Автореферат

Наименование Разработка web-приложений для мобильных систем

Автор А.В.Михалькевич

Специальность Задача веб-разработчика – сделать онлайн-контент удобным для просмотра с любых устройств. Особенности ТОП-5 популярных браузеров нужно знать глубоко. Адаптивная кроссбраузерная вёрстка и быстрая загрузка страниц – это стандарт. Web-разработчик обязан следить за изменением алгоритмов ранжирования Google и Яндекс, чтобы сайты держались высоко в списке выдачи.,

Анотация

Anotation in English

Ключевые слова

Количество символов 94820

Содержание

Введение

1 Обзор технологий

В данном разделе предоставлен обзор используемых технологий для разработки web-приложений.

1 .1 Введение в разработку web-приложений для мобильных устройств

Сами технологии, используемые в программировании, можно разделить на две большие группы: серверные, и клиентские.

Причем, если ранее, на заре зарождения web, ожидалось, что основную нагрузку на себя возьмет серверная часть, т.к. сервера будут мощнее компьютеров пользователей. Оказалось же, что компьютеры пользователей по характеристикам ничем не отличаются от серверов. Поэтому нагрузка решений между сервером и клиентом распределена примерно поровну.

Так же и в рынке IT, выделяется 2 направления: FrontEnd (или клиентская) и BackEnd (или серверная). И 3 группы технологий (для backend, для frondend и общий).

FrontEnd:

По сути, в понятие FrontEnd входит всего три технологии: HTML, CSS и JavaScript. Но перечень знаний и умений frontend-разработчика значительно шире. Это:

1. Умение разрабатывать разные типы верстки: табличная и блочная. Резиновая и фиксированная верстка.
2. Гибкая блочная верстка
3. Адаптивная или мультимедийная верстка.
4. Знать и уметь пользоваться такими форматами данных, как JSON (основной формат) и XML.
5. Селекторы: классы, идентификаторы, тэги, вложения, атрибуты и фильтры.
6. Bootstrap 3, API Bootstrap – использование библиотек для адаптивной (или мультимедийной) верстки.
7. Использование в коде архитектурных шаблонов MVC (модель-вид-контроллер), HMVC (иерархические модель-вид-контроллер), MV-VM (модель-вид-пользователь-вид-модель) и MVW (модель-вид-что-то еще).
8. Библиотека запросов jQuery. jQuery позволяет сделать из любого селектора объект.
9. Фрэймворки Angular, React.
10. CoffeeScript
11. Less: переменные, миксины или функции, расширения, импорт, вложенность, соединение в одно свойство несколько свойств.

12. Системы сборки FrontEnd-а. Gulp, grunt, webpack.
13. HTML5 и API HTML5: видео и аудио, холст, перетаскивание, работа с файлами, геолокация (в том числе работа с GOOGLE и YANDEX-картами), web-хранилища, взаимодейтсвие с сервером. Вспомогательный инструментарий.
14. Основы SEO.
15. Schema.org, генератор schema.org микроформат данных, микроданные.

BackEnd

1. Linux-подобные операционные системы. Знать, уметь с ними работать, настраивать и администрировать.
2. Языки программирования.

PHP +

Java +

Python +

Perl -

Ruby -

Asp.net -

1. СУБД (это ПО предназначенное для работы с базами данных):

MySQL (PHPMyAdmin)

NoSQL, SQL-lite

Oracle

MS-SQL

1. Один, лучше 2-3 фрэймворка. Для PHP – это Laravel, YII, ZendFramework Symfony. На сегодняшний день лучшим PHP фрэймворком является Laravel. Для Python - это Flex и Django.
2. Системы управления контентом CMS: Drupal и 1Сбитрикс MoDex, Joomla, WordPress, Magenta, OpenCart.
3. Менеджер зависимостей. Для PHP - Composer. Для Python - pip.
4. Web-интерфейсы для серверных решений: PHPMyAdmin, Webmin, cloud, media-center.
5. Передача потоковых данных между браузерами, с помощью проекта с открытым исходным кодом WebRTC.
6. Платежные системы. Подключение платежных систем.

7. Парсинг. Библиотеки и классы для парсинга внешних страниц. CURL. Имитация действий пользователя.
8. Умение делать тесты, профилировать, дебагировать приложение.
9. Умение делать аудит, анализ и консультирование проектируемых и уже существующих сайтов. Ajile-технологии разработки проектов. Умение разрабатывать проекты без технического задания.
10. Основы хакинга. PHP-include (вставка исполняемых хакерских php-скриптов на взломанном сайте/сервере, бывает локальный и удаленный), SQL-injection (вставка исполняемого sql-кода в строку запроса, url), XSS (это вставка html/JavaScript/VBScript-кода в сгенерированные сервером страницы; пассивный — срабатывает только при передаче пользователю сформированной хакером ссылке; активный — непосредственная вставка html-кода в сгенерированную страницу).

Инструментарий

1. IDE:NetBeans, PHPShtorm, WebShtorm, – или любая другая интегрированная среда разработки.
2. OpenServer – Локальный сервер (Содержит встроенные технологии:PHP, Apache, MySQL и системы управления MySQL).
3. Git – система контроля версий. Кроме git сущестуют и другие системы контроля версий, такие как Mercurial, SVN, Subversion. github.com использует git, bitbucket.org – git и mercurial. Основной и наиболее востребованной системой контроля версий является GIT.
4. Composer – локальный менеджер зависимостей.
5. Firefox, Chrome – Браузер.
6. Firebug – Отладчик кода на стороне клиента
7. FireFTP – или любой другой FTP-клиент.
8. Node.js – программная платформа
9. Npm — менеджер зависимостей для Node
10. Фрэймворк Laravel

***

Frontend и backend программисты постоянно сталкиваются с новыми технологиями, которые приходится изучать.

На что в первую очередь следует обрадить внимание:

Возможности технологии. Для чего и в каких случаях рационально использовать данную технологию.

Особенности синтаксиса.

Документация.

Платформенная настройка. Установка, обновление и удаление неоходимых инструментов.

Hello World. Если вы разобрались как вывести hello world на изучаемой технологии, значит вы уже изучили половину этой технологии. На изучение оставшихся, хотя бы 30%, могут уйти годы практики.

Работа с выводом (шаблонами)

Базы данных, хранение данных

Регистрация авторизация

Обработка изображений

Формы, обработка данных из форм.

Безопасность и защита отдельных страниц.

События

Клиент-сервер

Чтобы упростить изучение новых технологий, можно отметить, что существует только один программный язык: язык логики, все вышеперечисленные технологии являются частными случаями. В ходе развития IT что-то может появляться новое, что-то устаревает. Сами потребности отсеивают ненужное технологическое нагромождения и приводят к новому. Нужно быть вовлеченным в разработку, чтобы остановить свой выбор на правильных технологиях, которые действительно упрощают решение задач.

Важно, для самообразования использовать не только Интернет, а задействовать как можно больше органов восприятия (ведение конспекта, схематизация, пересказ и т.д.).

1 .2 Платформы мобильных устройств

Платформы мобильных устройств: Android, iOS, Windows Phone. Статистика использования мобильных устройств. Установка приложений. Использование языка Java и JavaScript. Магазины приложений: AppStore и Google Play. Альтернатива магазину приложений. Понятие кроссплатформенной разработки. Фрагментация операционной системы. Процесс создания мобильного приложения. Формулировка и описание задачи.

