

Сервис контейнер

№ урока: 3 **Курс:** Веб разработка на PHP Symfony

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

Обзор, цель и назначение урока

Цель прохождения урока - научиться пользоваться такими компонентами как Service Container, DI (Dependency Injection), сервисы внутри Service Container и Compiler Pass. Кроме этого, в начале урока затрагиваются такие принципы как DIP и IoC. Данные принципы нужно обязательно знать разработчику-новичку, поскольку это уже относится даже к шаблонам проектирования, в свою очередь применение данных принципов делает код более чистым и понятным.

Изучив материал данного занятия, учащийся сможет:

- Понимать, как можно регистрировать собственные сервис классы и также посторонние в Symfony приложении, что позволит ему расширять функционал для разработки и писать независимые классы, которые служат для выполнения какой-то одной операции (например рассылки).
- Пользоваться уже установленными внутренними сервисами Symfony приложения.
- Уметь использовать принципы Сервис контейнера и Dependency Injection на практике в Symfony, а также в других фреймворках, благодаря самому понимаю принципов.
- Сможет использовать теги в фреймворке для того, чтобы можно было указывать какой сервис и что должен делать.

Содержание урока

1. Service container, DI, DIP, IoC
2. Установка
3. Autowiring
4. Биндинг параметров
5. Тегирование сервисов
6. Compiler pass

Резюме

- Service Container - является контейнером внутри каждого Symfony приложения и содержит в себе уже по умолчанию установленные сервисы (классы или даже целые библиотеки), которые позволяют полноценно обеспечить процесс разработки. Сервисы предоставляют самые разные классы для разных задач во время процесса разработки.
- Можно также создать и свой собственный сервис в приложении с целью, чтобы как-то расширить внутренний функционал системы.
- Внедрение зависимостей — это один из принципов шаблонов проектирования, которые на сегодняшний день очень часто используется. Суть внедрения зависимостей заключается в том, что какие-то классы используют методы других классов. Простым примером может послужить класс А - который для реализации своего полного функционала использует некоторые функции из класса Б. Это и есть внедрение зависимости, ведь класс А зависим от некоторых функций класса Б.

- DI или Dependency Injection - является шаблоном проектирования, который даёт внешнюю зависимость классу. Работает по принципу внедрения зависимостей и абсолютно полностью поддерживается в Symfony.
- Autowiring или же автоматическое определение зависимостей, представляет собой возможность управлять сервисами из сервис контейнера с минимальными затратами на саму конфигурацию сервисов.
- Тегирование сервисов предоставляет возможность давать инструкции сервисам к каким событиям они могут принадлежать. либо указывать, что они должны делать в системе.

Закрепление материала

- Что такое Service Container и для чего он используется в Symfony приложения?
- Как можно воспользоваться классом какого-то сервиса в контроллере?
- Как можно создать собственный сервис в Symfony приложении?
- Зачем используется класс ContainerBuilder?
- Как можно в config/services.yaml добавлять параметры к сервисам?
- Какие есть функции для того, чтобы работать с параметрами сервисом в контроллере?

Дополнительное задание

Задание

Создать свой небольшой сервис в приложении, которое будет генерировать случайное число. После этого зарегистрировать сервис в самом приложении и протестировать как он работает в любом из доступных контроллеров приложения.

Задание

Создать в созданном сервисе 1-2 параметра и прописать им значение именно через файл конфигурации сервисов.

Самостоятельная деятельность учащегося

Задание 1

Создать и зарегистрировать свой собственный маленький сервис в приложении, который будет складывать два числа и вызываться в любом из доступных контроллеров.

Задание 2

К созданному сервису создать 2 параметра. Этим параметра задать значение через файл конфигурации config/services.yaml. После этого в любом из доступных контроллеров сделать проверку на их существование и вывод их значения.

Задание 3

Создайте в сервисе параметр в качестве константы и также дайте данному параметру глобальное значение.

Рекомендуемые ресурсы

Официальная документация разработчиков Symfony

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

Документация Symfony на русском языке (много уже устаревшего материала)

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