

WEB DEVELOPER CHEMNITZ

Regelmäßiges monatliches Treffen chemnitzer Web-Entwickler.
Erfahrungsaustausch zu aktuellen Technologien in interessanten Vorträgen.

AGENDA

19.00 Uhr • **Alexander Widera** Java News

19.05 Uhr • **Lars Wiedemann** Frontend News

19.15 Uhr • **Daniel Schröder** PHP News

//

19.20 Uhr • **Morris Jobke** Vue.js

//

20.00 Uhr • **Bettina Helgenlechner** React Portals

20.20 Uhr • **Martin Kutter** React Context API

21.00 Uhr • **Community** Networking

Java NEWS

Frontend NEWS

Version 3.0

Lighthouse

- csv export
- First Contentful Paint measurement



Version 1.0

WebRTC

- feature completed



WebRTC

Polly.js

- from netflix
- HTTP interactions
 - record
 - replay
 - stub
- cli support
- Framework Support:
 - ember

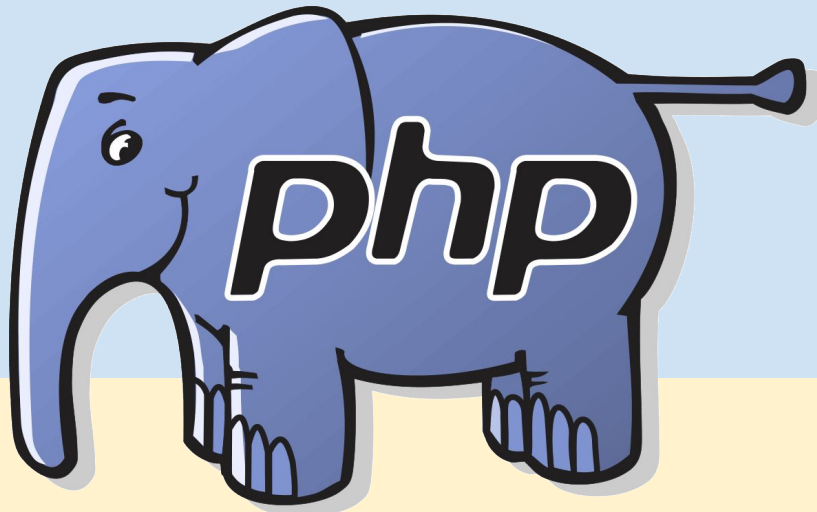


PHP NEWS

Alpha Releases

PHP 7.3

- Alpha 1 and 2 released
- Improved PHP GC
- Removed support for BeOS
- Add `net_get_interfaces()` function
- Add
`DateTime::createFromImmutable()`
method
- Implemented flexible heredoc and
nowdoc syntax
- Added support for references in `list()`
and array destructuring
- Implemented PCRE2
- Many bug fixes



GitHub, Magento Acquisitions

- Microsoft acquires GitHub (7.5 billion USD)
- Adobe acquires Magento (1.68 billion USD)



Microsoft



GitHub



+



Magento®

September 21st & 22nd

PHP Developer Days

- 2 days at International Congress Center Dresden
- TDD full-day workshop
- Event Sourcing full-day workshop
- 4 more half-day workshops
- 10 conference talks
 - SPL (Standard PHP Library) vs. PHP 7
 - Zero Downtime DB Deployments
 - Async Request Processing
 - Kubernetes
 - Prometheus
 - Profiling
 - 4 more talks



php developer days 2018

September 21st & 22nd 2018 in Dresden, Germany

85	14	12	23
Days	Hours	Minutes	Seconds

HOME

SCHEDULE

WORKSHOPS

TALKS

SPEAKERS/TRAINERS

BUY TICKETS

SCHOLARSHIPS

THE VENUE



REACT CONTEXT API

Context provides a way to pass data through the component tree without having to pass props down manually at every level.

When to Use?

To share “global” data for a tree of React components e.g.

