

Testing and Error Handling



Christian Wenz

Consultant, Software Architect

@chwenz

Faking APIs



Mocking



Stubbing

Mocking Fetch Calls with Mirage JS

<https://miragejs.com/>

index.js

```
new Server({
  routes() {
    this.get(
      "/events",
      () => [ /* ... */ ]
    );
  }
});
```

Stubbing APIs with json-server

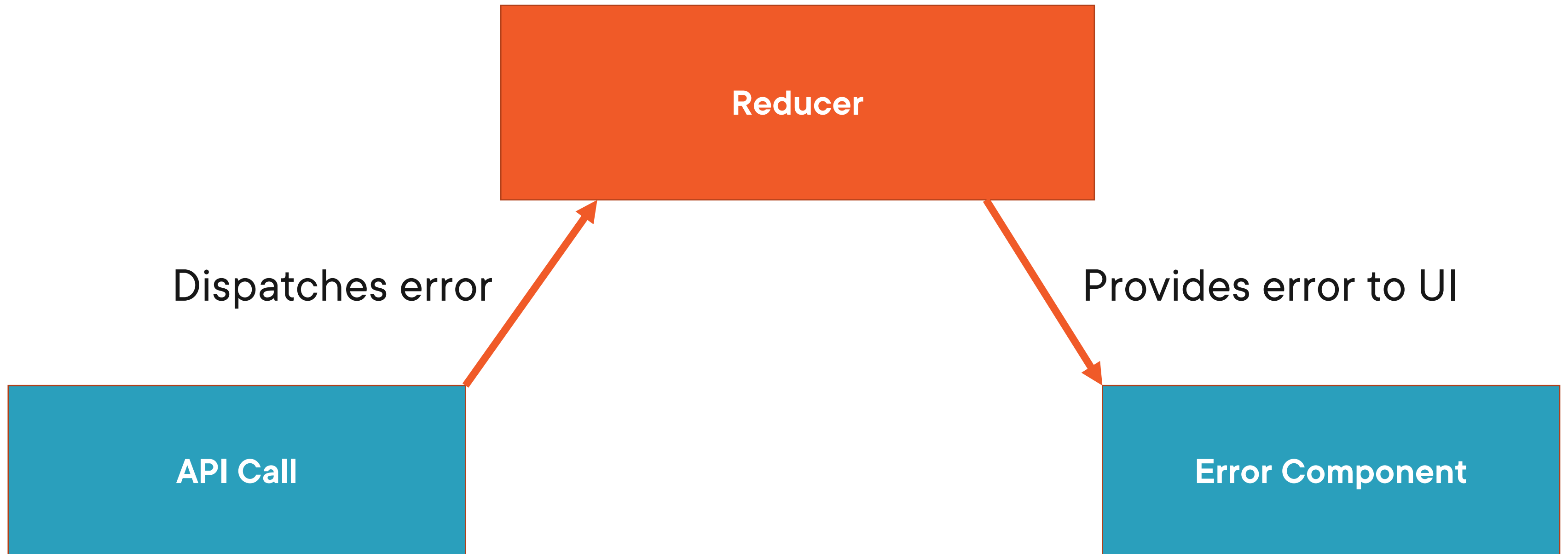
<https://github.com/typicode/json-server>

globoticket.json

```
{  
  "events": [  
    { /* ... */ }  
  ]  
}
```

```
> json-server --watch globoticket.json
```

Centralized Error Handling



Error Boundaries

```
class ErrorBoundary extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = { error: null };  
  }  
  render() {  
    if (this.state.error) {  
      <p>Error!</p>  
    }  
    return this.props.children;  
  }  
}
```

```
<ErrorBoundary fallback={<div>Error!</div>}>  
  <Suspense fallback={<div>Loading ...</div>}>  
    <Eventlist />  
  </Suspense>  
</ErrorBoundary>
```

Outlook: Using Suspense

Experimental feature

Summary



Faking APIs

Implementing centralized error handling

Using Suspense and error boundaries