Создание админ панели к блогу

№ урока:	7	**Курс:**	Создание веб приложений на PHP

Средства обучения:	Eclipse PHP Development Tools или любая другая система для работы с кодом, Open Server, Command Line. любой веб-браузер

Обзор, цель и назначение урока

В этом уроке идёт рассмотрение того, как можно создать отдельную админ панель к приложению и также создать систему ролей для пользователей, которая и позволит контролировать доступ к админ панели приложения.

Изучив материал данного занятия, учащийся сможет:

- С использованием Doctrine ORM написать собственную админ панель, которая отделяется от остального приложения.
- Реализовать ролевую систему, которая будет проверять доступ для админку.
- Писать собственные страницы для админ панели, для лучшего управления данными приложения.

Содержание урока

1. Логика ролей в приложении
2. Реализация ролевой системы
3. Конфигурация ролей

Резюме

- Для реализации ролевой системы понадобится создать отдельную таблицу (roles) в базе данных, которая будет хранить в себе все возможные роли для пользователей приложения. Важным также является заметить, что в таблице пользователей (users) приложения нужно будет добавить поле role, суть которого будет заключаться в хранении id роли из таблицы ролей. Также в таблице ролей будет отдельное поле permissions, где будет храниться JSON-объект с прописанными параметрами доступа к админ панели. Базовым параметром для доступа к админ панели является admin_access, который может принимать только значение true или false.
- После реализации нужных таблиц, следующим шагом будет создание контроллера самой админ панели. Он будет отвечать только за главную страницу админ панели, авторизацию и также иметь в себе функцию для проверки того, имеет ли пользователь нужную роль - через БД.
- Любой другой контроллер, который будет отвечать за самые разные функции в админ панели, должен будет всегда иметь в себе проверку на роль текущего пользователя именно через БД. Это надежнее, чем проверять через сессию.
- Когда в админ панели нужно будет реализовать какое-то управление определенным контентом, то удобнее всего создавать CRUD контроллеры, которые будут иметь в себе все формы для работы с данными. Это очень часто используется в приложениях, которые пишутся на фреймворках и используют уже готовую админ панель.
- Часто вместо того, чтобы самому писать админ панели, можно воспользоваться уже готовым решением с github. Примером может послужить репозиторий chetans9/core-php-admin-panel, где уже есть готовое приложение с готовой админ панелью и авторизацией. Но, например, в современных фреймворках, таких как Laravel, тоже есть

CyberBionic systematics

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

готовые инструменты для создания админ панели, примером может послужить Orchid Admin panel. Многие готовые инструменты уже предоставляют возможность сразу же создать через консоль CRUD контроллеры на готовые таблицы в БД.

Закрепление материала

- Зачем в таблице roles существует поле permissions и как оно используется?
- Как реализуется проверка в контроллерах админ панели без использования сессии?
- Почему в админ панели чаще используются CRUD контроллеры?
- Зачем лучше использовать готовые инструменты для создания админ панели?
- Почему JSON объект удобен для хранения параметров доступа?

Дополнительное задание

- Используя текущую админ панель, которая была сделана на протяжении всего урока - написать дополнительный контроллер в админ панели для редактирования данных из таблицы roles. Здесь применить стандартный CRUD функционал и Doctrine. Важным является то, чтобы была форма добавления и редактирования полей, а также отдельная страница для просмотра всех существующих ролей в приложении. И, кроме этого, также добавить все это в меню админ панели.
- Написать дополнительный контроллер к уже готовой админ панели из урока, который будет иметь CRUD функционал для работы с таблицей posts. Здесь создать форму для создания постов, просмотра всех текущих постов и их удаления.

Самостоятельная деятельность учащегося

- Для начала учащему потребуется создать новую таблицу для приложения, где будут храниться все роли для пользователей приложения, и также добавляет новое поле в таблицу с ролью пользователя.
- Учащий сам создает полностью контроллер для авторизации и главной страницы админ панели, где вместе с этим идёт проверка роли пользователя. Параллельно с этим студент создает новый контроллер, который будет предназначен для работы с ролями приложения в самой админ панели.
- Когда контроллер админ панели готов вместе с формой для входа в админ панель, и также создан layout самой админ панели, учащий создает отдельный контроллер для админ панели, суть которого будет заключаться в том, чтобы работать с данными пользователей приложения. Для этого учащий также создает все формы и создает отдельные маршруты. Также учащемуся необходимо самому создать проверку текущей роли пользователя.
- В конце урока идёт рассмотрение некоторых готовых инструментов для использования уже готовой админ панели в приложении.

Рекомендуемые ресурсы

Официальный сайт IDE - PHP Development Tools
https://www.eclipse.org/pdt/

Официальный сайт разработчиков - Composer
https://getcomposer.org/

Официальный сайт разработчиков локального сервера - Open Server
https://ospanel.io/
Официальный сайт разработчиков локального сервера - MAMP
https://www.mamp.info/en/windows/

CyberBionic systematics
CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Официальный сайт разработчиков - Git
https://git-scm.com/

Официальная документация разработчиков - Doctrine ORM
https://www.doctrine-project.org/projects/orm.html

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com