PHP ile TCP Soket ve Asenkron Uygulamalar

iletisim@kursadaltan.com

https://github.com/kursadaltan/phpkonf-2023

Hakkımda

Kürşad ALTAN

Webmaster

PHP Full-Stack Developer

IoT Developer

Back-End Developer

MEDIANOVA

İnternet bağlantısını evimizden dünyanın öbür ucuna nasıl sağlıyoruz?

MEDIANOVA

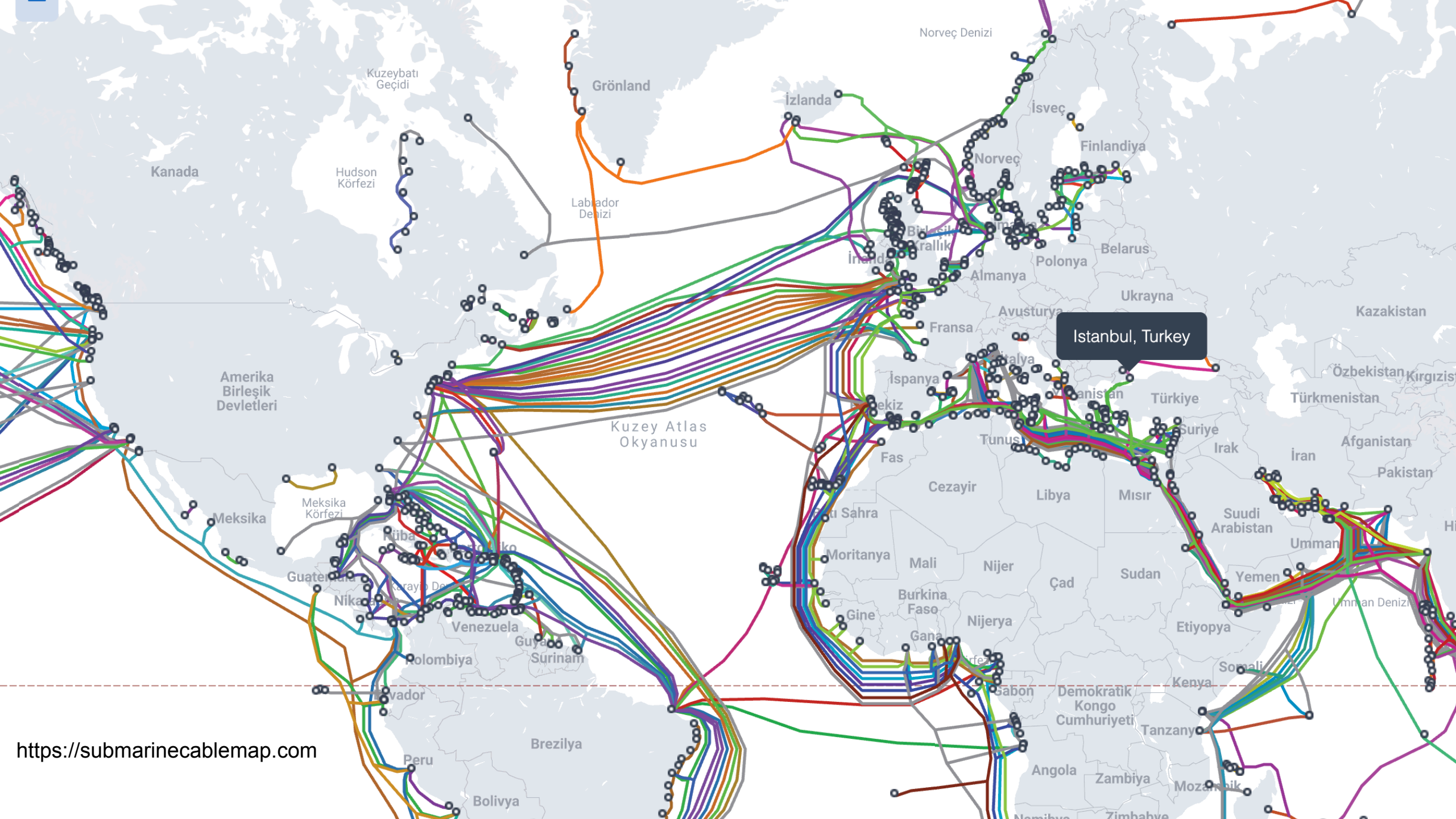
İnter(arasında) net(Ağ-Bağlantı)

