

1

Настройка среды разработки

Инструментов и ресурсов для разработки клиентской части существует немало, причем все время появляются новые. Выбор наилучших из них — непростая задача для разработчика с любым уровнем опытности. По мере работы с проектами в данной книге мы научим вас пользоваться некоторыми из наших любимых инструментов.

Для начала понадобятся три основных инструмента: браузер, текстовый редактор и хорошая справочная документация для используемых при разработке клиентской части технологий. Пригодятся также несколько дополнительных инструментов, которые не являются предметами первой необходимости, но все же сделают разработку более приятной и удобной.

При работе с книгой мы рекомендуем вам использовать то же программное обеспечение, что и мы, чтобы извлечь максимум пользы из наших инструкций и снимков экранов. Данная глава освещает процесс установки и настройки браузера Google Chrome, текстового редактора Atom, платформы Node.js и нескольких плагинов и дополнительных инструментов. Вы также узнаете, где найти хорошую документацию, и пройдете ускоренный курс по использованию командной строки в операционной системе Windows и на компьютерах Mac. В следующей главе воспользуемся всеми этими ресурсами в нашем первом проекте.

Установка Google Chrome

На вашей машине уже должен быть установленный по умолчанию браузер, но лучше всего использовать для разработки клиентской части браузер Google Chrome. Если у вас пока еще нет свежей версии Chrome, вы можете скачать ее на сайте www.google.com/chrome/browser/desktop (рис. 1.1).

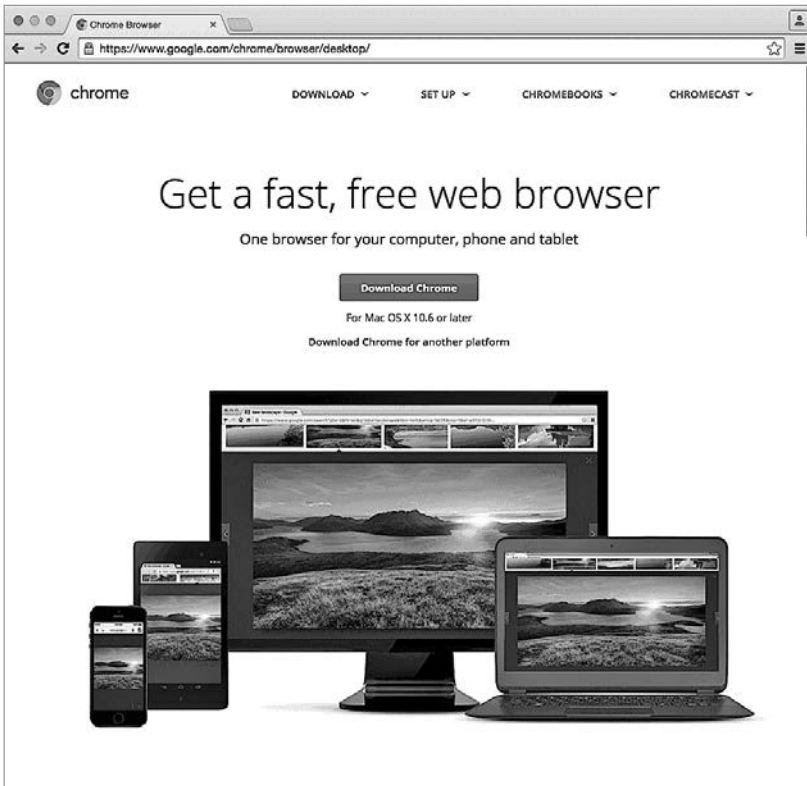


Рис. 1.1. Скачивание Google Chrome

Установка и настройка Atom

Один из лучших текстовых редакторов для разработки клиентской части — Atom от GitHub. Он предоставляет широкие возможности изменения конфигурации, для него существуют многочисленные плагины, облегчающие написание кода, к тому же скачать его можно бесплатно.

Скачать редактор Atom для Windows или Mac можно с сайта atom.io (рис. 1.2).

Следуйте инструкциям по установке для вашей платформы. После установки Atom вы, вероятно, захотите установить для него несколько плагинов.

Плагины для редактора Atom. Основное, для чего нам нужен текстовый редактор, — поиск по документации, автодополнение и линтинг кода (через минуту мы расскажем об этом подробнее). Atom предоставляет некоторые из этих возможностей по умолчанию, но установка парочки плагинов заметно улучшит положение дел.

