**Вариативная самостоятельная работа (ВСР)**

**Настройка дистанционного ресурса на платформе ApostropheCMS.**

Апостроф - это 100% JavaScript, как в браузере, так и на сервере. Асинхронный характер Node.js позволяет разрабатывать быстрые сайты. Apostrophe использует MongoDB для богатой, интуитивно понятной модели хранения документов и построен на основе надежной платформы Express для добавления пользовательских маршрутов в любой модуль. Легко создавать шаблоны с помощью Nunjucks, и компилировать таблицы стилей LESS.

Если установлено необходимое ПО для Apostrophe, мы можем приступить к созданию вашего первого проекта!

## Генерация тестового проекта

Первое, что вы захотите сделать, это взять инструмент, который делает загрузку нового проекта на одном дыхании! apostrophe-cli

# Install the apostrophe-cli module

npm install apostrophe-cli -g

**Вы получили ошибку «Отказано в доступе»?** В Linux это может быть связано с тем, что глобальные пакеты npm должны быть установлены от имени пользователя root при обычной установке.

Теперь можно использовать apostrophe-cli для создания нового проекта:

# Create a project  
apostrophe create-project test-project

# Alternatively, you can install a robust, themed, fully-baked project with  
apostrophe create-project --boilerplate https://github.com/apostrophecms/apostrophe-open-museum test-project

**Важно: вместо** test-project **используйте «короткое имя» вашего собственного проекта, содержащее только буквы, цифры, дефисы и / или подчеркивания. Он будет использоваться по умолчанию в качестве имени базы данных MongoDB и основы для имен** файлов **cookie и т. Д.**

После создания проекта вам просто нужно подключиться, установить зависимости и запустить сервер.

# cd into the new project

cd test-project

# Install our dependencies (most notably, Apostrophe)

npm install

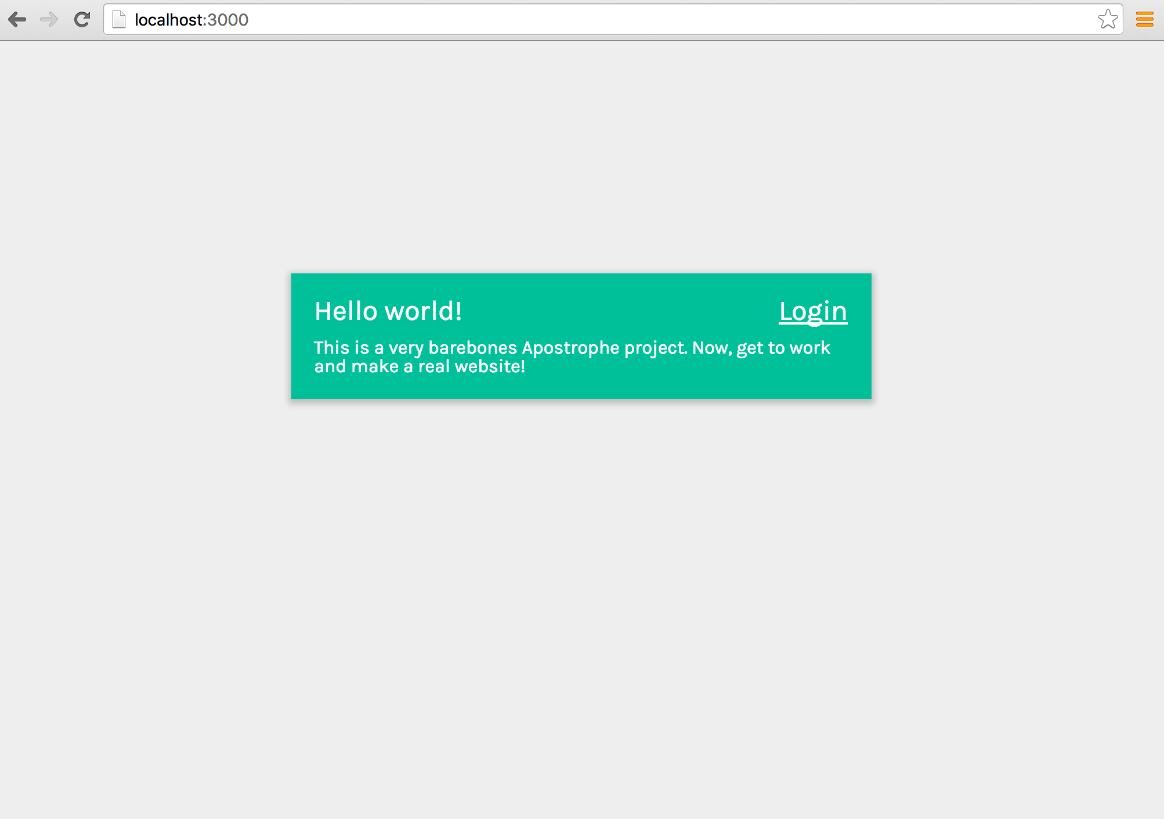
# Add an admin user to the admin group; prompts for password

