Async in Redux



Cory House

@housecor

reactjsconsulting.com



Here's the plan



Why a mock API?

Mock API setup

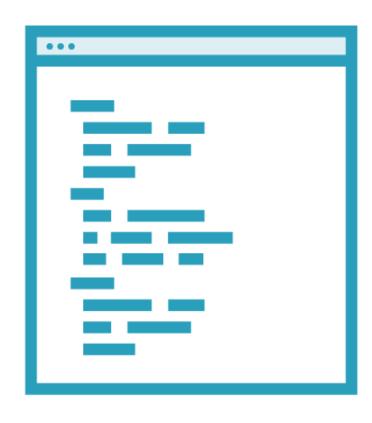
Redux middleware

Async libraries

Implement Thunks



Why a Mock API?



Start before the API exists

Independence

Backup plan

Ultra-fast

Test slowness

Aids testing

Point to the real API later



Demo



Setup Mock API



Redux Middleware



Redux Middleware



You can write custom logic that runs here.





Handling async API calls

Logging

Crash reporting

Routing



Custom Logger Middleware

```
// Logs all actions and states after they are dispatched.
const logger = store => next => action => {
  console.group(action.type)
  console.info('dispatching', action)
  let result = next(action)
  console.log('next state', store.getState())
  console.groupEnd()
  return result
}
```



Async in Redux



Redux Async Libraries

redux-thunk Return functions from action creators

redux-promise Use promises for async



Comparison

Thunks

Sagas

Functions

Generators

Clunky to test

Easy to test

Easy to learn

Hard to learn



```
export function deleteAuthor(authorId) {
  return dispatch => {
    return AuthorApi.deleteAuthor(authorId)
    .then(() => {
        dispatch(deletedAuthor(authorId));
     })
     .catch(handleError);
  };
}
```

Thunk

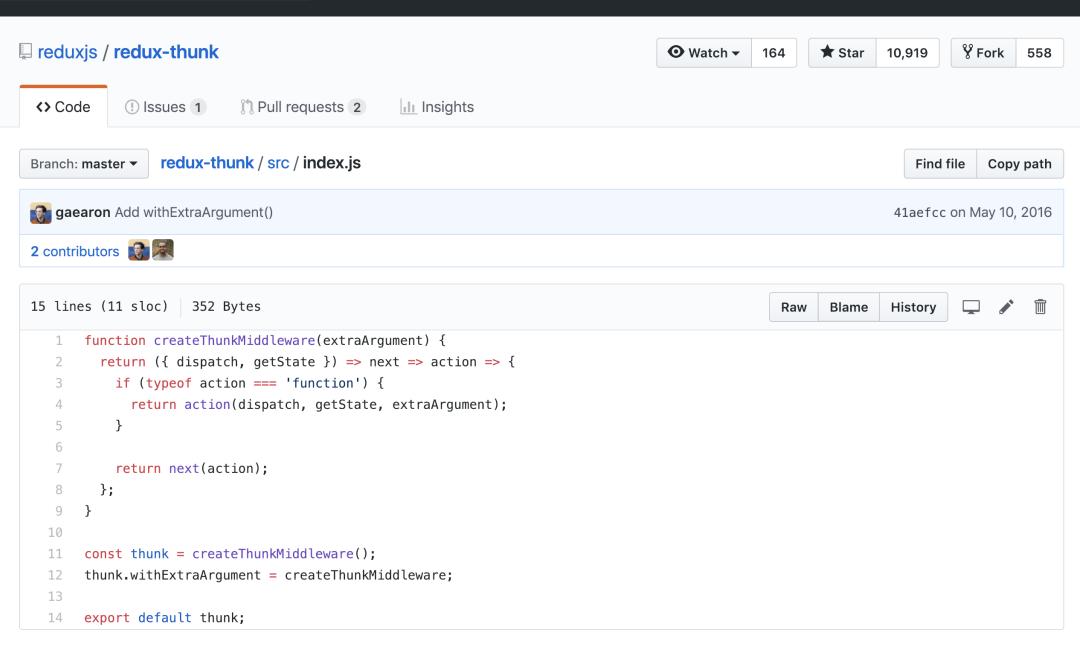


```
export function deleteAuthor(authorId) {
  return (dispatch, getState) => {
    return AuthorApi.deleteAuthor(authorId)
    .then(() => {
        dispatch(deletedAuthor(authorId));
    })
    .catch(handleError);
  };
}
```

Thunk







Middleware isn't *required* to support async in Redux.
But you probably want it.



```
export function deleteAuthor(dispatch, authorId) {
  return AuthorApi.deleteAuthor(authorId)
    .then(() => {
     dispatch(deletedAuthor(authorId));
     })
     .catch(handleError);
}
```

Thunk



Why Use Async Middleware?

Consistency

Purity

Easier testing



Demo



Let's get thunky.



Wrap Up



Why a mock API?

Set up Mock API

- Express and json-server

Redux middleware

Async libraries

- Thunks, Redux-Promise, and Sagas

Naming async actions

Consider SUCCESS and ERROR

First thunk, complete!

Next up: Async writes