Fiziksel Bağlantı
İnternet, ülkeler arasında ve ülke içinde bir çok noktadan birbirlerine kablolarla bağlantılı bir ağ mevcuttur.

Adres Tanımlama
Her bağlantı noktasının bir dijital adresi vardır. Bu adresler IP adresi olarak isimlendirilir

Sunucular
TCP/IP adresi üzerinden yapılan isteklere yanıt veren sunucular hazırlanır.

İstemciler
İnternete bağlı ve talep eden konumunda olan cihazlar istemci olarak adlandırılır.



https://submarinecablemap.com

TCP/IP Protokolü

Web Sunucuları (HTTP)

E-Posta Sunucuları (IMAP,POP3)

Dosya Transferi Protokolleri (FTP)

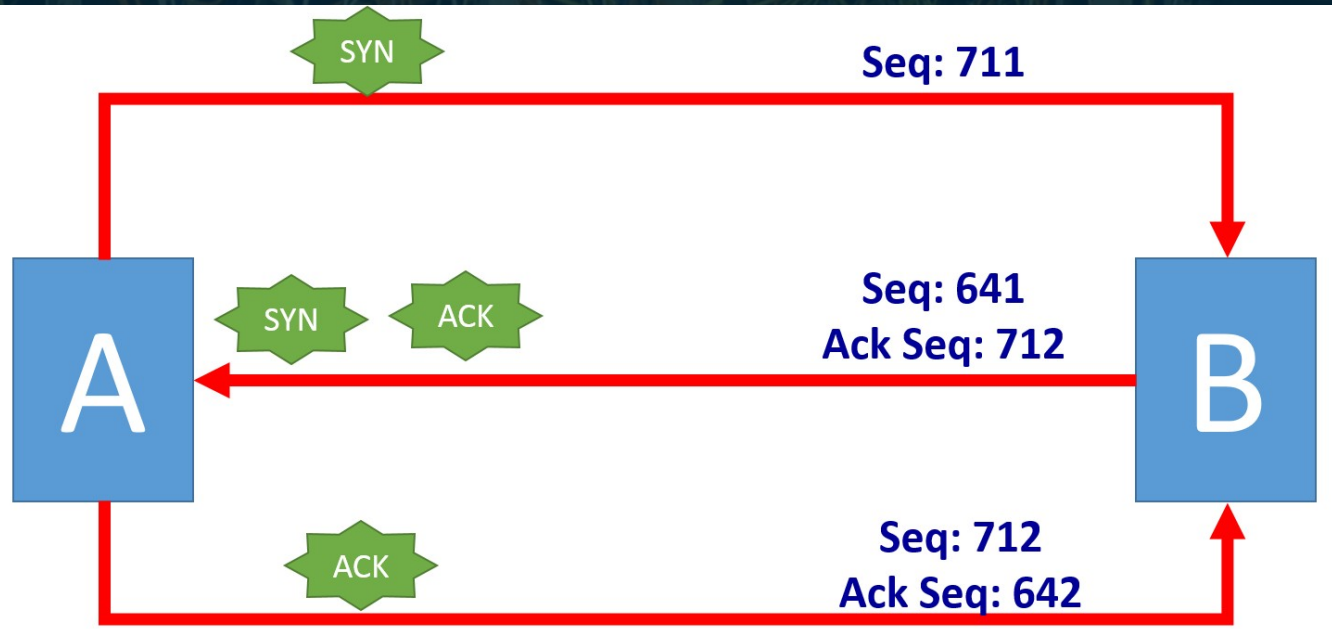
Sesli İletişim Protokolleri (VoIP)

Uzaktan Erişim ve Kontrol (SSH, Telnet)

Nesnelerin İnterneti Cihazları (IoT)

TCP/IP Bağlantısı Nasıl Sağlanır?