Кроссплатформенная разработка

Если разработчики в процессе написания приложения пользуются принятым для конкретной платформы языком программирования, будь то Objective-C и Swift для iOS или Java или Kotlin для Android, такое приложение будет называться нативным (от англ. native — родной, естественный).

Преимущества нативных приложений:

- скорость работы и отклика интерфейса. Приложение реагирует на нажатия мгновенно, практически отсутствуют задержки в анимации, скроллировании, получении и выводе данных;
- понятный и простой доступ к функциям и датчикам устройства. Для разработчика не представляет проблемы работа с геолокацией, пуш-уведомлениями, съёмкой фото и видео через камеру, звуком, акселерометром и другими датчиками;
- возможность углублённой работы с функциями смартфона. Как и в предыдущем пункте, такие вещи, как анимации, создание сложных интерфейсов и работа нейросетей прямо на устройствах реализуются, может быть, и не просто, но прогнозируемо;
- родной для платформы интерфейс. Нативные приложения обычно оперируют «платформенными» элементами интерфейса: меню, навигация, формы и все остальные элементы дизайна берутся от операционной системы и потому привычны и понятны пользователю.

Недостаток один — дороговизна разработки и поддержки, в том числе потому, что для каждой платформы надо писать свой код.

Кроссплатформенные приложения пишутся сразу для нескольких платформ на одном языке, отличном от нативного. Как такой код может работать на разных устройствах? Тут тоже есть два подхода.

Первый заключается в том, что на этапе подготовки приложения к публикации он превращается в нативный для определённой платформы с помощью транспилера. Фактически один кроссплатформенный язык программирования «переводится» на другой.

Второй — в том, что к получившемуся коду добавляется определённая обёртка, которая, работая уже на устройстве, на лету транслирует вызовы из неродного кода к родным функциям системы.

Предполагается, что большая часть такого кода может переносится между платформами — очевидно, что, например, логика совершения покупок, сохранения товара в корзину, просчёта маршрута для такси, написания сообщения в мессенджер не меняется в зависимости о того, Android у клиента или iOS. Нужно лишь доработать UI и UX для платформ, но сейчас, в определённых пределах, даже это можно объединить — например, меню-гамбургер активно используется как на Android, так и на iOS. Так что даже внесений исправления в интерфейс для того, чтобы приложение отвечало духу и букве нужной платформы — вопрос желания, необходимой скорости и качества разработки.

Преимущества:

- стоимость и скорость разработки. Так как кода надо писать заметно меньше, то и стоимость работ снижается;
- возможность использовать внутренние ресурсы компании. Как мы покажем дальше, разработку кроссплатформенных приложений зачастую можно осуществить силами уже существующих у вас программистов.

Недостатки:

- неродной интерфейс или, как минимум, необходимость работы с интерфейсом каждой

платформы отдельно. У каждой системы свои требования к дизайну элементов и иногда они взаимоисключающи. При разработке это необходимо учитывать;
проблемы в реализации сложных функций или возможные проблемы работы даже с простыми процедурами в силу ошибок самих фреймворков разработки. Кроссплатформенная среда лишь транслирует запросы к системным вызовам и интерфейсам в понимаемый ею, системой, формат, и потому на этом этапе возможны как сложности с пониманием, так и возникновение ошибок внутри самого фреймворка;
скорость работы. Так как кроссплатформенная среда является «надстройкой» над кодом (не всегда, но в определённых ситуациях), в ней возникают свои задержки и паузы в отработке действий пользователя и выводе на экран результатов. Это было особенно заметно несколько лет назад на смартфонах, более маломощных относительно сегодняшних, однако сейчас, с ростом производительности мобильных устройств, этим уже можно пренебречь.

Как видите, эти два метода практически являются зеркальным отражением друг друга — то, что плюсы у нативной разработки приложений, минусы у кроссплатформенной, и наоборот.

PhoneGap

Одно из самых популярных направлений в кроссплатформенном программировании, которое часто по-народному называют PhoneGap. Фактически создаётся мобильный сайт, который «оборачивается» небольшим платформенным кодом, транслирующим вызовы от системы к приложению и обратно.

1 .3 Шаблоны проектирования для мобильных устройств

Шаблоны проектирования для мобильных устройств. Шаблоны проектирования в объектно-ориентированном программировании (ООП). В процессе написания кода web-разработчики сталкиваются с архитектурным шаблоном проектирования MVC. Однако при разработке под мобильные системы используется другой шаблон проектирования - MVP. Рассмотрим оба шаблона.

MVC

Для разработки web-приложений чаще всего используется архитектурный шаблон проектирования MVC. Ключевым понятием в MVC является маршрутизатор, куда попадает запрос. Задача маршрутизатора которого перенаправить запрос в контроллер, или экшн контрллера. Контроллер может обращаться к модели за данными, и должен вернуть представление (или перенаправление на другой маршут, который, в свою очередь, также заканчивается либо элементом представления либо перенаправлением).

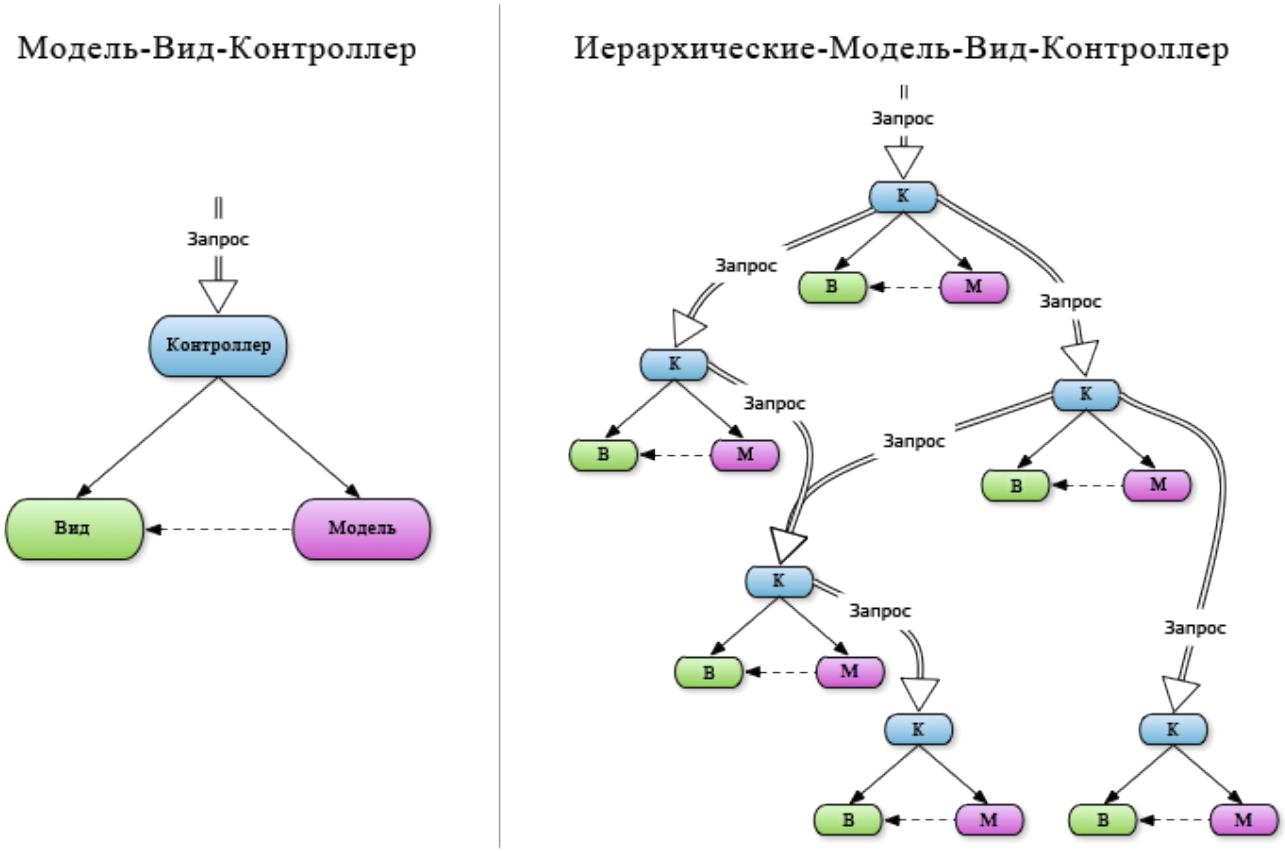
Модель (англ. Model) - модели данных, которые многие и без того используют без фреймворков. Фактически, обычные классы для работы с разными данными.

