

Utilizing Reflection API

For Enhanced API Documentation and Code Analysis

Selim Can CABA

Senior Software Developer
Wingie Enuygun Group

Ben Kimim?

Senior Software Developer at Wingie Enuygun Group

github.com/phpdev

linkedin.com/in/selimcancaba

x.com/selimcancaba



Wingie Enuygun Group

En İyi Kullanıcı Deneyimi ile Uçak Bileti, Otobüs Bileti, Otel ve Araç Kiralama'da Binlerce Seçenek Sunuyoruz



20M+

Aylık Ziyaretçi
156 farklı ülkeden
gezinler, Wingie
Enuygun üzerinden
rezervasyon yapıyor.



400+

Çalışan



150+

IT Departmanı
Çalışanı



200+

İçerik Sağlayıcı

150+
Tech Team

Agile
Organisation

WEG Tech Stack



Reflection API Nedir?



Read



Modify

Başlıklar

Reflection API - Read

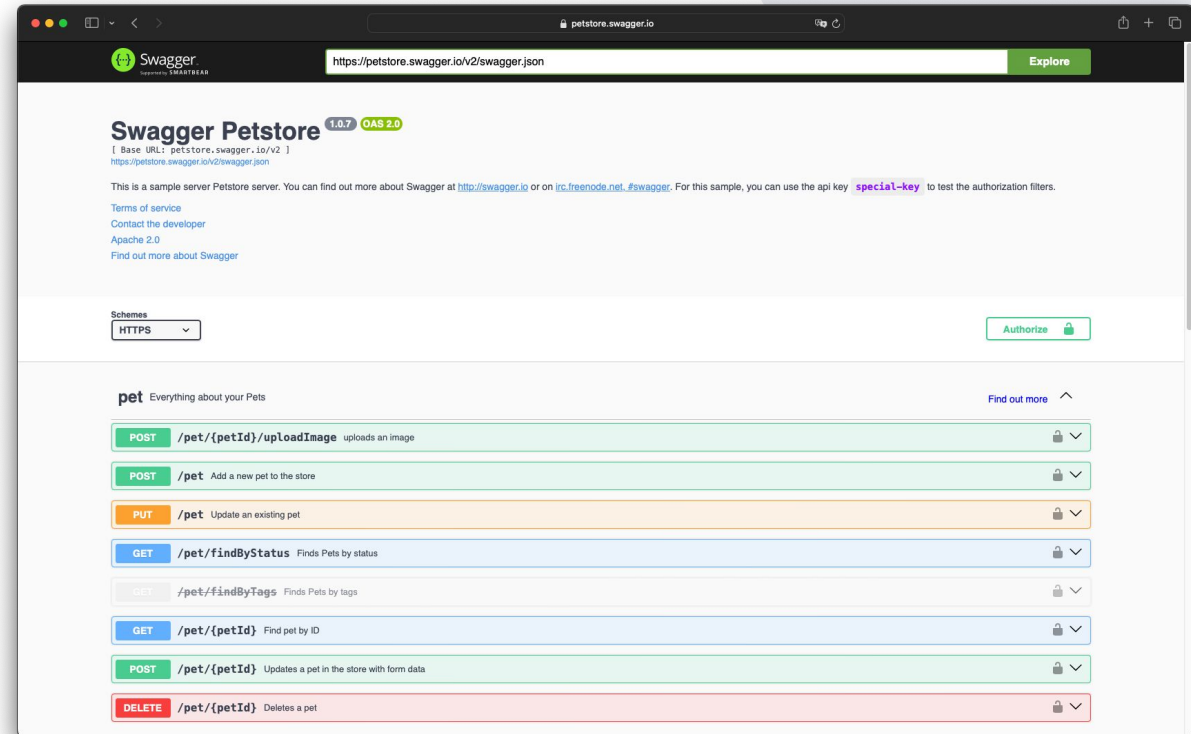
- Annotation & Attributes
- Request Mapping
- Response Mapping
- API Dokümantasyonu
- Projeler