node app.js apostrophe-users:add admin admin

# Go go go!

node app.js

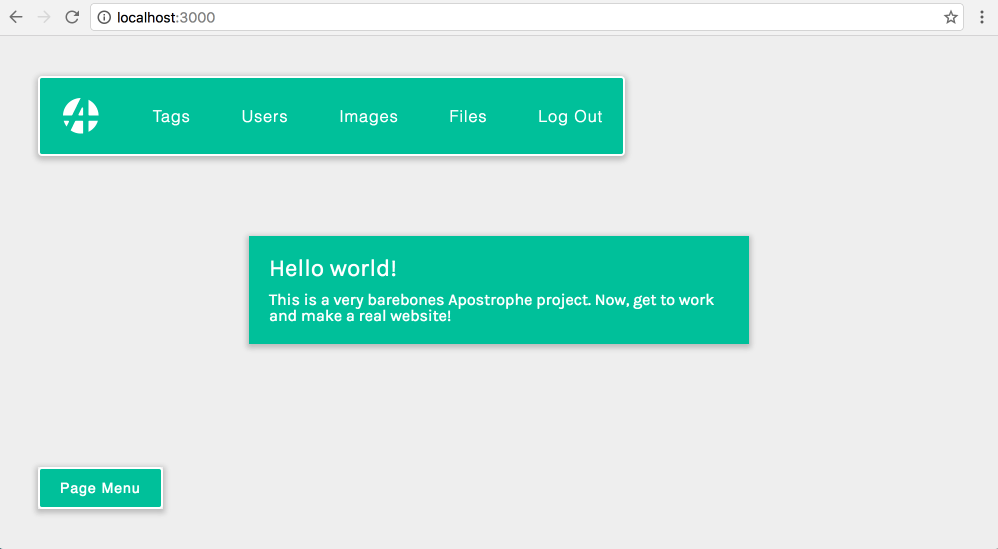
Теперь у вас есть тестовый проект http://localhost:3000 и он работает! У вас есть возможность посетить и просмотреть свой сайт.



Начальная страница шаблона

## Вход в систему

Мы можем войти через браузер по адресу http://localhost:3000/loginapostrophe-users:add admin. **Используйте пароль, который вы выбрали минуту назад,** когда запускали задачу (см. Выше), и **имя пользователя**. После входа в систему мы снова перенаправляемся на домашнюю страницу, но с двумя дополнительными элементами: админ-бар, плавающий в верхней левой части страницы, и меню страницы, плавающее в нижней левой части.



Зашли на шаблон

## Давайте сделаем это интересным

Теперь давайте сделаем это более интересным! Время добавить редактируемую область контента на страницу.

Откройте ваш текстовый редактор lib/modules/apostrophe-pages/views/pages/home.html и просто добавьте вызов: apos.area

{% extends "layout.html" %}

{% block title %}Home{% endblock %}

{% block main %}

<div class="main-content">

<h3>Hello world!

{% if not data.user %}

<a class="login-link" href="/login">Login</a>

{% endif %}

</h3>

<p>This is a very barebones Apostrophe project. Now, get to work and make a real website!</p>

{{ apos.area(data.page, 'body', {

widgets: {

'apostrophe-images': {

size: 'full'

},

'apostrophe-rich-text': {

toolbar: [ 'Styles', 'Bold', 'Italic', 'Link', 'Unlink' ],

styles: [

{ name: 'Heading', element: 'h3' },

{ name: 'Subheading', element: 'h4' },

{ name: 'Paragraph', element: 'p' }

]

}

}

}) }}

</div>

{% endblock %}

## Перезапуск вашего сайта

**Теперь перезапустите сайт,** чтобы увидеть влияние изменений:

1. Нажмите «CTRL-C» в окне терминала, где работает node app.js.
2. Запустите снова node app.js (совет: просто нажмите стрелку вверх, затем введите).
3. Обновите браузер.

