Инструкция

Приложение сделано на основе базового Advanced template фреймворка Yii2. Докумментация

Как поднять

1) Склонировать проект

git clone git clone git@bitbucket.org:vintageua/pivdenniy-back.git

либо

git clone https://{\$user_name}@bitbucket.org/vintageua/pivdenniy-back.git если нету ssh-ключа

2) В корне приложение лежит файл инициализации проекта. Выполнить комманду php init. При инициализации нужно выбрать рабочее окружение: Development или Production

3) Подтянуть библиотеки через композер коммандой:

composer install - если композер установлен глобально

php composer.phar install - если композер установлен под проект Установка composer

4) После инициализации (п. 2) приложения будут сформированы нужные локальные файлы для работы приложения и припишутся нужные права на папки.

common/config/main-local.php - хранит основную иформацию по подключению к БД:



Тут же можно задать данные по подключения SMTP мейлера для отправки писем с уведомлениями (если будут). Подробнее об отправке опишу ниже.



```
'class' => 'rmrevin\yii\postman\Component',
 'driver' => 'smtp',
 'default_from' => ['r.testmailer@gmail.com', 'Mailer'], // email + µмя
posepa
 'subject_prefix' => '',
 'subject_suffix' => null,
 'table' => '{{%postman_letter}}',
 'view_path' => '/email',
 'smtp_config' => [
 'host' => 'smtp.gmail.com', // cepBep
 'port' => 587, // порт
 'auth' => true,
 'user' => 'r.testmailer@gmail.com', // email, c которого будет
 ocymecrbлятьcя отправка писем
 'password' => '', // пароль
 'secure' => 'tls',
 'debug' => false,
 ]
],
....
]
```

5) Накатить миграции на базу данных командой php yii migrate

6) B frontend/config/params-local.php добавить URL для REST API:



Настройка веб-сервера

1) Для корректной работы сайта, рекоммендую использовать связку под nginx + php-fpm (>= 7.2)

2) На сервер должна быть возможность создавать симлинки и выполнять консольные комманды

3) Стоит обратить внимание на структуру приложения. По факту, у нас есть разделение проекта на 3 независимых приложения: backend, frontend и api. Все исходники, классы, стили и т.д. находятся в соответствующих директориях. Frontend - демонстративный функционал создания заявок и работы REST API.

/frontend/web, /frontend/web и api - наши web-root директории, в которых хранятся наши входные скрипты index.php. Веб сервер (nginx) нужно сконфигурировать (создать хосты) на эти две директории. В них же хранится вся статика. Настроить сжатие и кеширование, нужные редиректы при необходимости.

4) Для удобства, можно вынести эти симлинки на эти директории в любое место на сервере и

сконфигурировать сервер под них.

Отправка почты

Для того, что бы не отправлять письма "на лету" и ускорить сабмиты форм на сайте - мы используем компонент, который позволяет нам эти письма просто сохранять в БД, и отправлять адрессатам отложенно.

По-этому, обязательно, нужно настроить cron комманду на отпавку еще не отправленных писем

Например, */2 * * * /path/to/application/yii send/email - для проверки и отправки каждых 2 минуты. Время можно настроить под себя.

Последующие правки

По договоренностям, со стороны Vintage надается read-доступ к репозиторию проекта. В тот момент, когда будуь вноситься правки в проект, вы сможете просто подтягивать изменения через git. Если правки и изменения будут касаться БД - мы используем систему миграций. С вашей стороны просто нужно будет выполнить комманду php yii migrate для внесения изменений в БД.