

# Handling Errors and Debugging Components



**Peter Kellner**

Developer and Author

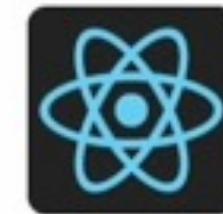
[peterkellner.net](http://peterkellner.net) | [linkedin.com/in/peterkellner99](https://linkedin.com/in/peterkellner99) Mastodon:  
[@pkellner@peterkellner.net](mailto:@pkellner@peterkellner.net)



## Basics of React Developer Tools

Analyzing the component over-rendering problem

Handling production time errors with Error Boundaries



### React Developer Tools

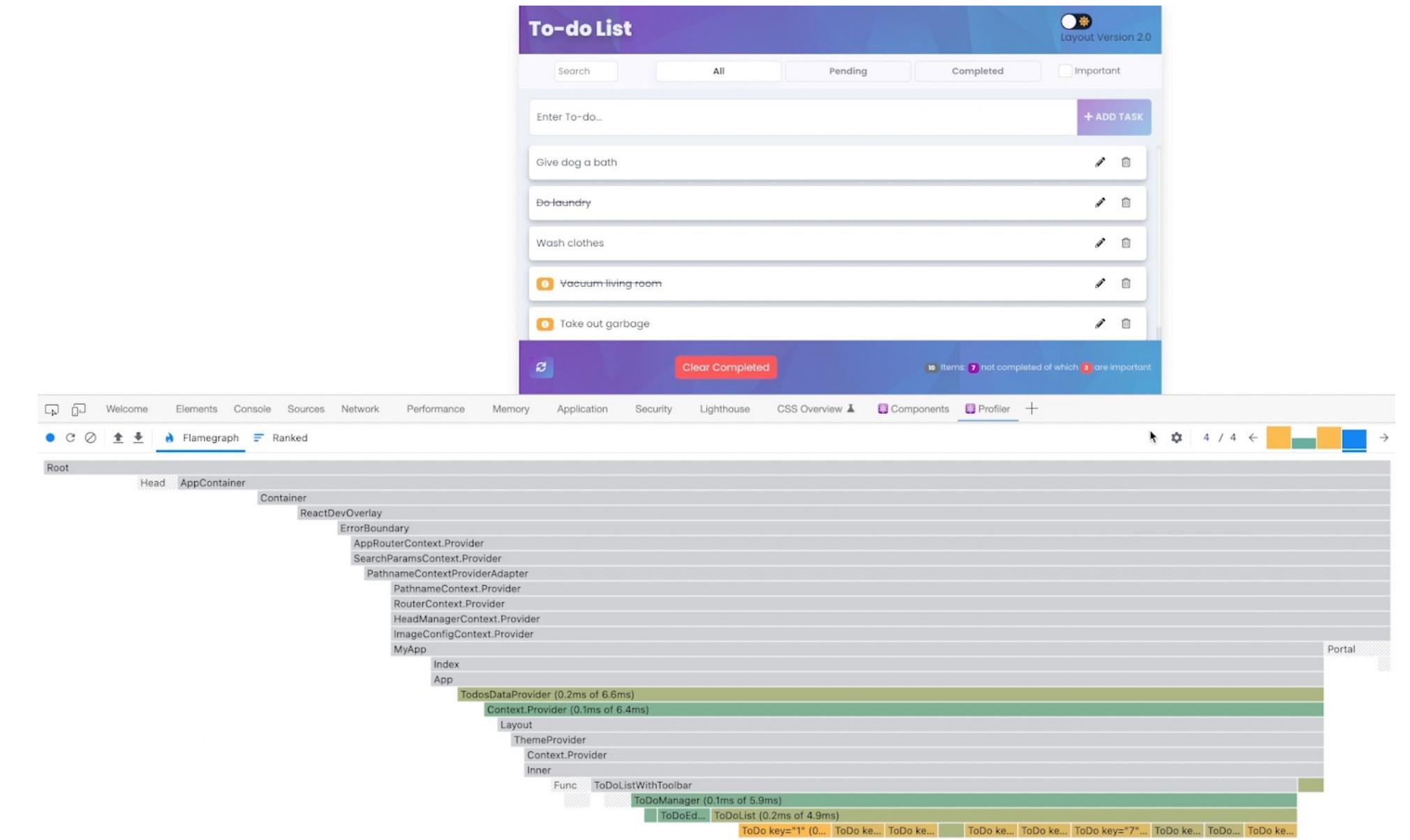
Featured

★★★★★ 1,405 ⓘ | [Developer Tools](#) | 3,000,000+ users

# Basics of React Developer Tools

## Analyzing the component over-rendering problem

## Handling production time errors with Error Boundaries



## Basics of React Developer Tools

Analyzing the component over-rendering problem

Handling production time errors with Error Boundaries

```
const ToDo = (props) => {
  return (
    <ErrorBoundary>
      <Inner {...props} />
    </ErrorBoundary>
  )
}
```

React Developer Tools - Chrome

https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihf

Home > Extensions > React Developer Tools

# React Developer Tools

Featured

★★★★★ 1,405 | Developer Tools | 3,000,000+ users

Overview Privacy practices Reviews Support Related

The screenshot shows the React Developer Tools extension page on the Chrome Web Store. The main heading is "React Developer Tools" with a "Featured" badge. Below it are the rating (5 stars, 1,405 reviews), category (Developer Tools), and user count (3,000,000+ users). A horizontal navigation bar includes "Overview" (which is active), "Privacy practices", "Reviews", "Support", and "Related". Below this is a large preview image showing the DevTools interface. The interface features a top navigation bar with tabs like Elements, Sources, Console, Components (which is selected), Profiler, Performance, Network, and more. The main area displays a "todos" application with a "Try React DevTools" input field. On the left, a component tree shows the "App" component structure, including "Header", "MainSection", "Todolist", and "Footer" components. The "TodoTextInput" component is highlighted in blue. On the right, the "Components" panel shows detailed information for the "TodoTextInput" component, including its props (newTodo checked, onSave(), placeholder: What needs to be done?), hooks (State: Try React DevTools), and rendered by (Header, App).