- current user
- preferred language
- UI theme



When to Use?

To share “global” data for a tree of React components e.g.

- current user
- preferred language
- UI theme



When to Use?

To share “global” data for a tree of React components e.g.

- current user
- preferred language
- UI theme



When to Use?

To share “global” data for a tree of React components e.g.

- current user
- preferred language
- UI theme



When to Use?

To share “global” data for a tree of React components e.g.

- current user
- preferred language
- UI theme



How?

1. Create Context
2. Provide Context
3. Consume Context

```
const BackgroundContext = React.createContext('purple');
```

```
class App extends React.Component {  
  render() {  
    return (  
      <BackgroundContext.Provider value="lightpurple">  
        <Section />  
      </BackgroundContext.Provider>  
    );  
  }  
}
```

```
function Section(props) {  
  // A component in the middle doesn't have to pass the theme down explicitly  
  // anymore.  
  return (  
    <div>  
      <Button />  
    </div>  
  );  
}
```

```
function Button(props) {  
  // Use a Consumer to read the current theme context.  
  // React will find the closest theme Provider above and use its value.  
  return (  
    <BackgroundContext.Consumer>  
      {background => <button data-background={background} />}  
    </BackgroundContext.Consumer>  
  );  
}
```

Updating Provider

You can pass a function down through the context to allow consumers to update the context.

```
const ThemeContext = React.createContext({
  theme: "dark",
  toggleTheme: () => {},
});

class App extends React.Component {
  constructor(props) {
    super(props);

    this.toggleTheme = () => {
      this.setState(state => ({
        theme: state.theme === themes.dark ? themes.light : themes.dark,
      }));
    };

    this.state = {
      theme: "dark",
      toggleTheme: this.toggleTheme,
    };
  }

  render() {
    return (
      <ThemeContext.Provider value={this.state}>
        <SomeComponent />
      </ThemeContext.Provider>
    );
  }
}
```

Multiple Provider

React will find the closest Provider above and use its value. You can combine multiple different providers.



Multiple Provider

React will find the closest Provider above and use its value. You can combine multiple different providers.

```
class App extends React.Component {
  render() {
    const { signedInUser, theme } = this.props;

    return (
      <ThemeContext.Provider value={theme}>
        <UserContext.Provider value={signedInUser}>
          <SomeComponent />
        </UserContext.Provider>
      </ThemeContext.Provider>
    );
  }
}

function SomeOtherComponent() {
  return (
    <ThemeContext.Consumer>
      {theme => (
        <UserContext.Consumer>
          {user => (
            <SomePage user={user} theme={theme} />
          )}
        </UserContext.Consumer>
      )}
    </ThemeContext.Consumer>
  );
}
```

HOC

Use Higher Order Components to provide contexts that are consumed by many components (e.g. theme or localization).

```
// theme-context.js
const ThemeContext = React.createContext('light');

export function withTheme(Component) {
  return function ThemedComponent(props) {
    return (
      <ThemeContext.Consumer>
        {theme => <Component {...props} theme={theme} />}
      </ThemeContext.Consumer>
    );
  };
}
```

```
// button.jsx
import { withTheme } from './theme-context';

function Button({theme, ...rest}) {
  return <button className={theme} {...rest} />;
}

const ThemedButton = withTheme(Button);
```

IMMER LETZTER DONNERSTAG IM MONAT

Donnerstag 26. Juli, Lokomov - Melde dich bei [Meetup.com](https://www.meetup.com) an.
Teile deine Erfahrungen! Wer ist als nächstes dran?



[goo.gl/F7yd6v](https://www.meetup.com/de-DE/Web-Developer-Chemnitz/polls/1263552/)

<https://www.meetup.com/de-DE/Web-Developer-Chemnitz/polls/1263552/>

weitere Termine: Do 26. Juli, Do 30. August, Do 27. September



<https://github.com/ChemnitzerWebDevs/slides>