**Настройка окружения**

Для начала стоит установить базовое программное обеспечение необходимое для работы Apostrophe. В следующих примерах мы установим все необходимое и создадим пустой сайт с использованием Apostrophe CMS.

**Установка программного обеспечения**

Дальнейшие инструкции предназначены для операционной системы **MacOS X**, которую большинство разработчиков используют в качестве тестирования своего кода, который в итоге будет развернут на операционной системе Linux. Смотрите данную статью [Установка Apostrophe под windows](https://js-node.ru/site/article?id=78) если вы являетесь пользователем операционной системы **Windows**. Пользователи **Linux** могут установить те же packages, которые приведены в данной статье [Размещение Apostrophe в продакшн](https://js-node.ru/site/article?id=61).

Этот туториал предназначен для операционной системы Mac версии OSX 10.9 или лучше.

Нам потребуется данное программное обеспечение:

* XCode (обязателен для компиляции под Mac)
* Homebrew (для легкой установки)
* Git
* Node & NPM
* MongoDB
* Опционально: Imagemagick (для обработки изображений с поддержкой анимированных GIF-ов)

**Установка XCode**

XCode предоставляется бесплатно компанией Apple. [Установка XCode с помощью App Store](https://itunes.apple.com/ru/app/xcode/id497799835?mt=12)

**XCode Command Line Tools**

Вам также необходим инструмент командной строки XCode, который можно установить по следующему маршруту Xcode ->Preferences ... -> Downloads.

**Использование терминала**

С этого момента все, что отформатировано подобным образом, должно запускаться в командной строке терминала. Базовое знакомство с терминалом будет очень полезно для разработки под node и Apostrophe

**Установка Homebrew**

**Homebrew** в значительной мере облегчит отслеживание и обновление пакетов программного обеспечения.

Выполнив данную команду можно установить Homebrew:

/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"

Возникли проблемы ? Обратитесь к официальной документации [Homebrew](https://brew.sh/)

**Установка Node**

Одна команда сделает все, что необходимо (если Homebrew был установлен)

brew install node

**Настройка PATH**

Командная строка ищет исполняемые файлы в директориях перечисленных в переменной среды PATH. Нам нужно добавить в $PATH путь к нашей node директории.

Создайте текстовый файл (если он еще не создан) с именем .profile в вашей **home** (корневой) директории и откройте его в текстовом редакторе.

touch ~/.profile && open ~/.profile

Добавьте в него две строки:

export NODE\_PATH="/usr/local/lib/node"

export PATH="/usr/local/share/npm/bin:$PATH"

Следует проверить успешность данных действий; для этого откройте новое окно терминала и выполните команду:

echo $PATH

Начиная с добавленной вами информации должны вывестись текущие настройки PATH разделенные двоеточием.

**Установка NPM (Node Package Manager)**

Выполните команду, чтобы проверить установлен ли NPM

npm

Если возникла ошибка "not found".

curl -L https://npmjs.org/install.sh | sh

**Установка git**

Git нужен для управления исходным кодов. В Mac он по умолчанию установлен, но если вы сомневаетесь в его наличии или хотите обновиться до последней версии, выполните команду:

brew install git

**Установка MongoDB**

В качестве базы данных ApostropheCMS использует MongoDB. Установить MongoDB с помощью homebrew можно следующим образом:

brew install mongo

Во время установки будут выводиться инструкции, которые вы должны выполнить, чтобы монго запускалось при входе в систему. Прочтите, что там написано, потому что последний шаг может изменяться.

Теперь откройте новое окно терминала и выполните:

mongo

Это должно запустить интерактивную сессию монго в консоле. Из которой можно выйти путем нажатия ctrl+c.

"Эй, нет соединения с MongoDB!". Возможно вы не следовали инструкциям, которые выводились после установки.

**Установка ImageMagick**

Выполните для установки:

brew install imagemagick

Теперь в командной строке вам доступны следующие команды convert и identify, которые Apostrophe использует, чтобы быстро обрезать и изменять масштаб изображения. *Если вы пропустите этот шаг, то Apostrophe все равно будет обрабатывать и загружать изображения, но это будет на порядок медленнее. Спасибо [Jimp](https://www.npmjs.com/package/jimp)*.

### Создание проекта

Теперь, когда все необходимое установлено, можно приступить к созданию тестового сайта.

Первое, что нам потребуется - это глобально установить данную утилиту apostrophe-cli, которая с легкостью позволяет генерировать новые проекты.

npm install apostrophe-cli -g

Переходим в директорию где будет находиться наш сайт и генерируем его следующей командой:

apostrophe create-project test-project

**Важно! Вместо test-project можно использовать любое другое "short name" (короткое имя) содержащее только буквы, цифры, дефис и/или нижнее подчеркивание. Это имя будет использоваться для названия базы данных в MongoDB, названия cookie и т.д. (В большинстве случаев дефис более распространен, чем нижнее подчеркивание).**

Далее нужно перейти в корень созданного нами проекта, установить зависимости и запустить сервер.