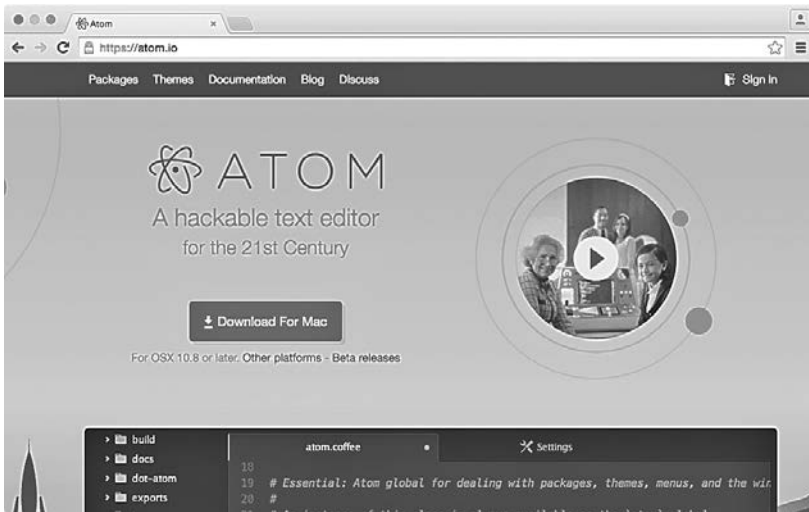


Рис. 1.2. Скачивание Atom

Откройте Atom и перейдите на экран Settings (Настройки). На Mac это можно сделать, выбрав Atom ► Preferences (Atom ► Установки) или нажав сочетание горячих клавиш Command+, (то есть клавиша Command плюс запятая). В Windows можно получить доступ через File ► Settings (Файл ► Настройки) или нажатием сочетания горячих клавиш Ctrl+.,.

В левой части экрана Settings (Настройки) нажмите + Install (Установить) (рис. 1.3).

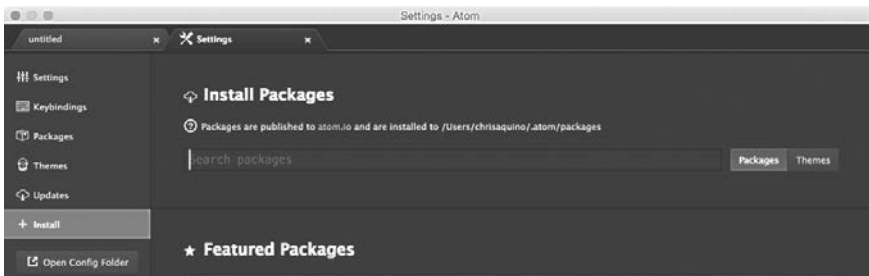


Рис. 1.3. Экран Install Packages (Установка пакетов) редактора Atom

Здесь можно выполнить поиск пакетов плагинов по названию. Начнем с поиска по запросу `emmet`.

Написание большого количества кода HTML может быть утомительным и чревато ошибками. Плагин `emmet` (рис. 1.4) позволяет писать хорошо форматированный HTML с помощью удобной сокращенной записи. Нажмите кнопку Install (Установить) для установки `emmet`.

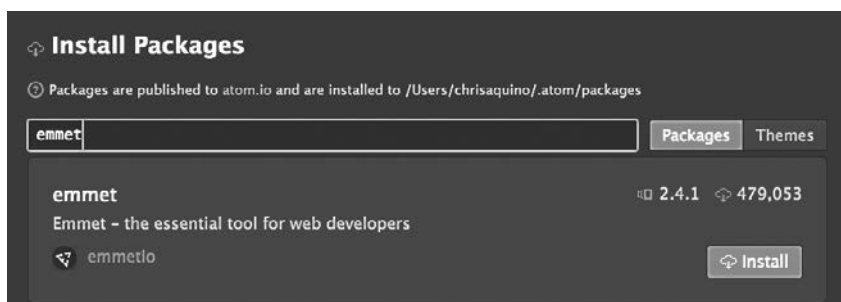


Рис. 1.4. Установка emmet

Далее выполните поиск по запросу `atom-beautify`. Плагин `atom-beautify` (рис. 1.5) помогает расставлять в коде отступы, которые повышают удобочитаемость. Нажмите кнопку `Install` (Установить) для установки плагина.

