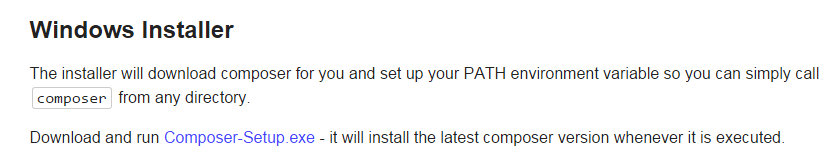
**RATCHET PUSH INTEGRATION SETUP (WINDOWS/XAMPP)**

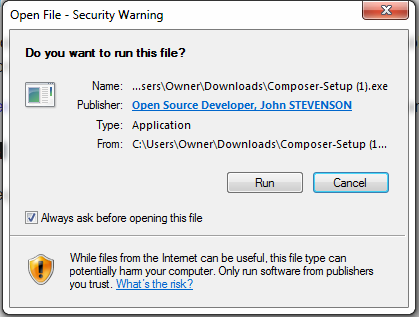
1. Install Composer
   1. Go to <https://getcomposer.org/download/>
   2. Scroll down to Windows Installer
   3. Click Composer-Setup.exe



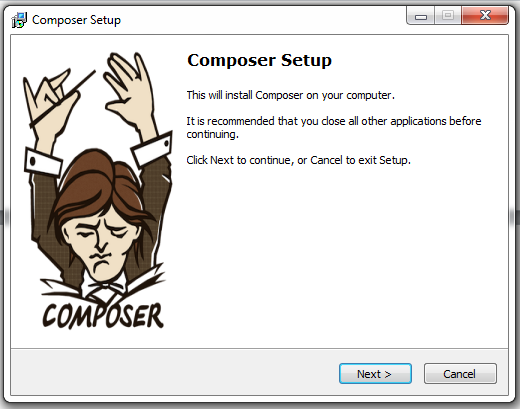
* 1. Click the file once it is downloaded

