

# Cordova Workshop

## Step 1

Go to the [node.js installer page](#) and click on the LTS (Long term support) option.

Node.js® is a JavaScript runtime built on [Chrome's V8 JavaScript engine](#).



## Download for Windows (x64)

**8.11.3 LTS**

Recommended For Most Users

**10.7.0 Current**

Latest Features

[Other Downloads](#) | [Changelog](#) | [API Docs](#)

[Other Downloads](#) | [Changelog](#) | [API Docs](#)

Or have a look at the [Long Term Support \(LTS\) schedule](#).

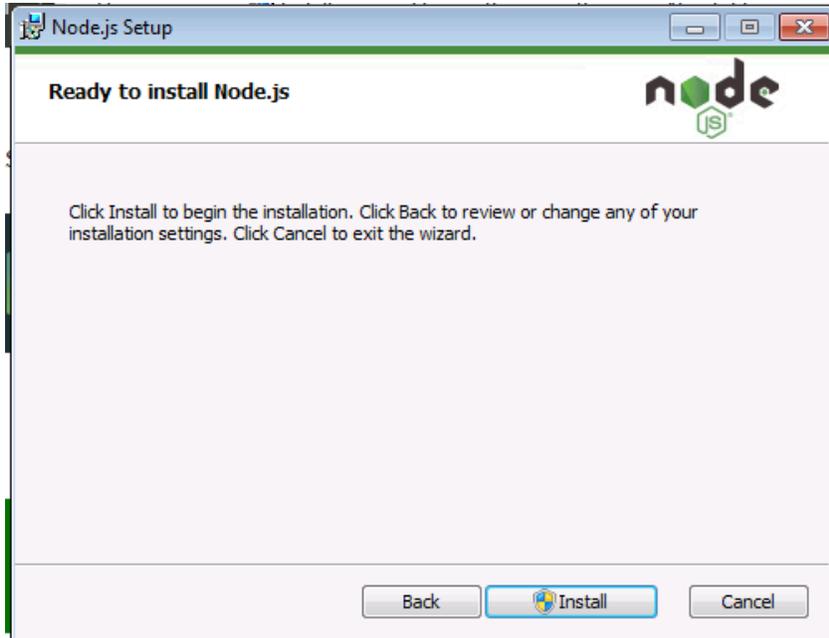
## Step 2

Navigate to the downloads folder and double click on the downloaded file.



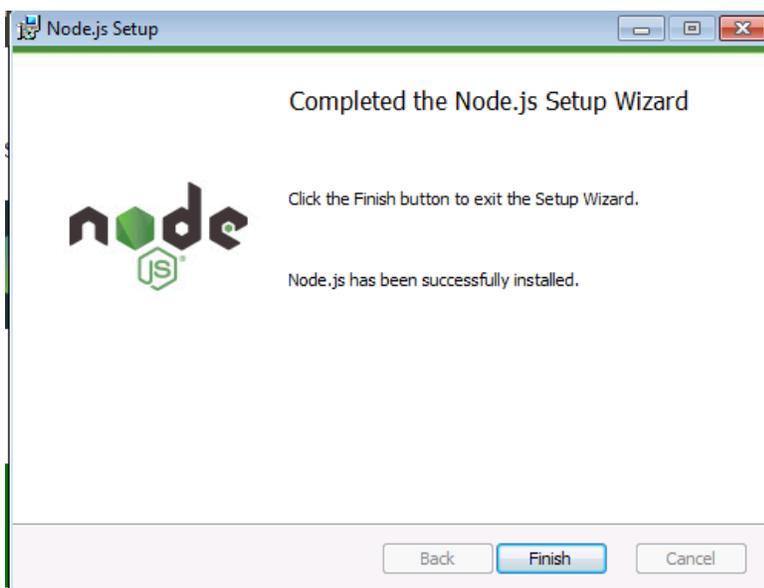
### Step 3

Click install.