# Переходим в корень проекта

cd test-project

# Устанавливаем зависимости

npm install

# Чтобы зайти в админскую часть добавляем нового пользователя под именем admin; нужно будет ввести пароль

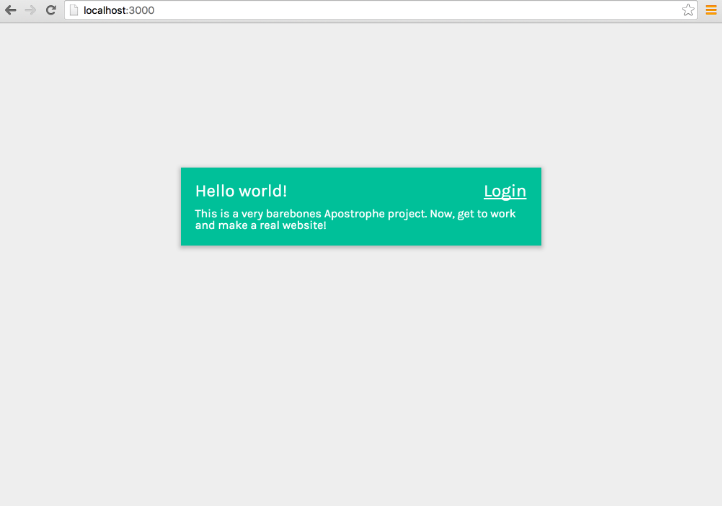
# База данных MongoDB должна быть запущенна

node app.js apostrophe-users:add admin admin

# Запускаем

node app.js

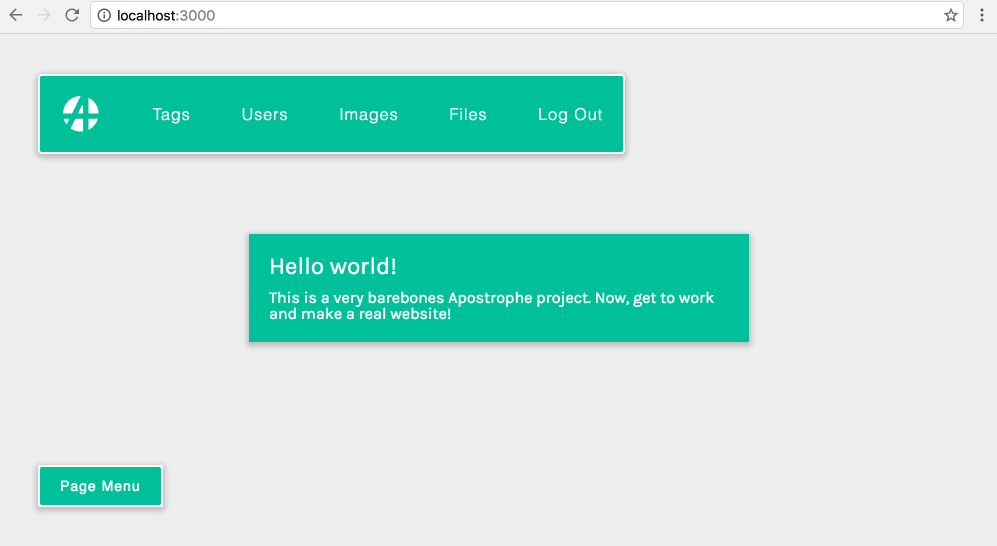
Теперь у нас есть готовый и запущенный проект! Вам следует перейти по этой ссылке http://localhost:3000 и увидеть простую домашнюю страницу, очень простую! Что же, давайте исправим это! Добавим больше страниц и редактируемых областей.



### Вход в систему

Вход в систему можно осуществить по данному пути http://localhost:3000/login. Используя пароль, который вы указали, когда выполняли команду apostrophe-users:add и username (имя пользователя) admin.

После входа в систему вы будите перенаправлены на главную страницу, но в этот раз будут отображены несколько дополнительных элементов, таких как панель администратора, находящаяся в верхнем левом углу и меню страницы располагающееся в нижнем левом углу.



### Сделаем это более интересным

Пришло время добавить область редактируемого контента на нашу страницу.

Откройте lib/modules/apostrophe-pages/views/pages/home.html в вашем текстовом редакторе и добавьте следующий контент:

{% extends data.outerLayout %}

{% 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 %}

#### **Перезагружаем сайт**

Теперь можно видеть на что повлияли изменения:

* Нажимаем "control-C" в том окне терминала, в котором запущен node app.js (вам нужно оставлять node app.js запущенным пока ваш сайт работает. В продакшене будут другие способы о которых мы поговорим позже)
* Запускаем node app.js снова (подсказка: нажимаем клавишу вверх, затем enter)
* Перезагружаем браузер