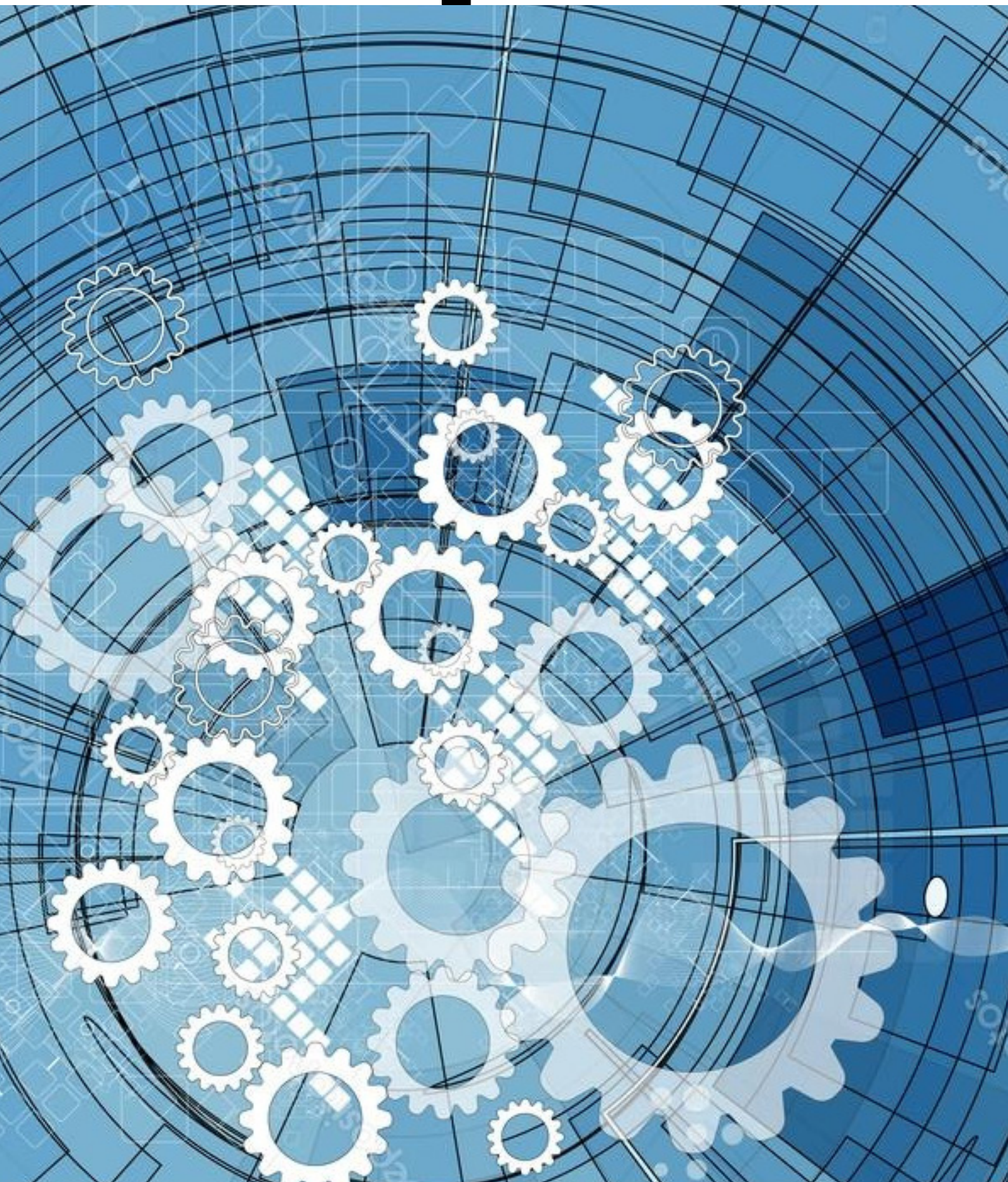




СТАНЬ ЭКСПЕРТОМ С НАМИ !

expert106



При разработке любого программного продукта перед командой разработчиков прежде всего стоит задача грамотного выбора программной платформы, определяющей структуру программной системы.

Для этого нужно учесть достаточно большое количество характеристик, от «как быстро всё будет работать» до «а необходима ли нам эта фича?». И так каждый раз. Именно в моменты мозгового штурма команда сравнивает удобство фреймворка, скорость, набор фич, которые реализованы в нем или в совместимых с ним модулях.



Но какой же всё-таки лучше, быстрее и производительнее? Разработчики постоянно проводят сравнение фреймворков, чтобы прояснить для себя этот вопрос.

Рассмотрим же такой фреймворк, как RNPixie.



RNPixie —фреймворк, поддерживающий схему HMVC.