Представление (англ. View) - представления, или вид, в котором отображаются данные.

Контроллер (англ. Controller) – основной вызываемый класс, содержащий базовую логику приложения.

По концепции MVC, когда пользователь делает запрос, запрос сперва попадает в контроллер

(Controller). Затем в конроллере может происходить вызов модели (Model) и затем передача данных в шаблон представления (View).



MVP

MVP — шаблон проектирования пользовательского интерфейса, который был разработан для облегчения автоматического модульного тестирования и улучшения разделения ответственности в презентационной логике (отделения логики от отображения):

Модель — данные;
Вид — реализует *отображение* данных (из Модели), обращается к Presenter за обновлениями, перенаправляет события от пользователя в Presenter;
Представитель (англ. Presenter) — реализует взаимодействие между Моделью и Видом и содержит в себе всю бизнес-логику; при необходимости получает данные из хранилища и преобразует для отображения во View.

Обычно экземпляр Вида (Представление) создаёт экземпляр Представителя, передавая ему ссылку на себя. При этом Представитель работает с Видом в абстрактном виде, через его интерфейс. Когда вызывается событие Представления, оно вызывает конкретный метод Представителя, не имеющего ни параметров, ни возвращаемого значения. Представитель получает необходимые для работы метода данные о состоянии пользовательского интерфейса через интерфейс Вида и через него же передаёт в Вид данные из Модели и другие результаты своей работы.

2 Клиентское программирование

Рассматриваются вопросы программирования под само устройство, под Android или под мобильное устройство. А также вопросы, связанные с адаптивностью web-приложения.

2 .1 Мобильные технологии

Мобильные технологии. Мобильный клиент и web-сервер.

У мобильных пользователей мало времени, небольшой объем внимания, маленький экран, и множество отвлекающих факторов. Принимая во внимание эти ограничения, мы можем понять, чем отличаются мобильные приложения от сайтов.

Большинство мобильных систем, как и серверов работают на ядре Unix

Особенности Unix, отличающие данное семейство от других ОС, приведены ниже.

Файловая система древовидная, чувствительная к регистру символов в именах, очень слабые ограничения на длину имён и пути.
Нет поддержки структурированных файлов ядром ОС, на уровне системных вызовов файл есть поток байтов.
Командная строка находится в адресном пространстве запускаемого процесса, а не извлекается системным вызовом из процесса интерпретатора команд (как это происходит, например, в RSX-11).
Понятие «переменных окружения».
Запуск процессов вызовом fork(), то есть возможность клонирования текущего процесса со всем состоянием.
Понятия stdin/stdout/stderr.
Ввод-вывод только через дескрипторы файлов.
Традиционно крайне слабая поддержка асинхронного ввода-вывода, по сравнению с VMS и Windows NT.
Интерпретатор команд есть обыкновенное приложение, общающееся с ядром обыкновенными системными вызовами (в RSX-11 и VMS интерпретатор команд выполнялся как специальное приложение, специальным образом размещённое в памяти, пользующееся специальными системными вызовами, поддерживались также системные вызовы, дающие возможность приложению обращаться к своему родительскому интерпретатору команд).
Команда командной строки есть не более чем имя файла программы, не требуется специальная регистрация и специальная разработка программ как команд (что являлось обычной практикой в RSX-11, RT-11).
Не принят подход с программой, задающей пользователю вопросы о режимах своей работы, вместо этого используются параметры командной строки (в VMS, RSX-11, RT-11 программы работали также с командной строкой, но при её отсутствии выдавали запрос на ввод параметров).
Пространство имён устройств на диске в каталоге /dev, поддающееся управлению администратором, в отличие от подхода Windows, где это пространство имён размещается в памяти ядра, и администрирование этого пространства (например, задание прав доступа) крайне затруднено из-за отсутствия его постоянного хранения на дисках (строится каждый раз при загрузке).
Широкое использование текстовых файлов для хранения настроек, в отличие от двоичной

базы данных настроек, как, например, в Windows.
Широкое использование утилит обработки текста для выполнения повседневных задач под управлением скриптов.
«Раскрутка» ОС после загрузки ядра путём исполнения скриптов стандартным интерпретатором команд.
Широкое использование именованных каналов (pipe).
Все процессы, кроме init, равны между собой, не бывает «специальных процессов».
Адресное пространство делится на глобальное для всех процессов ядро и на локальную для процесса части, нет «групповой» части адресного пространства, как в VMS и Windows NT, как и возможности загрузки туда кода и его исполнения там.
Использование двух уровней привилегий процессора вместо четырёх в VMS.
Отказ от использования оверлеев в пользу деления программы на несколько программ поменьше, общающихся через именованные каналы или временные файлы.
Отсутствие APC и аналогов, то есть произвольных (а не жёстко перечисленных в стандартном множестве) сигналов, не доставляемых до явного пожелания процесса их получить (Windows, VMS).
Концепция сигнала уникальна для Unix, и крайне сложна в переносе на другие ОС, такие как Windows.

Linux-Dictionary

2 .2 Инструментарий

Установка необходимого инструментария. Программирование под браузер, и программирование под мобильное приложение. Интегрированная среда разработки. Android Studio. Панель управления Android. Инструменты для дизайна мобильных приложений. Proto. Платформа для прототипирования мобильных приложений. Создание интерактивных прототипов и симуляция основных пользовательских действий с поддержкой основных браузеров. Инструмент для создания мокапов на HTML5 с простым интерфейсом и набором форм, кнопок, полей, контейнеров и основных элементов интерфейса. Инструмент быстрого создания прототипов на основе простой разметки с нуля. Альтернативные инструменты дизайна мобильных приложений.

Инструментарий, используемый на стороне сервера, и инструментарий используемый на стороне клиента.

IDE:NetBeans, PHPShtorm, WebShtorm, – или любая другая интегрированная среда разработки.
AndroidStudio – или любая другая интегрированная среда разработки под выбранную мобильную технологию.
Nodepad++ - Блокнот. (ссылка для скачивания)
Git – система контроля версий (Ссылка для скачивания)
Composer – локальный менеджер зависимостей.
Firefox – Браузер.
Firebug – Отладчик кода на стороне клиента
FireFTP – или любой другой FTP-клиент.

2 .3 Эмулятор

Установка эмулятора. Виды эмуляторов. Инструменты для тестирования приложений мобильных операционных систем: Google Android Emulator, Android SDK Emulator, MobiOne Developer, TestiPhone, iPhoney, BlackBerry Simulator, Genymotion Android Emulator .

