

# Laravel

# GitHub



Laravel GitHub



## 1.

- API
- API
- **API**

## 2.

|  | HTTP/REST gRPC RabbitMQ |
|--|-------------------------|
|  | Consul Nacos Eureka     |
|  | Docker + Kubernetes     |
|  | Prometheus + Grafana    |

# GitHub

## 1. Laravel Microservice Boilerplate

- **Stars:** 800+
- API
  - API **API** Kong
  - gRPC
  - **JWT**
  - Docker Compose

- `api-gateway` `user-service` `order-service` `docker`

```

/api-gateway #
/user-service #
/order-service #
/docker #

```

## 2. Laravel + DDD + CQRS

- **Stars:** 1k+
- `laravel`
  - `ddd` `ddd`
  - `cqrs` `cqrs`
  - `event-sourcing` `event-sourcing`
  - `laravel-octane` `laravel-octane`

## 3. Lumen

- **Stars:** 500+
- `lumen`
  - `api` `lumen`
  - `swagger`
  - `jwt` + `oauth2`



## 1. `api-gateway`

- `api-gateway` **HTTP/gRPC**

```

// Guzzle
$response = Http::post('http://user-service/api/users', [
    'name' => 'John Doe'
]);

```

- `order-service`

```

//
event(new OrderCreated($order));

```

```
// 00000000
class ProcessPayment implements ShouldQueue
{
    public function handle(OrderCreated $event)
    {
        // 00000
    }
}
```

## 2. API 0000

- Kong 0000 0

```
# 0000
routes:
- name: user-service
  paths: /users/*
  service: user-service
- name: order-service
  paths: /orders/*
  service: order-service
```

## 3. 00000

- Saga 00 00000000

```
// 00 Saga 000000
$saga = new CreateOrderSaga();
$saga->addStep(new ReserveProductStep());
$saga->addStep(new ProcessPaymentStep());
$saga->execute();
```

## 4. 0000000000

- Consul 00 0

```
// 0000
Consul::agent()->service()->register([
    'Name' => 'user-service',
    'Address' => '192.168.1.100',
```

```
'Port' => 8000
]);

// []

$services = Consul::catalog()->services();
```



# DevOps

## 1. Docker

- **docker-compose.yml** 

```
version: '3'
services:
  user-service:
    build: ./user-service
    ports: ["8001:80"]
  order-service:
    build: ./order-service
    ports: ["8002:80"]
  api-gateway:
    image: kong:latest
    ports: ["8000:8000"]
```

## 2. Kubernetes

- **Deployment** 

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: user-service
spec:
  replicas: 3
  selector:
    matchLabels:
      app: user-service
  template:
```



