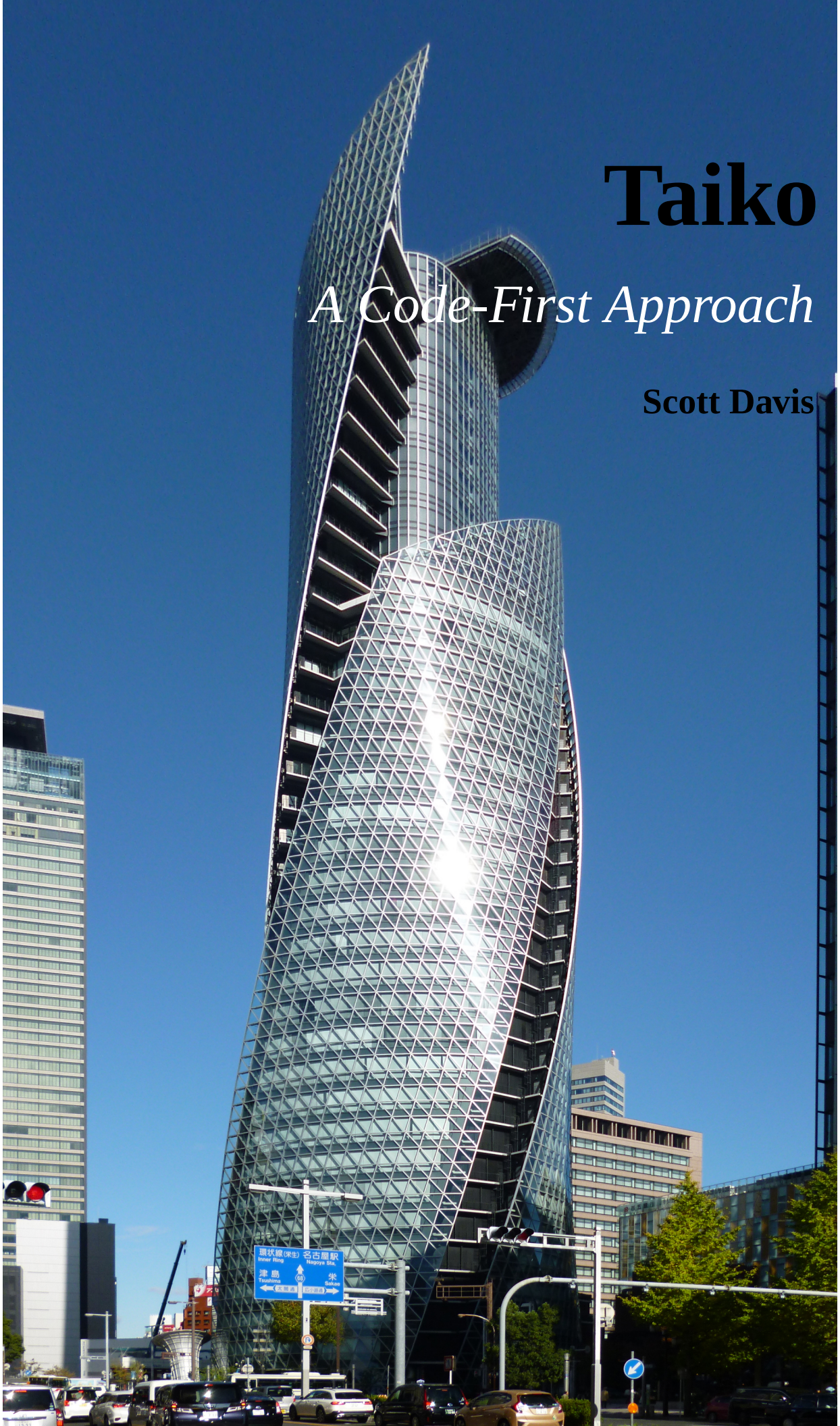

Taiko

A Code-First Approach

Scott Davis

Table of Contents

Introduction

Taiko is a JavaScript-based Domain Specific Language (DSL) for automatically driving your web browser just like a typical user does. If a user goes to your website, clicks on a link, fills in some form fields, and clicks the "submit" button, you can script up that behavior in Taiko and replay it in a reliable, automated way.

Installation and Configuration

Installing Taiko couldn't be easier. It's a single command: `npm install -g taiko`. But there's plenty more that you can do to configure and customize Taiko once it's installed.