Бесплатные эмуляторы для Android

Пожалуй, самый известный эмулятор Android, особенно популярный среди геймеров. Хотя вы можете работать с самыми разными приложениями в BlueStacks, программу создали с прицелом на игры. Она успешно запускает даже самые тяжёлые из них, если у вас достаточно мощный компьютер. С другой стороны, сам BlueStacks занимает больше места и дольше загружается, чем большинство других эмуляторов.

Бесплатная версия отображает спонсорские рекомендации игр. При желании вы можете подписаться на премиум?вариант BlueStacks за 3,33 доллара в месяц, чтобы отключить рекламу и получить доступ к техподдержке.

Перед началом установки настоятельно рекомендуется закрыть все сторонние программы и максимально разгрузить оперативную память компьютера.

1. Загрузите эмулятор Bluestacks с официального сайта. Для загрузки будет доступна конкретная версия под вашу ОС (Windows или Mac OS X).

2. После окончания загрузки, запустите файл установки. Согласитесь с условиями лицензионного соглашения и выберите желаемую директорию для установки эмулятора. Нажмите «Далее».

3. С целью обеспечения стабильности работы Bluestacks, рекомендуется оставить включенными два пункта: «Доступ к магазину приложений» и «Коммуникации приложения». Нажмите «Установить». На экране отобразится окно с прогрессом. Дождитесь окончания процесса. Во время установки эмулятор Bluestacks автоматически откроется в полноэкранном режиме и загрузит необходимые компоненты. Убедитесь, что во время выполнения этого этапа стабильно работает интернет-соединение.

4. Как только установка будет завершена, отобразится меню с популярными приложениями и чартами. Среда Android уже запущена. Воспользовавшись поиском, найдите интересующее вас приложение. После щелчка по иконке приложения, Bluestacks предложит пройти авторизацию в магазине Google Play.

Укажите данные учетной записи, которую вы используете во всех сервисах Google, предварительно выбрав пункт «Существующий». Авторизовавшись, в соответствующем меню снимите галочки напротив пунктов «Резервное копирование и восстановление» и «Я хочу получать новостную рассылку». Это значительно ускорит процесс настройки эмулятора. Нажмите «Далее» (кнопка-треугольник с направлением вправо).

5. Для настройки аккаунта, Bluestacks достаточно подтвердить используемую в магазине Google Play учетную запись. Нажмите «Продолжить».

6. Включение «Синхронизации приложений» осуществляется через повторный ввод логина и пароля от учетной записи Google в предложенном веб-интерфейсе.

Это заключительный этап настройки, по завершению которого будет открыта страница установки выбранного ранее приложения в магазине Google Play. Нажмите «Установить».

Через несколько секунд (зависит от размера файла и скорости соединения) приложение будет готово к запуску. Установленные приложения попадают на «Домашний» экран, перейти к которому можно с помощью аппаратных клавиш, расположенных в левом нижнем углу. Они полностью повторяют функциональность таковых на смартфонах и планшетах с операционной системой Android: «Назад», «Домой» и «Диспетчер приложений».

На этом настройку BlueStacks можно считать завершенной. После того как вы настроили эмулятор, для установки приложений из загруженных apk-файлов достаточно кликнуть по ним два раза. Система автоматически будет ассоциировать этот формат с эмулятором BlueStacks, а его открытие приведёт к немедленной установке приложения. Оно появится на экране лончера эмулятора сразу после завершения этого процесса.

Больше информации об эмуляторе и его возможностях вы можете найти в соответствующей теме на нашем форуме. Рассмотренный вариант является наиболее простым с точки зрения установки, но, в то же время, имеет несколько программных ограничений, связанных с потребностью в мощном компьютере при запуске «тяжелых» игр. Другие варианты установки Android на компьютер мы рассмотрим в следующих инструкциях.

2 .4 Ресурсы

Мобильные ресурсы

Есть два типа мобильных ресурсов:

1. Приложения
2. Сайты

Оба типа доступны на смартфонах и планшетах. Подробное описание форматов приводится в разделе о технических характеристиках

Ресурсы в мобильных приложениях

К основным методам таргетинга на мобильные приложения с пользовательским интерфейсом относятся:

> Таргетинг на отдельные мобильные приложения. Применяйте Планировщик кампаний в КМС и поиск по названию приложения.
> Таргетинг на устройства (на уровне кампании).
> Тематический таргетинг. Хотя тематический таргетинг технически осуществим, иногда у системы недостаточно данных для правильной классификации мобильных приложений.

Как правило, мобильные приложения не поддерживают файлы cookie, поэтому механизмы

таргетинга на их основе, используемые для ремаркетинга, в данном случае неэффективны.

Однако поскольку Ad Exchange передает идентификаторы IDFA и AdID в запросы ставок в режиме реального времени, ремаркетинг можно проводить путем размещения списка пользователей, содержащего IDFA и AdID, на своем сервере. При получении запроса ставки понадобится проверять, присутствует ли в списке идентификатор, указанный в запросе.

Подробнее о таргетинге на мобильные приложения с помощью IDFA и AdID...

Геотаргетинг

Вы можете адресовать свою рекламу пользователям мобильных устройств, осуществляющим доступ через Wi-Fi. Геотаргетинг можно настраивать как на уровне стран, так и на уровне почтовых индексов. Однако при выборе конкретных операторов мобильной связи таргетинг поддерживается только на уровне стран.

Таргетинг на все доступные ресурсы

Таргетинг на все доступные ресурсы распространяется и на ресурсы AdMob.

2 .5 Архитектура приложения

Проектирование и дизайн приложения. Выделение контроллеров, моделей и элементов представления приложения.

Архитектура приложения должна соответствовать концепции MVC. Рассмотрим основные папки приложения.

App/

Основная рабочая папка, содержащая классы приложения.

Папка app содержит папки *Console, Events, Exceptions, Http, Jobs, Listeners, Providers*.

Каталог *Console* содержит artisan-команды.

Каталог *Events* предназначен для классов событий. События могут быть использованы для того, чтобы определить, что какое-то действие произошло.

Каталог *Exceptions* содержит классы обработчики исключений.

Каталог Http содержит все фильтры, контроллеры, а также запросы.

Каталог Jobs содержит файлы, управляющие синхронностью загрузки приложения.

Каталог Listeners содержит классы прослушивателей событий. Слушатели тесно связаны с событиями. Например, событие UserRegistered должно быть обработано слушателем в

SendWelcomeEmail.

Каталог Providers содержит сервис-провайдеры приложения.

bootstrap/

Папка для конфигурационных файлов автозагрузки.

config/

Папка конфигурационных файлов.

database/

В папке database находятся папка migrations (для файлов миграций баз данных), seeds (предворительные данные таблиц базы данных) и factories (настройки для моделей).

public/

Корневая папка проекта. В этой папке находится файл index.php и .htaccess, которые загружаются первыми, а также папкимедиафайлов (css, js, изображения и др.)

resources/

Папка для шаблонов. Данная папка содержит элементы представления - *views*, файлы переводов - *lang*, и вспомогательные файлы - *assets*

routes/

Папка для маршрутов приложения, маршрутизатор.

storage/

Папка для хранения временных файлов, создаваемых фрэймворком.

