Heroku - это облачная платформа, которая позволяет компаниям создавать, поставлять, контролировать и масштабировать приложения - мы самый быстрый способ перейти от идеи к URL-адресу, минуя все эти головные боли в инфраструктуре.

Heroku - это сервис, который позволяет компаниям тратить время на разработку и развертывание приложений, которые сразу же начинают приносить прибыль.

 Услуги Heroku можно использовать только для разработки и запуска приложений в инфраструктуре служб Heroku. Вы не можете получить доступ к Услугам Heroku для целей подачи иска о нарушении прав интеллектуальной собственности в отношении SFDC или с целью создания продукта или услуги, конкурирующих с Сервисами Heroku.

Учетные записи получают базу по 550 часов в месяц, в которой могут работать ваши бесплатные диноды. В дополнение к этим базовым часам счета, которые [проверяют с помощью кредитной карты](https://devcenter.heroku.com/articles/account-verification), получат дополнительные 450 часов ежемесячной бесплатной динозависимости.

Использованные свободные динамограммы не подлежат передаче. Если вы передаете приложение, которое использовало часы для другой учетной записи, то часть, которую вы использовали, когда приложение находилось в вашей учетной записи, останется до тех пор, пока она не сбрасывается в следующем месяце.

Когда вы используете, все ваши бесплатные динозавры в течение определенного месяца, все бесплатные приложения в вашем аккаунте будут вынуждены спать всю оставшуюся часть месяца.

Развертывание с использованием Heroku Git

Интерфейс командной строки Heroku (CLI) упрощает создание и управление приложениями Heroku непосредственно с терминала. Это важная часть использования Heroku.

**Проблема со связыванием Open Server с PHPStorm 9.0.2**

Настройка **PHPStorm 9.0.2**

Перевод скрина:

Создайте php.ini-файл в каталоге 'D:\OpenServer\modules\php\PHP-7 или C:\Windows',.

Шаблоны конфигурационных файлов можно найти в корневой папке php:

"php.ini-development "- настройки по умолчанию

"php.ini-production " - рекомендуемые настройки.

Скрин в папке Рисунки.

Установил Composer. При проверке правильности установки командой «composer –V» выдало: «"""" не является внутренней или внешней командой, исполняемой программой или пакетным файлом». Проблему решило изменение в настройках Open Server. Файл php.exe при установке Composer я загрузил в папку D:\OpenServer\modules\php\PHP-7. А в этой директории Open Server-а имеются файлы composer.phar и composer.bat устаревшей версии и они подхватываются в Composer. Чтобы их не удалять, на всякий случай, я их только переименовал, добавив «1».

**Алгоритм**

//Создал репозиторий на GITHab, загрузил туда файлы.

Создал директорию d:/GitHub/Alice.

Поддержка Heroku PHP будет применяться к приложениям только тогда, когда приложение имеет файл с именем composer.json в корневом каталоге.

Heroku использует Composer для управления зависимостями в проектах PHP. Если ваше приложение имеет какие-либо зависимости, composer.json файл должен указать их. Также этот файл инструктирует Героку, какую версию PHP использовать. Если такую инструкцию не указать, то Heroku выберет самую последнюю возможную версию PHP 5. Даже если приложение не имеет зависимостей от [Composer](http://getcomposer.org/) , оно должно содержать хотя бы **пустое** ( {}) composer.json, чтобы быть распознанным как приложение PHP.

В редакторе создал файл composer.json и загрузил в папку /Alice.

Создал файл блокировки composer.lock-

composer install

и загружается папка vendor/ .

Затем требуется убедиться, что новые требования «заморожены» composer.lock, запустив: composer update.

**Профайл**

PROCFILE представляет собой механизм объявления о выполнении команд с помощью приложения на платформе Heroku. Его следует назвать точно Procfile, а не иначе и без указания расширения.

Файл должен быть помещен в корневую директорию вашего приложения. Он не будет функционировать, если он помещен в подкаталог.

PROCFILE представляет собой текстовый файл в корневом каталоге приложения, который определяет типы процессов и прямо заявляет, что команда должна быть выполнена, чтобы запустить приложение.

Файл Procfile содержит несколько деклараций типа процесса, каждая из которых начинается с новой строки. Каждый тип процесса представляет собой объявление команды, которая выполняется при запуске dyno этого типа процесса.

