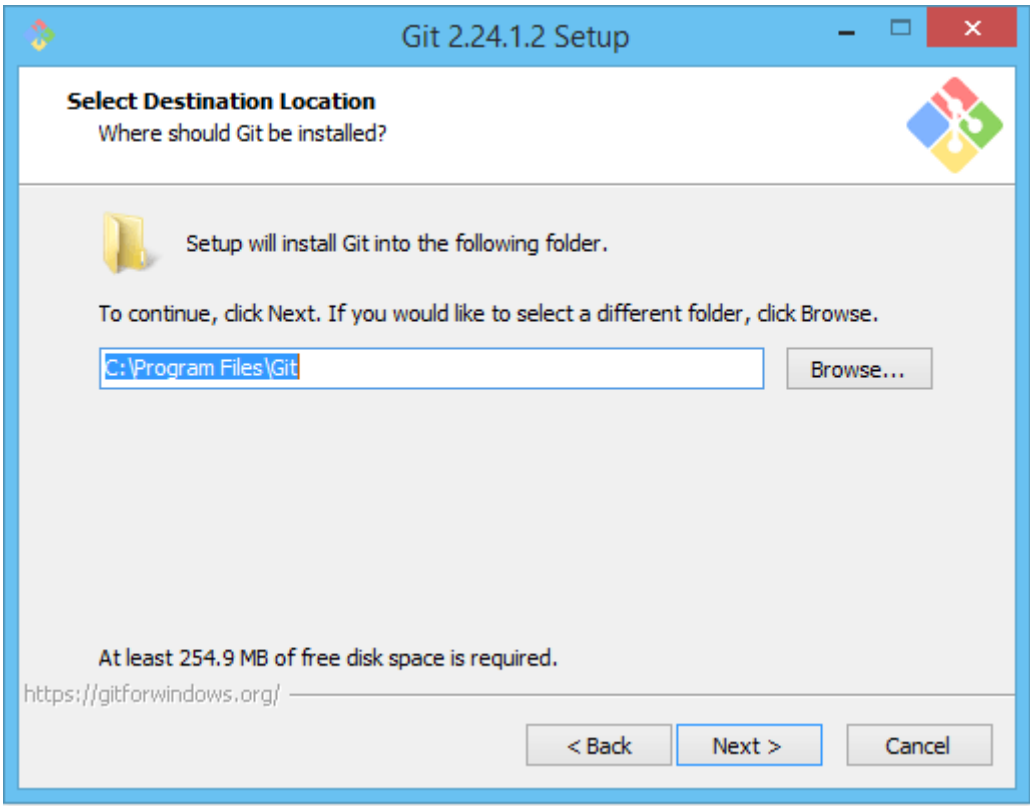
Step 1: To install the Git, you can visit its official website that are as follows: http://git-scm.com

Go ahead and follow the instructions on its official website. The web page will appear as:



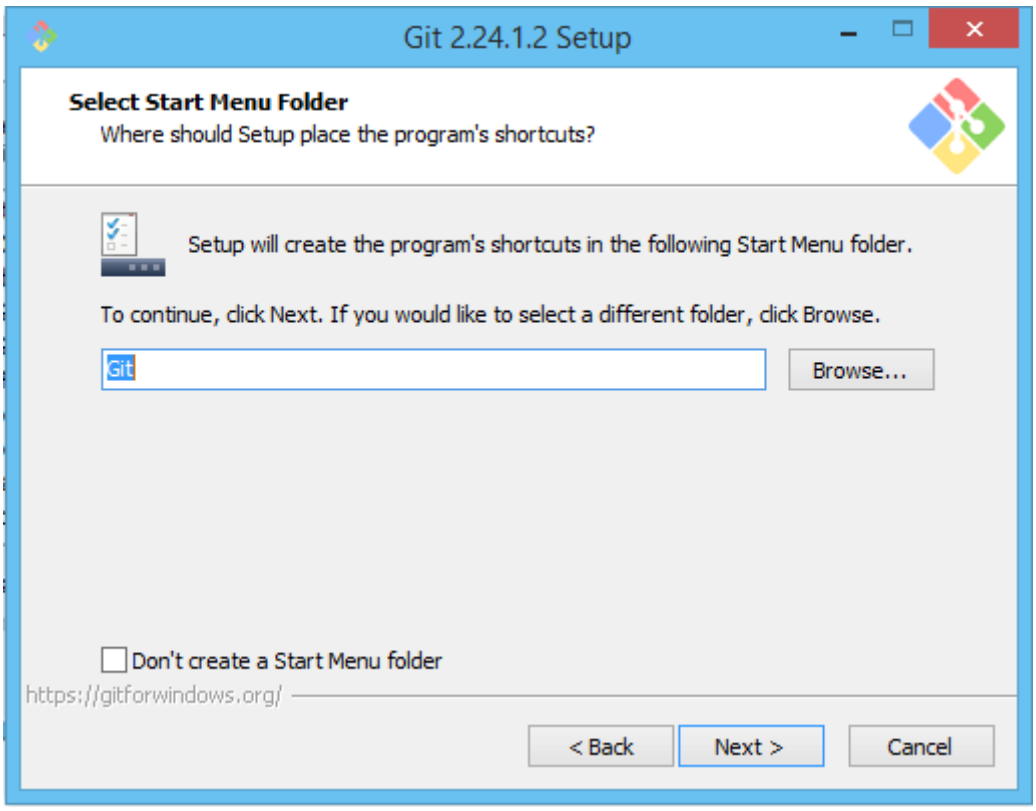
Step 2: Click on the latest release version **2.24.1** for Windows. The pop-up window will be shown after installation:

Accept the license agreement and click on the **Next** Button to continue.

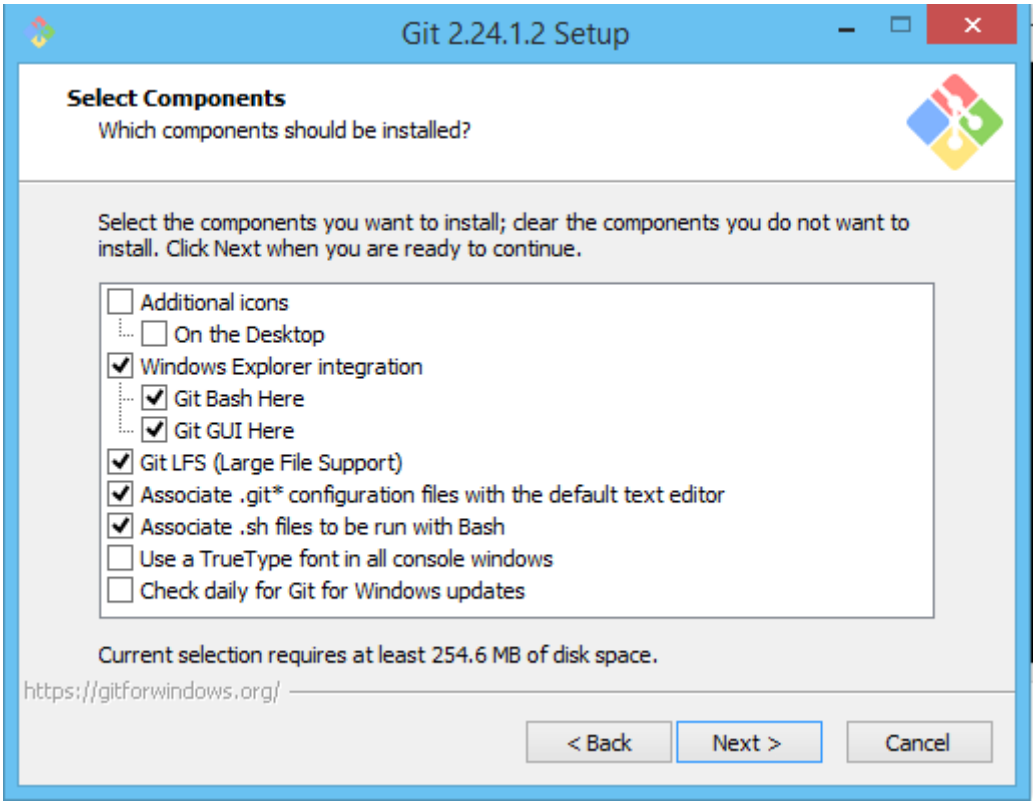


Step 3: Here, you can select the path where you want to install **Git** and then click on the **Next** button to continue.

Step 4: After that, you can specify the shortcut name that is displayed for the Start Menu folder and then click on the **Next** button to continue.

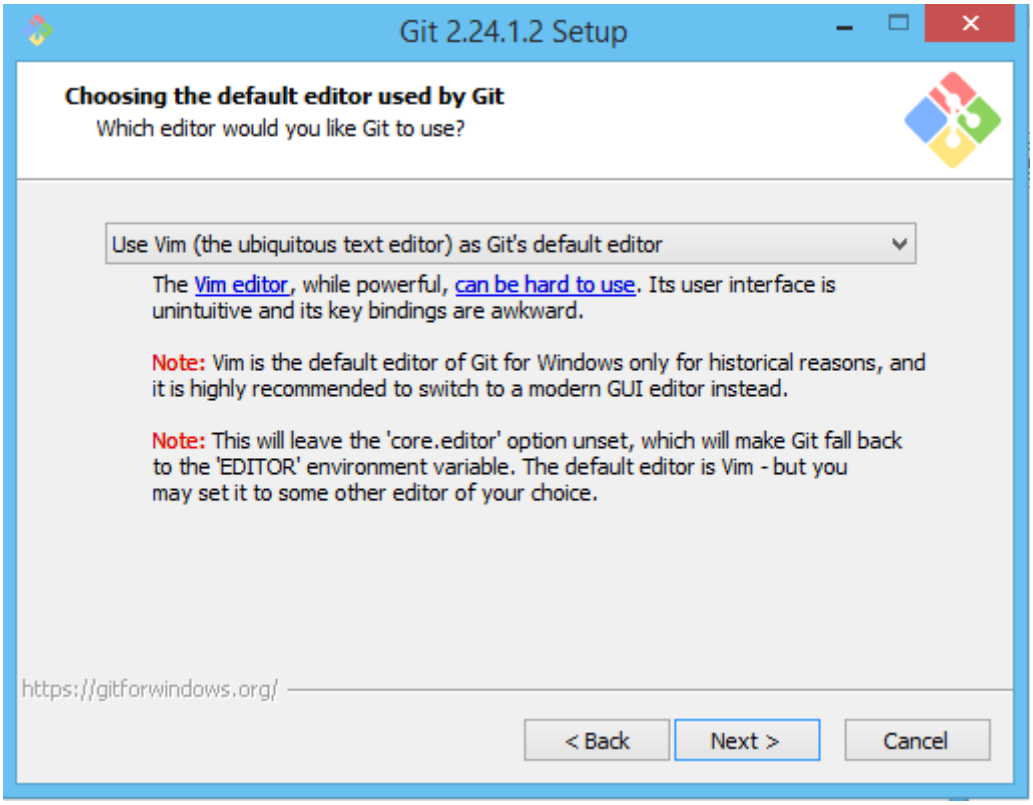


Step 5: Default components required for installation are automatically selected. You can also select your additional components.