Папки внутри storage должны быть доступны веб-серверу для записи. Если вы устанавливаете фреймворк на Linux или MacOS, открыть папки на запись можно командой chmod -R 777 storage.

tests/

Папка содержит файлы автоматических тестов.

vendor/

Папка содержит composer – зависимости.

2 .6 Вспомогательные библиотеки

На стороне клиента, для адаптивности используются следующие библиотеки: bootstrap и jQuery

Bootstrap

Bootstrap – это платформа с открытым исходным кодом, поддерживаемая людьми, которые действительно хорошо разбираются в CSS. Они позаботились о том, чтобы обеспечить бесперебойную работу платформы, и они охватывают несколько общих функций дизайна, которые могут вам понадобиться. Это делает проектирование намного проще и вам не нужно слишком беспокоиться о том, как выглядит ваш сайт в разных браузерах, потому что

инфраструктура уже позаботится об этом. Это экономит вам много времени.

Простота использования

Bootstrap предоставляет файлы LESS для тех, кто привык к предварительной обработке CSS, но если вы этого не знаете или не хотите использовать, вы можете иметь простые файлы CSS. Все, что вам нужно сделать, это загрузить файлы с сайта Bootstrap, распаковать их и включить в заготовки ваших файлов HTML. Это дает вам доступ ко всей структуре, и вы можете начать использовать предварительно определенные классы Bootstrap.

Высоко гибкий

Bootstrap дает разработчикам гибкость в разработке. Это CSS-структура с предопределенными классами для макета, сеткой, различными компонентами CSS и функциями Javascript. Все они включены и разработчик обладает гибкостью и свободой использования только тех классов, которые необходимы в разметке. Это делает его очень гибким, поскольку разработчики могут использовать только те объекты, которые необходимы в разметке, и отключить не нужные.

Реагирующая сетка

Это самая сильная часть структуры бутстрапа. Bootstrap предлагает систему с 12 колонками. Система сетки “реагирует”, что означает, что она настраивается в зависимости от разрешения устройства клиента. Эти сетки имеют дополнительные классы ячеек, которые были определены в синхронизации с разрешением устройства, которое они представляют. Эти ячейки имеют классы xs, sm, md и lg, каждый из которых представляет собой разрешение устройства. Все, что разработчик должен сделать, это включить эти классы, определяя видимость элемента в разметке и, следовательно, создать отзывчивый веб-сайт. Интеллектуальная сетка позволяет быстро создавать адаптивные веб-сайты.

Полный список компонентов

Bootstrap содержит все компоненты, которые вам потребуются для вашего сайта. Он включает в себя выпадающие меню, панель навигации, индикатор выполнения, оповещения, ярлыки, значки и т. Д. Он включает все компоненты, которые могут вам потребоваться, и вы можете легко включить их в свою разметку. Это экономит много времени, так как вам не придется разрабатывать их с нуля и беспокоиться о кросс-браузерности и совместимости с другими устройствами. Все, что нужно, чтобы заставить их работать – включить правильные классы в свою разметку, и все будет на месте.

Читайте также: Bootstrap - первые шаги

Использование Javascript Libraries

Bootstrap содержит более дюжины плагинов javascript, разработанных и готовых к использованию. Существуют различные популярные функции, которые были включены. К ним относятся Tabs, ScrollSpy, всплывающие подсказки, всплывающие окна, оповещения и многое другое. Использование и настройка их не сложна.

Частые обновления

Bootstrap выпускает больше обновлений, чем любая другая инфраструктура. Команда разработчиков bootstrap начинает работу над решением, как только сталкивается с какой-либо

проблемой. С обновлениями Bootstrap, выпускаемыми довольно часто, вы можете быть уверены, что работаете с новейшими инструментами. Это также обеспечивает более широкий диапазон совместимости.

Подробная документация

Bootstrap имеет очень подробную документацию и обширное сообщество, поддерживающее ее. Даже если разработчик новичок в Bootstrap, документация обеспечивает отличную поддержку в обучении без каких-либо проблем. Документация включает примеры и демонстрации, которые помогают понять концепции и привыкнуть к Bootstrap за короткий промежуток времени. И если даже разработчик застопорился в каком либо вопросе, есть огромное сообщество и множество форумов, которые помогут решить все вопросы.

Последовательность

Bootstrap была разработана с идеей дать разработчикам централизованный код разработки. Bootstrap предоставляет вам конечный результат, который является однородным на всех платформах. Вам не нужно больше беспокоиться о проблемах совместимости с Internet Explorer, Google Chrome или Firefox и это дает вам полную свободу действий.

jQuery

JQuery - это кроссплатформенная JavaScript-библиотека, предназначенная для упрощения клиентского сценария HTML. JQuery - самая популярная в настоящее время библиотека JavaScript, которая устанавливается на 65% из 10 миллионов самых посещаемых сайтов в Интернете. JQuery - это бесплатное программное обеспечение с открытым исходным кодом, лицензированное по лицензии MIT.

jQuery предлагает:

1. Обход документа HTML

Когда браузер отображает веб-страницу, это визуальное представление того, что известно как DOM (или объектная модель документа). Эта модель может быть концептуально смоделирована как структура данных дерева, где есть определенные узлы с корнями и листьями.

2. Манипуляция документами HTML

Когда дело доходит до фактического манипулирования DOM, у jQuery есть *много функций*, которые позволяют нам изменять то, что видят наши посетители.

Некоторые из этих функций просты, например, позволяют нам показывать (`show`) или скрывать (`hide`) элементы, которые не отображаются при загрузке страницы. Другие функции позволяют нам создавать новые элементы и добавлять (`append`) их к существующему элементу или добавлять их перед (`prepend`) всем списком.

3. Обработка событий

Если вы новичок в JavaScript, то одна вещь, которая имеет ключевое значение для понимания того, как она работает со страницей, отображаемой в веб-браузере, состоит в том, что она реагирует на различные события.

То есть, когда пользователь нажимает на элемент, нажимает клавишу или щелкает мышью, тогда браузер выдает сигнал, соответствующий произошедшему событию. Это позволяет нам использовать преимущества взаимодействия пользователя с браузером.

В частности, каждый раз, когда пользователь *что-то* делает на странице, мы можем реагировать с помощью пользовательского события. Проблема заключается в том, что не каждый браузер реализует события одинаково (поэтому существует потребность в спецификации, но это контент для другой статьи).

К счастью, jQuery делает это намного проще, определяя последовательное имя для всех событий, чтобы мы могли использовать одно и то же имя для события, на которое мы пытаемся ответить, и оно будет работать во всех основных браузерах.

4. Анимация

Когда jQuery впервые вышел, Flash все еще был относительно популярен, и общие анимации в Интернете не были полностью мертвыми.

Когда мы говорим об анимации в контексте jQuery, мы не говорим о том же типе эффектов или поведения, которые мы привыкли видеть со старой технологией. Вместо этого мы говорим об эффектах, которые дают пользователям обратную связь, *что-то* случилось на экране. Кроме того, он менее инвазивный и может придать правильное чувство стиля странице или приложению при правильном использовании (хотя конечно чем-то можно и злоупотреблять).

Вы можете просмотреть API всех эффектов на этой странице, но стоит заметить, что эффекты jQuery могут варьироваться в любом месте от обработки простого выцветания и выхолащивания элементов или скользящих элементов до более сложного, например манипулирования очередью зарегистрированных эффектов для запуска потом на определенном элементе.

Конечно, в последнем случае предполагается, что у вас есть небольшой опыт работы с API-интерфейсами эффектов, но это естественно, если у вас есть достаточно времени для работы с библиотекой и документацией.