- Bağlantı İstemi (SYN)
- Bağlantı Kabulü (SYN-ACK)
- Bağlantı Onayı (ACK)



MEDIANOVA

Paketlerin doğruluğu?

TCP/IP Protokolü iletişimin doğruluğunu garanti eder.

İsteği paketlere bölerek iletilmesini güvenceye alır.

Zaman aşımına uğrayan paketleri tespit eder.

Paketleri Checksum ve CRC ile denetler

TCP Soket Uygulamaları

Düşük Seviyeli İletişim Mekanizması

Talebin Karşılanması İçin İhtiyaç Duyulması

Özel Protokol Uygulamaları

TCP Socket Uygulamaları

1. Anlık Mesajlaşma Uygulamaları
2. Veri Toplama ve Analiz Uygulamaları
3. Canlı Anket Uygulaması
4. Nesnelerin İnterneti (IoT) Uygulamaları

PHP ve TCP Soket

PHP ile TCP Soket uygulamaları geliştirmek eğlencelidir.

MEDIANOVA

Canlı ortamlarda çalışmak için hazır.

Kararlı sürümleri, uzun süreli desteklenir.

Herhangi bir ekstra eklentiye ihtiyaç duymaz.

Performans için PHP8+ ve PHP7+ önerilir. PHP5.3+ ile çalışabilir.

Event-driven, non-blocking I/O with PHP

ReactPHP is a low-level library for event-driven programming in PHP. At its core is an event loop, on top of which it provides low-level utilities, such as: Streams abstraction, async DNS resolver, network client/server, HTTP client/server and interaction with processes. Third-party libraries can use these components to create async network clients/servers and more.

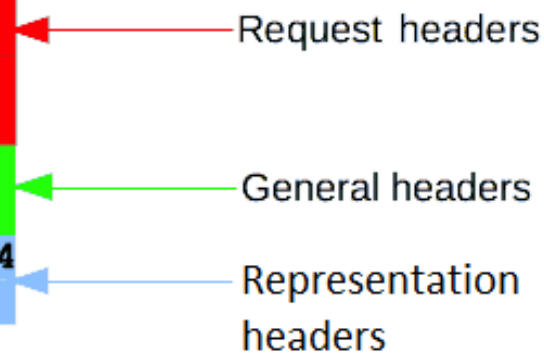
GitHub Twitter # Gitter

```php
<?php

// $ composer require react/http react/socket # install example using Composer
// $ php example.php # run example on command line, requires no additional web server

require __DIR__ . '/vendor/autoload.php';

$http = new React\Http\HttpServer(function (Psr\Http\Message\ServerRequestInterface $request) {
    return React\Http\Message\Response::plaintext(
        "Hello World!\n"
    );
});

$socket = new React\Socket\SocketServer('127.0.0.1:8080');
$http->listen($socket);

echo "Server running at http://127.0.0.1:8080" . PHP_EOL;
```

This simple web server written in ReactPHP responds with "Hello World!" for every request.

HTTP Protokolü

```
POST / HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (Macintosh;… )… Firefox/51.0
Accept: text/html,application/xhtml+xml,…,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=-12656974
Content-Length: 345

-12656974
(more data)
```



