



Веб разработка на PHP Symfony

Сериализатор

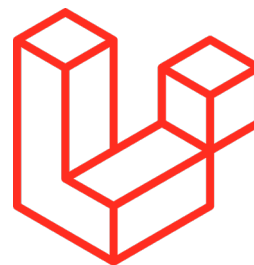
Веб разработка на PHP Symfony

Introduction



Демьян Костельный
Middle PHP Developer

 demian-kostelny-613b90151



Веб разработка на PHP Symfony

Тема урока

Сериализатор

Веб разработка на PHP Symfony

План урока

1. Понятия сериализации и нормализации
2. Сериализация объекта
3. Десериализация объекта
4. Контекст
5. Вложенная сериализация
6. Циклические зависимости
7. Встроенные возможности

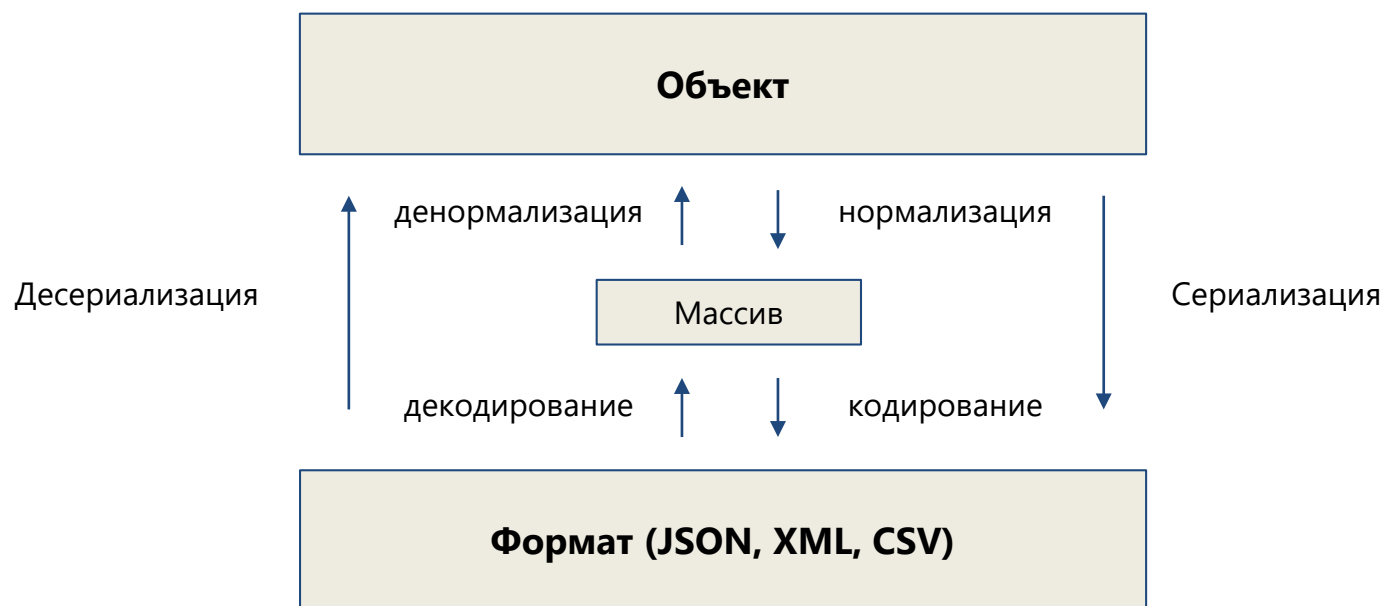
Понятие сериализации и нормализации

Веб разработка на PHP Symfony

Понятие сериализации и нормализации

В Symfony есть компонент, который называется Serializer (Сериализатор), и сейчас мы поговорим о таких понятиях, как сериализация и нормализация.

Процесс сериализации представляет из себя превращение объекта сразу же в такие форматы как JSON, XML, CSV. В случае же нормализации объект сначала нормализируется и становится массивом, а уже после этого кодируется в нужный формат из показанных выше.