5. Ajax

Если вы не знакомы с Ajax, то, по сути, веб-страница может сделать вызов на сервер, обработать ответ и обновить часть страницы, не обновляя всю страницу. Хотя технология существует уже довольно давно, это все еще что-то действительно крутое и может обеспечить некоторые действительно аккуратные функциональные возможности в контексте страницы или веб-приложения при правильном и эффективном использовании.

Хотя поддержка Ajax не так плоха, как это было пять или десять лет назад, реализация API в разных браузерах может немного меняться. Это означает, что нам нужно написать код Ajax специально для браузера, который корпорация Майкрософт предоставляет, который Google предоставляет, который предоставляет Apple, предоставляемый Chrome, и так далее.

По крайней мере, так обстоит дело без jQuery. Благодаря поддержке Ajax, мы можем использовать Ajax несколькими способами, не сталкиваясь с несогласованностью кросс-браузера. На самом деле тривиально легко обрабатывать запросы `GET` и `POST`, а также возможность совершать гораздо более сложные вызовы с помощью метода `$.ajax`.

Как только вы привыкли к тому, что API доступен в ядре вашего приложения или в вашем распоряжении, трудно представить, что вы не работаете с ним (или с чем-то подобным).

Слово о расширяемости

Одной из особенностей, которую предлагает множество серверных фреймворков и библиотек, является возможность создания расширений для базовой кодовой базы. Современные клиентские библиотеки и платформы позволяют это, и jQuery не является исключением.

3 Серверное программирование

Основной серверный язык программирования - PHP

3 .1 Локальный сервер OpenServer

Установка, настройка, обновление OpenServer.

Архитектура расположения каталогов программного комплекса. Типы данных: динамических данные пользователя (настройки, временные файлы, логи т.д.) и статичные данные (модули, программы, служебные файлы). Изменяемые папки domains и userdata. Неизменная папка modules. Установка обновлений.

Основные компоненты:

OSPanel **5.3.5**
Apache **2.2.31** / **2.4.38** / **2.4.41**
Bind **9.14.5**
ConEmu **19.07.14**
FTP FileZilla **0.9.60**
Ghostscript **9.27**
Git **2.23.0**
HeidiSQL **10.2.0.5599**
Nginx **1.17.3**
NNCron Lite **1.17**
Sendmail **32**
Wget **1.20.3**

Системы управления базами данных:

MariaDB **5.5.63** / **10.0.38** / **10.1.38** / **10.2.22** / **10.3.13**
Memcached **1.2.6** / **1.4.5**
MongoDB **2.4.14** / **2.6.12** / **3.0.15** / **3.2.22** / **3.4.19** / **3.6.11** / **4.0.6** / **4.2.0**
MySQL **5.1.73** / **5.5.62** / **5.6.43** / **5.7.25** / **8.0.15**
PostgreSQL **9.2.24** / **9.3.25** / **9.4.21** / **9.5.16** / **9.6.12** / **10.7** / **11.2**
Redis **2.8.2402** / **3.0.504** / **3.2.100**

PHP модули:

PHP **5.2.17** + расширения
PHP **5.3.29** + расширения
PHP **5.4.45** + расширения
PHP **5.5.38** + расширения
PHP **5.6.40** + расширения
PHP **7.0.33** + расширения
PHP **7.1.32** + расширения
PHP **7.2.22** + расширения
PHP **7.3.9** + расширения

PHP приложения:

Adminer **4.7.3**
PHPMemcachedAdmin **1.3**
PHPMyAdmin **4.9.0.1**
PHPPgAdmin **7**
PHPRedisAdmin **1.11.4**

Компоненты сборки представлены в 32-битной и 64-битной (частично) версиях.

Системные требования

Поддерживаемые версии Windows (32-бит и 64-бит): Windows 7 SP1 и все более новые версии;
Минимальные аппаратные требования: 500 МБ свободной RAM и 3 ГБ свободного места на HDD;
Требуется наличие Microsoft Visual C++ 2005-2008-2010-2012-2013-2017 Redistributable Package;

Возможности управляющей программы

Незаметная работа в трее Windows;
Быстрые старт и остановка;
Автостарт сервера при запуске программы;
Несколько режимов управления доменами;
Монтирование виртуального диска;
Поддержка управления через командную строку;
Поддержка профилей настроек;
Удобный просмотр логов всех компонентов;
Переключение HTTP, MySQL и PHP модулей;

Подробная и понятная документация;
Доступ к доменам в один клик;
Быстрый доступ к шаблонам конфигурации;
Мультиязычный интерфейс;
Автозапуск программ по списку;

Особенности комплекса

Не требует установки (портативность);
Возможность работы с USB накопителя;
Одновременная работа с Denwer, Xampp и т.д.;
Работа на локальном/сетевом/внешнем IP адресе;
Поддержка SSL без всякой дополн. настройки;
Создание домена путем создания обычной папки;
Поддержка кириллических доменов;
Поддержка алиасов (доменных указателей);
Защита сервера от внешнего доступа;
Punycode конвертер доменных имён;
Пакет из более 40 портативных программ;
Планировщик заданий (cron);
Создание локального поддомена без потери видимости основного домена в сети интернет;

3 .2 База данных MySQL

База данных MySQL. Подключение к MySQL.

PHPMyAdmin подключение к MySQL. Автоинкремент числовых значений. Типы данных. Кодировка. Взаимодействие с PHP.

На занятиях рассматриваются основные операции MySQL:

создавать базу данных
создавать таблицу
записывать в таблицу данные
извлекать данные из таблицы различными способами
редактировать данные из таблицы
удалять данные
работать с несколькими таблицами сразу

3 .3 Язык программирования PHP. Фрэймворк Laravel.

Основы PHP. Строки и функции PHP. Классы и объекты PHP. Переход на PHP7. Синтаксис. Типы данных. Переменные и константы. Выражения. Операторы PHP Конструкции PHP

ООП и фрэймворки. Фрэймворк PHP Laravel.

Маршрутизация. Создание контроллеров. Подключение базы данных MySQL. Модели. Миграции (migration) и первоначальная загрузка данных (seeds). ServiceProvider – вспомогательные классы библиотек ядра Laravel, внешних и внутренных (директория\App).

Middleware, как промежуточное программное обеспечение.

Требования к установке

Официальный сайт laravel – http://laravel.com

У Laravel всего несколько требований к вашему серверу:

PHP >= 7.2
Mcrypt PHP Extension
OpenSSL PHP Extension
MbstringPHPExtension

В некоторых операционных системах может понадобиться ручная установка PHP JSON extension.

Перед установкой необходимо обновить composer

```
Composer selfupdate
```

Далее необходимо перейти в корневую серверную папку

```
cd domains
```

После чего можно установить Laravel

```
Composer create-project laravel/laravel -—prefer-dist
```

Laravel Установлен.

3 .4 Тестирование

Laravel предоставляет очень удобный API для создания HTTP-запросов к вашему приложению, проверки вывода, и даже заполнения форм.

При выполнении тестов Laravel автоматически задаст настройки среды `testing`. Также при тестировании Laravel автоматически настроит сессии и кэш на драйвер `array`, а значит данные сессий и кэша не сохранятся при тестировании.

При необходимости вы можете определить другие значения настроек для тестовой среды. Переменные среды `testing` можно настроить в файле `phpunit.xml`, но перед запуском тестов не забудьте очистить кэш настроек с помощью Artisan-команды `config:clear`!