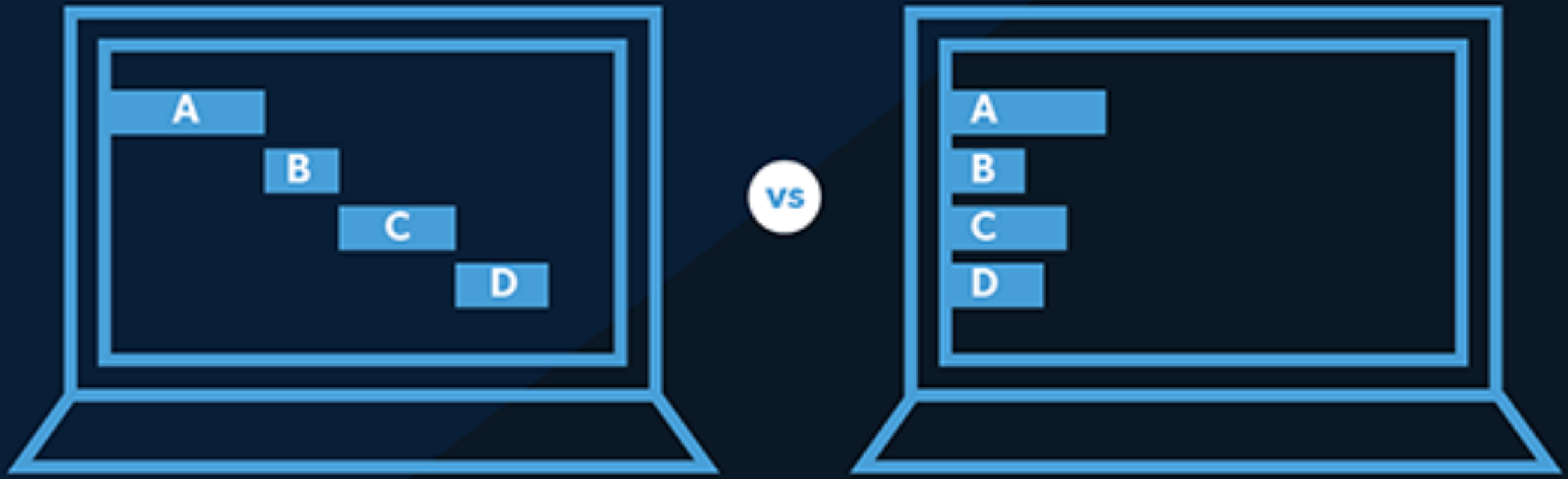
```php
<?php

$socket = new React\Socket\SocketServer('0.0.0.0:6666');

$socket->on('connection', function (React\Socket\ConnectionInterface $connection) {
    $connection->write("Merhaba " . $connection->getRemoteAddress() . "!\n");
    $connection->write("PHPKonf2023'e Hoşgeldiniz !\n");

    $connection->on('data', function ($data) use ($connection) {
        $connection->close();
    });
});
```

```php
<?php

$socket = new React\Socket\SocketServer('0.0.0.0:6666');
$server = new React\Socket\LimitingServer($socket, 1);
$socket->on('connection', function (React\Socket\ConnectionInterface $connection) {
    $connection->write("Merhaba " . $connection->getRemoteAddress() . "!\n");
    $connection->write("PHPKonf2023'e Hoşgeldiniz !\n");

    $connection->on('data', function ($data) use ($connection) {
        $data = trim(preg_replace('/[^\w\d \.\,\-\!\?]/u', '', $data));

        if ($data == 'ACCESS') {
            $connection->write('TRUE' . PHP_EOL);
        } else {
            $connection->write('FALSE' . PHP_EOL);
        }
        $connection->end($data . PHP_EOL);
        // $connection->close();
    });

    $connection->on('close', function () use ($connection) {
        echo '[' . $connection->getRemoteAddress() . ' güle güle]' . PHP_EOL;
    });
});

$socket->on('error', function (Exception $e) {
    echo 'Error: ' . $e->getMessage() . PHP_EOL;
});
```

```
→  phpkonf telnet localhost 6666
Trying ::1...
Connected to localhost.
Escape character is '^]'.
Merhaba tcp://172.18.0.1:55218!
PHPKonf2023'e Hoşgeldiniz !
ACCESS
TRUE
ACCESS
Connection closed by foreign host.
```

```php
<?php

use React\EventLoop\Loop;

$connector = new React\Socket\Connector([
        'tcp' => true,
        'tls' => true,
        'unix' => true,
        'dns' => true,
        'timeout' => true,
        'happy_eyeballs' => true,
]);
$timeoutConnector = new React\Socket\TimeoutConnector($connector, 3.0);

$promise = $timeoutConnector->connect('127.0.0.1:6666')->then(function (React\Socket\ConnectionInterface $connection) {
    $connection->write('ACCESS');
    $connection->on('data', function ($data) {
        echo $data;
    });

    $connection->on('close', function () {
        echo 'Connection closed' . PHP_EOL;
    });

    Loop::addTimer(5, function () use ($connection) {
        $connection->end();
    });

}, function (Exception $e) {
    echo 'Error: ' . $e->getMessage() . PHP_EOL;
});
```