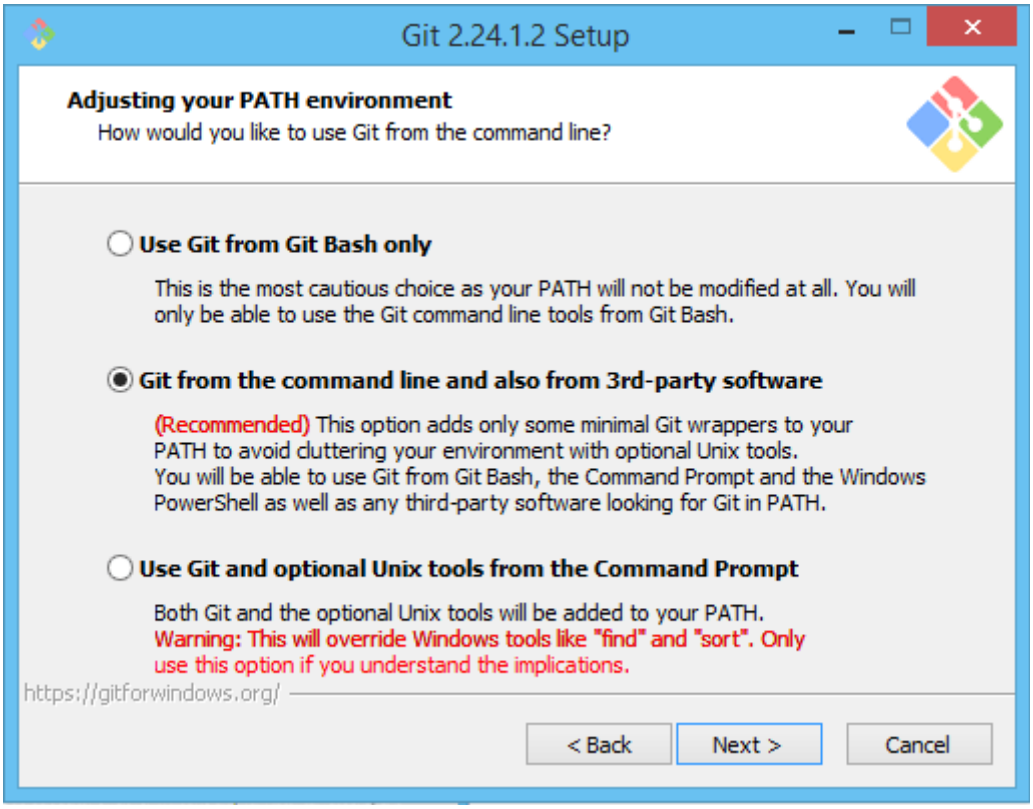
Click on the **Next** button to continue.

Step 6: You can choose the editor for using the Git. By default, Vim editor is being used.



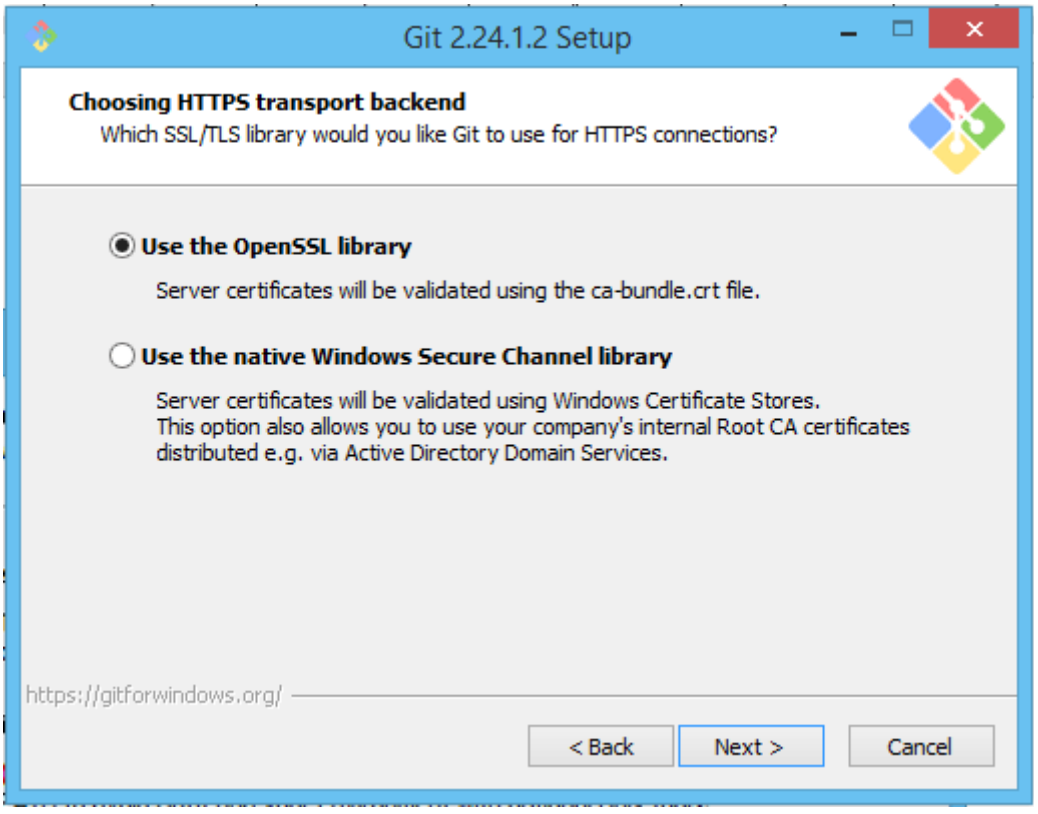
Click on the **Next** Button to continue.

Step 7: You can adjust the path environment of Git from the below three options. The default path for the Git installation is from the command line and 3rd party software.



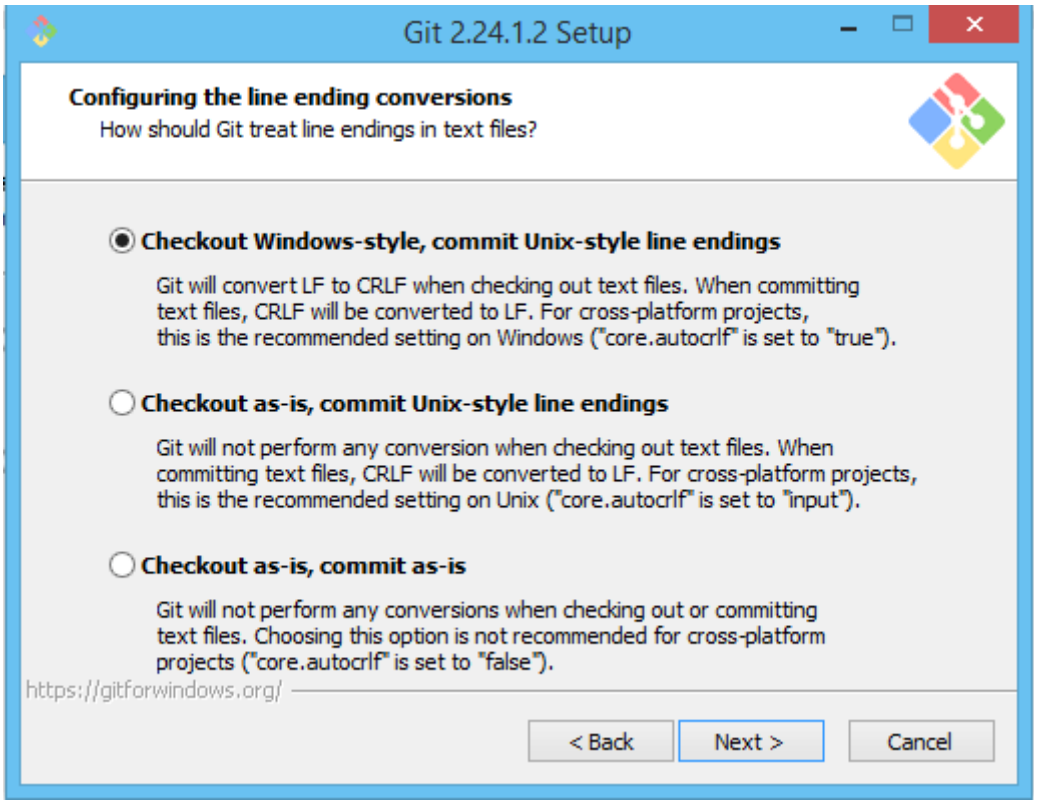
Click on the **Next** button to continue.

Step 8: The default library is being selected for the HTTPS connections.