Для создания теста используйте Artisan-команду `make:test`:

```
php artisan make:test UserTest
```

Эта команда поместит новый класс `UserTest` в папку `tests`. Далее вы можете объявлять методы тестов как вы обычно объявляете их для PHPUnit. Для запуска тестов просто выполните команду `phpunit` в терминале.

4 Командная разработка

Основы разработки с использованием систем контроля версий.

4 .1 Система контроля версий GIT

GIT – система контроля версий. Оснавная задача систем контроля версий – создавать локальные репозитории и хранить информацию о файлах проекта в трех состояниях: новый файл, измененный, зафиксированный.

Преимущества Git перед другими системами контроля версий:

Высокая производительность;

Развитые средства интеграции с другими системами контроля версий;

Продуманная система команд, позволяющая удобно встраивать git в скрипты;

Качественный веб-интерфейс «из коробочки»;

Репозитории git могут распространяться и обновляться общесистемными файловыми утилитами архивации и обновления.

Основные команды git

git – проверяем правильность установки git.
git init – инициализация пустого репозитория
git config --global user.name "Alex" – имя пользователя для git
git config --global user.email "mikhalkevich@ya.ru" – email пользователя для git
git status – текущий статус репозитория
git add * - добавить все файлы текущей папки и подпапок в область видимости git.
git commit –m “first commit” – фиксация изменений
git remote add project https://github.com/User/project - создание переменной project
git push project master – заливаем файлы на удаленный репозиторий ветку master
git pull project master – скачиваем ветку master
git clone https://github.com/User/project - клонирование удаленного проекта на локальный компьютер.
git branch Alex – создание ветки Alex
git checkout Alex – переключение на ветку Alex
git merge Alex – заливаем изменения, которые внес разработчик под веткой Alex в текущую ветку (master)

4 .2 Удаленный репозиторий http://github.com

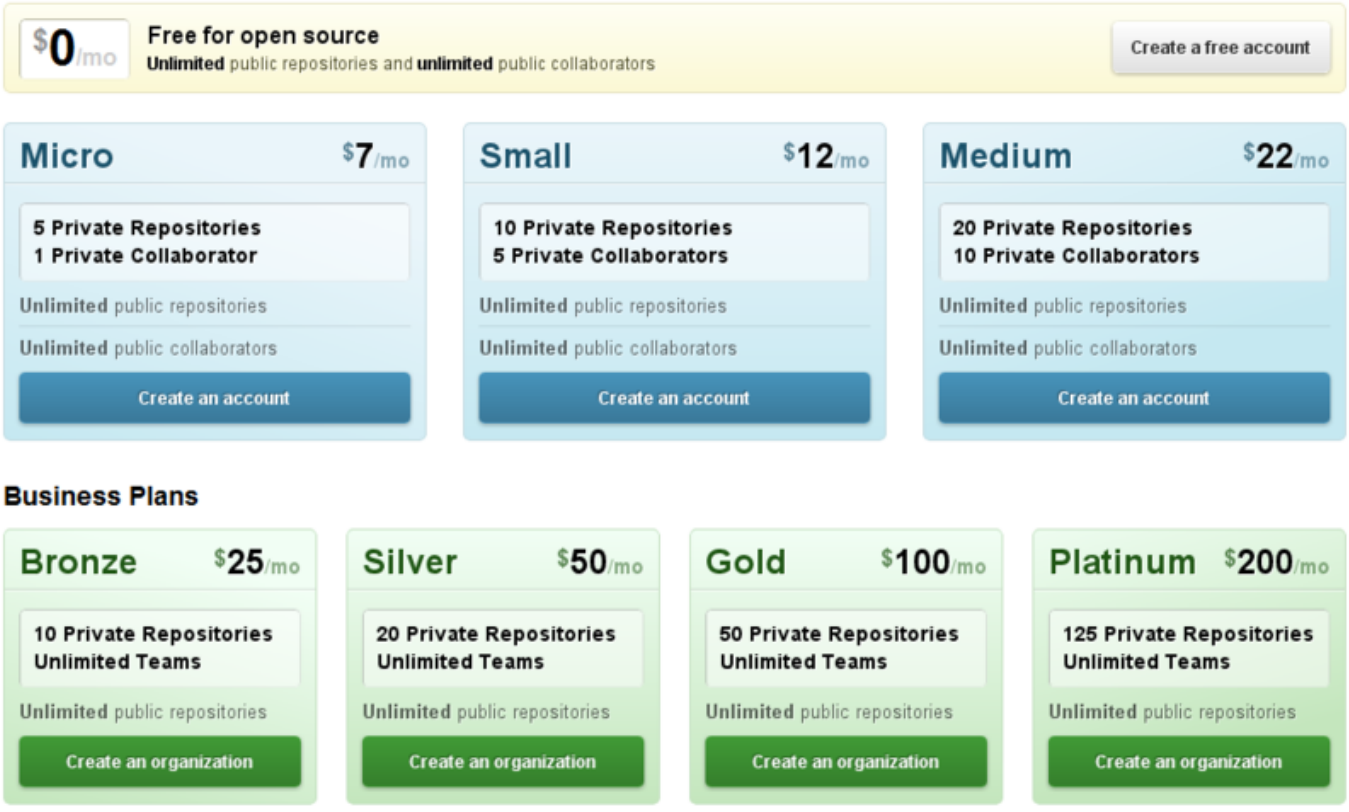
Использование удаленного репозитория для git

github
SOCIAL CODING

Распределенные системы контроля версий (DVCS) постепенно замещают собой централизованные. Если вы еще не используете одну из них — самое время попробовать.