## Используйте Nodemon для просмотра изменений файлов

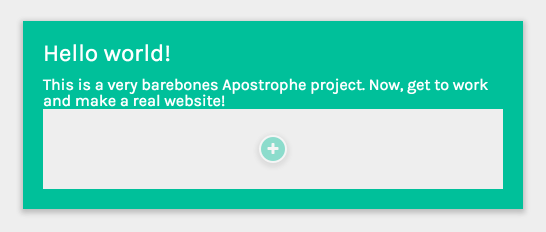
Чтобы избежать перезапуска вручную в будущем, вы можете использовать [Nodemon](https://nodemon.io/) для просмотра файлов проекта.

1. Запустите npm install -g nodemon, чтобы установить его глобально.
2. Используйте nodemon в корневой папке вашего проекта. Nodemon будет работать node app.js по умолчанию.

Теперь, каждый раз, когда вы вносите изменения в файл в вашем проекте, Nodemon перезапускает ваше приложение.

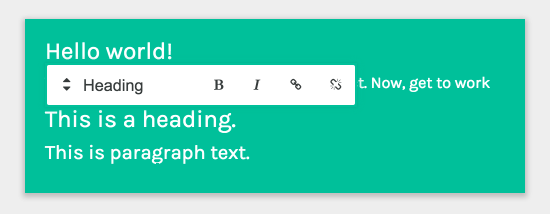
Если вы установили Apostrophe с использованием apostrophe-cli описанного выше, файл предоставляет базовую конфигурацию Nodemon package.json.

## Работа с областями



### Добавление расширенного текста

Нажмите знак «+» и выберите «расширенный текст». Вам представлен дружественный редактор:

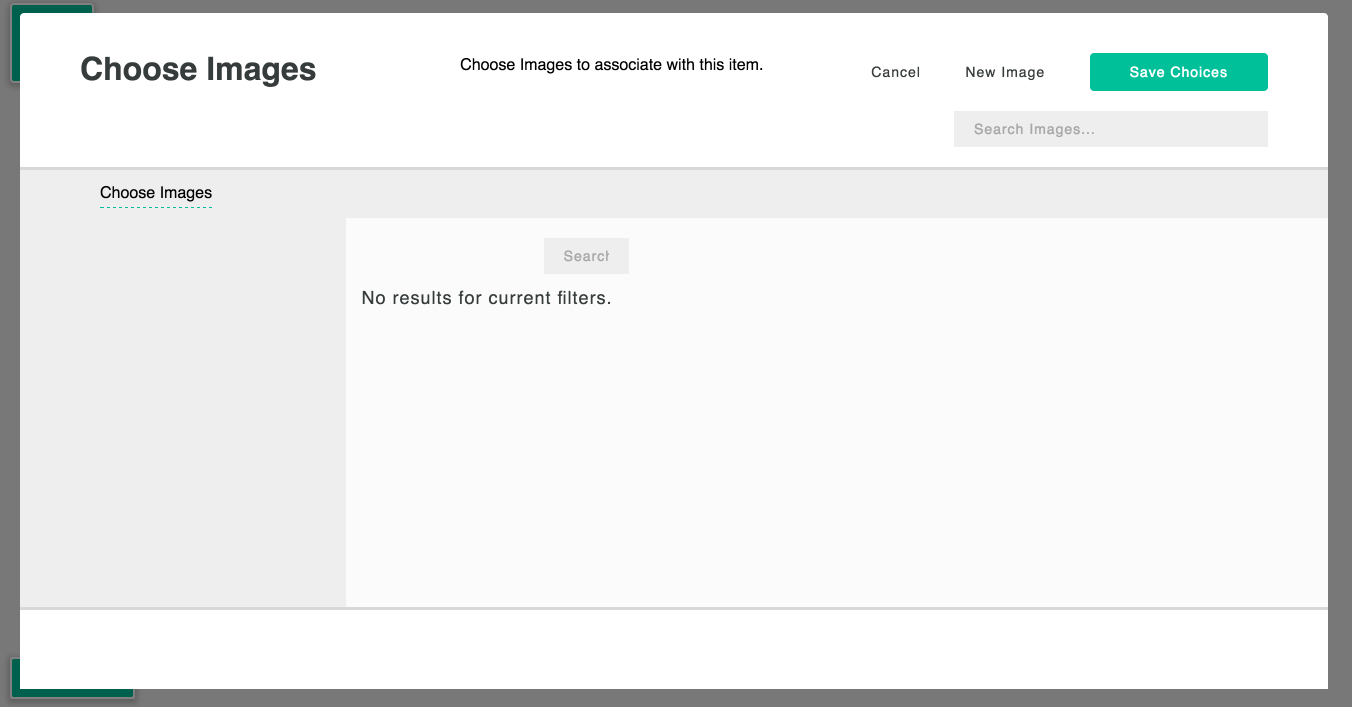


Отредактируйте, как считаете нужным, и попробуйте обновить страницу. Обратите внимание, что ваши изменения уже сохранены. Нет кнопки сохранения, потому что сохранение происходит автоматически.

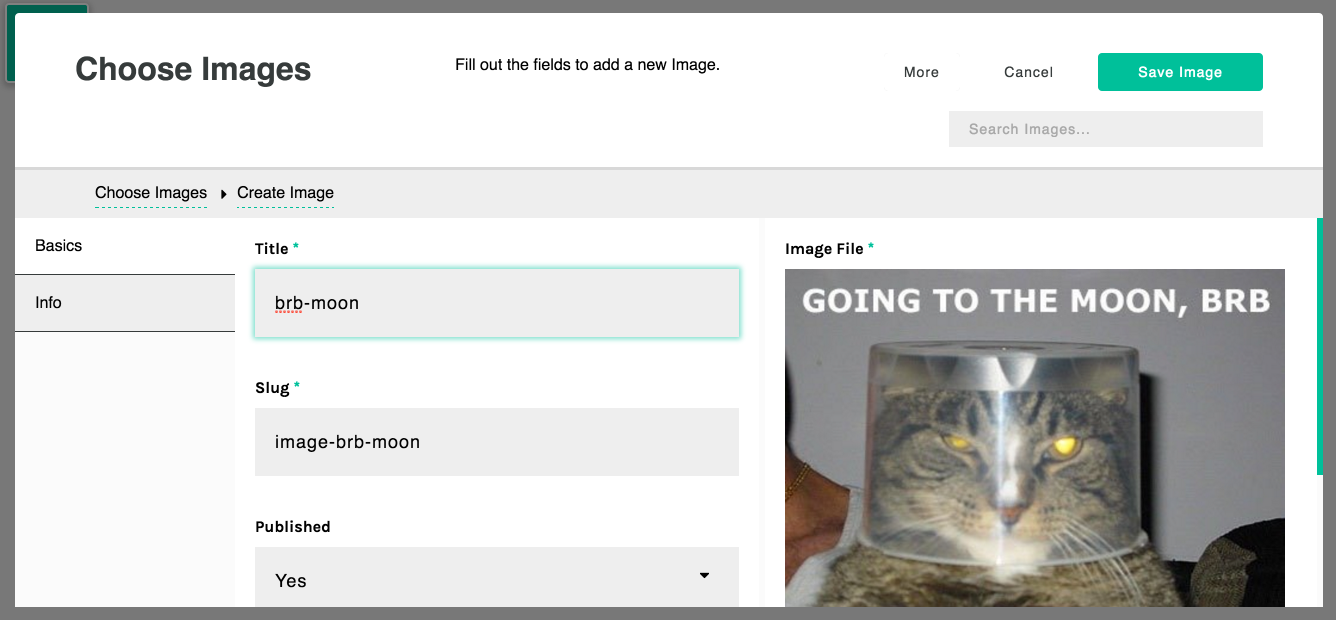
### Добавление слайд-шоу

Теперь давайте добавим слайд-шоу. Нажмите за пределами редактора форматированного текста, и вы увидите два новых знака «+»: один над текстом, а другой под ним. Нажмите один из них и выберите «Изображения».

Вы увидите библиотеку изображений, которая изначально пуста:



Нажмите кнопку «Новое изображение» в правом верхнем углу, и вы увидите:



Нажмите «Загрузить файл», чтобы выбрать файл GIF, JPEG или PNG для загрузки с вашего компьютера. Также заполните поле заголовка. Затем нажмите «Сохранить изображение».

Снова нажмите «Новое изображение» и загрузите второй файл. Затем установите флажок рядом с каждым из них:

