

Laravel

API development in Laravel



Rekening houden met:

- CORS(Cross-origin resource sharing)
- Token based authentication?
- REST(GET, POST, PUT, DELETE methods)
- Geen sessions gebruiken

2 composer packages voor laravel die API development makkelijk maken

- jwt-auth
- <u>laravel-cors</u>

Deze les



- API development
- API Authenticatie met HTTP
- API Authenticatie met JWT

API authentication met HTTP



- Door Middel van HTTP header
- Gebruikers gegevens versturen in het formaat username:password en base64 encoded



HTTP HEADER Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ IN LARAVEL(middleware die je bijvoorbeeld auth.once.basic kan noemen) public function handle(\$request, Closure \$next) { return Auth::onceBasic() ?: \$next(\$request); }

API authentication met JWT

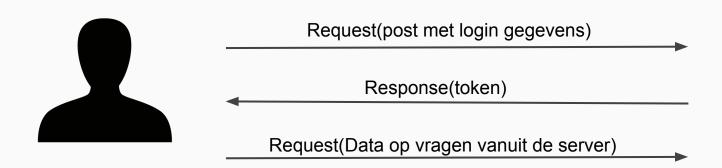




- JSON web tokens
- Gebaseerd op een standaard
- Implementaties voor veel verschillende platformen
- Gebruikers data mee geven aan de token

Een token kan er zo uit zien.

eyJhbGciOiJIUzI1NilsInR5cCl6lkpXVCJ9. eyJzdWliOilxMjM0NTY3ODkwliwibmFtZSl6lkpvaG4gRG9lliwiYWRtaW4iOnRydWV9. TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ







Header

```
base64enc({
    "alg": "HS256",
    "typ": "JWT"
})
```

Payload

```
base64enc({
    "iss": "toptal.com",
    "exp": 1426420800,
    "company": "Toptal",
    "awesome": true
})
```

Signature

```
HMACSHA256(
base64enc(header)
+ '.' +,
base64enc(payload)
, secretKey)
```