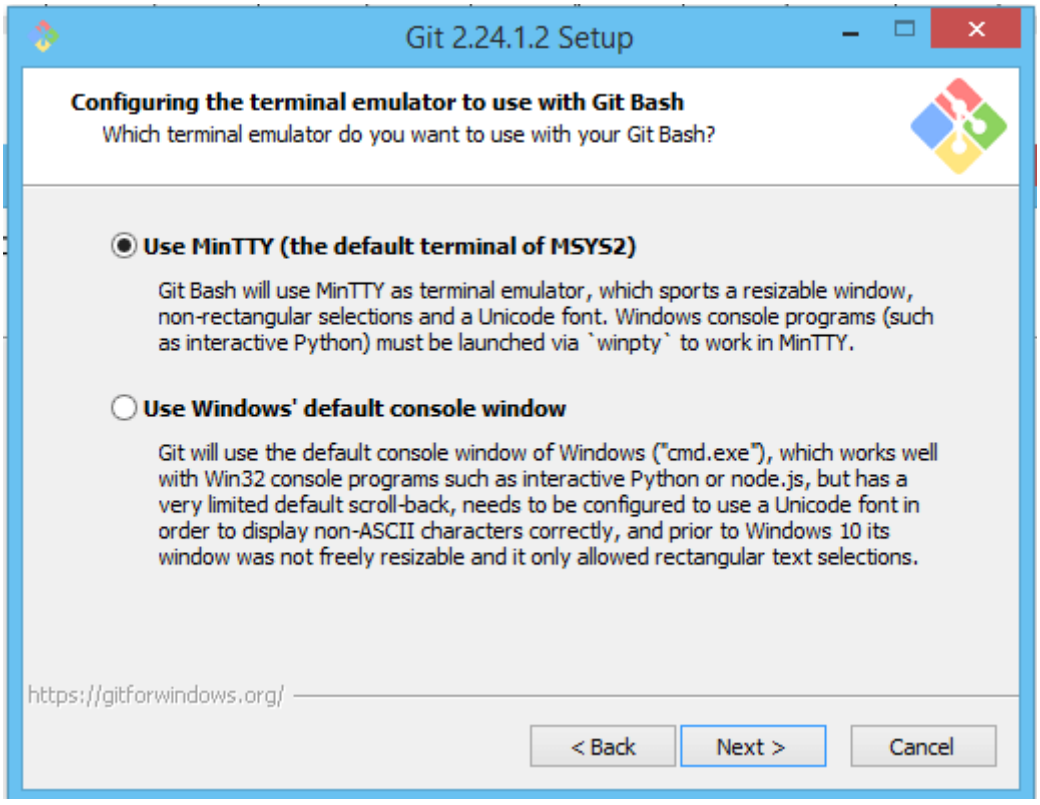


Click **Next** to continue.

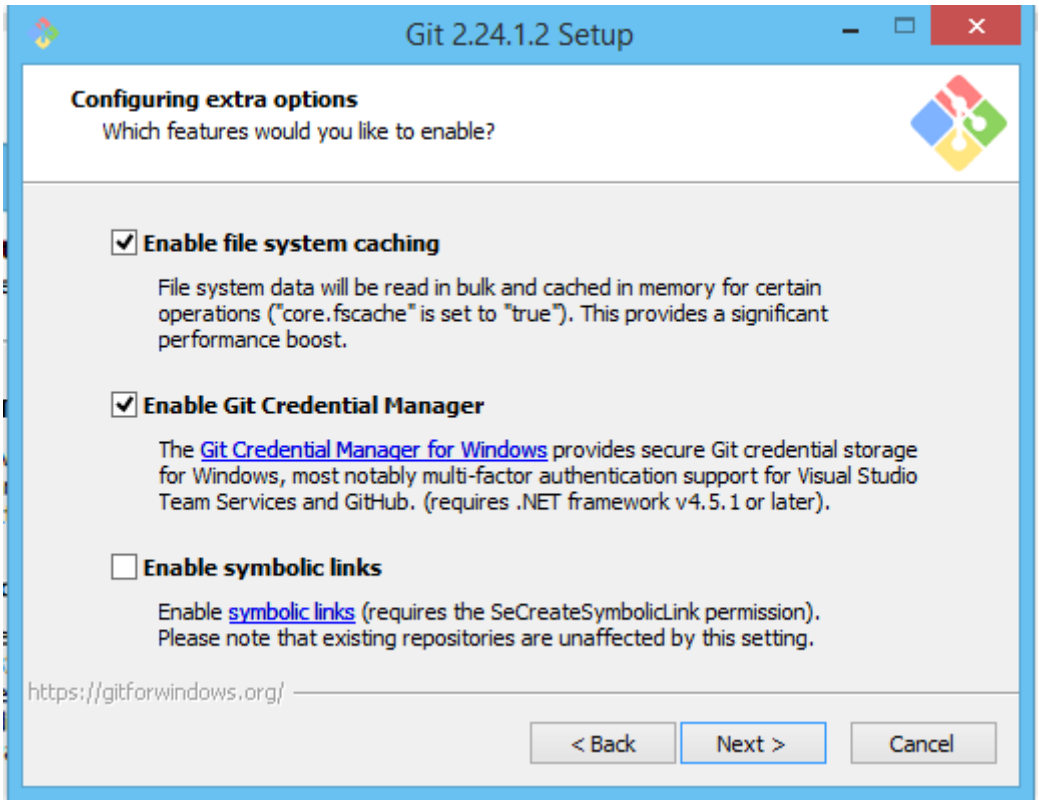
Step 9: Select line ending options and then click on the **Next** button.



Step 10: Select the terminal emulator that is to be used with Git Bash and click on the **Next** button to continue.

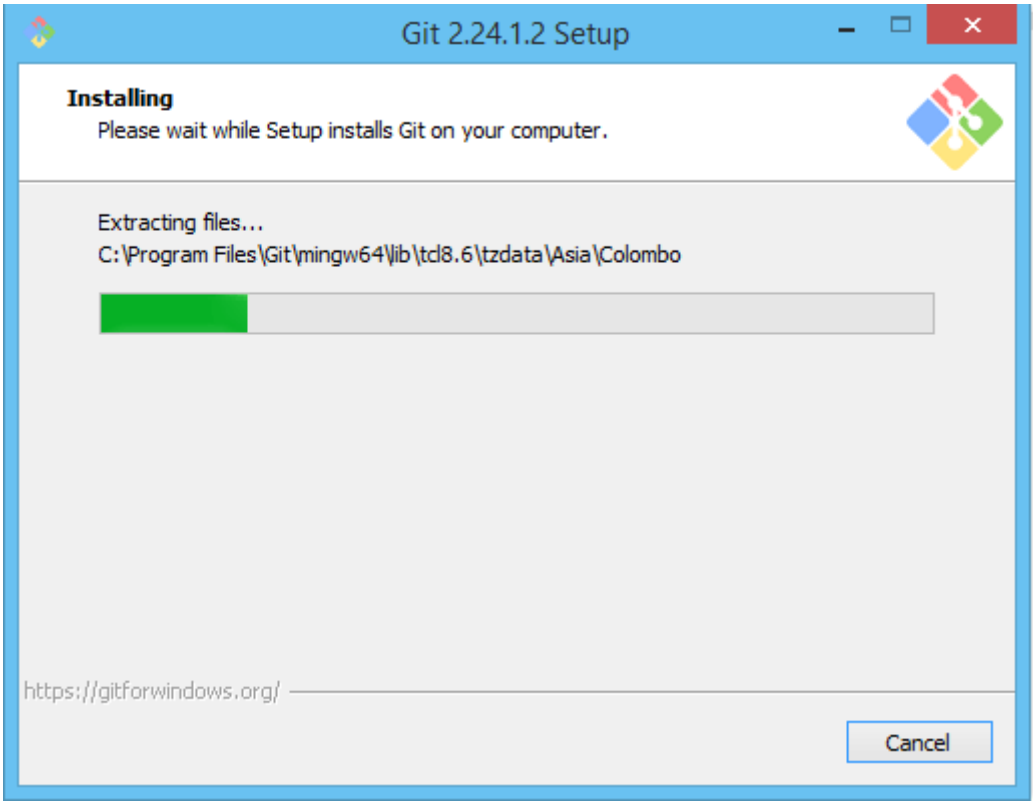


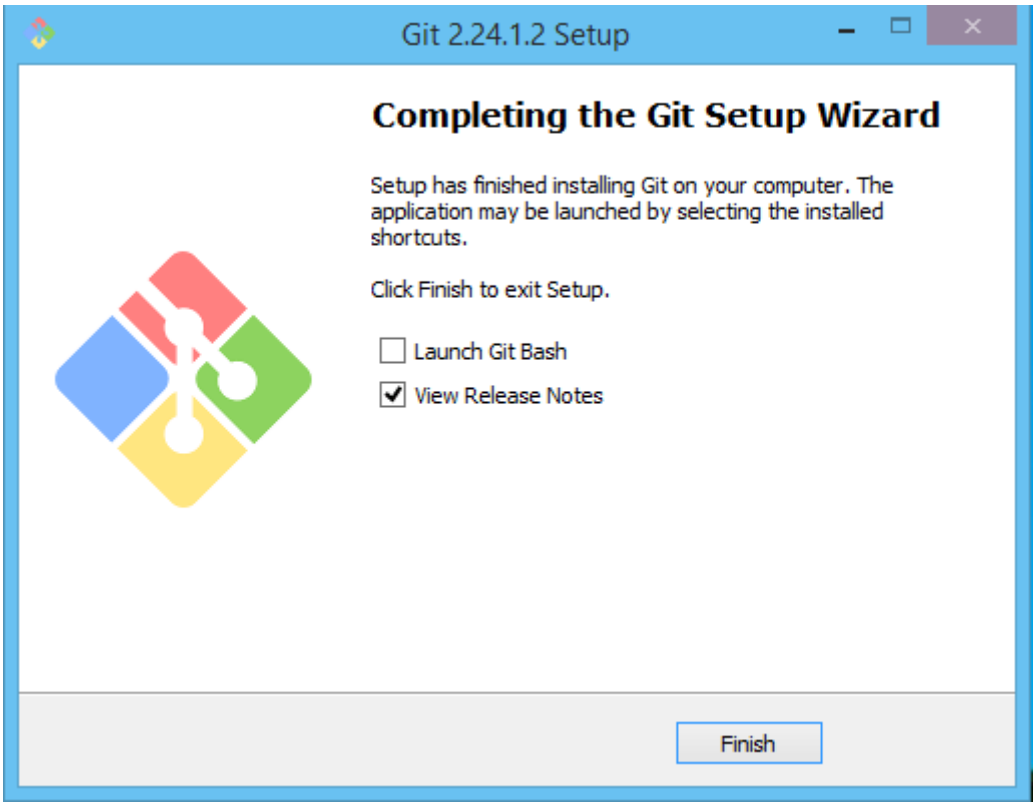
Step 11: This is the last step that provides some additional feature that you want to add in your Git application.