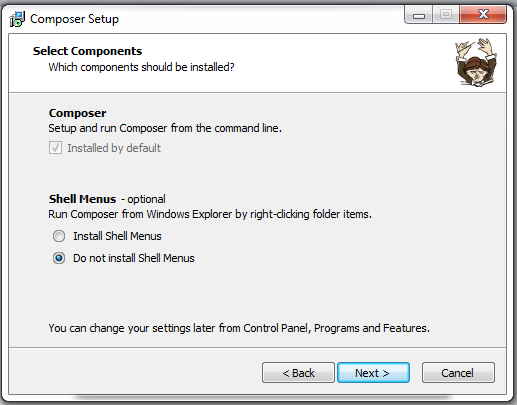


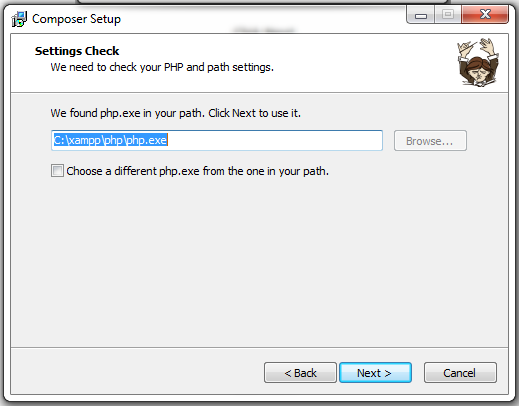
* 1. Click Run



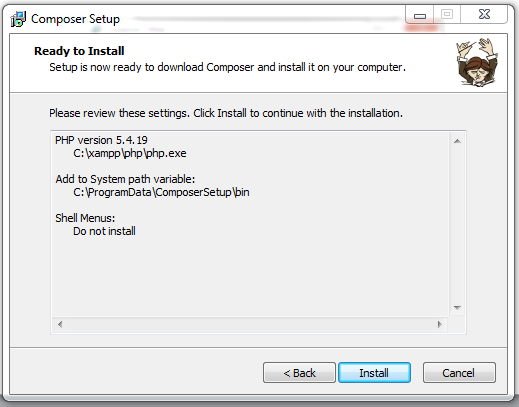
* 1. Go through Composer Setup



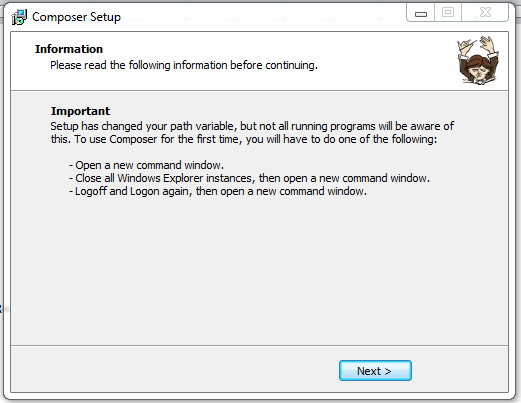




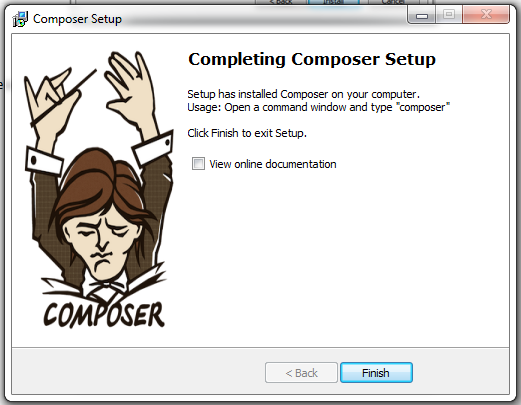
* 1. Click Install



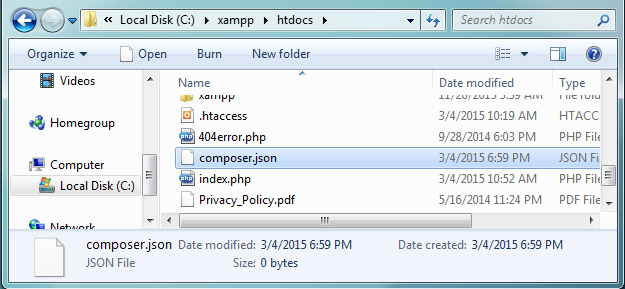
* 1. Read the following and Click Next



* 1. Click Finish



1. Create composer.json file
   1. Go to C:\xampp\htdocs\
   2. Create a file named composer.json



* 1. Copy and paste the following into composer.json

{

"autoload": {

"psr-0": {

"MyApp": "src"

}

},

"require": {

"cboden/ratchet": "0.3.\*",

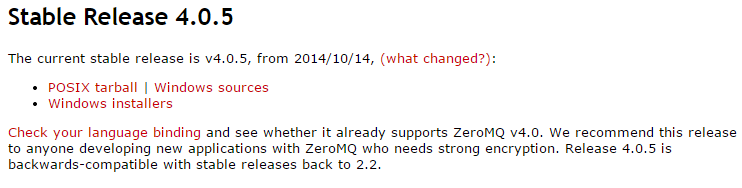
"react/zmq": "0.2.\*|0.3.\*"

}

}

* 1. Save the file

1. Install ZeroMQ
   1. Go to <http://zeromq.org/area:download>
   2. Find the most recent stable release
   3. Click Windows sources



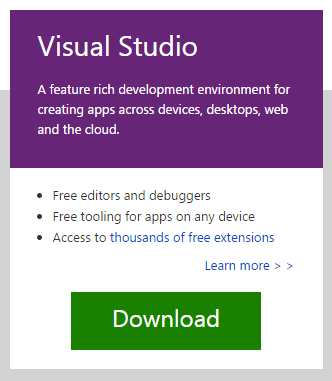
* 1. Click the down arrow and select Show in folder



