

WebSockets útmutató

Telepítés

Majd a telepítő-útmutatóban találod, amelyet a tanárnak csinállok.

Használat

Szerver oldalon

Szerver létrehozása

Létre kell hoznunk egy olyan PHP fájlt, amely a szerveret képzi (vagyis ez fog futni, és közben figyelni). Ez a fájl kibővíti az alap-libraryt, amellyel kapcsolatokat kezeljük. Legyen ez a server.php néven a fájl, amelyet futtatnunk kell a PHP-vel.

```
require dirname(__DIR__) . '/vendor/autoload.php';

use Ratchet\Server\IoServer;
use Ratchet\Http\HttpServer;
use Ratchet\WebSocket\WsServer;
use MyApp\Chat; //Ezt helyettesítsd be a projekt nevével – a
'composer.json' fájlban megtalálod a „name”-nél

//Itt jön a fő megvalósítás
class Chat implements \Ratchet\MessageComponentInterface {
    protected $clients;

    public function __construct() {
        $this->clients = new \SplObjectStorage;
    }

    public function onOpen(\Ratchet\ConnectionInterface $conn) {
        // Store the new connection
        $this->clients->attach($conn);
        echo "New connection! ({\$conn->resourceId})\n";
    }

    public function onMessage(\Ratchet\ConnectionInterface $from,
$msg) {
        foreach ($this->clients as $client) {
            if ($from !== $client) {
                // Send the message to all clients except the sender
                $client->send($msg);
            }
        }
    }

    public function onClose(\Ratchet\ConnectionInterface $conn) {
        // Remove the connection
        $this->clients->detach($conn);
        echo "Connection {$conn->resourceId} has disconnected\n";
    }
}
```

```

    }

    public function onError(\Ratchet\ConnectionInterface $conn,
\Exception $e) {
        echo "An error has occurred: {$e->getMessage()}\n";
        $conn->close();
    }
}

//Végül meg:
$server = IoServer::factory(
    new HttpServer(
        new WsServer(
            new Chat()
        )
    ),
    8080
);

$server->run();

```

A kliens oldalon pedig a JavaScript kód így fog kinézni:

```

var ws = new WebSocket('ws://localhost:8080');

ws.onopen = function() {
    console.log('WebSocket connection established');
    ws.send('Hello Server!');
};

ws.onmessage = function(event) {
    console.log('Received from server: ' + event.data);
};

ws.onclose = function() {
    console.log('WebSocket connection closed');
};

ws.onerror = function(error) {
    console.error('WebSocket error: ' + error);
};

```