Click on the **Next** button to continue.

Now, the installation will begin:





Installing Cordova:

You can install the Cordova module by using the Node Package Manager(**npm**), utility of Node.js. This module will automatically be downloaded from the **npm** utility of Node.js.

On Windows:

You can run the following command in command prompt to install the Cordova application on Windows:

1. C: \>npm install -g cordova

This **-g** flag is responsible for installing the Cordova module globally. Otherwise, it will install Cordova in the **node_modules** subdirectory of current working directory.

After installing it, you can run cordova on the command line with no arguments. If you want to check the version of cordova, you can simply check it from the below command:

1. cordova --version

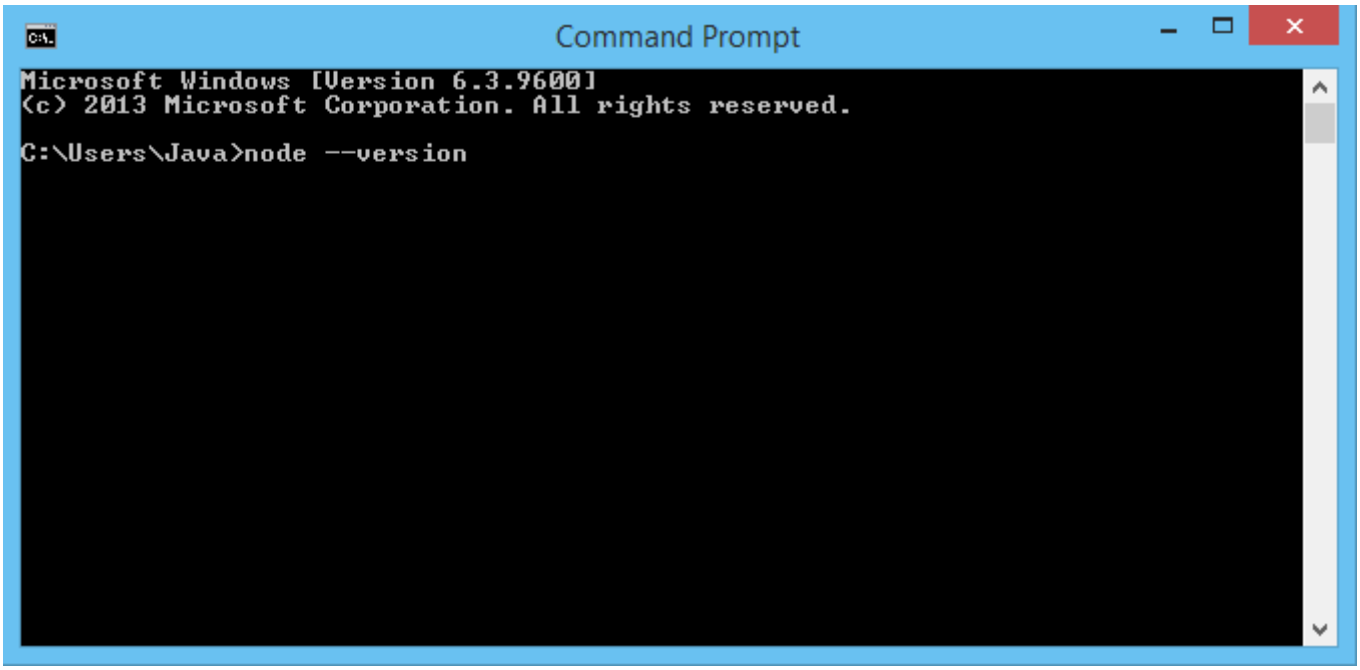
If you see the version number in the command prompt that means your cordova application is properly installed.

Creating first Cordova Application

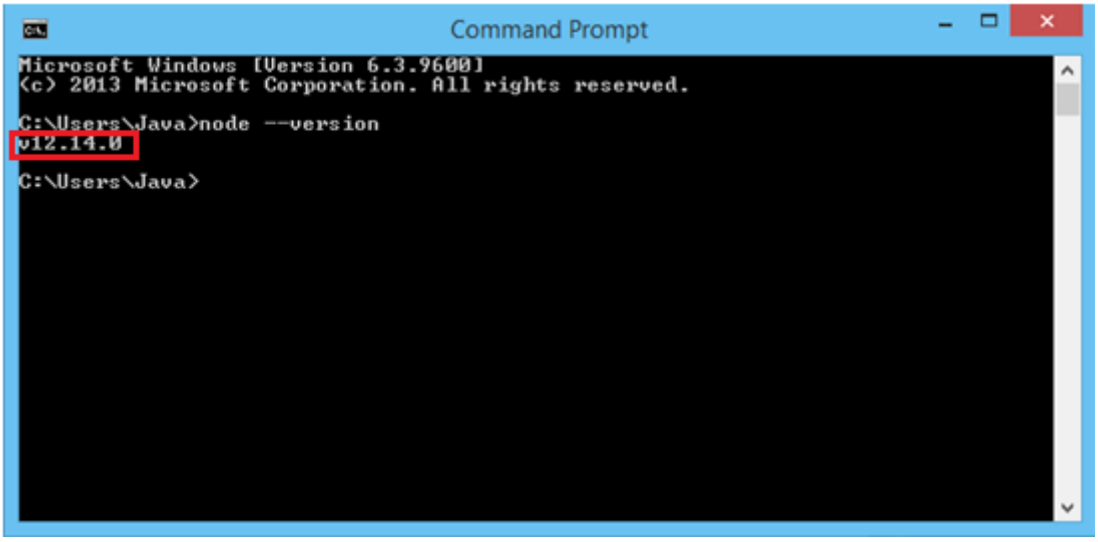
All the required components for creating a Cordova application are already installed. So, let's create the first Cordova application. For creating a Cordova application, you must follow these steps:

Step 1: First of all, check the node.js version in command prompt to check whether it is properly installed or not.

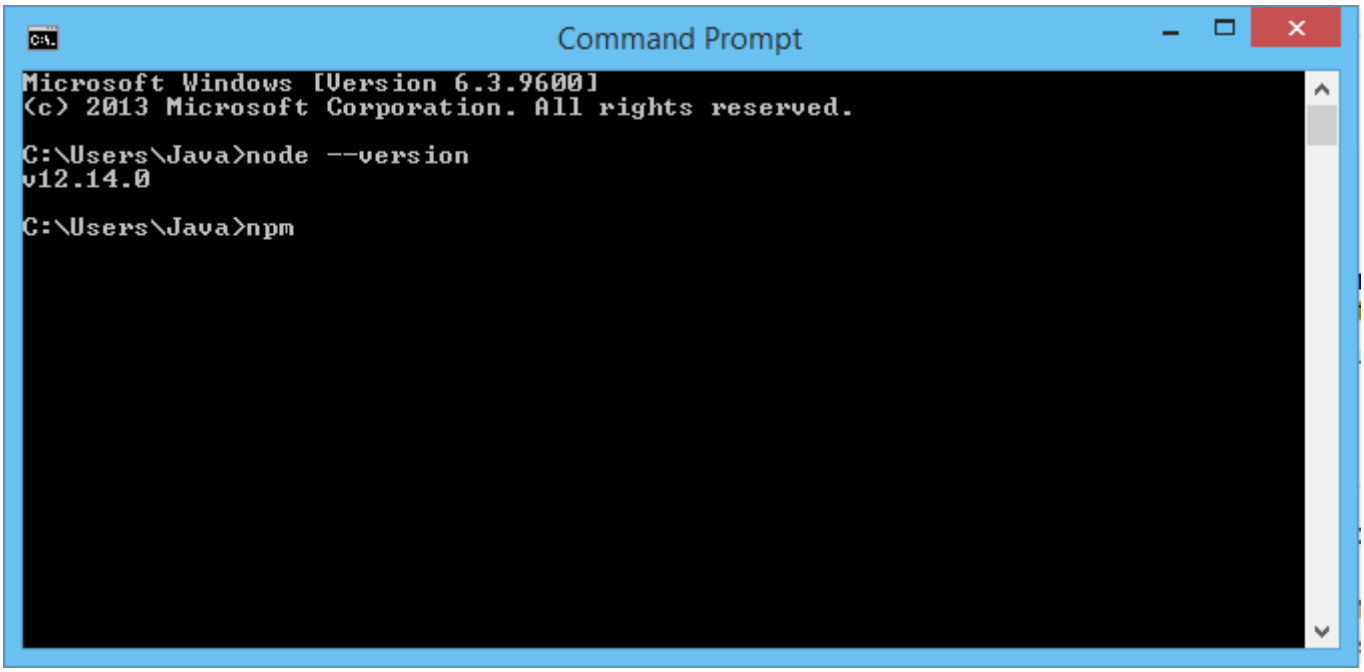
Type the following command in command prompt **-> node --version**



