

Сериализатор

№ урока: 10 **Курс:** Веб разработка на PHP Symfony

Средства обучения: Eclipse PHP Development Tools, командная строка, любой современный браузер

Обзор, цель и назначение урока

В данном уроке идёт рассмотрение компонента Serializer, который позволяет в приложении объекты классов конвертировать в нужные форматы, например такие, как: XML, JSON или CSV. Это полезно использовать при передаче запросов через API.

Изучив материал данного занятия, учащийся сможет:

- Использовать компонент Serializer для того, чтобы объекты классов преобразовывать в такие форматы JSON, XML и CSV.
- Использовать контекст при сериализации объектов, для того чтобы задавать правила конвертирования объекта.
- Использовать вложенную сериализацию для конвертирования объектов с разными уровнями, т.е. конвертировать объекты, которые содержат в себе другие объекты.
- Делать десериализацию полученного объекта, например переводить из формата JSON в обычный объект PHP.

Содержание урока

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

Резюме

- В Symfony есть встроенный компонент Serializer, который позволяет брать объекты классов PHP и конвертировать их в такие форматы как JSON, XML и CSV. Это удобно использовать при отправке запросов ответов на API запросы, либо же просто отправлять JSON данные на интересующее нас API. Перед тем как начать использовать данный компонент рекомендуется убедиться в том, что он установлен, для этого ввести команду: `composer require symfony/serializer`.
- Сериализация представляет из себя перевод объекта PHP класса в нужный формат, к примеру мы можем взять объект любого класса и перевести его в формат JSON. Делается это с помощью использования уже заранее подключенного класса сериализатора и функции `serialize()`, где в первом аргументе мы указываем какой объект именно хотим конвертировать, а во втором - указываем в какой формат именно хотим конвертировать.
- Нормализация — это процесс, когда мы превращаем объект в массив. Аналогично денормализация - наоборот, массив превращается в объект.
- Десериализация объекта — это процесс, применяемый когда нам требуется перевести объект в каком-то формате (например XML) в обычный PHP объект. Для того чтобы это

сделать - нам требуется объект в закодированном формате и класс, в объект которого мы хотим перевести закодированный объект.

- Если будет потребность в том, чтобы как-то изменить процесс сериализации, который уже идёт по умолчанию, то можно воспользоваться контекстом. Контекст представляет из себя массив, в котором мы задаём правила при сериализации для какого-то определенного формата. Для каждого формата есть свои параметры, которые можно изменить при сериализации либо десериализации - все это можно найти в официальной документации Symfony на странице описания компонента Serializer.
- Компонент Serializer также позволяет пользоваться вложенной сериализацией, т.е. это когда нужно закодировать объект, в котором имеются и другие объекты. Хорошо также работает данная особенность и с использованием массивов.

Закрепление материала

- Что такое сериализации и десериализация объекта и зачем её использовать?
- Зачем нужно использовать нормализацию?
- Что нужно для того, чтобы перевести закодированный объект в формате JSON в обычный PHP объект определенного Entity?
- Что нужно сделать чтобы перевести в формат XML объект, который имеет в себе другие объекты?
- Что такое контекст в сериализации и для чего он используется?

Дополнительное задание

Задание

Сначала разобрать пример сериализации объекта в формат JSON. То есть берём уже готовый Entity объект, который имеет в себе заполненными все поля и конвертируем его через созданный сериализатор в формат JSON. После этого разобрать пример десериализации объекта с использованием уже класса, который имел изначально объект. Для десериализации мы пользуемся созданным нами сериализатором в отдельной переменной. После всего этого идёт рассмотрение контекста и то, как можно его использовать при сериализации, чтобы задать отдельные параметры конвертирования объекта в другой формат.

Самостоятельная деятельность учащегося

Задание 1

Достать любой доступный Entity из приложения и создать новый объект этого Entity который будет иметь в себе заполнены все поля. После того как объект был создан, подключить класс сериализации в контроллер и сделать перевод объекта в формат JSON.

Задание 2

Взять предыдущий объект и уже перевести в формат CSV, но при этом нужно воспользоваться контекстом. Позволяется воспользоваться документацией, чтобы посмотреть какие параметры можно задавать в контексте. В контексте можно воспользоваться любыми двумя параметрами, которые изменят в итоге внешний вид объекта в формате CSV.

Задание 3

Создать объект Entity (желательно с полем relation), который будет иметь в себе заполненные поля и будет хранить в себе другой объект. После создания объектов правильно сделать их сериализацию в формат JSON.

Рекомендуемые ресурсы

Официальная документация Symfony

<https://symfony.com/doc/current/index.html>

Документация Symfony на рус. языке (имеет в себе некоторые устаревшие материалы)

<https://symfony.com.ua/doc/current/index.html>