

{@a top}

Set the Document Title

Your app should be able to make the browser title bar say whatever you want it to say. This cookbook explains how to do it.

See the .



To see the browser title bar change in the live example, open it again in the Plunker editor by clicking the icon in the upper right, then pop out the preview window by clicking the blue 'X' button in the upper right corner.

The problem with `<title>`

The obvious approach is to bind a property of the component to the HTML `<title>` like this:

```
<title>{{ThisDoesNot_Work}}</title>
```

Sorry but that won't work. The root component of the application is an element contained within the `<body>` tag. The HTML `<title>` is in the document `<head>`, outside the body, making it inaccessible to Angular data binding.

You could grab the browser `document` object and set the title manually. That's dirty and undermines your chances of running the app outside of a browser someday.

Running your app outside a browser means that you can take advantage of server-side pre-rendering for near-instant first app render times and for SEO. It means you could run from inside a Web Worker to improve your app's responsiveness by using multiple threads. And it means that you could run your app inside Electron.js or Windows Universal to deliver it to the desktop.

Use the `Title` service

Fortunately, Angular bridges the gap by providing a `Title` service as part of the *Browser platform*. The [Title](#) service is a simple class that provides an API for getting and setting the current HTML document title:

- `getTitle() : string` — Gets the title of the current HTML document.
- `setTitle(newTitle : string)` — Sets the title of the current HTML document.

You can inject the `Title` service into the root `AppComponent` and expose a bindable `setTitle` method that calls it:

Bind that method to three anchor tags and voilà!



Here's the complete solution:

Why provide the `Title` service in `bootstrap`

Generally you want to provide application-wide services in the root application component, `AppComponent`.

This cookbook recommends registering the title service during bootstrapping, a location you reserve for configuring the runtime Angular environment.

That's exactly what you're doing. The `Title` service is part of the Angular *browser platform*. If you bootstrap your application into a different platform, you'll have to provide a different `Title` service that understands the concept of a "document title" for that specific platform. Ideally, the application itself neither knows nor cares about the runtime environment.