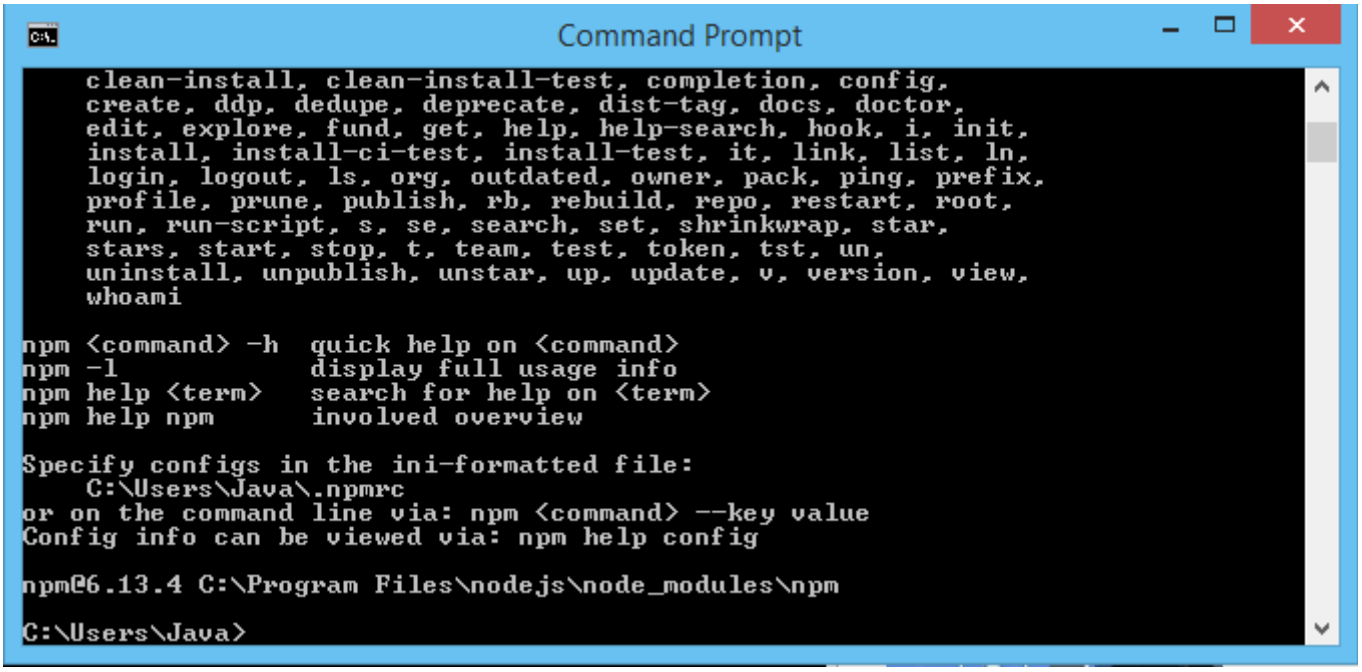
If it shows the current version of node.js application, it means that node.js is properly installed on your system.



Step 2: Then, type **npm** on the command prompt.



If the **npm** module is properly installed on your system, then the command will be executed:

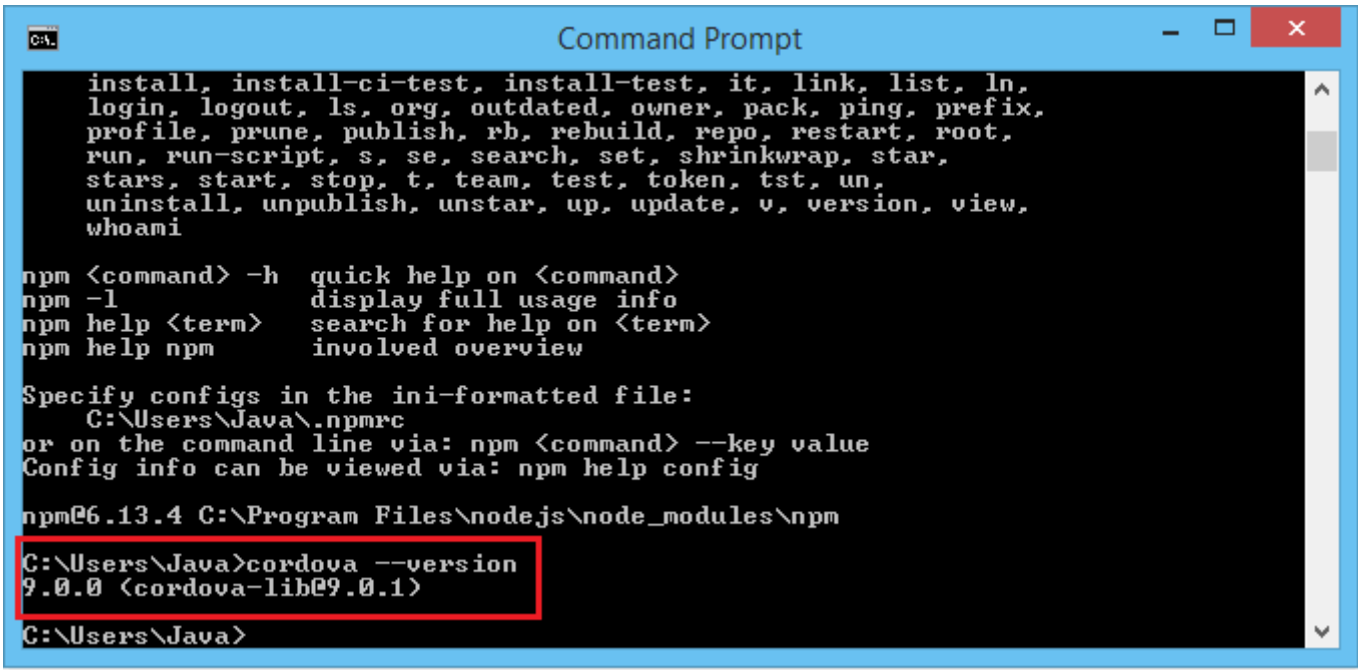


Step 3: We have already installed Cordova in our system. If you did not install it, you can simply do that by this specific command->

1. npm install -g cordova

This command install Cordova in your system. You can also check the version of Cordova by using the below command:

1. cordova --version



Step 4: Now, create the Cordova application on the desktop. First, we have to change our default directory to the desktop.

Type the below command on command prompt to change the default directory.

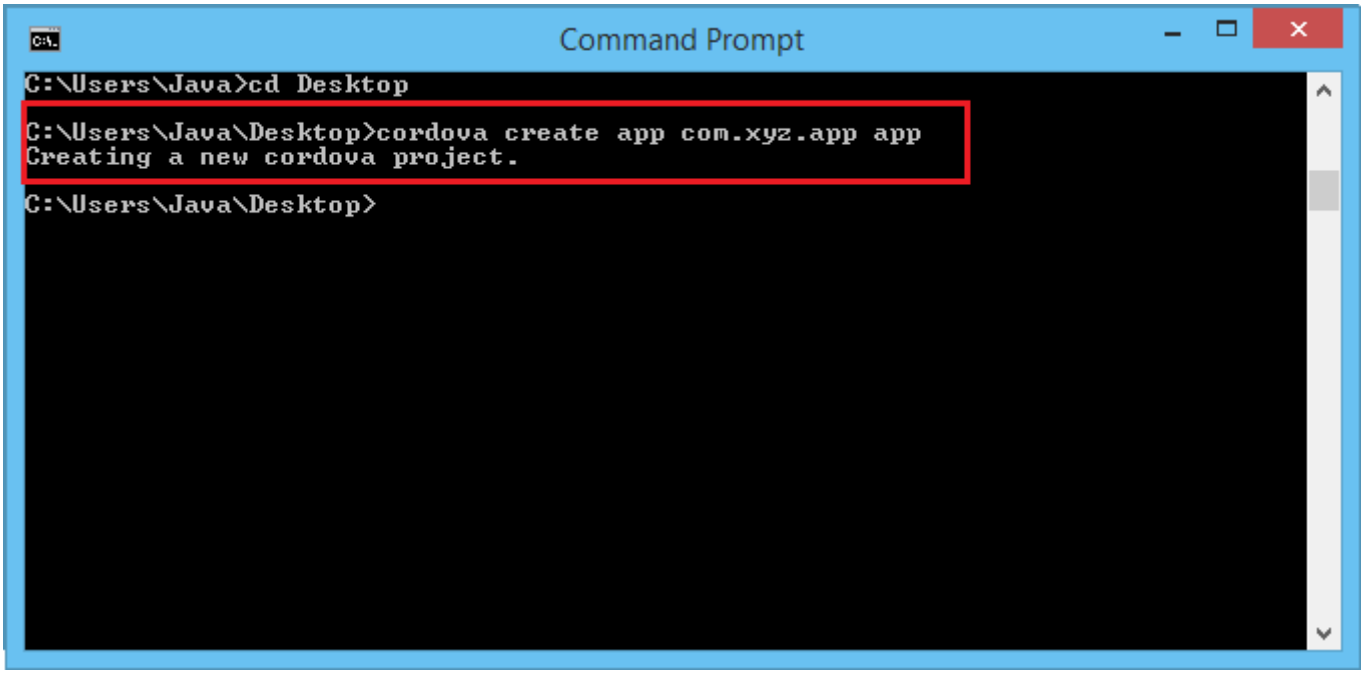
1. cd Desktop

The above command is responsible for creating our application on the Desktop.

Step 5: To create our first Cordova application, we have to write the below command on the command prompt.

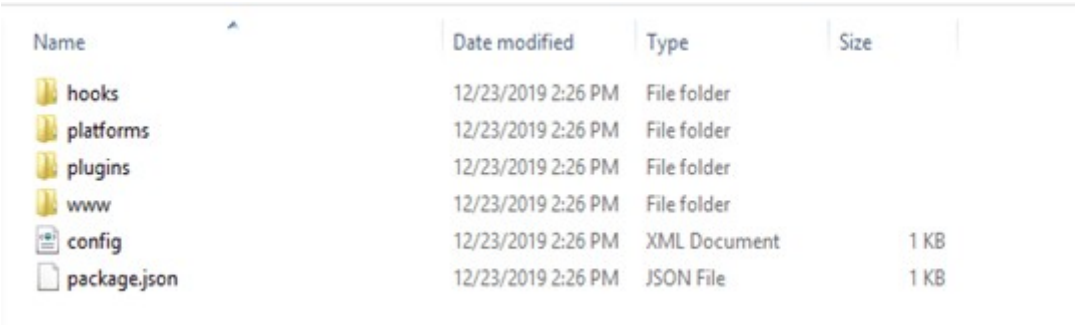
1. cordova create app com.xyz.app app

The above command creates our first application on the desktop with necessary files, folders and directory structure.



