

REACT

Partha Das Chowdhury



Lifecycle Methods

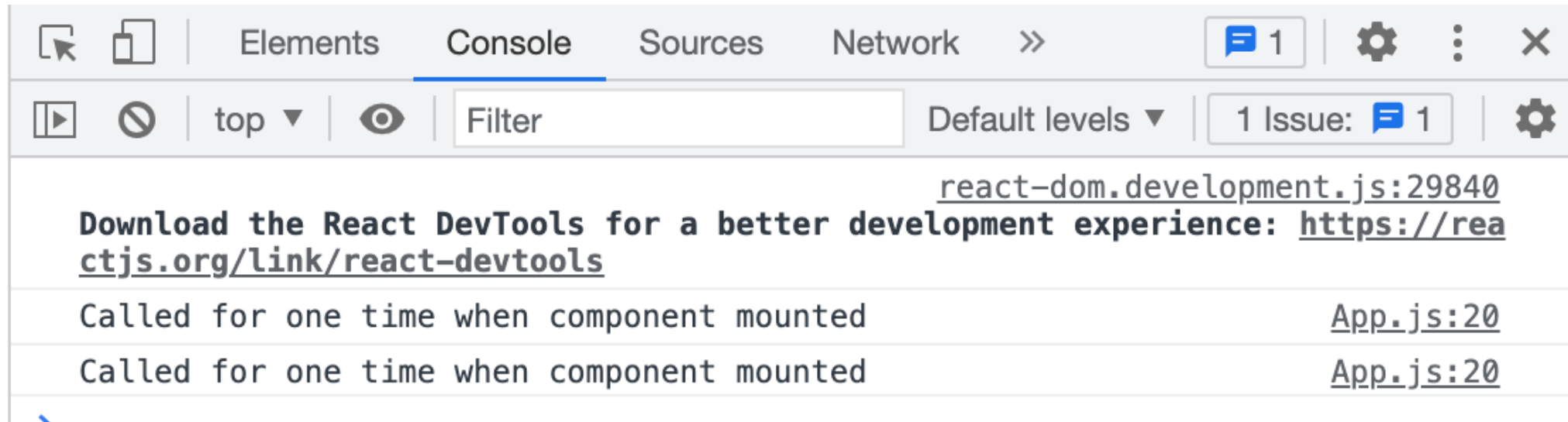
```
class App extends Component {
  state = {
    count: 0,
    v: false,
  };
  componentDidMount() {
    console.log("Called for one time when component mounted");
  }

  shouldComponentUpdate(nextProps, nextState) {
    if (nextState.count !== this.state.count) {
      console.log("Before Count changed");
    }
    console.log("Before Called every time");
    return true;
  }

  componentDidUpdate(prevProp, prevState) {
    if (prevState.count !== this.state.count) {
      console.log("After Count changed");
    }
    console.log("After Called every time");
  }

  render() {
    return (
      <div>
        Count: {this.state.count}
        <button onClick={() => this.setState({ count: this.state.count + 1 })}>
          Increase
        </button>
        <button onClick={() => this.setState({ v: !this.state.v })}>
          Toggle
        </button>
      </div>
    );
  }
}
```

Lifecycle Methods



Lifecycle Methods

react-dom.development.js:29840	
Download the React DevTools for a better development experience: https://reactjs.org/link/react-devtools	
Called for one time when component mounted	App.js:20
Called for one time when component mounted	App.js:20
Before Count changed	App.js:25
Before Called every time	App.js:27
Before Count changed	App.js:25
Before Called every time	App.js:27
After Count changed	App.js:32
After Called every time	App.js:34
>	

Lifecycle Methods

```
const App = () => {
  const [count, setCount] = useState(0);
  const [v, setV] = useState(false);

  // componentDidMount – equivalent, not exact
  useEffect(()=>{
    console.log('Called for one time when component mounted')
  }, [])

  // dependency updates
  useEffect(()=>{
    console.log('After Count changed')
    return () => {
      console.log('Before Count changed')
    }
  }, [count])


  // componentDidUpdate – equivalent, not exact
  useEffect(()=>{
    console.log('After Called every time')
    return () => {
      console.log('Before Called every time')
    }
  })

  return <div>Count: {count}
    <button onClick={()=>setCount(count+1)}>Increase</button>
    <button onClick={()=>setV(!v)}>Toggle</button>
  </div>
}
```

Lifecycle Methods

react-dom.development.js:29840	
Download the React DevTools for a better development experience: https://reactjs.org/link/react-devtools	
Called for one time when component mounted	App.js:57
After Count changed	App.js:62
After Called every time	App.js:70
Before Count changed	App.js:64
Before Called every time	App.js:72
Called for one time when component mounted	App.js:57
After Count changed	App.js:62
After Called every time	App.js:70
>	

Useful VS Code Extensions




Prettier - Code formatter

✓ Prettier | 30,232,858 | ★★★★★

Code formatter using prettier

[Disable](#) [Uninstall](#) ⚙️

This extension is enabled globally.



Simple React Snippets

✓ Burke Holland | 2,608,437 | ★

Dead simple React snippets you will actually

[Disable](#) [Uninstall](#) ⚙️

This extension is enabled globally.

Happy REACT - ing