PHP ve Asenkron Programlama



```php
<?php

use React\EventLoop\Loop;

Loop::addTimer(0.5, React\Async\async(function () {
    echo 'Merhaba' . PHP_EOL;
    React\Async\await(React\Promise\Timer\sleep(1.0));
    // sleep(1);
    echo 'Dünya' . PHP_EOL;
}));

Loop::addTimer(1.0, function () {
    echo 'Yeni' . PHP_EOL;
});

// t=0,5s'de "Merhaba" yazdırır
// t=1,0s'de "Yeni" yazdırır
// t=1,5s'de "Dünya" yazdırır
```

```php
<?php

use Psr\Http\Message\ResponseInterface;
use React\Http\Browser;

$browser = new Browser();

$start = microtime(true);
$promise = $browser->get('https://download.samplelib.com/mp4/sample-5s.mp4')
->then(function (ResponseInterface $response) use ($start) {
    echo 'SampleLib Time: ' . (microtime(true) - $start) . PHP_EOL;
    return "SampleLib size = " . $response->getBody()->getSize();
}, function (Exception $e) {
    echo 'Error: ' . $e->getMessage() . PHP_EOL;
});

$medianova = microtime(true);
$promise = $browser->get('https://sample-videos.mncdn.org/mp4/sample-5s.mp4')
->then(function (ResponseInterface $response) use ($medianova) {
    echo 'Medianova Time: ' . (microtime(true) - $medianova) . PHP_EOL;
    return "Medianova.com size = " . $response->getBody()->getSize();
}, function (Exception $e) {
    echo 'Error: ' . $e->getMessage() . PHP_EOL;
});
```

```
Çıktı :
Medianova Time: 1.0883800983429
Medianova.com size = 2848208
SampleLib Time: 8.1532490253448
SampleLib size = 2848208
```

MEDIANOVA

```php
<?php

use React\EventLoop\Loop;
use React\MySQL\Factory;
use React\MySQL\QueryResult;
use function React\Async\await;

include __DIR__ . '/vendor/autoload.php';

$loop = Loop::get();
$factory = new Factory($loop);
$connection = $factory->createLazyConnection('root:phpkonf@db/phpkonf');

$start = microtime(true);

foreach(range(1, 120000) as $i) {
    $json = addslashes(json_encode(["value" => rand(210,230), "measure" => "voltage"]));
    $connection->query('INSERT INTO raw_data (`data`, `device_id`, `created_at`) VALUES ("' . $json .'", 10, now())')
        ->then(function (QueryResult $command) {
        }, function (Exception $error) {
            echo 'Error: ' . $error->getMessage() . PHP_EOL;
        });

    $kmemReal = round((memory_get_usage() / 1024) / 1024);
    if ($kmemReal >= 32) {
        await(React\Promise\Timer\sleep(0.01));
    }
}
echo 'Time: ' . (microtime(true) - $start) . PHP_EOL;
$connection->quit();
```