* 1. Right click the .zip file and select Extract All…
  2. Click Extract
  3. Download Visual Studio (Skip steps 3h-3n if you already have Visual Studio on your computer)
  4. Go to <https://www.visualstudio.com/en-US/products/visual-studio-express-vs#2010-Visual-CPP>
  5. Click Free Visual Studio



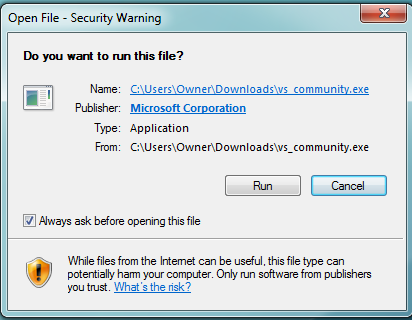
* 1. Click Download



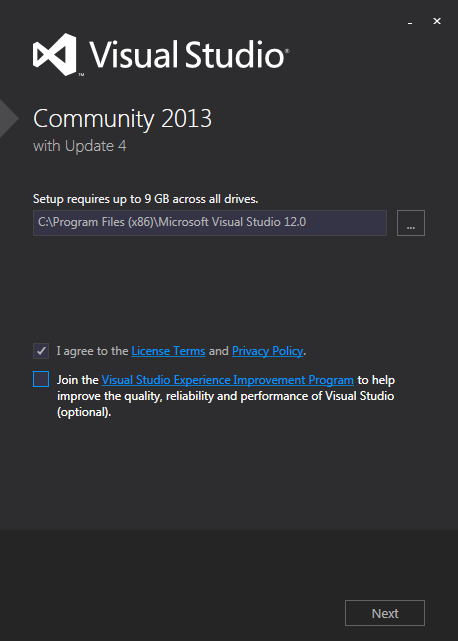
* 1. Click the file once it is downloaded