Reflection API - Modify

- Entity Manipulation

Reflection API - Read

- 1250+ repositories
- 1000+ endpoints



Reflection API - Read

```
1  /**
2   * Searches for flights based on input parameters.
3   *
4   * @Route("/search", methods={"POST"})
5   */
6  public function search(Request $request): JsonResponse
7  {
8      return $this->json(
9          $this->search->handle($request->getPayload())
10     );
11 }
```

```
$reflectionMethod = new \ReflectionMethod(SearchController::class, 'search');
```

```
ReflectionMethod {#286 ▼
  +name: "search"
  +class: "App\Controller\SearchController"
  returnType: "Symfony...\JsonResponse"
  parameters: {▼
    $request: ReflectionParameter {#337 ▼
      +name: "request"
      position: 0
      typeHint: "Symfony...\Request"
    }
  }
  extra: {▼
    file: "/srv/app/src/Controller/SearchController.php"
    line: "21 to 26"
    docComment: ""
    /**\n
      * Searches for flights based on input parameters.\n
      *\n
      * @Route("/search", methods={"POST"})\n
      */
    ""
  }
  isUserDefined: true
  modifiers: "public"
}
```


Annotation & Attributes



Doctrine Annotations

```
/**
 * Searches for flights based on input parameters.
 *
 * @Route("/search", methods={"POST"})
 */
public function search(Request $request): JsonResponse
{
    return $this->json(
        $this->search->handle($request->getPayload())
    );
}
```

Reflection API + PhpDoc Parser

PHP 8 Attributes

```
/*
 * Searches for flights based on input parameters.
 */
#[Route('/search', methods: ['POST'])]
public function search(Request $request): JsonResponse
{
    return $this->json(
        $this->search->handle($request->getPayload())
    );
}
```

Reflection API

Request Mapping - Controller

Manual Mapping

```
/*
 * Updates a user based on the input parameters.
 */
#[Route('/users', methods: ['PATCH'])]
public function update(
    Request $request,
): JsonResponse
{
    return $this->json(
        $this->update->handle($request->query, $request->getPayload())
    );
}
```

Mapping with Attributes

```
/*
 * Updates a user based on the input parameters.
 */
#[Route('/users', methods: ['PATCH'])]
public function update(
    #[MapQueryString] UserUpdateFilters $userUpdateFilters,
    #[MapRequestPayload] UserUpdateRequest $userUpdateRequest,
): JsonResponse
{
    return $this->json(
        $this->update->handle($userUpdateFilters, $userUpdateRequest)
    );
}
```

Request Mapping - Schemas

Query String

```
class UserUpdateFilters
{
    #[Assert\Choice(choices: ['admin', 'user'])]
    public string $role;

    #[Assert\Choice(choices: ['en', 'tr'])]
    public string $preferredLanguage;
}
```

Request Body

```
class UserUpdateRequest
{
    #[Assert\Length(min: 3, max: 10)]
    public string $name;

    #[Assert\Length(min: 20, max: 200)]
    public string $address;
}
```

Request Mapping - Array to DTO

```
public function handle(InputBag $query, InputBag $payload): Response
{
    // Build user update filter and validate it
    $filters = $this->buildUserUpdateFilters($query);

    // Build user update request and validate it
    $request = $this->buildUserUpdateRequest($payload);

    // Find users
    $users = $this->searchUsers($filters);

    // Update users
    return $this->bulkUpdate($users, $request);
}
```

InputBag
Associative array

```
public function handle(Filters $filters, Request $request): Response
{
    // Find users
    $users = $this->searchUsers($filters);

    // Update users
    return $this->bulkUpdate($users, $request);
}
```