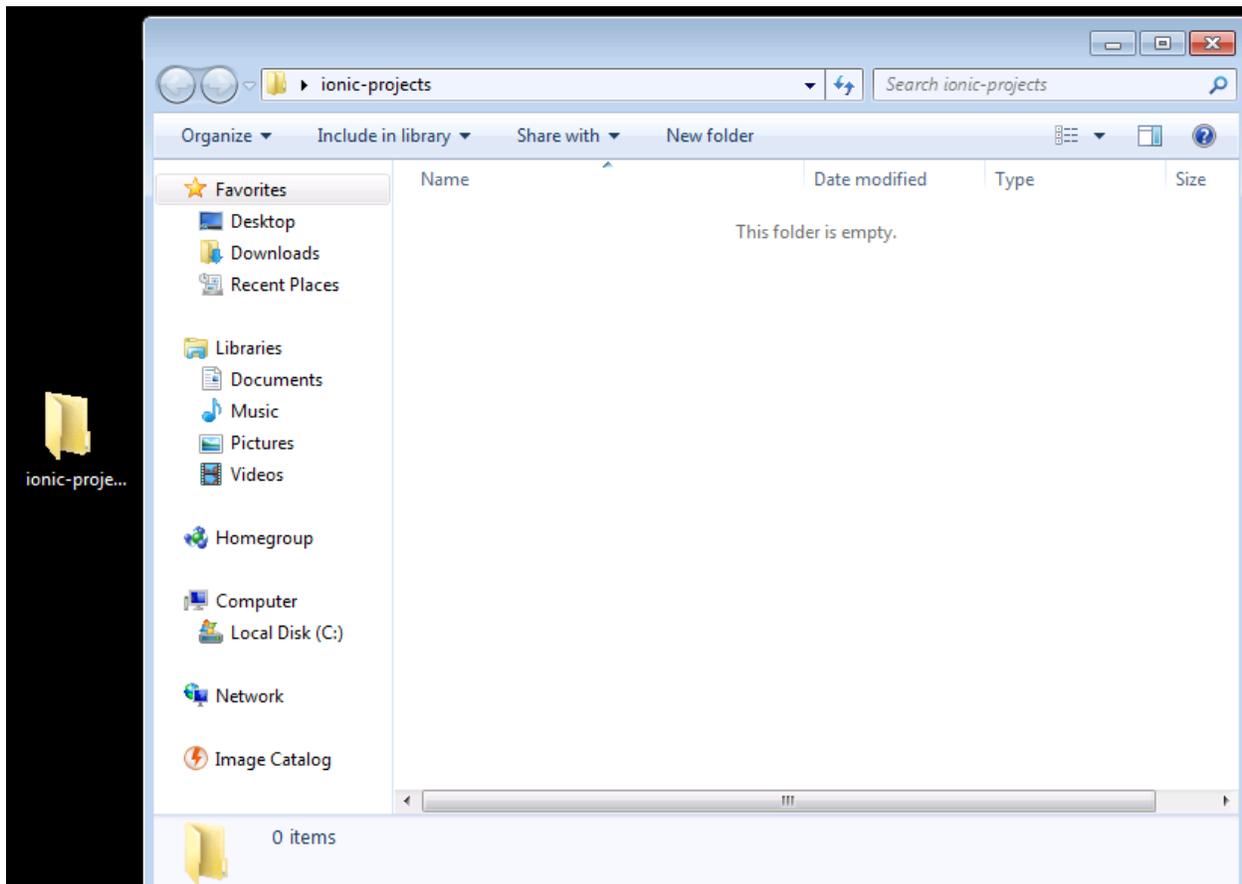
### Step 4

Once you have successfully installed node.js you are ready to install Cordova.



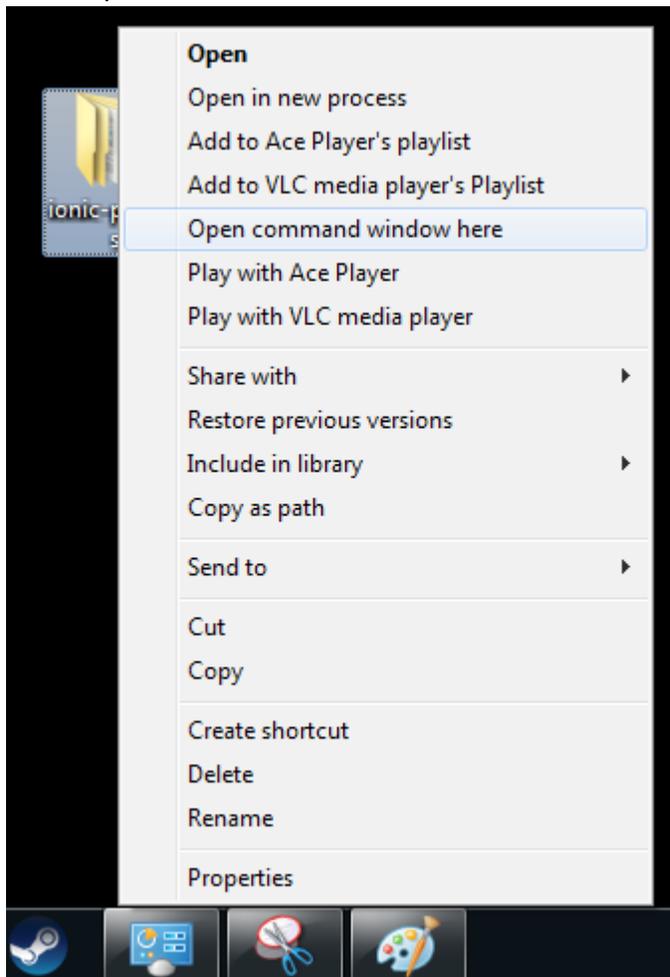
## Step 4

On your desktop, right click and create a folder called "cordova-projects"



## Step 5

While holding the shift key, right click on the cardova-projects folder.  
Click "open in command window here"



## Step 7

In the terminal, run the command  
**npm install -g cordova**

## Step 8

Run the command

**cordova create mqtt com.app.mqtt "MQTT App"**

## Step 9

Run the command

```
cd mqtt
```

## Step 10

Run the command

```
cordova platform add android
```

## Step 11

Download the git.exe at this [link](#)

## Step 12

Add the git.exe file to your PATH

Locate the git.exe file

- Right-Click on **My Computer**
- Click **Advanced System Settings**
- Click **Environment Variables**
- Then under **System Variables** look for the **path** variable and click **edit**
- Add the path to git's bin and cmd at the end of the string

If your unsure about this step please ask for help. You can also take a look at the "[workshop setup tutorial](#)" step 25 as a reference.