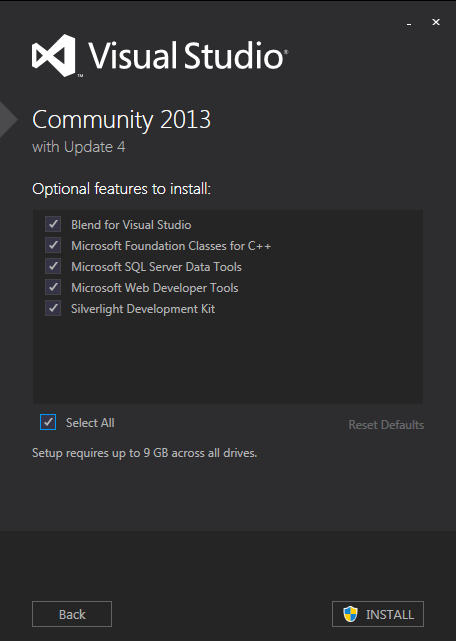
* 1. Click Run



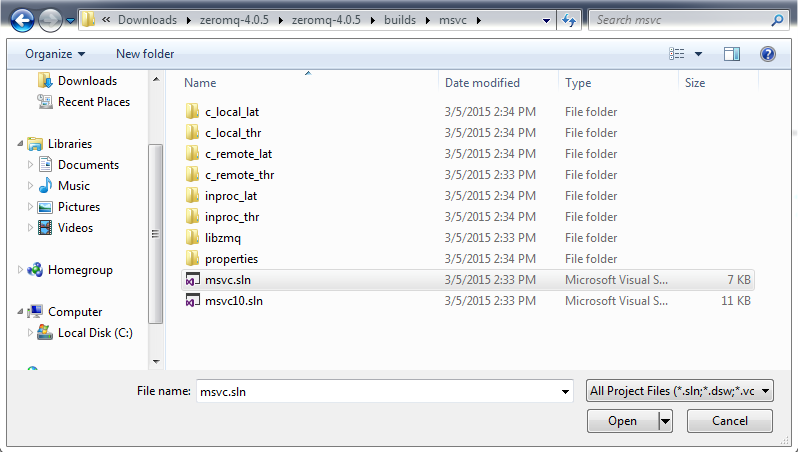
* 1. Go through Visual Studio Setup



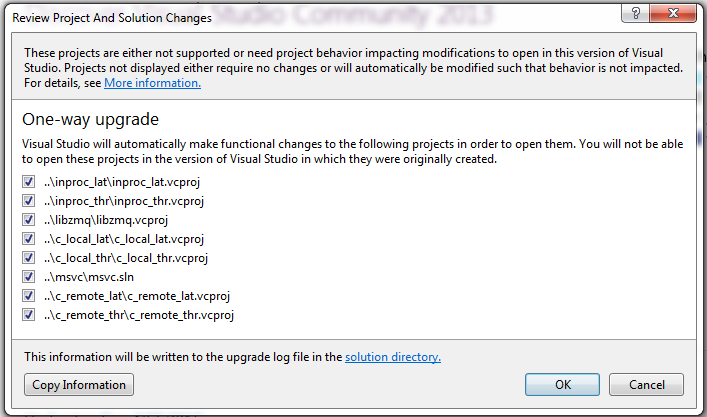
* 1. Click Install (This may take a while)



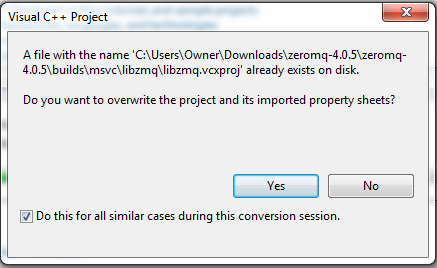
* 1. Open Visual Studio
  2. Select File 🡪 Open 🡪 Project/Solution…
  3. Open file at Downloads\zeromq-4.0.5\zeromq-4.0.5\builds\msvc\msvc.sln