Installing Taiko

```
$ npm install -g taiko

/Users/scott/.nvm/versions/node/v12.14.1/bin/taiko ->
/Users/scott/.nvm/versions/node/v12.14.1/lib/node_modules/
  taiko/bin/taiko.js

> taiko@1.0.7 install
/Users/scott/.nvm/versions/node/v12.14.1/lib/node_modules/taiko
> node lib/install.js

Downloading Chromium r724157 - 117.6 Mb [====================] 100%
0.0s

> taiko@1.0.7 postinstall
/Users/scott/.nvm/versions/node/v12.14.1/lib/node_modules/taiko
> node lib/documentation.js

Generating documentation to lib/api.json
+ taiko@1.0.7
added 73 packages from 114 contributors in 50.835s
```

When you install Taiko, notice that you get a known-compatible version of Chromium installed as well. Chromium is an open-source, bare-bones web browser that, as you might've guessed by the name, is the core of the Google Chrome browser. Interestingly, Chromium is also the foundation of the Opera browser, the Microsoft Edge browser, and many others. Chromium-based browsers make up roughly two-thirds of the browser market, so using Chromium with Taiko covers the widest possible swath of typical web users.

Running the Taiko REPL

```
$ taiko

Version: 1.0.7 (Chromium:81.0.3994.0)
Type .api for help and .exit to quit

> openBrowser()
 ¬ Browser opened
> goto('wikipedia.org')
 ¬ Navigated to URL http://wikipedia.org
> click('Search')
 ¬ Clicked element matching text "Search" 1 times
> write('User (computing)')
 ¬ Wrote User (computing) into the focused element.
> press('Enter')
 ¬ Pressed the Enter key
> click('Terminology')
 ¬ Clicked element matching text "Terminology" 1 times
> closeBrowser()
 ¬ Browser closed
> .exit
```

The Taiko REPL (Read Evaluate Print Loop) is an interactive terminal shell that allows you to experiment with a live browser. When you type `openBrowser()`, a browser window should open on your computer. When you type `goto('wikipedia.org')`, you should end up on the Wikipedia website.

The Taiko REPL is the perfect way to experiment with Taiko whether you are brand new to the DSL or an experienced user. Once you are confident that your code works (because you've just watched it work), you can save it and run it outside of the REPL, either manually or as a part of your automated CD pipeline.

Saving Code from the Taiko REPL

```
$ taiko

> openBrowser()
 ¬ Browser opened
> goto('wikipedia.org')
 ¬ Navigated to URL http://wikipedia.org
> closeBrowser()
 ¬ Browser closed
> .code

const { openBrowser, goto, closeBrowser } = require('taiko');
(async () => {
    try {
        await openBrowser();
        await goto('wikipedia.org');
    } catch (error) {
        console.error(error);
    } finally {
        await closeBrowser();
    }
})();

// If you provide a filename,
//   .code saves your code to the current directory
> .code visit-wikipedia.js
```

At any point in the Taiko REPL, you can type `.code` to see what the JavaScript will look like once you run your Taiko code outside of the REPL. Notice that this is modern asynchronous JavaScript — every command will `await` completion before moving on to the next step.

If you'd like to save this code for running outside of the REPL, simply provide a filename like `.code visit-wikipedia.js`. This will save the JavaScript code to the current directory.

Running Taiko Code Outside of the REPL

```
$ taiko visit-wikipedia.js

 ¬ Browser opened
 ¬ Navigated to URL http://wikipedia.org
 ¬ Browser closed
```

When you type `taiko` without a filename, it launches the Taiko REPL. When you type `taiko visit-wikipedia.js`, it runs the Taiko commands in the file.

You might have noticed that typing `openBrowser()` in the Taiko REPL actually opens a browser that you can see. By default, running Taiko commands outside of the REPL runs the browser in "headless mode". This means that the browser isn't actually shown on screen, but its behavior in headless mode is identical to its behavior with a visible browser. This is ideal for running Taiko commands in an automated server environment where there most likely isn't a screen to display the progress.

If you'd like to see the browser when running Taiko commands outside of the REPL, type `taiko --observe visit-wikipedia.js`. The `--observe` command-line flag, in addition to showing the browser, also inserts a 3 second (3000 millisecond) delay between steps to make them easier to observe. If you'd like to adjust this delay, use the `--wait-time` command-line flag — `taiko --observe --wait-time 1000 visit-wikipedia.js`.