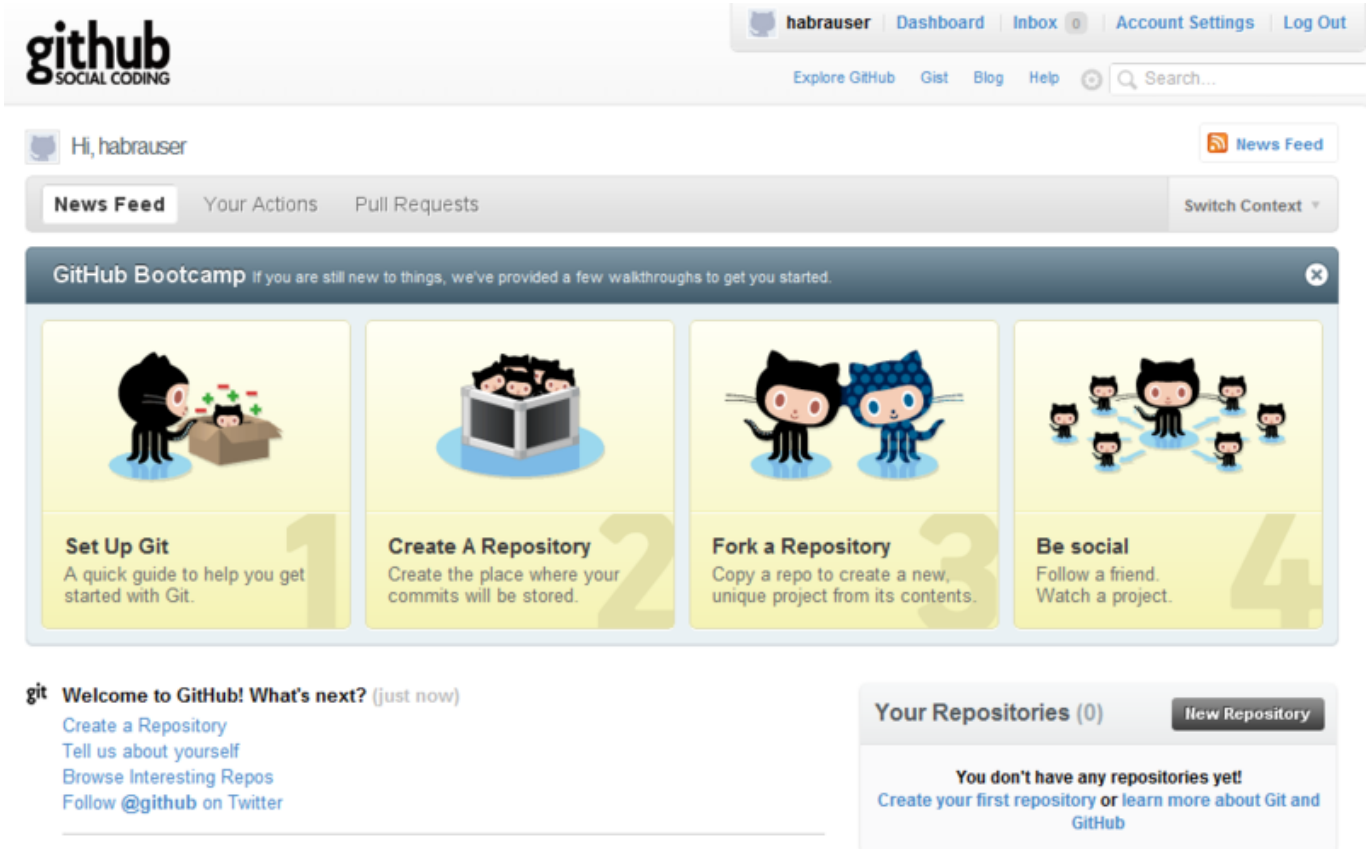
github.com позиционируется как веб-сервис хостинга проектов с использованием системы контроля версий git, а также как социальная сеть для разработчиков. Пользователи могут создавать неограниченное число репозиториев, для каждого из которых предоставляется wiki, система issue tracking-a, есть возможность проводить code review и многое другое. GitHub на данный момент является самым популярным сервисом такого рода, обогнав Sourceforge и Google Code.

Для open-souce проектов использование сайта бесплатно. При необходимости иметь приватные репозитории, есть возможность перейти на платный тарифный план:



Для регистрации по ссылке github.com/signup/free и вводим свои данные.

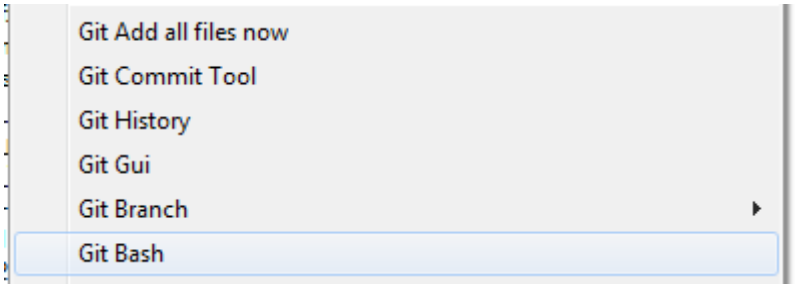
После регистрации мы попадаем в кабинет пользователя:



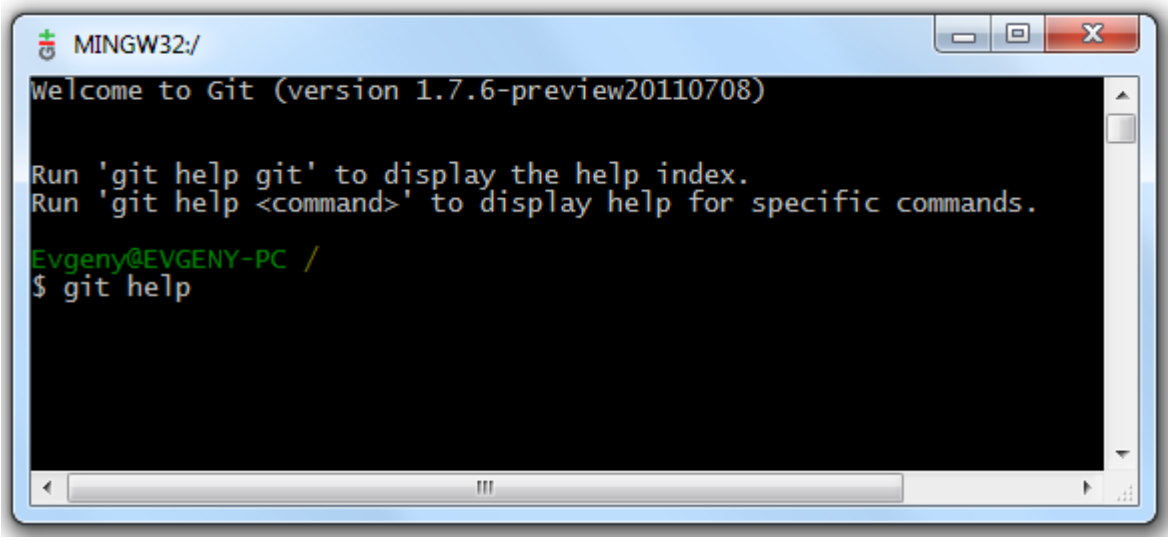
Сейчас у нас нет ни одного репозитория, и мы можем либо создать новый репозиторий, либо ответвиться (fork) от уже существующего чужого репозитория и вести собственную ветку разработки. Затем, при желании, свои изменения можно предложить автору исходного репозитория (Pull request).

Но для начала установим git и настроим его для работы с сайтом.

Для Windows, качаем и устанавливаем msysgit. Это консольная версия git для Windows (далее расказ будет вестись на примере этой ОС).
Инструкция для MacOS X (eng)
Инструкция для Linux (eng)



или через Git Bash.lnk в папке с установленой программой:



Прописываем в консоли свои данные и настройки переносов строк:

```
git config --global user.name "ваше имя"
git config --global user.email "ваша почта"
git config --global core.autocrlf true
git config --global core.safecrlf true
```

Для тех, кто предпочитает gui — для Windows существует несколько таких инструментов для работы с git. Два основных — это SmartGit (кроссплатформенный) и TortoiseGit.
В кабинете пользователя создадим новый репозиторий, жмем New Repository (https://github.com/repositories/new), вводим данные и жмем Create Repository.

GitHub позволяет работать с репозиториями тремя способами: SSH, HTTP и Git Read-Only, соответственно предоставляя ссылки трех видов для нашего репозитория:

Для того, чтобы просто забрать репозиторий на локальную машину, достаточно внутреннего протокола git (третья ссылка). Это наиболее быстрый и эффективный способ, который обеспечивает анонимный доступ только для чтения.

Репозиторий автора http://github.com/mikhalkevich

5 Презентация

Презентация проекта

Заключение

Список использованных источников

Приложения

1. [picture] 5e298a3a7ac1d_7-Ways-to-Structure-your-Presentation-To-Keep-your-Audience-Wanting-More-Infographic-2.jpg
2. [picture] 5e298a5eb2240_gdpr_right_to_be_forgotten.jpg
3. [picture] 5e341e168c896_css3.doc
4. [picture] 5e341e27e2260_jQuery.doc