

# 1 Hello World

Итак, приступим к освоению процесса разработки приложения с использованием React. В этой главе мы рассмотрим установку React и вы напишете свое первое веб-приложение Hello World.

## Установка

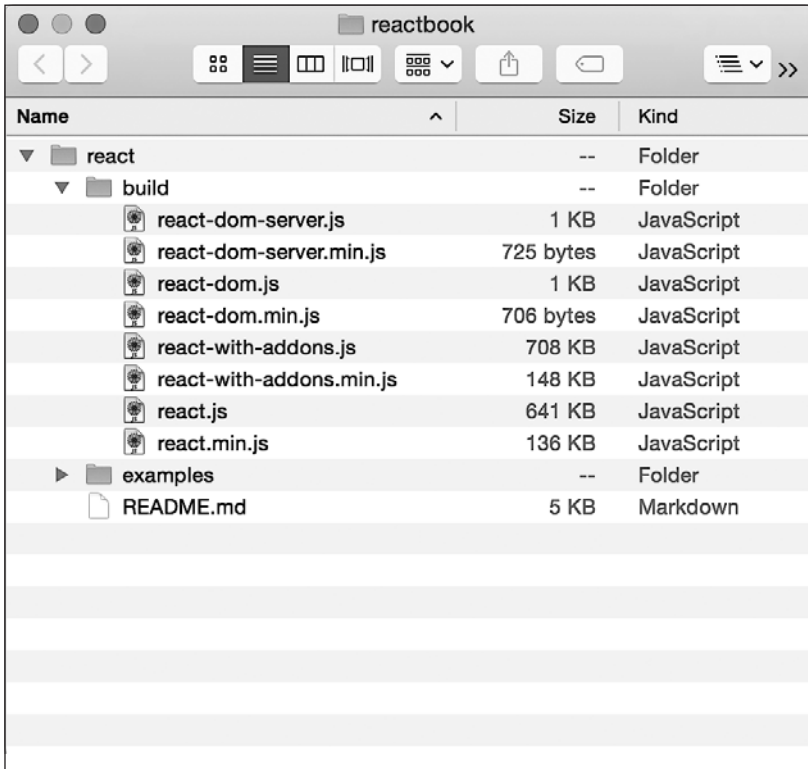
Для начала нужно получить копию библиотеки React. К счастью, сделать это проще простого.

Перейдите по адресу <http://reactjs.com> (оттуда вас перенаправят на официальную страницу GitHub), нажмите кнопку **Download** (Загрузить), потом кнопку **Download Starter Kit** (Загрузить начальный инструментарий) — и получите копию ZIP-файла. Распакуйте и скопируйте каталог, содержащийся в загрузке, в то место, где вы его легко сможете найти.

Например:

```
mkdir ~/reactbook
mv ~/Downloads/react-0.14.7/ ~/reactbook/react
```

Теперь ваш рабочий каталог (reactbook) приобретет вид, показанный на рис. 1.1<sup>1</sup>.



**Рис. 1.1.** Внешний вид вашего каталога React

<sup>1</sup> Вероятно, описанный автором процесс получения исходных файлов для формирования показанной на рисунке структуры рабочего каталога устарел и не может быть повторен. Реальнее всего получить все указанные файлы, загрузив сопровождающий книгу полный пакет инструментальных средств и примеров исходного кода с адреса <https://github.com/stoyan/reactbook/>, нажав кнопку Clone or download. — *Примеч. пер.*

Единственный файл, который вам понадобится для начала, — это `~/reactbook/react/build/react.js`. Остальные файлы будут рассмотрены чуть позже.

Следует отметить, что React не навязывает какую-либо структуру каталогов, вы можете свободно переместить файлы в другой каталог или переименовать `react.js`, как вам захочется.

## Привет, мир React

Начнем с простой страницы в вашем рабочем каталоге (`~/reactbook/01.01.hello.html`):

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello React</title>
    <meta charset="utf-8">
  </head>
  <body>
    <div id="app">
      <!-- здесь будет отображено мое приложение -->
    </div>
    <script src="react/build/react.js"></script>
    <script src="react/build/react-dom.js"></script>
    <script>
      // код моего приложения
    </script>
  </body>
</html>
```



---

Весь представленный в книге код можно найти в сопровождающем ее хранилище по адресу <https://github.com/stoyan/reactbook/>.

---

В этом файле заслуживают внимание только две особенности:

- включение библиотеки React и ее DOM-дополнения (с помощью тегов `<script src>`);
- определение места, где должно находиться ваше приложение на странице (`<div id="app">`).



---

С React-приложением всегда можно смешивать обычное HTML-наполнение, а также другие библиотеки JavaScript. На одной и той же странице также могут находиться сразу несколько React-приложений. Вам понадобится лишь место в объектной модели документов (DOM), на которое можно нацелить библиотеку React и сказать ей: «Твори здесь свое волшебство».

---

Теперь добавим код, который говорит hello. Для этого обновим содержимое файла `01.01.hello.html`, заменив строку `// код моего приложения` следующими строками:

```
ReactDOM.render(  
  React.DOM.h1(null, "Hello World!"),  
  document.getElementById("app")  
);
```

Загрузите `01.01.hello.html` в браузер — и увидите свое новое приложение в действии (рис. 1.2).

Поздравляю, вы только что создали свое первое React-приложение!

На рис. 1.2 также показано, в каком виде *сгенерированный* код демонстрируется в области инструментов разработчика браузера Chrome: можно увидеть, что содержимое контейнера `<div id="app">` заменено содержимым, созданным вашим React-приложением.

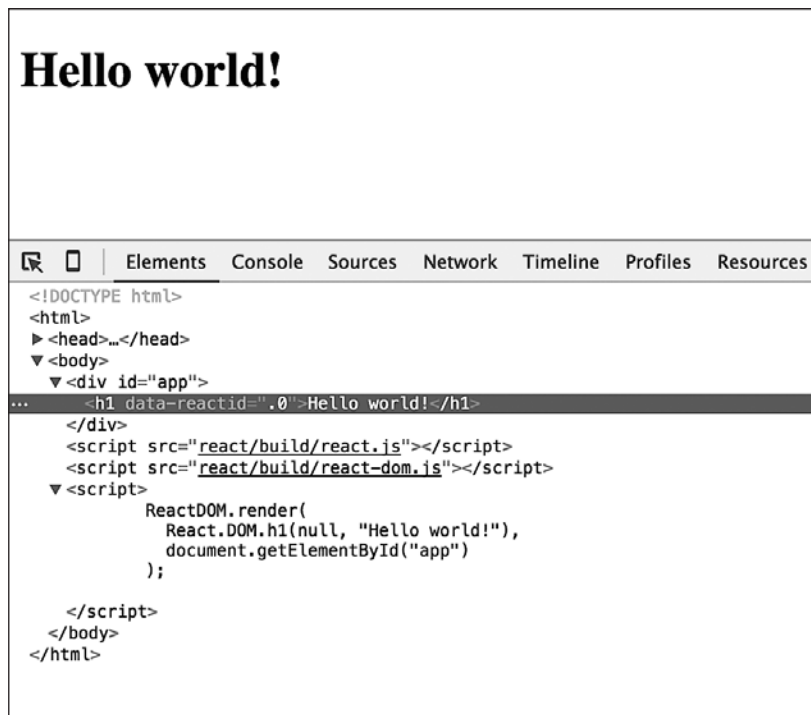


Рис. 1.2. Hello World в действии

## Так что же сейчас произошло?

В коде, заставившем работать ваше первое приложение, есть несколько любопытных мест.

В первую очередь в нем видно использование React-объекта. Через него происходит обращение ко всем доступным вам API-интерфейсам. API-интерфейс намеренно сделан минимальным, чтобы не пришлось запоминать слишком много имен методов.

Можно также посмотреть на ReactDOM-объект. В нем всего лишь горстка методов, наиболее полезным из которых является `render()`. Ранее эти методы были составной частью React-объекта, но начиная с версии 0.14 они выделены в самостоятельный объект, чтобы подчеркнуть тот факт, что отображение приложения является отдельной задачей. React-приложение можно создать для отображения в различных средах, например в HTML (в DOM-модели браузера), на холсте (canvas) или прямо в Android либо iOS.

Затем следует обратить внимание на концепцию *компонентов*. Ваш пользовательский интерфейс создается с использованием компонентов, которые сочетаются в любом (подходящем для вас) виде. В своем приложении вы, конечно же, займетесь созданием собственных привычных компонентов, но для начала React предоставляет вам надстройки над HTML DOM-элементами. Эти надстройки используются посредством `ReactDOM`-объекта. В нашем первом примере показано использование компонента `h1`. Он соответствует HTML-элементу `<h1>` и доступен по вызову `ReactDOM.h1()`.

И наконец, вы можете заметить хорошо известный DOM-доступ к контейнеру, осуществляемый с помощью метода `document.getElementById("app")`. Он используется, чтобы подсказать React, где именно на странице должно размещаться приложение. Это своеобразный мост, переброшенный между манипуляцией с известными вам DOM-объектами и React-территорией.



---

После переброски моста от DOM к React можно больше не беспокоиться о работе с DOM, поскольку React выполняет трансформацию своих компонентов в базовую платформу (DOM-модель браузера, холст, исходное приложение). Вам не нужно заниматься DOM-моделью, но это не означает, что вы не можете этого делать. React предоставляет вам «резервные переходы», если возникнет необходимость вернуться к истокам DOM-модели.

---