



Promisify

Medium 15mins 2.11k done

Before promises/ `async` / `await` became the standard, it was a convention for async APIs in JavaScript to accept callbacks as the last argument. Many async versions of Node.js APIs (e.g. `fs.readFile` and `fs.rm`) have such signatures. Node.js' `util.promisify` function was created to wrap around callback-based functions by returning `Promise`s so that they can be used with `async` / `await`.

Implement a function `promisify` that takes a function following the common callback-last error-first style, i.e. taking a `(err, value) => ...` callback as the last argument, and returns a version that returns promises.

Examples

```
1 // Example function with callback as last argument
2 // The callback has the signature `(err, value) => any`
3 function foo(url, options, callback) {
4   apiCall(url, options)
5     .then((data) => callback(null, data))
6     .catch((err) => callback(err));
7 }
8
9 const promisifiedFoo = promisify(foo);
10 const data = await promisifiedFoo('example.com', { foo: 1 });
```

Asked at these companies

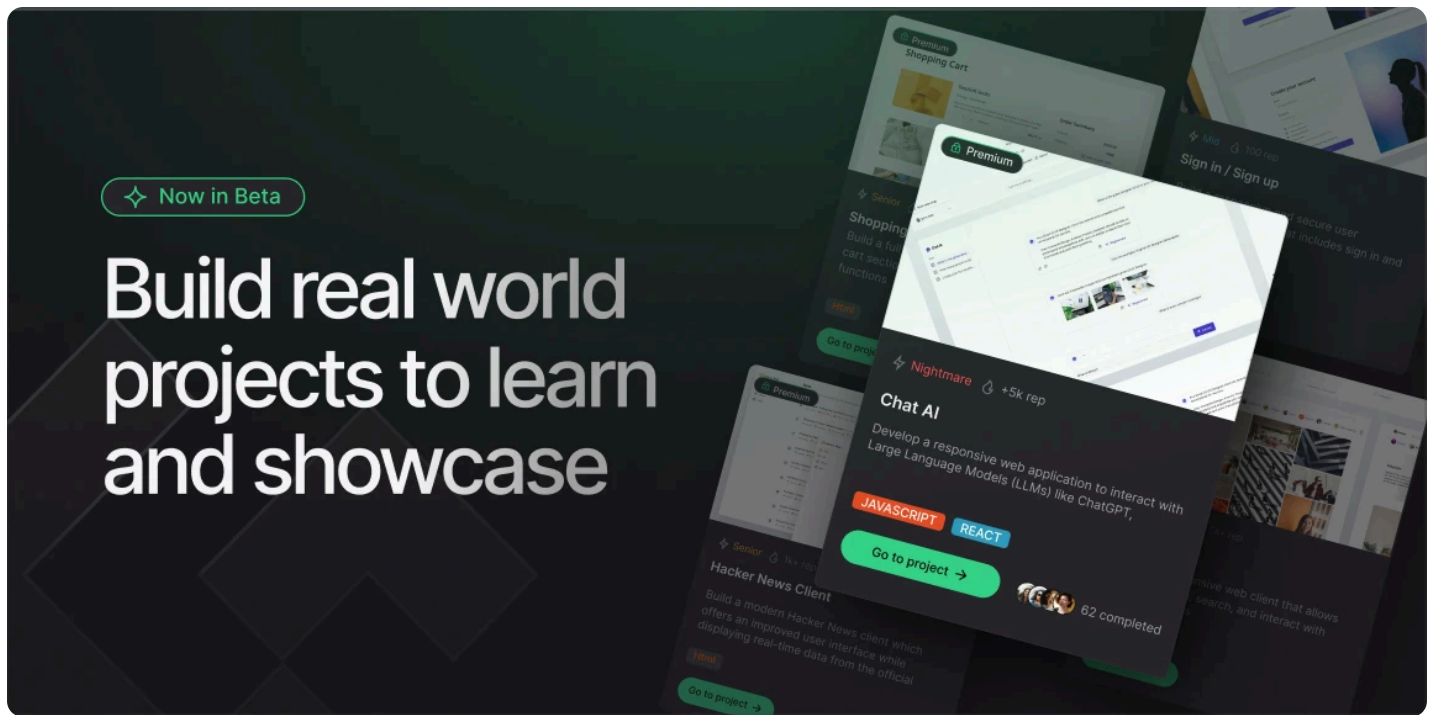
Amazon

Microsoft



Coding and submission is not supported on mobile devices. Use a wider screen to practice solving this question within the editor.

[View the solution](#)



Sponsor: [GreatFrontEnd Projects](#)

The fastest way to learn Front End – by building actual projects

Want to grow in your front end skills? Learn them the right way — build apps step-by-step while referencing best practices from Senior engineers.

- [80+ real world projects](#)
- Professional specs and designs by top-tier product managers and designers
- Guides and reference solutions from Senior Big Tech engineers
- Use a [front end skill roadmap](#) to build from beginner to pro

[Get 30% off now with BETA30 →](#)



Amazon

39/61



✓ Mark complete