Attributes
DTO for Request Schema

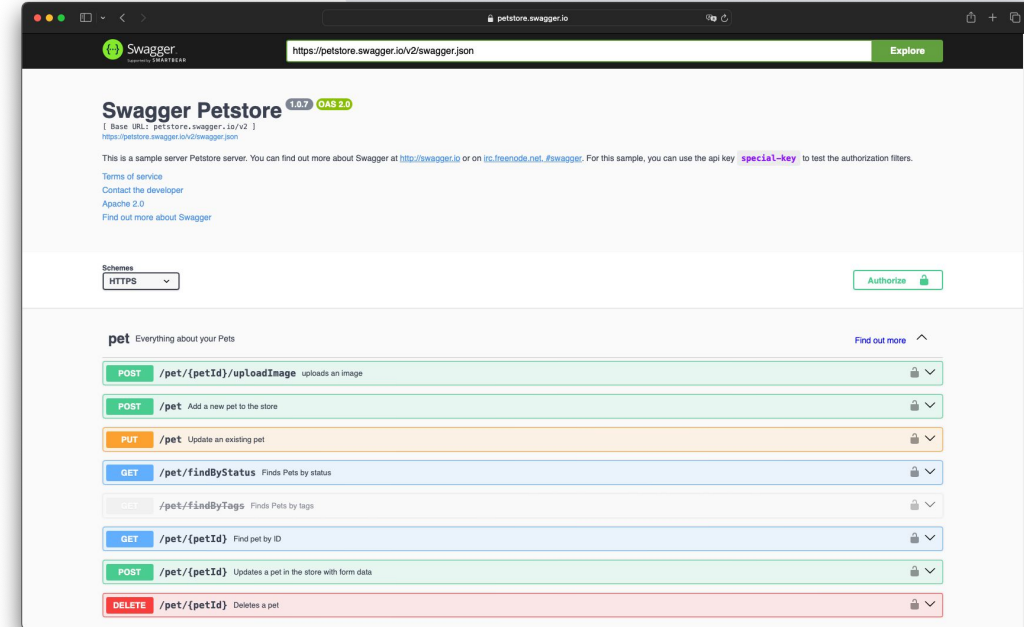
Response Mapping



```
/*
 * Updates a user based on the input parameters.
 */
#[Route('/users', methods: ['PATCH'])]
#[Response(UserUpdateResponse::class)]
public function update(
    #[MapQueryString] UserUpdateFilters $userUpdateFilters,
    #[MapRequestPayload] UserUpdateRequest $userUpdateRequest,
): JsonResponse
{
    return $this->json(
        $this->update->handle($userUpdateFilters, $userUpdateRequest)
    );
}
```

API Dokümantasyonu

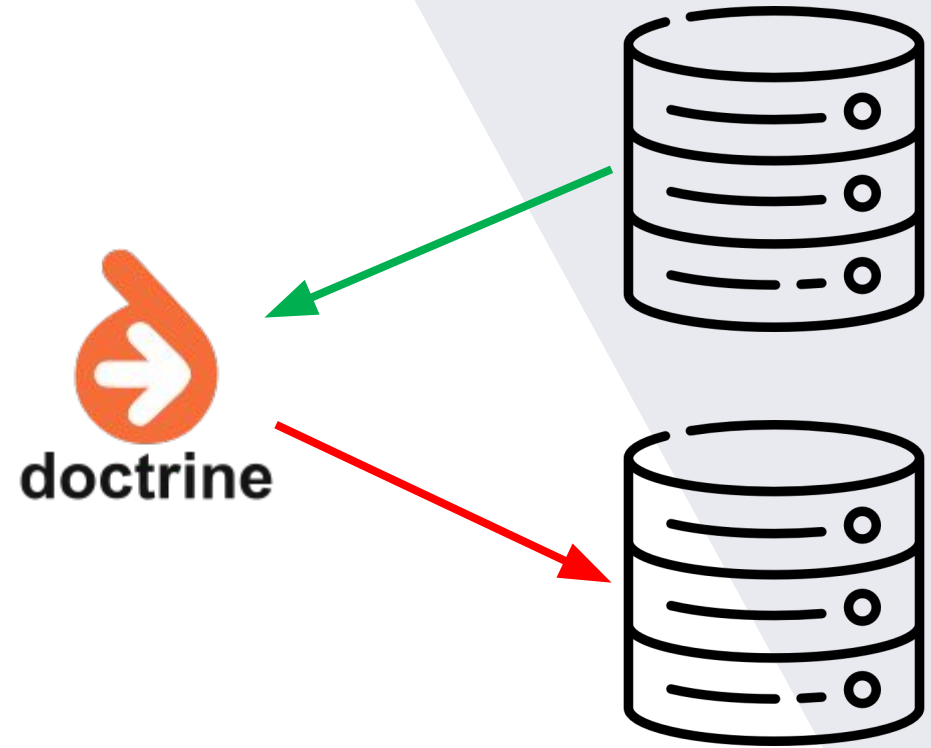
```
/*
 * Updates a user based on the input parameters.
 */
#[Route('/users', methods: ['PATCH'])]
#[Response(UserUpdateResponse::class)]
public function update(
    #[MapQueryString] UserUpdateFilters $userUpdateFilters,
    #[MapRequestPayload] UserUpdateRequest $userUpdateRequest,
): JsonResponse
{
    return $this->json(
        $this->update->handle($userUpdateFilters, $userUpdateRequest)
    );
}
```