- **com.xyz.app** specify the domain name of the application in reverse notation.
- **app** defines the name of application.

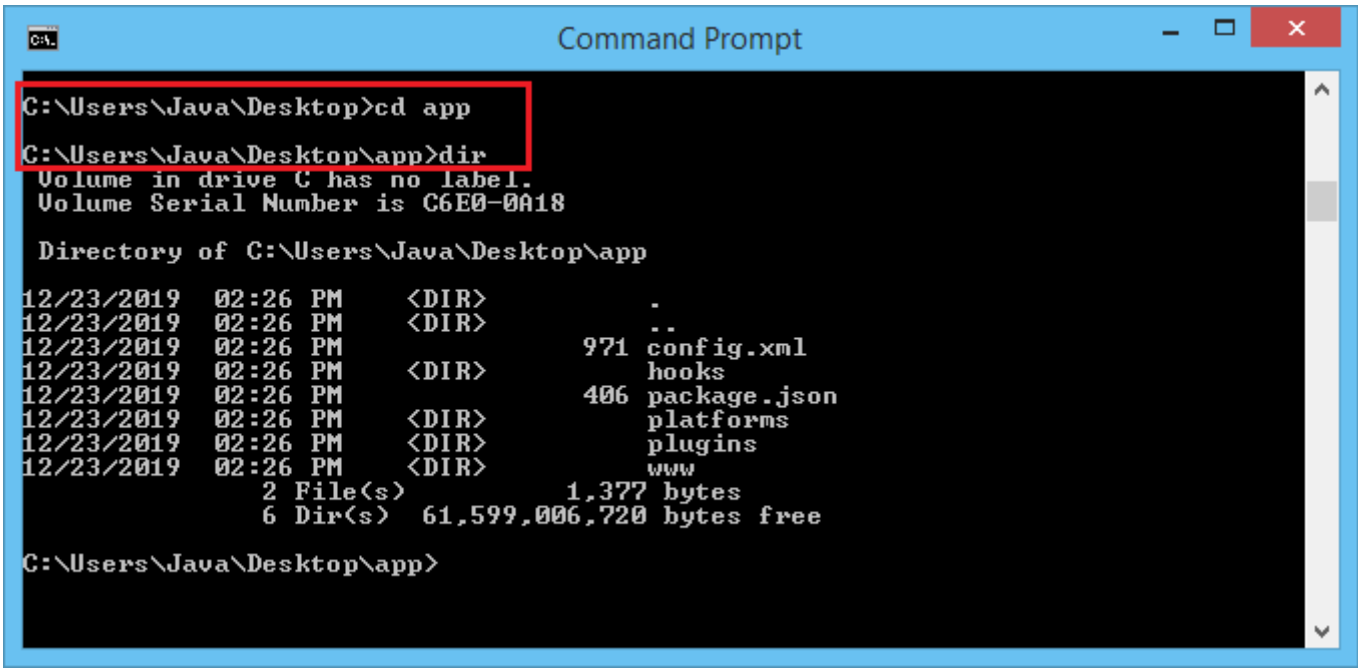
The above command makes a new folder **app** on the desktop that contains these folders:



Add Platform

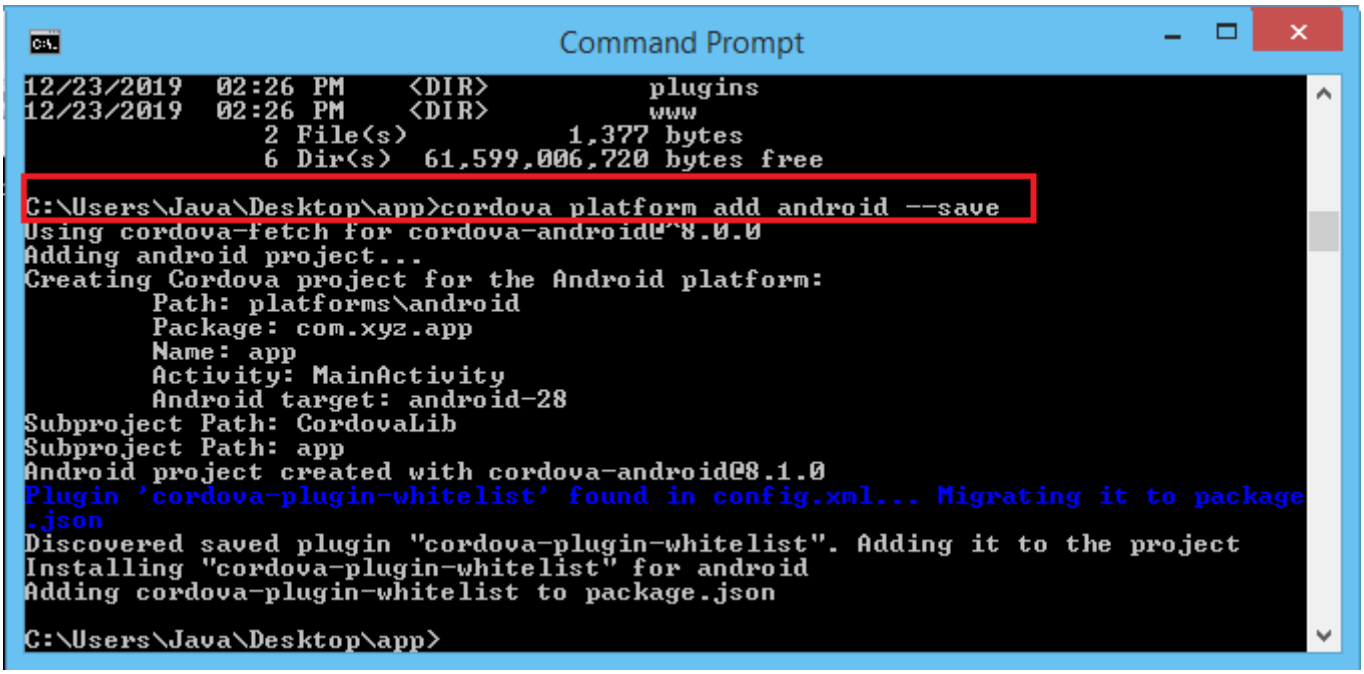
If we want to add any platform in our application, we can simply do it by following these steps:

Step 5: First of all, we have to move to that directory that we have created for building the application i.e., **app** and type **dir** for showing the current directory.



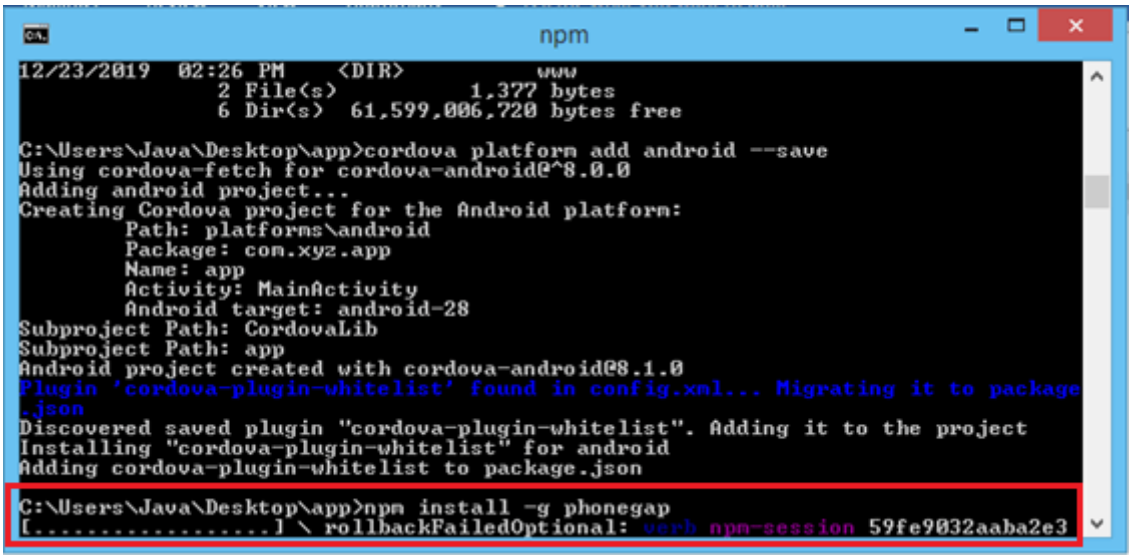
Step 6: By adding a new platform, we can simply deploy our application. Type the below command to create an Android platform.

