

Админ-панели для Laravel: сравнительная таблица

	October CMS	Laravel Nova	Orchid	Voyager
Доступы	Разграничение пользователей по ролям и деление на группы.	Групп нет: любой пользователь из штатной таблицы Users получает полный доступ в админку. Ограничения доступа придётся делать вручную.	Из коробки доступны таблицы с пользователями и таблица с ролями (плюс таблица расшивки, чтобы настроить их взаимоотношения). Из стандартного: доступ в админку, управление через медиаменеджер, доступ к редактированию пользователей и ролей.	Есть группы пользователей, можно добавлять свои права. Нет разграничения пользователей по типу контента: любой пользователь может вставлять любой контент (скрипты, embed, и т.д.).
Связи между сущностями	Штатные решения Laravel для связей разработчики отвергли и написали своё . Из документации до конца неясно, как это работает и как получить программный доступ к настройкам.	Есть штатные HasMany, BelongsTo и т.д. Древовидной структуры — нет. С большим объемом данных справляется нормально.	Штатные для Laravel отношения вроде HasMany, BelongsTo и т.д.	Стандартно через HasMany, HasOne, BelongsToMany, BelongsTo и т.д.; в админке дублируются связи (через создание полей в Bread), можно задать "обязательные" связи. Древовидных структур нет.
Формы	Для сущностей в базе данных формы описываются YAML-файлом с описанием всех полей и их атрибутов. Для публички форм нет, всё придётся делать руками.	Самих по себе форм нет. Зато есть стопка классов для определения полей редактирования сущностей таблиц — можно кастомить через наследование.	Есть свои формы для добавления или редактирования. Также есть FormBuilder, с которым можно создавать формы без привязок к сущностям БД, но с аплодом файлов придётся разбираться.	Формы для контента генерируются по описанию BREAD-сущностей. Форм для публички нет, приходится использовать сторонние решения.
Фильтры списков и сортировки	Отличный функционал фильтрации из коробки: любые типы полей и даже поиск по отношениям.	Неудобный функционал фильтрации из коробки: только по одной колонке.	Есть простой фильтр, сложный придётся писать руками, используя неудобный механизм с добавлением колонок. Сортировки работают штатно в алфавитном порядке.	Есть простой фильтр (поле, операция сравнения <, >, =, like и значение). Бывают баги из-за разных вариантов реализации: с рендерингом на сервере и рендерингом на клиенте через js. Сортировки — по значению выводимых полей (по колонке).
Редактор текста	Вместо одного из привычных текстовых редакторов — своя реализация. Но можно кастомизировать стили через админ-панель и список разрешённых html-тегов.	Из коробки доступен Trix editor со стандартным набором функций для модификации текста. На вид — выходец из Windows 98, редактирования исходного кода нет, возможностей для кастомизации, собственно, тоже. Есть поле для визуального редактирования Markdown.	Есть обычный — для визуального редактирования (SimpleMDE), есть классический — для редактирования как html (Quill). Есть широкий набор возможностей для построения графиков.	Встроенный редактор текста TinyMCE (относительно свежей версии) с джентльменским набором функций: параграфы, списки, таблицы, изображения, редактирование в режиме html-кода. Можно подключить плагины от TinyMCE.
Письма, почта, шаблоны	Есть редактор шаблонов в админке и отправка тестового письма.	Ничего готового нет.	Ничего готового нет.	Только через стандартные уведомления Laravel.