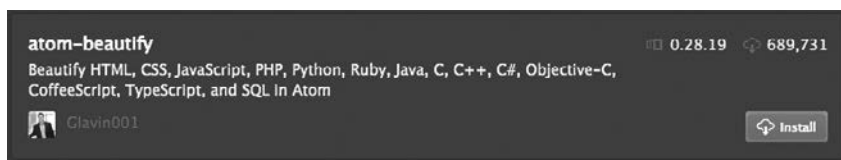


Рис. 1.5. Установка atom-beautify

Найдите и установите плагин `autocomplete-paths` (рис. 1.6). Очень часто приходится ссылаться в коде на другие файлы и каталоги проекта. Этот плагин предлагает имена файлов в меню автодополнения по мере набора текста.

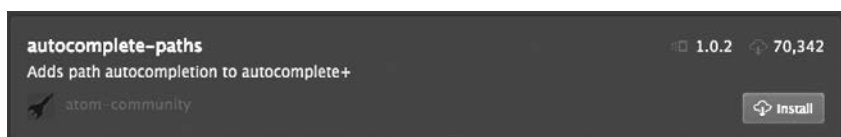


Рис. 1.6. Установка autocomplete-paths

Следующий плагин, который нужно установить, — пакет `api-docs` (рис. 1.7). Он даст возможность выполнять поиск в документации по ключевому слову. Плагин отображает документацию в отдельном окне в редакторе.

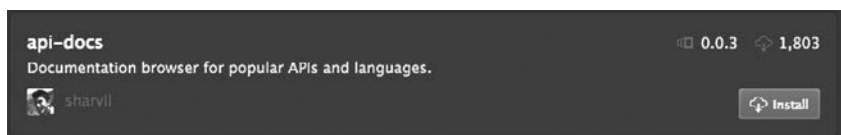


Рис. 1.7. Установка пакета api-docs

Далее найдите и установите пакет `linter` (рис. 1.8). *Линтер* — программа, проверяющая синтаксис и стиль кода. Убедитесь, что вы нашли и установили пакет, который называется просто `linter`. Это основной линтер, работающий с плагинами для конкретных языков. Он понадобится, чтобы использовать другие плагины-линтеры, указанные ниже.



Рис. 1.8. Установка пакета `linter`

Существует три дополнения для `linter`, которые пригодятся для проверки вашего кода, написанного на CSS, HTML и JavaScript. Начнем с дополнения `linter-csslint`. Оно обеспечивает синтаксическую корректность CSS и предлагает советы по написанию производительного CSS-кода.

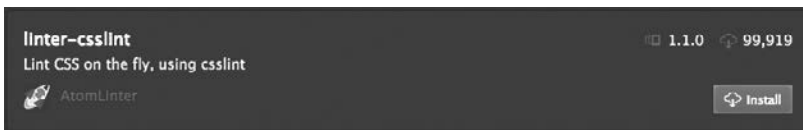


Рис. 1.9. Установка `linter-csslint`

Следующий дополнительный плагин для `linter` — `linter-htmlhint` (рис. 1.10), который подтверждает, что код HTML правильно сформирован. В случае рассогласования тегов HTML он выведет предупреждение.

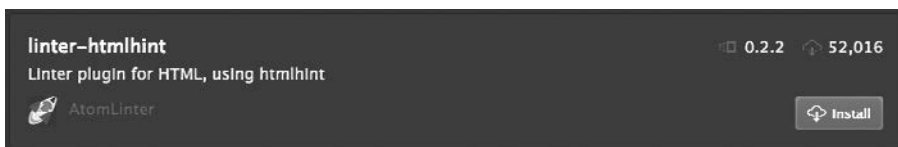


Рис. 1.10. Установка `linter-htmlhint`

Последний дополнительный плагин для `linter` — `linter-eslint` (рис. 1.11). Он проверяет синтаксис кода JavaScript и может быть настроен для проверки стиля и форматирования кода (например, количества пробелов в отступах строк или количества пустых строк до и после комментариев).

Теперь браузер Chrome и редактор Atom готовы для разработки клиентской части. Осталось организовать доступ к документации, разобраться с основами командной строки и скачать еще два инструмента.