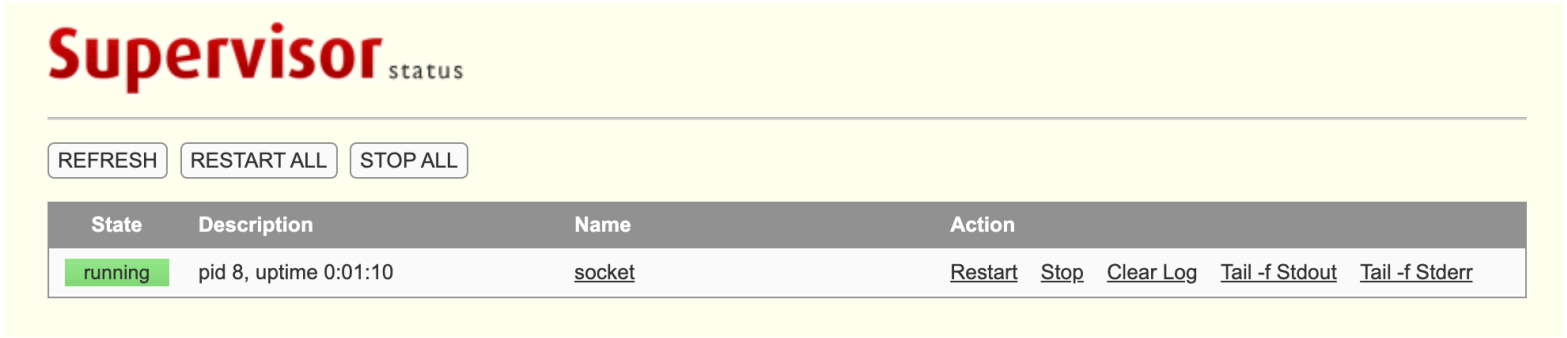
```php
<?php

use React\EventLoop\Loop;
use React\MySQL\Factory;

$loop = Loop::get();
$factory = new Factory($loop);
$connection = $factory->createLazyConnection('root:phpkonf@db/phpkonf');

$formatter = new \React\Stream\ThroughStream(function ($data) {
    return json_decode($data['data'], true)['value'] . ';' . strtotime($data['created_at']) . PHP_EOL;
});

$file = new \React\Stream\WritableResourceStream(fopen('data.txt', 'w'), $loop);

$connection->queryStream('SELECT * FROM raw_data')->pipe($formatter)->pipe($file);

echo 'OK';

$connection->quit();
```

```php
<?php

include __DIR__ . '/../vendor/autoload.php';

use React\EventLoop\Loop;
use React\Socket\Connector;
use React\Promise\Promise;

$loop = Loop::get();
$connector = new Connector($loop);
$promises = [];
$requests = ['ACCESS', 'PHP', 'Konf'];

$promises = array_map(function ($request) use ($loop) {
    $connector = new Connector($loop);
    $promise = $connector->connect('127.0.0.1:6666');

    return $promise->then(function (React\Socket\ConnectionInterface $connection) use ($request) {
        $connection->write($request . "\n");

        return new Promise(function ($resolve) use ($connection) {
            $buffer = '';
            $connection->on('data', function ($data) use (&$buffer) {
                $buffer .= $data;
            });
            $connection->on('close', function () use (&$buffer, $resolve, $connection) {
                $resolve($buffer);
                $connection->end();
            });
        });
    });
}, $requests);

\React\Promise\all($promises)->then(function ($responses) {
    foreach ($responses as $response) {
        echo "Response: " . $response . PHP_EOL;
    }
});

$loop->run();
//ChatGPT ile Yazıldı
```

```
Response: Merhaba tcp://127.0.0.1:59972!
PHPKonf2023'e Hoşgeldiniz !
TRUE
ACCESS

Response: Merhaba tcp://127.0.0.1:59974!
PHPKonf2023'e Hoşgeldiniz !
FALSE
PHP

Response: Merhaba tcp://127.0.0.1:59976!
PHPKonf2023'e Hoşgeldiniz !
FALSE
Konf
```