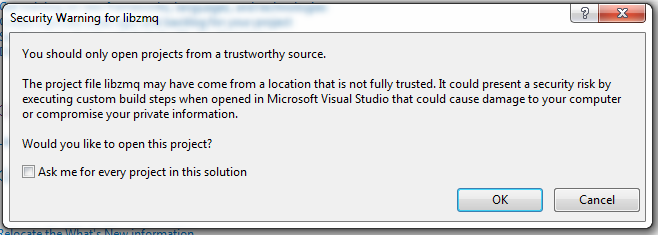
* 1. Click Ok for One-way upgrade



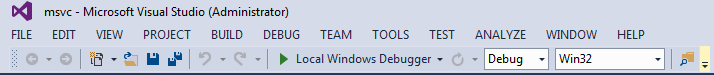
* 1. Check the checkbox and click Yes



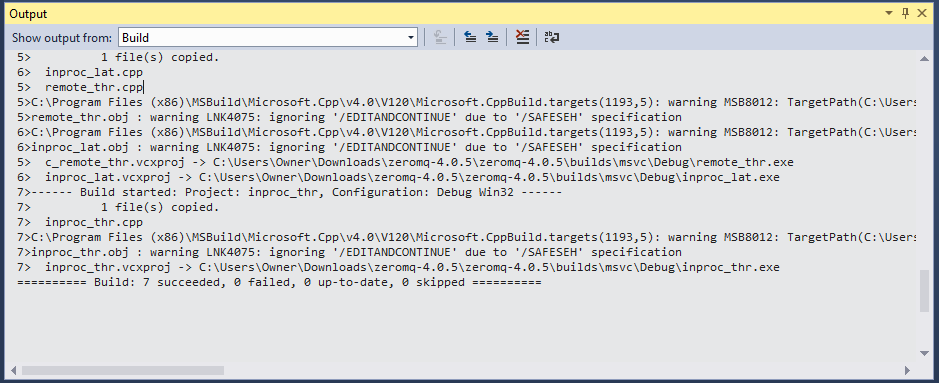
* 1. Uncheck the checkbox and click Ok



* 1. Select Debug and Win32 from the dropdown

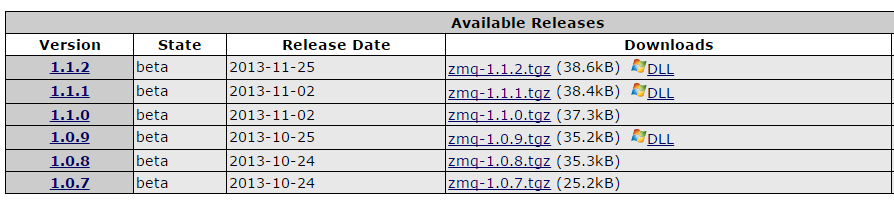


* 1. Select Build 🡪 Build Solution
  2. The output should look something like this:

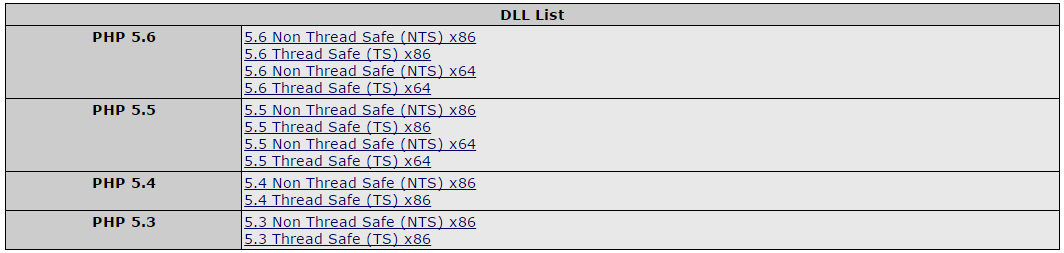


At this point I’m not sure if anything else needs to be done. Are there any additional steps that need to be taken to incorporate the ZMQ libraries into the Ratchet Push Integration Tutorial? Any help would be appreciated!

1. Install PECL extension
   1. Go to <http://pecl.php.net/package/zmq>
   2. Under Available Releases find the most recent Version
   3. Click DLL



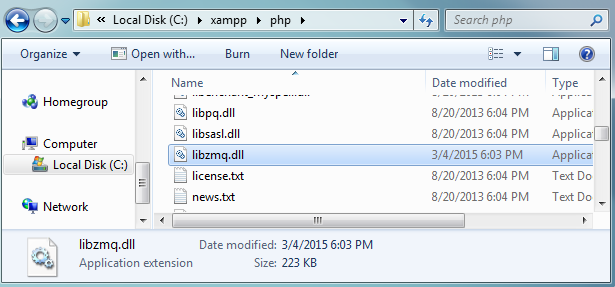
* 1. Click the correct link from the DLL list (If you are using XAMPP your DLL should be *TS & x86 version*)



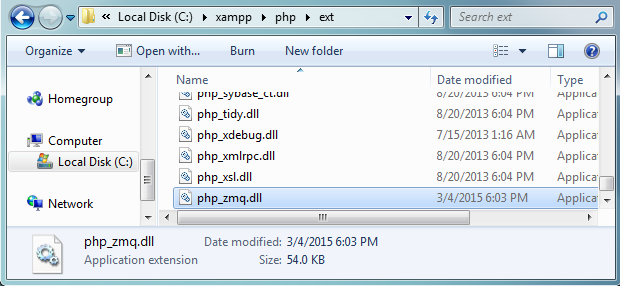
* 1. Click the file once it is downloaded