Файл Procfile содержит несколько деклараций типа процесса, каждый из которых содержит новую строку. Каждый тип процесса представляет собой объявление команды, которая выполняется при запуске dyno этого типа процесса. Например, если объявлен webтип процесса, то при запуске dyno этого типа будет выполняться команда, связанная с webтипом процесса. Это может означать, например, запуск веб-сервера.

Procfile будет выглядеть примерно так:

web: vendor/bin/heroku-php-apache2

Создал текстовый файл Procfile без расширения в папке /Alice.

Проверить работу команды можно из командной строки следующим образом:

heroku run web

Файл объявляет один тип процесса web и команду, необходимую для его запуска. Название web,, важно здесь. Он объявляет, что этот тип процесса будет прикреплен к [стекю HTTP-маршрутизации](https://devcenter.heroku.com/articles/http-routing) Heroku и будет получать веб-трафик при развертывании.

В командной строке Администратора (консоле) перейти из C:\Windows\system32> в директорию D:\GitHub\Alice где файлы приложения командой:

C:\Windows\system32>cd /D d:/GitHub/Alice Ключ /D поменяет не только текущий каталок, но и диск.

Итог: d:\GitHub\Alice>

Создать приложение в Heroku: heroku create

Развернуть свое приложение в Heroku:

git push heroku master.

Файлы из \Alice копируются в приложение Heroku.

**Полезные команды**

Чтобы попасть в корень приложения на Heroku команда: heroku run bash -a=<наименование приложения>

Удалить приложение <shielded> с Heroku команда: heroku apps:destroy --app shielded

**Webhook** – это способ, или механизм изменения поведения веб-страницы, или веб-приложения с помощью пользовательских обратных вызовов. Эти обратные вызовы могут поддерживаться, модифицироваться и управляться сторонними пользователями и разработчиками, которые необязательно могут быть связаны с исходным веб-сайтом или приложением.

heroku config:set -a fast-mountain-48254 HEROKU\_OAUTH\_ID=ad351582-e824-4467-b343-b722ede79696 HEROKU\_OAUTH\_SECRET=b7f459aa-bef9-441f-8922-8b758fdc5a64

Ваша подписка на веб-хост будет уведомлять вас о каждом развертывании приложения по вашему выбору. Вы можете выбрать один из существующих приложений Heroku или создать новый.

3. Добавьте веб-сайт выпуска для $ TRIGGER\_APP

Замените $ TRIGGER\_APP именем приложения, созданного на последнем шаге. Кроме того, вы можете использовать одно из существующих приложений Heroku в качестве триггерного приложения.

heroku webhooks: add --include api: release -l sync --url https://fast-mountain-48254.herokuapp.com/webhooks -s e84c9220e602c7ed3e20f9f3f650f43bb3feaa7df040dcf2da7d7c3c4f925c26 -a $ TRIGGER\_APP

Secret alice-ya: 2d3f54d3ab74cd3c787967128b0238e758d0fd92ff5365d071d891581c98

**Редактирование приложения на облачном хостинге Хероку**

1. Зайти в папку локального репозитория командой

cd /D <директория>

1. Выполнить изменения в локальных файлах.
2. Добавить измененные файлы в локальный репозиторий

git add .

1. Зафиксировать файлы в репозитории

git commit -m "<комментарий>"

1. Развернуть приложение с изменениями на сервере хостинга

git push heroku master

Если нет локальных файлов, то клонировать источник существующего приложения из Heroku с помощью Git , используйте heroku git:clone команду:

heroku git:clone –a <имя приложения>

cd <имя приложения>.

Heroku предоставляет услугу git в первую очередь для развертывания, а возможность клонирования из нее предлагается в качестве удобства. Мы настоятельно рекомендуем хранить ваш код в другом репозитории git, таком как GitHub, и относиться к нему как к каноническому.

**Алиса**

В браузере:

Страница недоступна

Сайт **agile-harbor-63876.herokuapp.com** пока не может обработать этот запрос.

HTTP ERROR 500

В логах приложения Application Logs запись:

PHP Fatal error: Call to undefined function mb\_internal\_encoding() in /app/index.php on line 9

Проблему решило удаление этой функции в файле index.php.