1. cordova platform add android --save



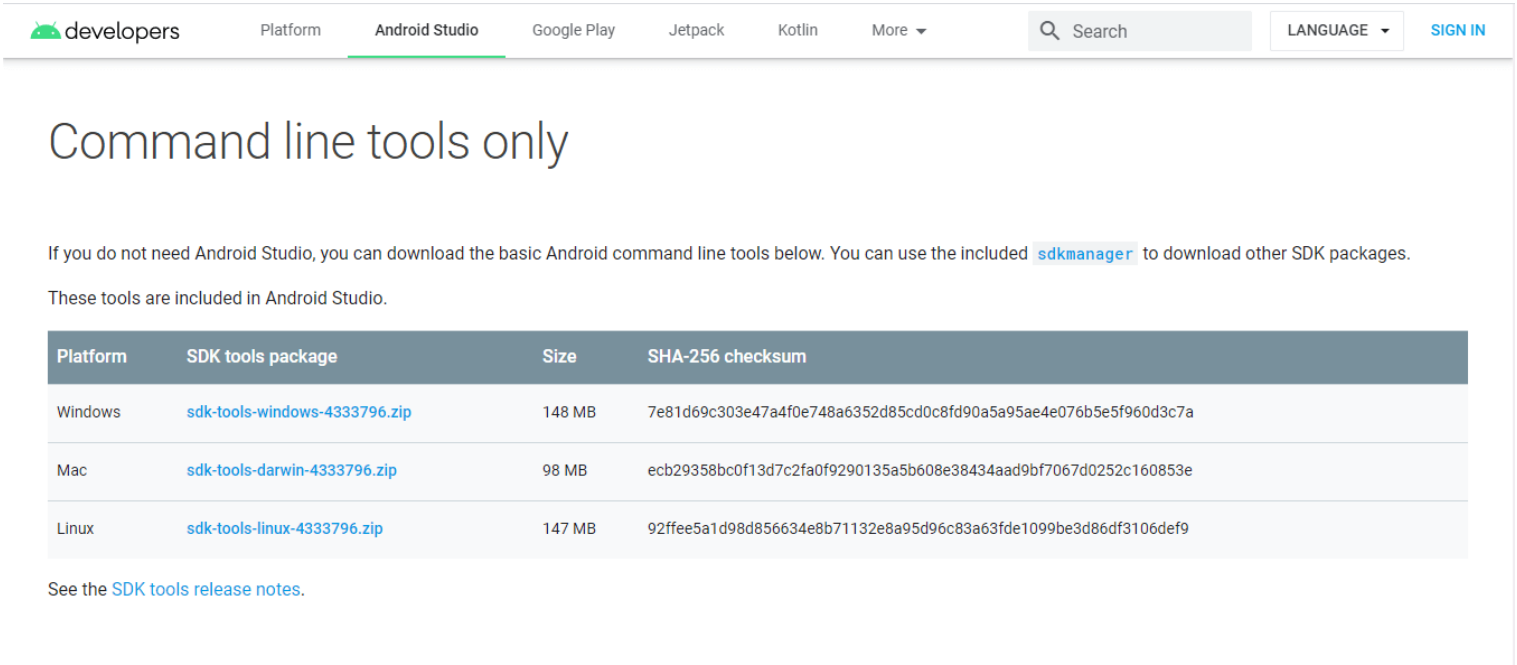
Step 7: We also need to install PhoneGap for serving the application. To install the PhoneGap module, we must have to type the below command:

1. npm install -g phonegap

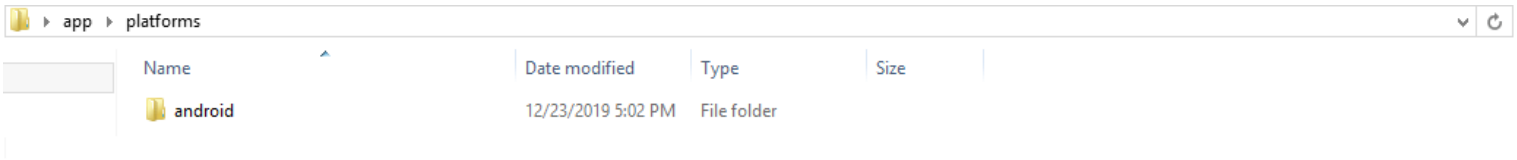


Step 8: After installing PhoneGap, we also need to install the Android SDK for adding a platform to our application. We can install it from its official website https://developer.android.com/studio.

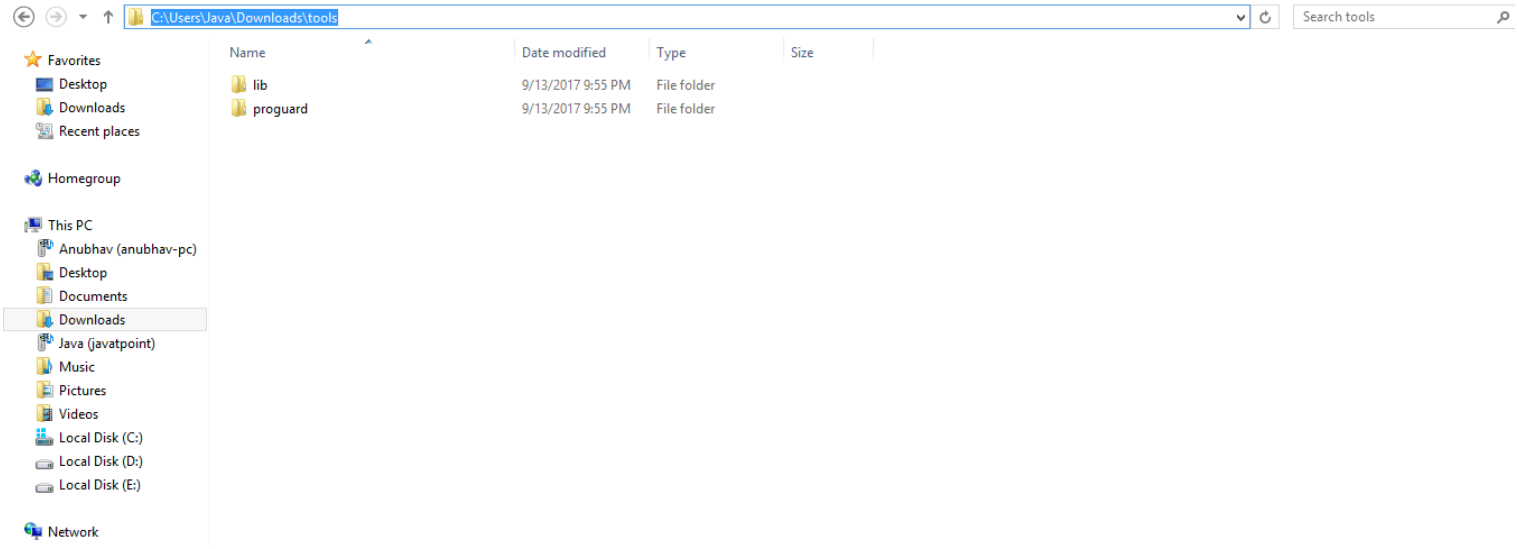
At bottom of the page, we can find **command line tools** and download the Android SDK tools package for windows platform.



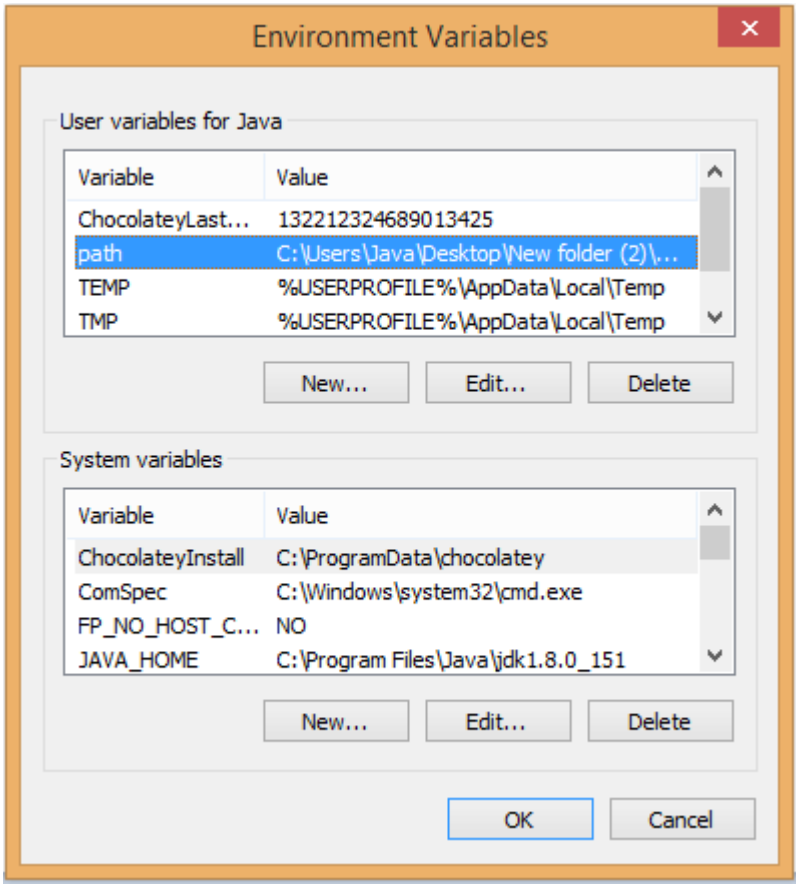
After installing Android platform for our application, we can a see that a new folder is created in **platforms** folder.



Step 9: Go to that folder where you have installed **Android SDK** in your system and click on **tools** folder and copy its path.