Это сравнительно новый фреймворк, созданный в 2012 году для сайтов-визиток.

Как и Symfony, RNPixie состоит из двух частей: библиотеки компонентов и базового проекта, правда в случае RNPixie базовый проект более тонкий и состоит всего из нескольких файлов.

Интересный факт о RNPixie: согласно исследованию Sitepoint 2015 года, это любимый фреймворк украинских PHP-разработчиков.

RNPixie также популярен среди группы респондентов до 18 лет. Наверное это благодаря картинке с феечкой на сайте :D

Нельзя не отметить, что RNPixie построен на независимых компонентах, которые можно легко использовать отдельно от фреймворка. Все они протестированы на 100%, имеют отличные показатели качества и требуют минимальных зависимостей

КОМПОНЕНТЫ

PHPixie



AUTH

библиотека аутентификации

Обрабатывает аутентификацию пользователей после криптографически защищен-



DATABASE

Доступ к базам данных с использованием интуитивно понятного построителя запросов с расширенной поддержкой функций.



DI

Простота настройки и доступа к Вашим пакетам. Может использоваться как контейнер для инъекций зависимостей и локатор обслуживания.



HTTP

Представление PSR-7 для HTTP-запросов и ответов. Обеспечивает удобные в использовании обертки для них и поддерживает обработку файлов cookie и сессий.



IMAGE

Простая библиотека обработки изображений, обеспечивающая общий интерфейс для GD, ImageMagick и GraphicsMagick.



MIGRATE

Легко обновить схему базы данных, отследить изменения и вставить данные с помощью простых команд.



ORM

Быстрое сопоставление данных с использованием оптимизированных запросов и минимальной конфигурации.



PAGINATION

Простые в использовании массивы, поддерживающие разбиение на страницы, запросы ORM и пользовательские источники данных.



TEMPLATE

Очень расширяемый механизм шаблонов, поддерживающий макеты.



VALIDATE

Проверка как простых форм, так и сложных структур данных с помощью интуитивно понятного API, расширяемых фильтров и простой обработки ошибок.

Для работы этого фреймворка необходимо создать скелет проекта, используя Composer.

После этого нужно настроить HTTP сервер на папку /web directory. В случае Nginx понадобится также настроить конфигурацию файлов.

Если вы используете Windows вам надо будет создать ярлык /web/bundles/app указывающий на /bundles/app/web.

На Linux и MacOS ярлык будет работать из коробки. Он нужен для того чтобы открыть доступ к веб файлам дефолтного бандла.



Такая острая необходимость в настройке доступа к бандлам нужна потому, что PHPixie 3 поддерживает систему бандлов, как например делает Symfony2.

Если вы еще с ними не работали представьте себе что сайт можно теперь разбить на логические части, которые потом легко переносить в другие проекты и делится через Composer.

Например, аутентификация пользователей может быть создана как отдельный бандл со своими шаблонами, стилями и изображениями а затем использована на нескольких проектах.

Бандлы «монтируются» в проект как например в линуксе диски монтируются в файловую систему.



- **BUNDLE SYSTEM**

Возможность повторно-го использования пакета, а также использование пакета как самостоятельной части с помощью Composer.

- **MONGODB SUPPPORT**

Поддержка компонентов базы данных, работающих с MongoDB, включая отношения между базами данных SQL и коллекциями Mongo.

- **LINEAR CODE FLOW**

Избегается использование событий.

Каждый новый класс должен иметь factory-метод (например в App\Pixie) для создания его экземпляров.

Такой подход позволяет легко заменить один класс другим, что особенно важно при написании юнит тестов.

- **BEST PRACTICES**

Разработаны с нуля в соответствии с принципами SOLID и отраслевыми

стандартами.

- **HARDENED SECURITY**

Использование криптографически защищенных хэшей, случайных токенов и безопасной обработки файлов cookie.

- **FIRST-HAND SUPPORT**

Нужна помощь? Просто напишите в чат и получите помощь в кратчайшие сроки.



Заключение

RNPixel начался как микро-структура и постепенно стал одним из самых популярных фреймворков fullstack, сохраняя при этом высокую производительность.

Это отчасти из-за строгой архитектуры, которая позволяет избегать распространенных ошибок, таких как зависимость от статических методов, глобального масштаба, синглтонов и других антипаттернов, что также гарантирует, что код легко читается, отлаживается, расширяется и тестируется.



Фактически, все компоненты RNPixel имеют полный пакетный охват тестирования.

Это фреймворк РНР, который никогда не будет стоять на вашем пути и обеспечит вам полный контроль над потоком выполнения, который легко освоить.

А существенных минусов как таковых у данного фреймворка нет, кроме того, что документации на русском для фреймворка мало (разве что посты на хабре).