

GABRIELE ROMANATO

Usare le Docker Engine API con PHP

Le Docker Engine API consentono di interagire con Docker a livello programmatico. Utilizzando PHP, possiamo sfruttare queste API per gestire container, immagini e altro.

Prerequisiti

- Docker installato e in esecuzione
- PHP (>=7.4) installato
- Composer installato

Installazione dell'SDK

Il modo più semplice per interagire con Docker in PHP è usare l'SDK `docker-php`. Installiamolo con Composer:

```
composer require docker-php/docker-php
```

Configurazione della Connessione

Creiamo una connessione con il demone Docker:

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

use Docker\Docker;
use Docker\API\Client;
use Http\Adapter\Guzzle6\Client as GuzzleClient;

$docker = Docker::createWithHttpClient(GuzzleClient::createWithConfig([
    'base_uri' => 'unix:///var/run/docker.sock',
]));
```

Se usiamo Windows, dobbiamo connetterci tramite TCP invece di UNIX socket:

```
$docker = Docker::createWithHttpClient(GuzzleClient::createWithConfig([
    'base_uri' => 'tcp://127.0.0.1:2375',
```

```
]);
```

Gestire i Container

Elencare i Container

```
$containerManager = $docker->containerList();  
foreach ($containerManager as $container) {  
    echo "ID: " . $container->getId() . "\n";  
}
```

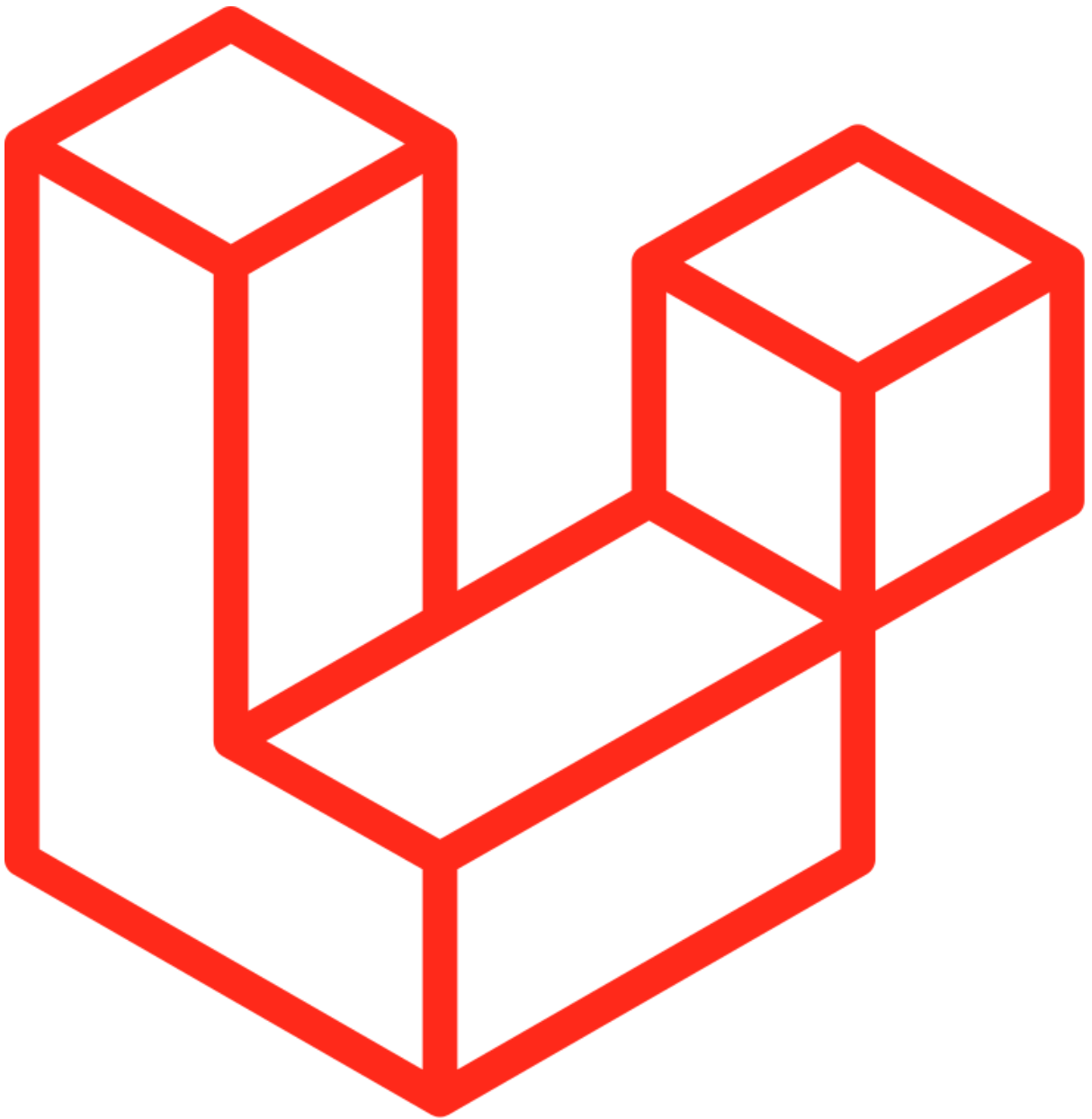
Creare un Container

```
use Docker\API\Model\ContainersCreatePostBody;  
  
$containerConfig = new ContainersCreatePostBody();  
$containerConfig->setImage('nginx');  
$containerConfig->setCmd(['echo', 'Hello from Docker']);  
  