# Browser Debugging Tools

**Standard debugger**

**Speaks HTML**

**React developer tools**

**Speaks React**

# React Developer Tools Compatibility



**Chrome based browser (Google Chrome and Microsoft Edge)**



**Firefox browser**



**Standalone app**

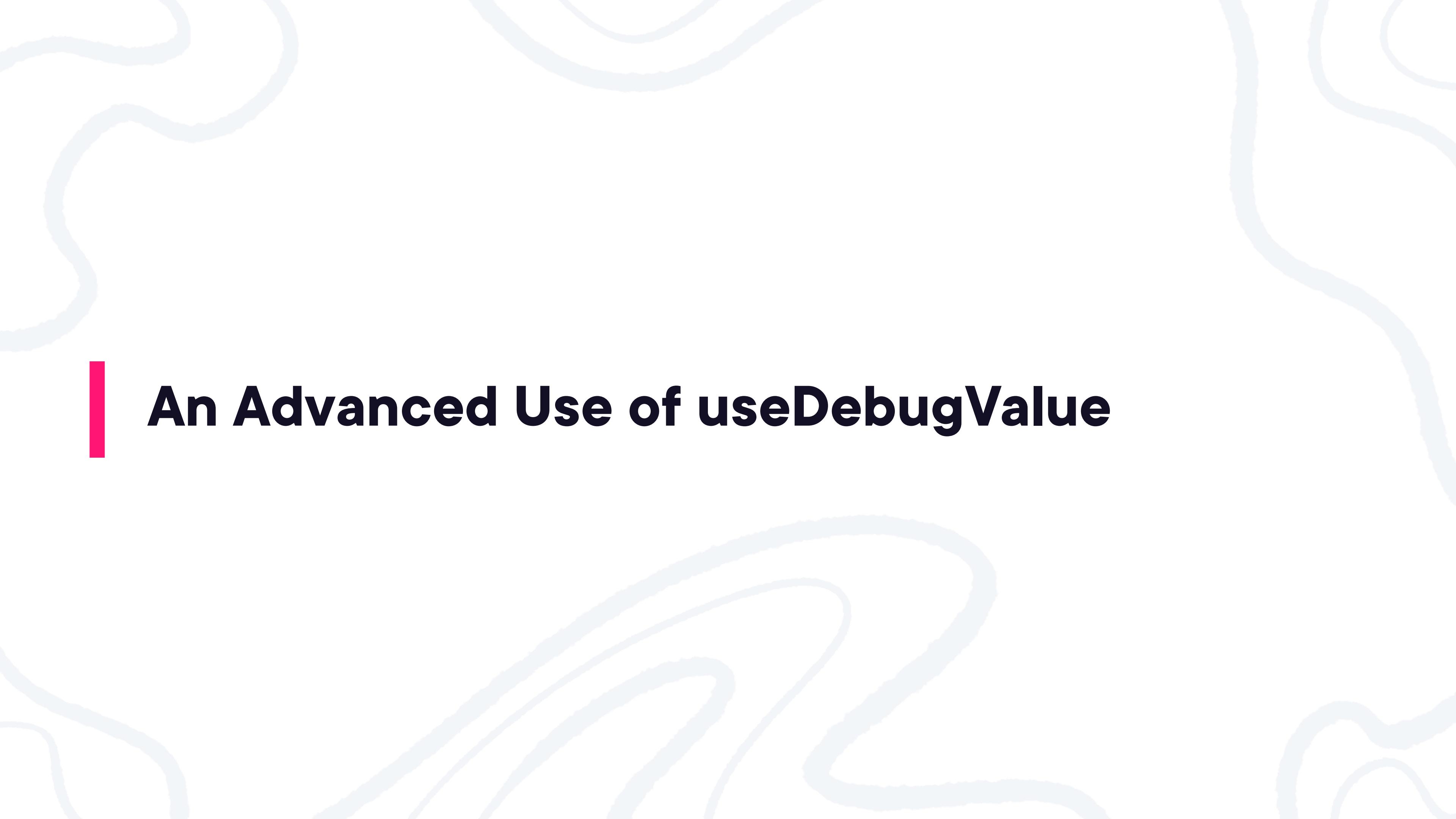
# Why do we need a clearer view into the browser?

# Why do we need a clearer view into the browser?

The browser does not know anything about **React code**

# The `useDebugValue` Hook Basics

# Integrating `useDebugValue` Into Custom Theme Hook



# An Advanced Use of `useDebugValue`

# useDebugValue Advanced Usage

```
function myCustomHook() {
  const [textState, setTextState] = useState("");
  useDebugValue(` ${textState} `);
  return <div>{textState}</div>
}
}
```

# useDebugValue

## Advanced

### Usage

```
function myCustomHook() {
  const [textState, setTextState] = useState("");

  useDebugValue(
    `${textState}`,
    (val) => {
      if (val === "...") {
        "Result ...";
      } else {
        "Other Result...";
      }
    });
}

return <div>{textState}</div>
);
}
```

# useDebugValue

## Advanced

### Usage

```
function myCustomHook() {
  const [textState, setTextState] = useState("");

  useDebugValue(
    `${textState}`,
    (val) => {
      if (val === "...") {
        return "Result ...";
      } else {
        return "Other Result...";
      }
    });
}

return <div>{textState}</div>
}
}
```



# The Profiler in React Developer Tools

# **Adding Error Boundaries for Catching Errors**

# **Adding Error Boundaries for Catching Errors**

# Takeaways

**Tools available to make debugging experiences better**

**React library catches runtime errors in our browsers**

**React Dev Tools added 2 tabs to Chrome Debug Tools**

- Component Viewer
- FlameGraph

**ErrorBoundary class to wrap components for better error handling**