## Changing the title tag

You can find the source HTML file in the `public` folder of the generated project. You may edit the `<title>` tag in it to change the title from “React App” to anything else.

Note that normally you wouldn’t edit files in the `public` folder very often. For example, adding a stylesheet is done without touching the HTML.

If you need to dynamically update the page title based on the content, you can use the browser `document.title` API. For more complex scenarios when you want to change the title from React components, you can use React Helmet, a third party library.

If you use a custom server for your app in production and want to modify the title before it gets sent to the browser, you can follow advice in this section. Alternatively, you can pre-build each page as a static HTML file which then loads the JavaScript bundle, which is covered here.

## Generating Dynamic `<meta>` Tags on the Server

Since Create React App doesn’t support server rendering, you might be wondering how to make `<meta>` tags dynamic and reflect the current URL. To solve this, we recommend to add placeholders into the HTML, like this:

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta property="og:title" content="__OG_TITLE__" />
    <meta property="og:description" content="__OG_DESCRIPTION__" />
  </head>
</html>
```

Then, on the server, regardless of the backend you use, you can read `index.html` into memory and replace `__OG_TITLE__`, `__OG_DESCRIPTION__`, and any other placeholders with values depending on the current URL. Make sure to sanitize and escape the interpolated values so that they are safe to embed into HTML!

If you use a Node server, you can even share the route matching logic between the client and the server. However duplicating it also works fine in basic cases.

## Injecting Data from the Server into the Page

Similarly to the previous section, you can leave some placeholders in the HTML that inject global variables, for example:

```
<!doctype html>
<html lang="en">
  <head>
    <script>
      window.SERVER_DATA = __SERVER_DATA__;
    </script>
```

Then, on the server, you can replace `__SERVER_DATA__` with a JSON of real data right before sending the response. The client code can then read `window.SERVER_DATA` to use it. **Make sure to sanitize the JSON before sending it to the client as it makes your app vulnerable to XSS attacks.**