Рис. 1.11. Установка linter-eslint

Где найти документацию и справочную информацию

Разработка клиентской части отличается от программирования для таких платформ, как iOS и Android. Помимо описания очевидных различий для технологий клиентской части, не существует официальной документации разработчика, не считая технических спецификаций. Следовательно, вам нужно будет где-то искать инструкции. Мы советуем ознакомиться с приведенными ниже ресурсами и обращаться к ним регулярно по мере работы с данной книгой и в дальнейшем при разработке клиентской части.

Сеть разработчиков Mozilla (Mozilla Developer Network, MDN) — наилучший справочник по всему, что относится к языкам HTML, CSS и JavaScript. Можно получить к ней доступ через devdocs.io (замечательный интерфейс для работы с документацией) (рис. 1.12). Он извлекает документацию из MDN для основных технологий клиентской части, причем может работать в режиме офлайн, так что вы сможете обратиться к нему даже при отсутствии подключения к Интернету.

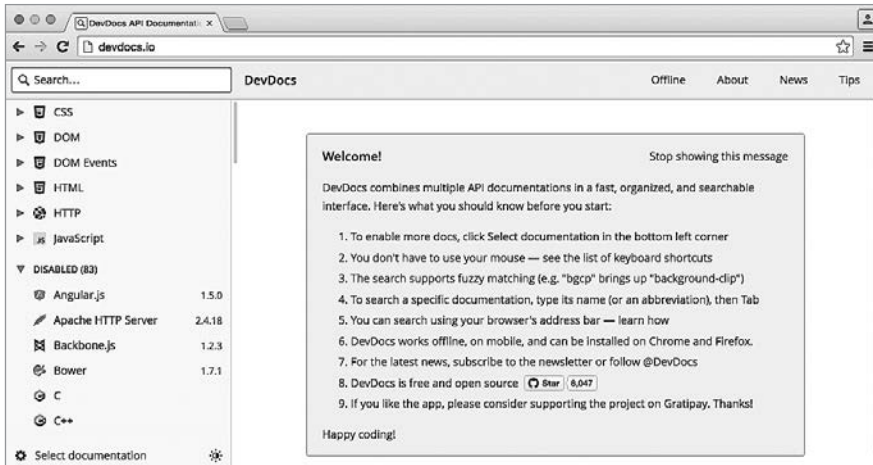


Рис. 1.12. Доступ к документации через devdocs.io

Обратите внимание, что браузер Safari пока не поддерживает механизм кэширования, используемый devdocs.io. Для доступа к нему вам понадобится другой браузер, например Chrome.

Можно также использовать сайт MDN, developer.mozilla.org/en-US (рис. 1.13), или добавить MDN в запрос поисковой системы для нахождения нужной информации.



Рис. 1.13. Сайт Mozilla Developer Network

Другой полезный сайт — stackoverflow.com (рис. 1.14). Официально он не является источником документации. Это место, где разработчики могут задавать друг другу вопросы насчет кода. Качество ответов разное, но зачастую они вполне исчерпывающие и весьма полезные. Так что, когда будете пользоваться этим ресурсом, не забывайте, что написанное там вовсе не истина в последней инстанции (из-за краудсорсинговой природы).

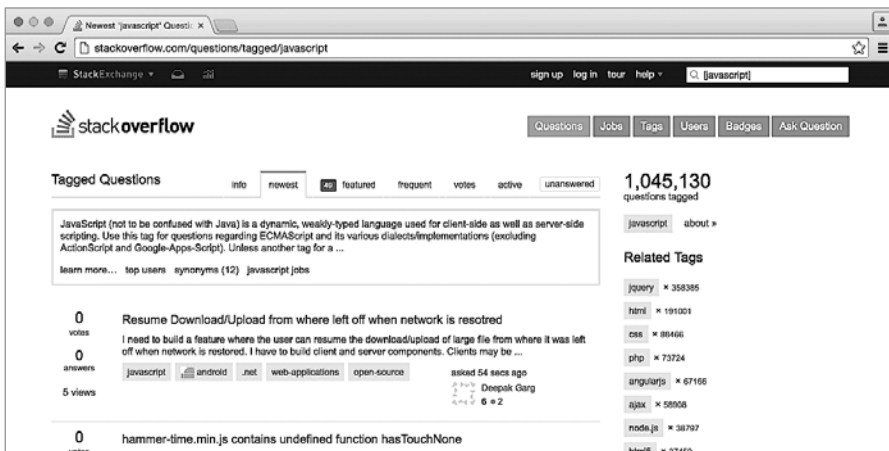


Рис. 1.14. Сайт Stack Overflow