06 **End-to-End Testing: External Services**

Testing React Apps

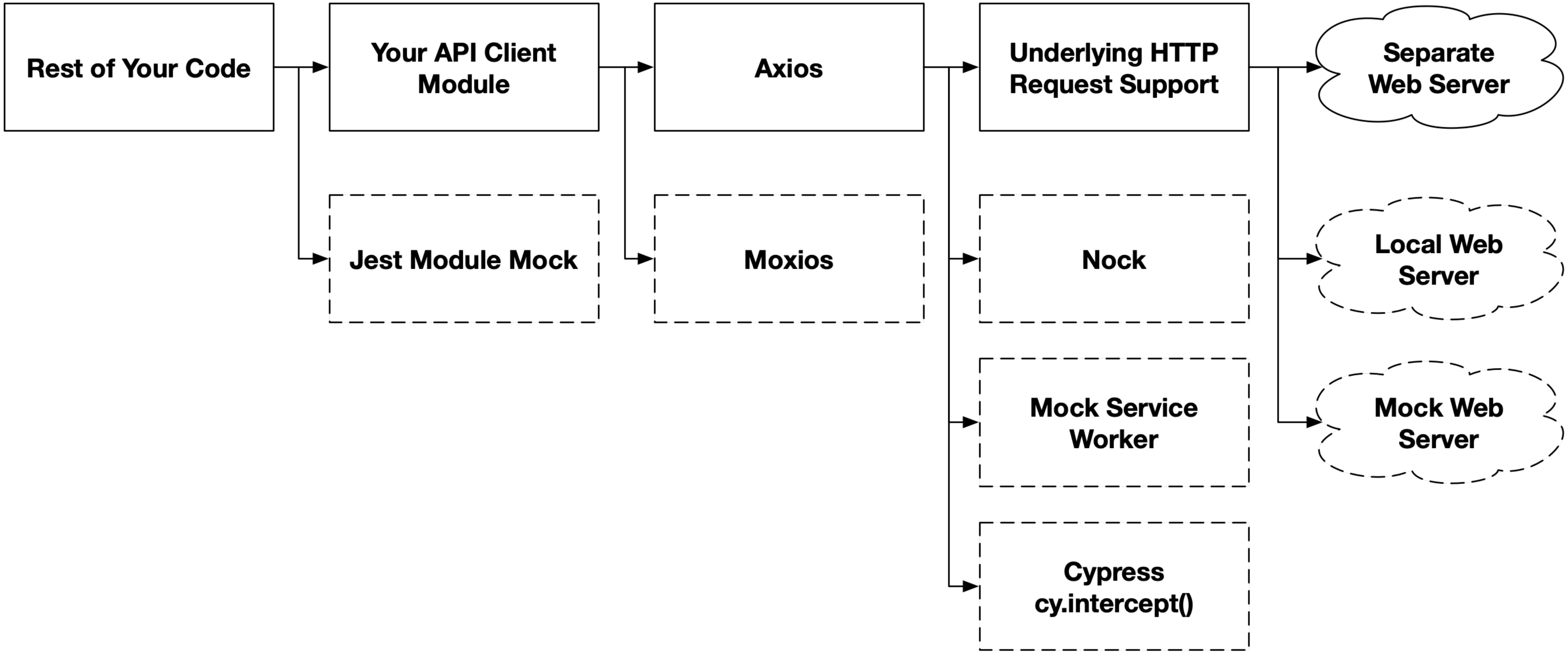
Connecting to the real backend:

Upsides?

Connecting to the real backend:

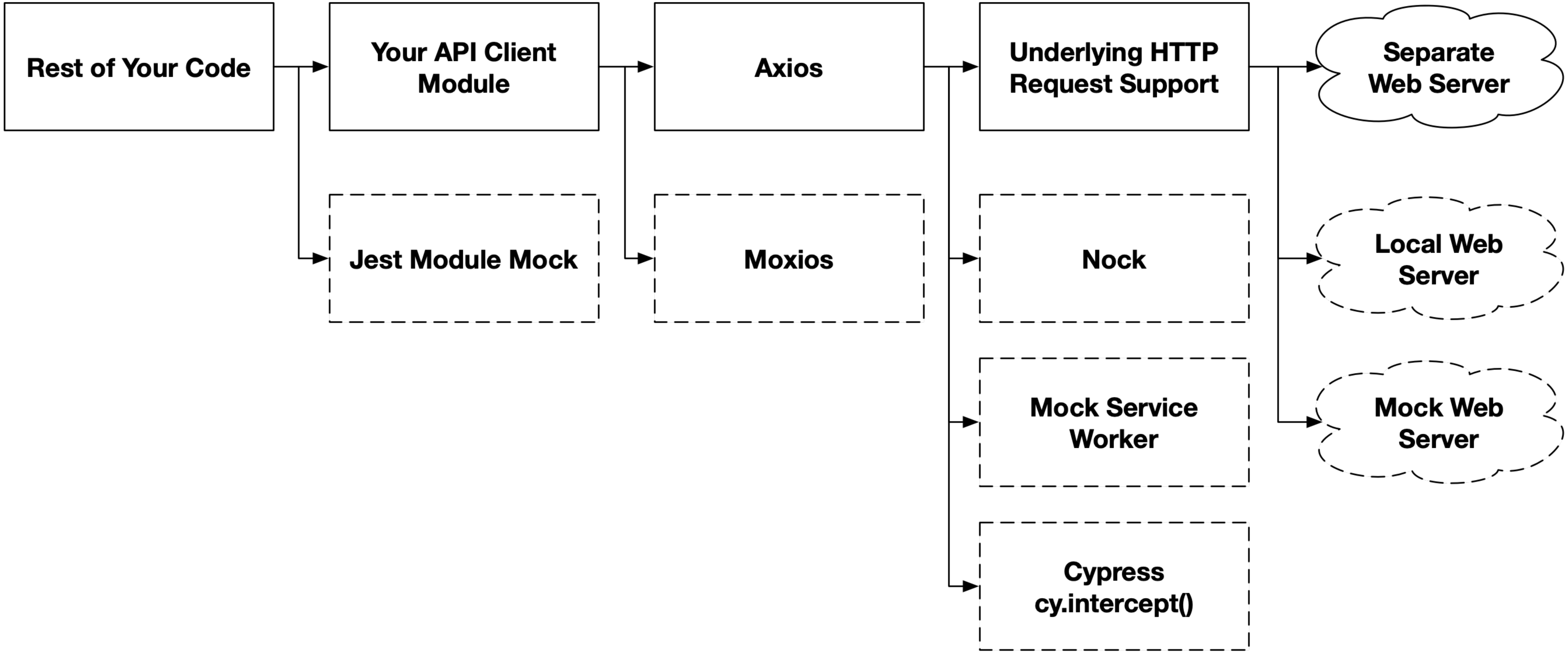
Downsides?

Options for Mocking Web Service Requests



Local web server?

Options for Mocking Web Service Requests



Rest of Your Code

Your API Client Module

Axios

Underlying HTTP Request Support

Separate Web Server

Jest Module Mock

Moxios

Nock

Local Web Server

Mock Service Worker

Mock Web Server

Cypress cy.intercept()

Intercept the requests and mock responses

Network Request Testing Strategies - Cypress Guides

cy.intercept()

Allows you to configure responses to be returned instead of real requests being made

To the code!

```
cy.intercept(
  'GET',
  'https://rn-testing-api.herokuapp.com/widgets',
  [
    {id: 1, name: 'Widget 1'},
    {id: 2, name: 'Widget 2'},
  ]
);
```

`cy.intercept(method, url, staticResponse)`

Testing Outgoing Data

```
cy.intercept(method, url, staticResponse).as('myNamedRequest');

// user actions

cy.wait('@myNamedRequest')
  .its('request.body')
  .should('deep.equal', {someKey: 'someValue'});
```

Don't Fear the `wait()`

Exercise 4

- react-testing-exercises
- `exercises/Exercise-4.md`

For help:
- Cypress Docs: Writing Your First E2E Test
- Cypress Docs: Network Requests - Routing
- Cypress Docs: Network Requests - Assertions

Recommendation:

- Automate testing against `cy.intercept()`
- Manually test integration with real service
- If you need to automate testing with real service, do a small separate test suite

To Discuss:

For a real app you work on, what would be the most valuable scenarios to end-to-end test?

Do you have any uncertainty about how you could automate those scenarios?