- Symfony Router ile Tüm Rotaları Listele
- Reflection API ile Attribute'leri Oku
- OpenAPI Modellerini Oluştur

Reflection API - Modify

- 2 veritabanı arasında veri transferi
- Kod seviyesinde veri transferi
- Doctrine ORM
- Entity?



Reflection API - Modify

```
class User
{
    #[ORM\Id]
    #[ORM\GeneratedValue]
    #[ORM\Column]
    private ?int $id = null;

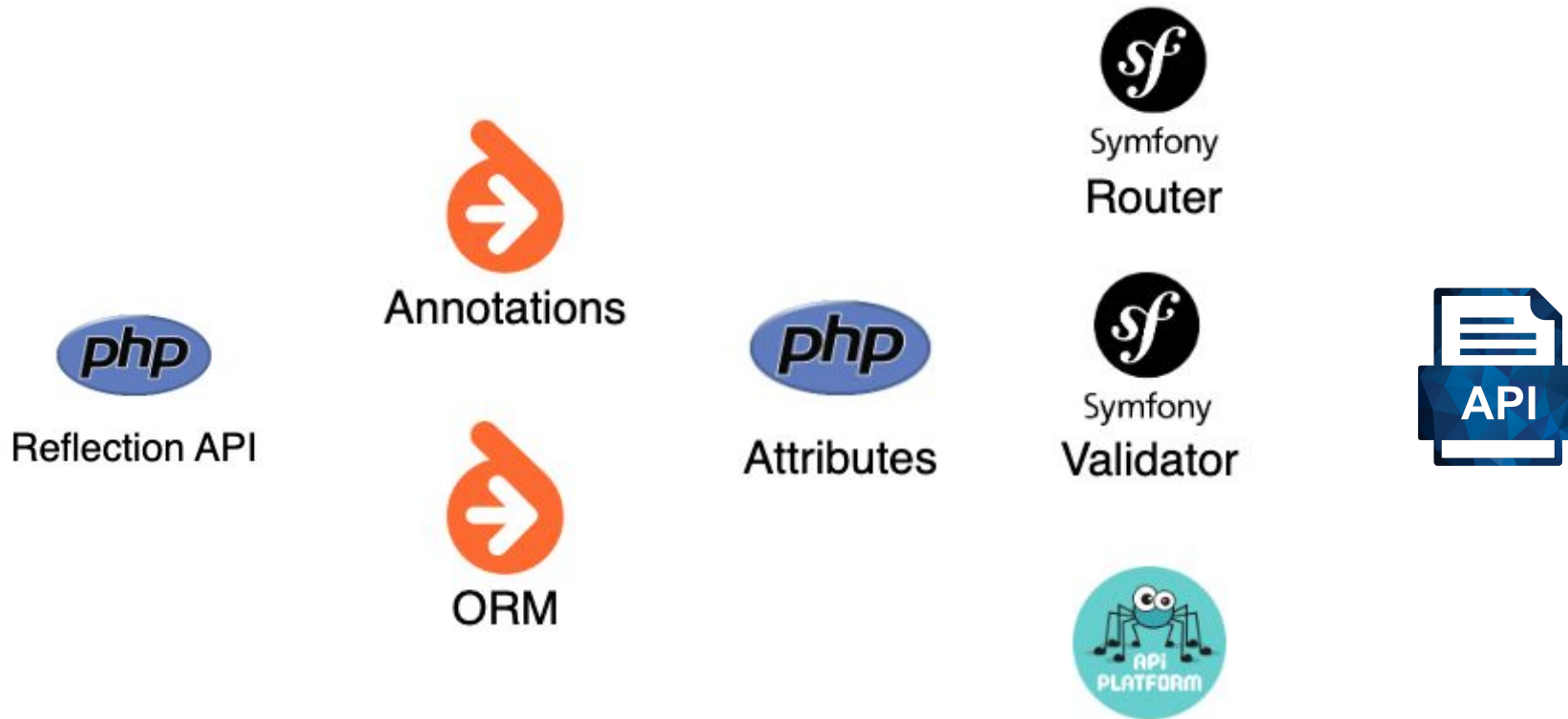
    public function getId(): ?int
    {
        return $this->id;
    }
}
```

```
// Read
$entity = $this->sourceEntityManager->getRepository($entityFqcn)->find(123);

// Set id to null
$reflectionProperty = new \ReflectionProperty($entityFqcn, 'id');
$reflectionProperty->setValue($entity, null);

// Write
$this->targetEntityManager->persist($entity);
```