Сериализация обекта

Веб разработка на PHP Symfony

Сериализация объекта

Перед тем как мы сможем использовать компонент Serializer, нужно установить его через Composer:

```
composer require symfony/serializer
```

После этого станет возможным использовать функции данного компонента:

```
use Symfony\Component\Serializer\Encoder\JsonEncoder;  
use Symfony\Component\Serializer\Encoder\XmlEncoder;  
use Symfony\Component\Serializer\Normalizer\ObjectNormalizer;  
use Symfony\Component\Serializer\Serializer;
```

```
$encoders = [new XmlEncoder(), new JsonEncoder()];  
$normalizers = [new ObjectNormalizer()];
```

```
$serializer = new Serializer($normalizers, $encoders);
```


Веб разработка на PHP Symfony

Сериализация объекта

Рассмотрим небольшой пример. Пусть у нас будет новый пост, данные которого мы хотим перевести в формат JSON, и делается все это прямо в контроллере:

```
$post = new Post();  
$post->setTitle('New post');  
$post->setDescription('Super nice description');  
$post->setAuthor('Admin');  
$post->setIsDraft(false);  
  
$jsonContent = $serializer->serialize($post, 'json'); // Переводим наш пост в формат JSON
```

Десериализация обекта

Веб разработка на PHP Symfony

Десериализация объекта

Теперь рассмотрим пример того как можно сделать десериализацию объекта, который был закодирован в формате XML:

```
use App\Model\Post; // Чтобы сделать десериализацию объекта - нужно чтобы была подключена его модель

$data = <<<EOF
    <post>
        <title>XML to object</title>
        <description>Here is description</description>
        <author>Admin</author>
        <isDraft>false</isDraft>
    </post>
EOF;

$post = $serializer->deserialize($data, Post::class, 'xml'); // Возвращаем объект в нормальное состояние
```

Веб разработка на PHP Symfony

Контекст

Веб разработка на PHP Symfony

Контекст

В таких форматах, как CSV, XML, Yaml - при сериализации есть третий параметр функции (который указывается массивом), с помощью которого можно задать больше правил для сериализации объекта.

Пример:

```
use Symfony\Component\Serializer\Encoder\CsvEncoder;

$csvEncoder = new CsvEncoder();

$context = [
    "csv_escape_formulas" => true,
    "output_utf8_bom" => false
];

$csvEncoder->encode($array, 'csv', $context);
// $array представляет из себя массив данных которые нужно перевести в формат CSV
```

Веб разработка на PHP Symfony

Контекст

Вообще контекст касается не только форматов, которые были показаны на предыдущем слайде, это также касается и всех остальных.

Если говорить ещё более проще, **контекст** - это то, что позволяет нам настроить полностью процесс кодирования/сериализации/нормализации так, как мы этого хотим.

Вложенная сериализация

Веб разработка на PHP Symfony

Вложенная сериализация

Вложенная сериализация представляет из себя сериализацию объекта на несколько уровней. При этом объект, который кодируется в нужный формат, имеет дочерние элементы, которые также кодируются.

Рассмотрим простой пример объекта класса, который имеет дочерние элементы:

```
$level1 = new Parent();  
$level1->title = 'level1';  
  
$level2 = new Parent();  
$level2->title = 'level2';  
$level2->child = $level1;  
  
$level3 = new Parent();  
$level3->title = 'level3';  
$level3->child = $level2;
```


Веб разработка на PHP Symfony

Вложенная сериализация

Если попробовать закодировать переменную \$level1 из предыдущего примера, то вот что получится:

```
$result = $serializer->normalize($level1, null, [AbstractObjectNormalizer::
    ENABLE_MAX_DEPTH => true]);
/*
$result = [
    'title' => 'level1',
    'child' => [
        'title' => 'level2',
        'child' => [
            'title' => 'level3',
        ],
    ],
];
*/
```