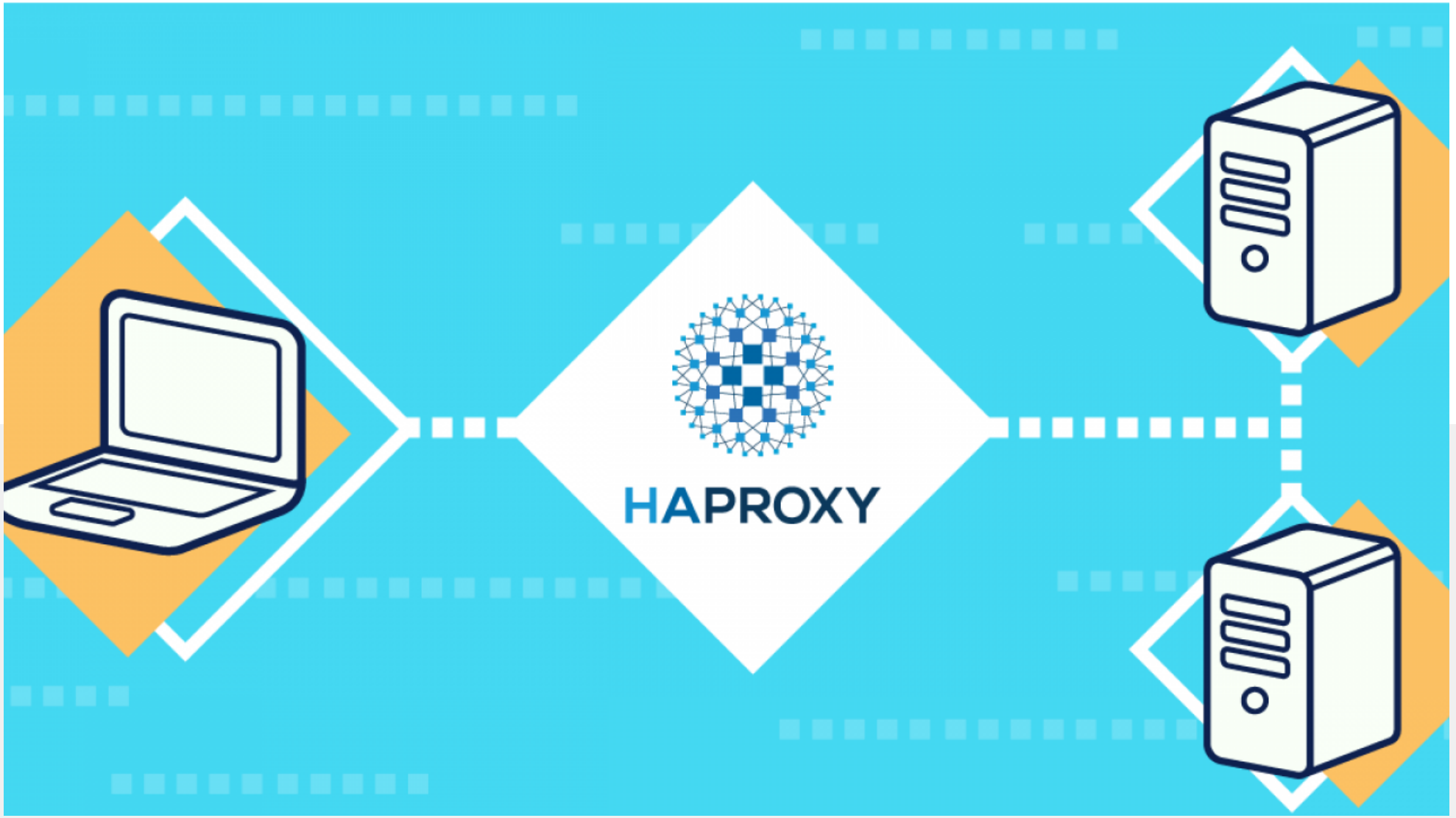
Süreç Yönetimi

MEDIANOVA




```
[program:socket]
process_name=%(program_name)s
command=php /usr/src/app/tcp-socket.php
autostart=true
autorestart=true
user=www-data
redirect_stderr=true
```

MEDIANOVA


HAPROXY

PHP ile TCP Soket ve Asenkron Uygulamalar

iletisim@kursadaltan.com

https://github.com/kursadaltan/phpkonf-2023