Reflection API ile Yapılan Projeler



Bonus

PHP AST

Code

```
<?php  
$var = 42;
```

Usage

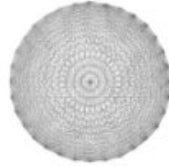
```
var_dump(ast\parse_code($code, $version=70));
```

```
object(ast\Node)#1 (4) {  
  ["kind"]=>  
  int(132)  
  ["flags"]=>  
  int(0)  
  ["lineno"]=>  
  int(1)  
  ["children"]=>  
  array(1) {  
    [0]=>  
    object(ast\Node)#2 (4) {  
      ["kind"]=>  
      int(517)  
      ["flags"]=>  
      int(0)  
      ["lineno"]=>  
      int(2)  
      ["children"]=>  
      array(2) {  
        ["var"]=>  
        object(ast\Node)#3 (4) {  
          ["kind"]=>  
          int(256)  
          ["flags"]=>  
          int(0)  
          ["lineno"]=>  
          int(2)  
          ["children"]=>  
          array(1) {  
            ["name"]=>  
            string(3) "var"  
          }  
        }  
      }  
      ["expr"]=>  
      int(42)  
    }  
  }  
}
```

PHP AST & Tokenizer



AST



PHP-Parser



PHPStan



RectorPHP



Phan



Tokenizer



PHP CodeSniffer



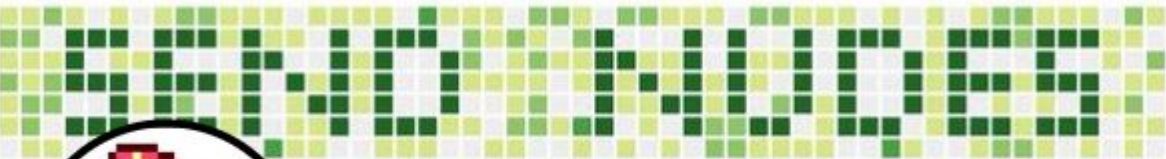
Performans

- Reflection API
- PHP-AST

Marco Pivetta - Doctrine

Contributions in the last year

Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec



Less More



@ocramius@mastodon.social 🇺🇦 ⚙️

@Ocramius

Consultant for [@RoaveTeam](#), [@getlaminas](#) team. I'm a tad blunt, but I mean no harm. Making unpopular decisions, so you don't have to.

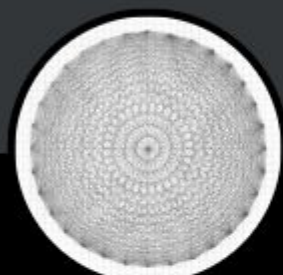
Daily curse of [@_nyunyu_](#).

📁 Science & Technology ⓘ 📍 Frankfurt, Germany
🔗 [github.com/Ocramius/](#) 📅 Joined April 2008

1,389 Following 19.1K Followers

Follow

Nikita Popov - php-ast



Nikita Popov

@nikita_ppv

LLVM toolchain engineer at [@RedHat](#). Open source contributor, mainly PHP / LLVM / Rust. Studied computer science and physics at TU Berlin.

📍 Berlin, Germany 🔗 [nikic.github.io](#) 📅 Joined December 2011

79 Following 21.2K Followers

Follow



Teşekkürler!

Selim Can CABA

Senior Software Developer

linkedin.com/in/selimcancaba

 **WINGIE**
ENUYGUN