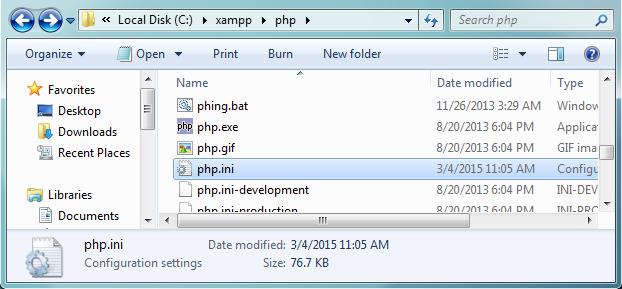
* 1. Copy libzmq.dll into C:\xampp\php\



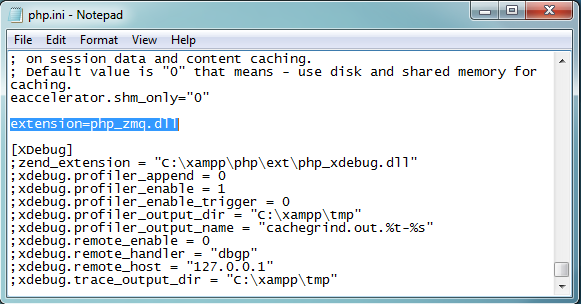
* 1. Copy php\_zmq.dll to C:\xampp\php\ext\



* 1. Open php.ini

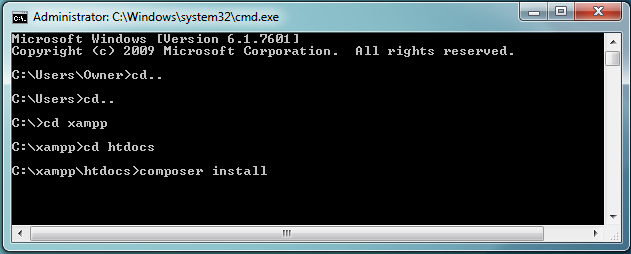


* 1. Add the following line to php.ini: extension=php\_zmq.dll



* 1. Save the file

1. Back to Composer
   1. Open cmd.exe
   2. Using the cd command go to C:\xampp\htdocs
   3. Type composer install and press enter

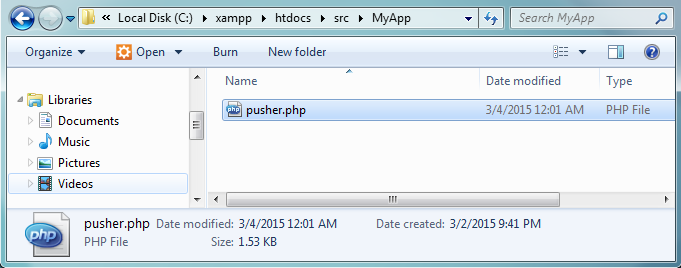


Composer will automatically install all of the dependencies for the Ratchet application. It should look something like this.



Now you are ready to create a Ratchet application.

1. Complete the Ratchet Push Integration Tutorial
   1. Create pusher.php
      1. Create the file path C:\xampp\htdocs\src\MyApp\pusher.php



* + 1. Copy and paste the following into pusher.php

<?php

namespace MyApp;

use Ratchet\ConnectionInterface;

use Ratchet\Wamp\WampServerInterface;

class Pusher implements WampServerInterface {

/\*\*

\* A lookup of all the topics clients have subscribed to

\*/

protected $subscribedTopics = array();

public function onSubscribe(ConnectionInterface $conn, $topic) {

$this->subscribedTopics[$topic->getId()] = $topic;

}

public function onUnSubscribe(ConnectionInterface $conn, $topic) {}

public function onOpen(ConnectionInterface $conn) {}

public function onClose(ConnectionInterface $conn) {}

public function onCall(ConnectionInterface $conn, $id, $topic, array $params) {

// In this application if clients send data it's because the user hacked around in console

$conn->callError($id, $topic, 'You are not allowed to make calls')->close();

}

public function onPublish(ConnectionInterface $conn, $topic, $event, array $exclude, array $eligible) {

// In this application if clients send data it's because the user hacked around in console