Веб разработка на PHP Symfony

Вложенная сериализация

Также работает и с массивами:

```
use App\Entity\Post;

$post1 = new Post();
$post1->setTitle('New post');
$post1->setContent('Super new cool description');
$post1->setAuthor('Demian Cool');

$post2 = new Post();
$post2->setTitle('New post 2');
$post2->setContent('Super new cool description 2');
$post2->setAuthor('Demian Cool 2');

$posts = [$post1, $post2];
$encoded_data = $serializer->serialize($posts, 'json');
```

Циклические зависимости

Веб разработка на PHP Symfony

Циклические зависимости

Циклические зависимости представляет собой кодирование объектов, которые взаимодействуют между собой.

К примеру, у нас может быть два объекта Person и Group.

Person хранит в себе данные об человеке который состоит в группе. Group имеет поле массива members, которое хранит в себе объекты из переменных Person (т.е. люди которые состоят в группе).

В свою очередь entity - Person имеет поле для группы, к которой он принадлежит. Рассмотрим пример как можно правильно закодировать данные объекты.

Веб разработка на PHP Symfony

Циклические зависимости

```
$person = new Person();
$person->setName('Mike');

$group = new Group();
$group->setName('Super group');
$group->setMembers([$person]);

$person->setGroup($group);

$encoder = new JsonEncoder();
$defaultContext = [
    AbstractNormalizer::CIRCULAR_REFERENCE_HANDLER => function ($object, $format, $context)
    {
        return $object->getName();
    },
];
$normalizer = new ObjectNormalizer(null, null, null, null, null, null, $defaultContext);

$serializer = new Serializer([$normalizer], [$encoder]);
var_dump($serializer->serialize($group, 'json'));
```

Веб разработка на PHP Symfony

Встроенные возможности

Веб разработка на PHP Symfony

Встроенные возможности

Мы уже рассмотрели основной функционал Сериализатора в Symfony.
А что можно сказать об встроенных функциях и возможностях в сериализаторе?
Давайте рассмотрим.

Информационный видеосервис для разработчиков программного обеспечения



Проверка знаний

TestProvider.com



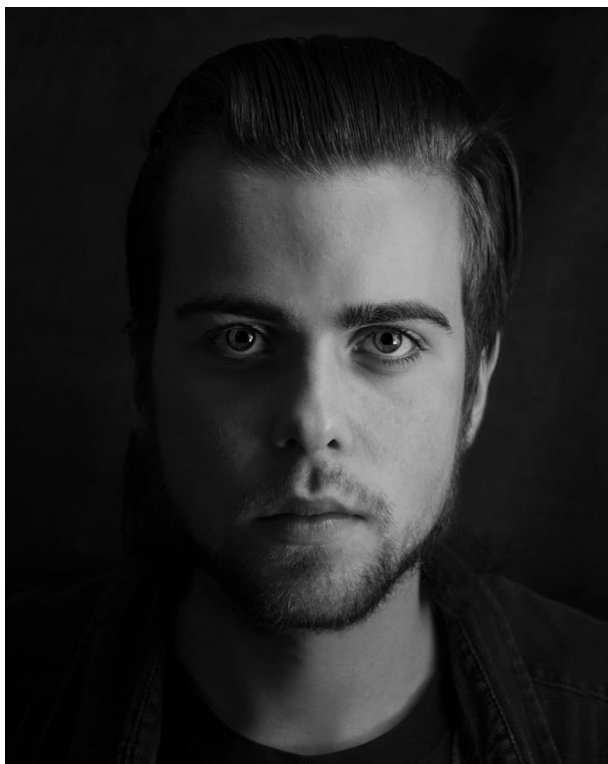
Проверьте как Вы усвоили данный материал на TestProvider.com

TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.

Веб разработка на PHP Symfony

Спасибо за внимание! До новых встреч!



Демьян Костельный
Middle PHP Developer