## Step 13

Now that you have git installed you can run the following command.

```
cordova plugin add cordova-plugin-geolocation
```

## Step 14

Build the project to fetch the plugin's dependencies.

```
cordova build
```

## Step 15

Click on the following link, it will take you to a github repository that has the starting code you need for this workshop.

<https://github.com/tom-flynn1/cordova-mobile-workshop>

Click on “clone or download” and then “Download zip”.

No description, website, or topics provided.

Edit

Add topics

The screenshot shows the GitHub repository interface for 'tom-flynn1/cordova-mobile-workshop'. At the top, it indicates '1 commit', '1 branch', '0 releases', and '1 contributor'. Below this, there are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. A dropdown menu is open for 'Clone or download', showing options for 'Clone with SSH' (with a copy icon), 'Use HTTPS', and 'Download ZIP'. The repository content shows a file tree with 'README.md' and 'Initial commit'. The main content area displays the title 'cordova-mobile-workshop'.

## Step 16

Locate the file in the downloads folder and unzip the contents.

Go back to the directory that your cordova app was initialized in.

It should have this structure.

Name	Size	Type	Modified
hooks	1 item	Folder	11:04
node_modules	3 items	Folder	11:06
platforms	1 item	Folder	11:05
plugins	4 items	Folder	11:06
res	3 items	Folder	11:04
www	4 items	Folder	11:04
config.xml	1.1 kB	Markup	11:06
package.json	628 bytes	Program	11:06

Click into the “www” folder and delete both the **index.html** file and the **index.js** file in the **js** directory.

Navigate back to the downloads folder where you unzipped the contents from the github

repository.

Copy both the index.html file and the “js” directory and paste them into the the “www” folder of your Cordova project.

You will also need to copy the image [iot-ul-logo.png](#) to the img directory in your cordova project.

## **Step 17**

Run the following command

**cordova build**

## **Step 18**

Connect your phone via usb to your desktop and run the command  
**cordova run android --device**

Ensure your device dev enabled.

## **Step 19**

Once the app loads on your phone click on “get location” button.

Then click the “print location” button.

You should see longitude, latitude and a timestamp printed to the screen.

## Part 2: MQTT

### Step 1

Using the same project folder as before

Run the command

```
cordova plugin add cordova-plugin-mqtt
```

### Step 2

Replace the index.html and index.js files with the ones found in “part 2” of the [github repository](#) you downloaded.

### Step 3

run the command

```
cordova run android --device
```

### Step 4

When the app loads, type into the following fields

“url” -> tcp://test.mosquitto.org

“Port” -> 1883

“Client-id” -> your name

### Step 5

Hit the connect button and then scroll down to the bottom of the page to see if you get a “successful connection” message.

If your not able to connect, make sure you “tcp://” does not have an uppercase T like this “Tcp://”

### Step 6

Under the “subscribe” button enter your name, this is the topic you will be subscribing to.

### Step 7

Under the “publish” button, enter some text and then hit the button to send a message to the “your name” topic.

You should see the received message at the bottom of the page.

## Part 3: MQTT

### Step 1

Similar to “Part 2: Step 2”, replace the index.js and index.html files with the ones found in “part 3” of the github repository.

### Step 2

Hit the “get location” button, followed by the “print location” button.

### Step 3

Enter into the following fields

“url” -> <local IP address of Raspberry Pi>

“Port” -> 1883

“Client-id” -> <your name>

### Example:

192.168.0.44

1883

James

### **Step 4**

Hit the connect button and then scroll down to the bottom of the page to see if you get a "successful connection" message.

If your not able to connect, ask for help.

### **Step 5**

Under the "subscribe" button enter "group-" followed by your group number

Example -> group-4

### **Step 6**

Under the "publish" button, enter some text and then hit the button to send a message to the "your name" topic.

You should see the received message at the bottom of the page. The message should contain latitude, longitude and a timestamp.

Mobile App

# Internet of Things @UL

## Geolocation

Get Position

## Print location

Print location

## Connect

P.S:- "mqtt://" protocol is not supported by this plugin. Instead use "tcp://" which works with any broker.

Url:

Port:

Client id

Connect

## Subscribe



## Subscribe

## Publish

## Activity Log

--> You got disconnected--> Success: you are connected to, tcp://192.168.1.6:1883  
--> Success: you have published to the topic, test  
--> Error: something is wrong,  
{ "type": "connectionLost", "message": "Connection lost (32109) - java.net.SocketException: Software caused connection abort", "connectionStatus": false }