$conn->close();

}

public function onError(ConnectionInterface $conn, \Exception $e) {}

/\*\*

\* @param string JSON'ified string we'll receive from ZeroMQ

\*/

public function onBlogEntry($entry) {

$entryData = json\_decode($entry, true);

// If the lookup topic object isn't set there is no one to publish to

if (!array\_key\_exists($entryData['category'], $this->subscribedTopics)) {

return;

}

$topic = $this->subscribedTopics[$entryData['category']];

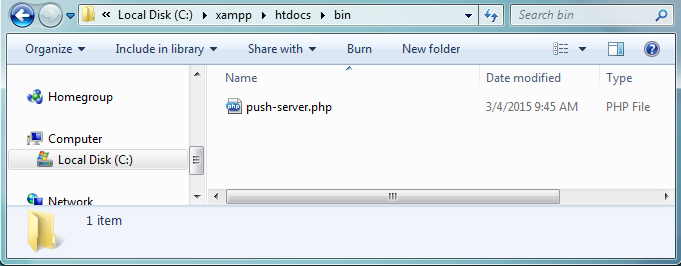
// re-send the data to all the clients subscribed to that category

$topic->broadcast($entryData);

}

}

* + 1. Save the file
  1. Create push-server.php
     1. Create the file path C:\xampp\htdocs\bin\push-server.php



* + 1. Copy and paste the following into push-server.php

<?php

require dirname(\_\_DIR\_\_) . '/vendor/autoload.php';

$loop = React\EventLoop\Factory::create();

$pusher = new MyApp\Pusher;

// Listen for the web server to make a ZeroMQ push after an ajax request

$context = new React\ZMQ\Context($loop);

$pull = $context->getSocket(ZMQ::SOCKET\_PULL);

$pull->bind('tcp://127.0.0.1:5555'); // Binding to 127.0.0.1 means the only client that can connect is itself

$pull->on('message', array($pusher, 'onBlogEntry'));

// Set up our WebSocket server for clients wanting real-time updates

$webSock = new React\Socket\Server($loop);

$webSock->listen(8080, '0.0.0.0'); // Binding to 0.0.0.0 means remotes can connect

$webServer = new Ratchet\Server\IoServer(

new Ratchet\Http\HttpServer(

new Ratchet\WebSocket\WsServer(

new Ratchet\Wamp\WampServer(

$pusher

)

)

),

$webSock

);

$loop->run();

* + 1. Save the file
  1. Add ZeroMQ to existing website
     1. Incorporate the following into your website

<?php

// post.php ???

// This all was here before ;)

$entryData = array(

'category' => $\_POST['category']

, 'title' => $\_POST['title']

, 'article' => $\_POST['article']

, 'when' => time()

);

$pdo->prepare("INSERT INTO blogs (title, article, category, published) VALUES (?, ?, ?, ?)")

->execute($entryData['title'], $entryData['article'], $entryData['category'], $entryData['when']);

// This is our new stuff

$context = new ZMQContext();

$socket = $context->getSocket(ZMQ::SOCKET\_PUSH, 'my pusher');

$socket->connect("tcp://localhost:5555");

$socket->send(json\_encode($entryData));

* 1. Add Client side code
     1. Incorporate the following into your website

<script src="http://autobahn.s3.amazonaws.com/js/autobahn.min.js"></script>

<script>

var conn = new ab.Session('ws://localhost:8080',

function() {

conn.subscribe('kittensCategory', function(topic, data) {

// This is where you would add the new article to the DOM (beyond the scope of this tutorial)

console.log('New article published to category "' + topic + '" : ' + data.title);

});

},

function() {

console.warn('WebSocket connection closed');

},

{'skipSubprotocolCheck': true}

);

</script>

* 1. Run the executable
     1. Open cmd.exe
     2. Using the cd command go to C:\xampp\htdocs
     3. Type php bin/push-server.php and press enter

