**Serving static assets**

Broadly speaking, modern web servers serve two different types of things. On the one hand, there are *dynamic assets*, which get created on the fly when they're requested. In an online banking account dashboard, the browser might request account data an endpoint at mybankingaccount.com/api/account/373737. The server would look for an account with the id 373737, and if the account can be located, send back a JSON object representing the state of the account at that time.

On the other hand, servers also serve *static assets*, which are files that don't change between requests (unless a new version of the file is uploaded). When you visit mybankingaccount.com/my-account in a web browser, the server might send back an index.html file. This HTML file might link to stylesheets or client-side JavaScript files that are located on the same server. These HTML, CSS, and JavaScript files would all be considered static assets.

In this reading, we'll learn how to configure an Express app to serve static assets with just a single line of code. Serving static assets from an Express app is straightforward, and most of your time will be spent implementing endpoints for dynamic content. That said, you still need to know how to serve static assets.

Before learning how, a quick note: there's no requirement that an Express app serve both dynamic and static content. It's common to host static assets on a content delivery network, separate from an Express app that provides an API, to improve load times for users. In your life as a professional developer, this architectural decision (*Should I serve static assets from the same server as API?*) will depend on the project requirements.