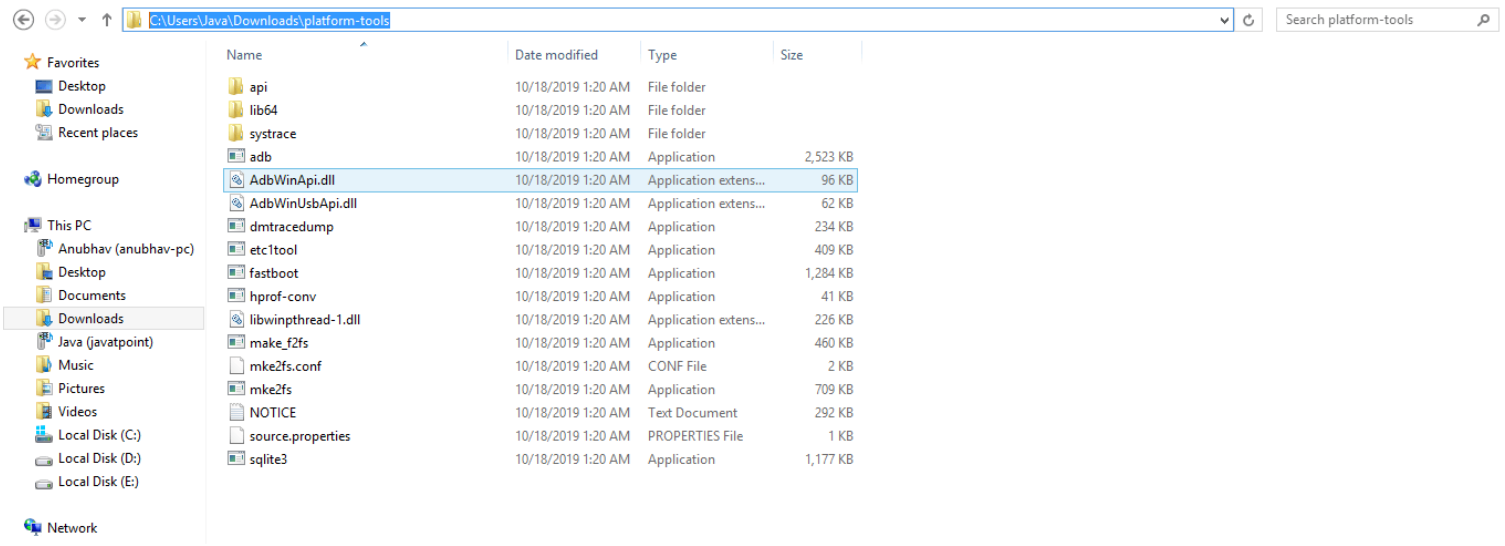


Step 10: Now, open **environment variables** in your system and click on the **path** variable.



Step 11: Add a new path by pasting the path in the **path** variable.

Step12: After that, copy the **platform-tools** path and paste it into the path variable.

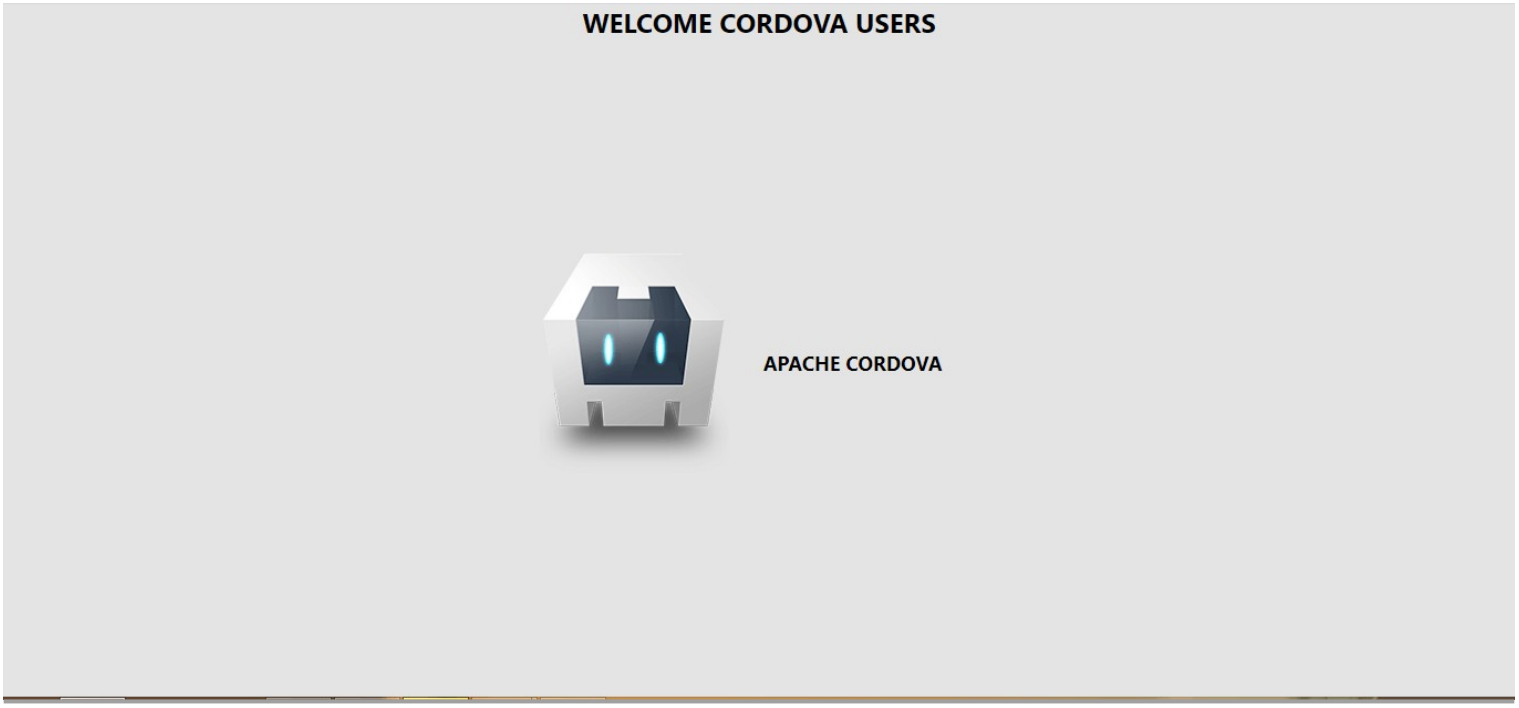


Step 13: Once you done all of that, run your app in the browser by typing the following command in command prompt:

1. phonegap serve

This command starts the server and listen IP address and port number. For running your application in browser, copy the IP address and paste it into the browser.

This will run your application in browser:



Config.xml file

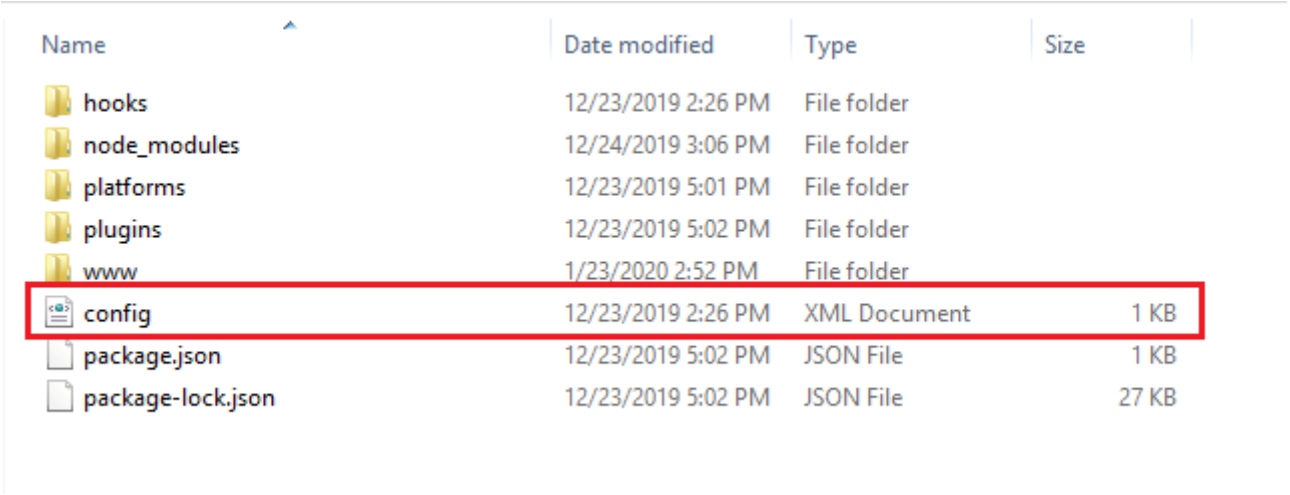
The **config.xml** file is the place where we can change the configuration of the app. When we created our app in the last tutorial, we set reverse domain and name. The values can be changed in the **config.xml** file. When we create the app, the default config file will also be created.

```
<?xml version='1.0' encoding='utf-8'?>
<widget id="com.example.hello" version="0.0.1" xmlns="http://www.w3.org/ns/widgets" xmlns:cdv="http://cordova.apache.org/ns/1.0">
    <name>CordovaApp</name>
    <description>
        A sample Apache Cordova application that responds to the deviceready event.
    </description>
    <author email="dev@cordova.apache.org" href="http://cordova.io">
        Apache Cordova Team
    </author>
    <content src="index.html" />
    <plugin name="cordova-plugin-whitelist" spec="1" />
    <access origin="*" />
    <allow-intent href="http://*/*" />
    <allow-intent href="https://*/*" />
    <allow-intent href="tel:*" />
    <allow-intent href="sms:*" />
    <allow-intent href="mailto:*" />
    <allow-intent href="geo:*" />
    <platform name="android">
        <allow-intent href="market:*" />
    </platform>
    <platform name="ios">
        <allow-intent href="itms:*" />
        <allow-intent href="itms-apps:*" />
    </platform>
</widget>
```

The following table explains configuration elements in **config.xml**.

This file is available at the top-level directory of an app:

app/config.xml



Before version 3.3.1- 0.2.0, the config.xml file was present at **app/www/config.xml**.

Configuration table of config.xml file:

The below table defines the elements used in the config.xml file:

Elements	Description
Widget	It defines the reverse domain value of an app that should be specified when creating the app.
Name	It defines the name of an app.
Description	It represents the description of an app.
Author	It represents the contact information that can be shown within app-store listings.
Content	It represents the starting page of an app at the top-level web assets directory. The default value is **index.html** that appears at the top-level **www** directory.
Plugin	It is an additional feature for enhancing the capabilities of Cordova. It can be defined as a package of code that helps to communicate with the native platforms.
Access	It is used to control access for a specific network domain. It has the default **origin** value ***,** which shows that the access is opened to any domain.
Engine	It specifies the details about the platform, which is restored during the implementation.
allow-intent	It is used for enabling the specific URLs to ask the app to open.
Hook	It represents your custom script that is called by Cordova when a particular action occurs. It is useful for extending the default Cordova functionality.
Platform	It represents a platform where we build our application.
resource-file	It installs the resource file into the system.