$container = $docker->containerCreate($containerConfig);  
$docker->containerStart($container->getId());
```

Conclusione

In questo articolo abbiamo visto come installare e configurare l'SDK per interagire con Docker Engine tramite PHP. Da qui, possiamo esplorare funzionalità più avanzate come gestione delle immagini, volumi e reti.

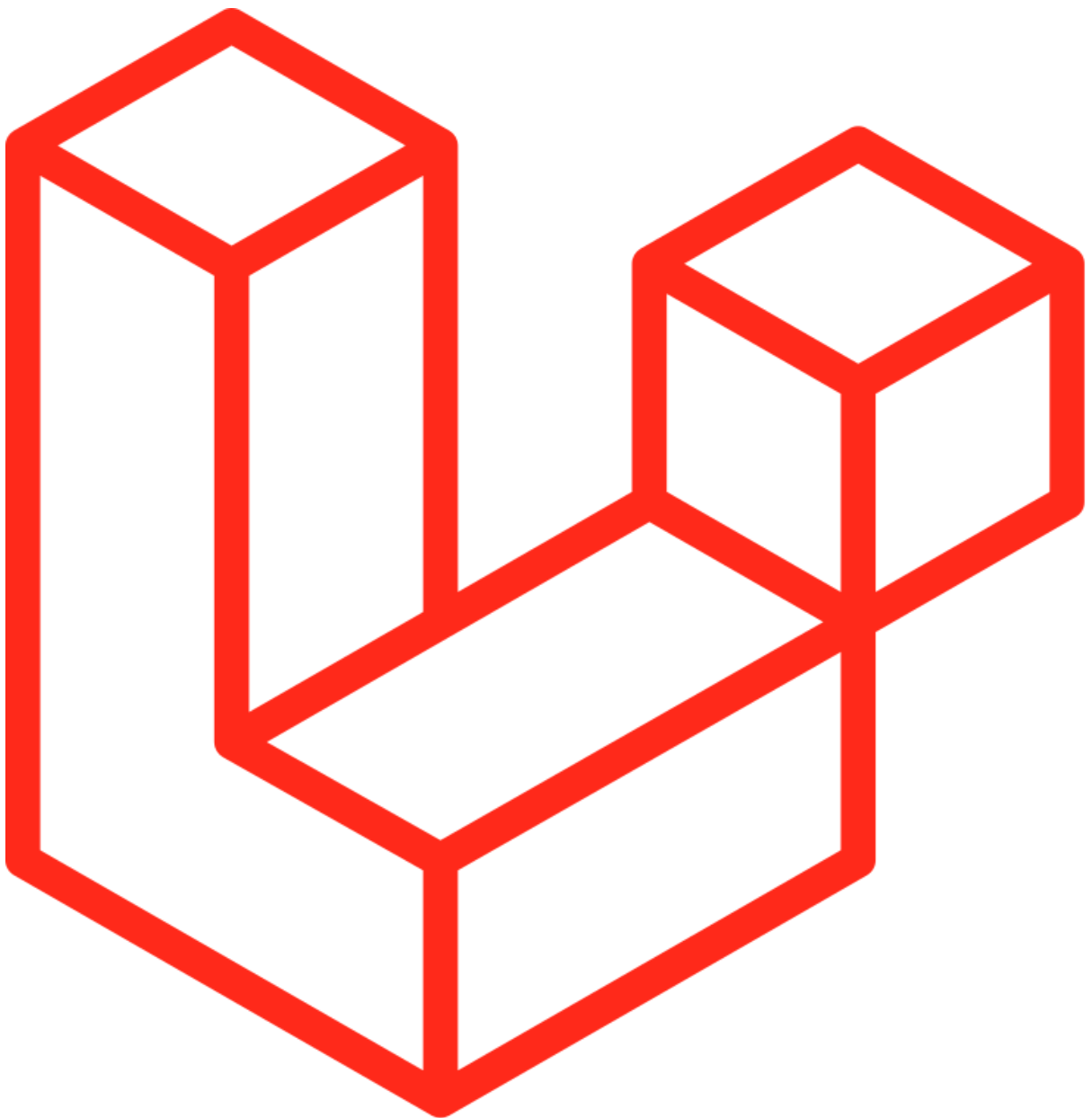
Applicazioni Correlate



•

Laravel Placeholder Image

Applicazione in Laravel per creare immagini segnaposto.
[Docker](#)[Docker Compose](#)[Composer](#)[PHP](#)[Laravel](#)[JavaScript](#)

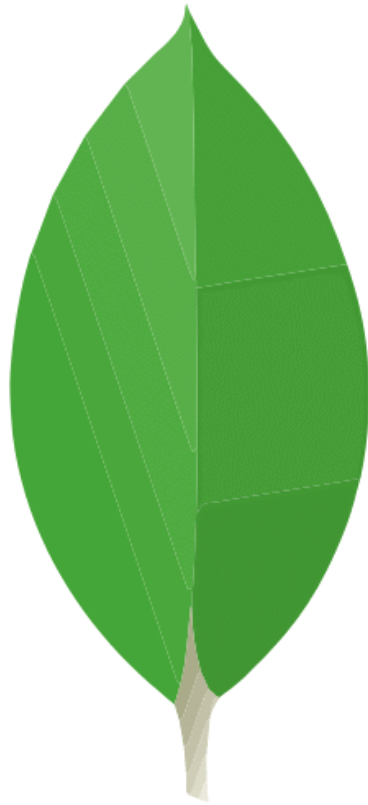


•

Laravel Shopping Cart

Applicazione che fa parte del progetto Laravel E-commerce e implementa la gestione del carrello in Laravel.

Docker Docker Compose Composer PHP Laravel



-

PHP MongoDB App

Applicazione basata su MongoDB con il driver PHP ufficiale.
DockerDocker ComposeComposerPHPMongoDB