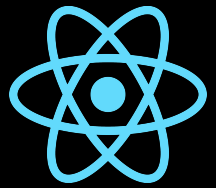
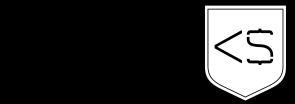
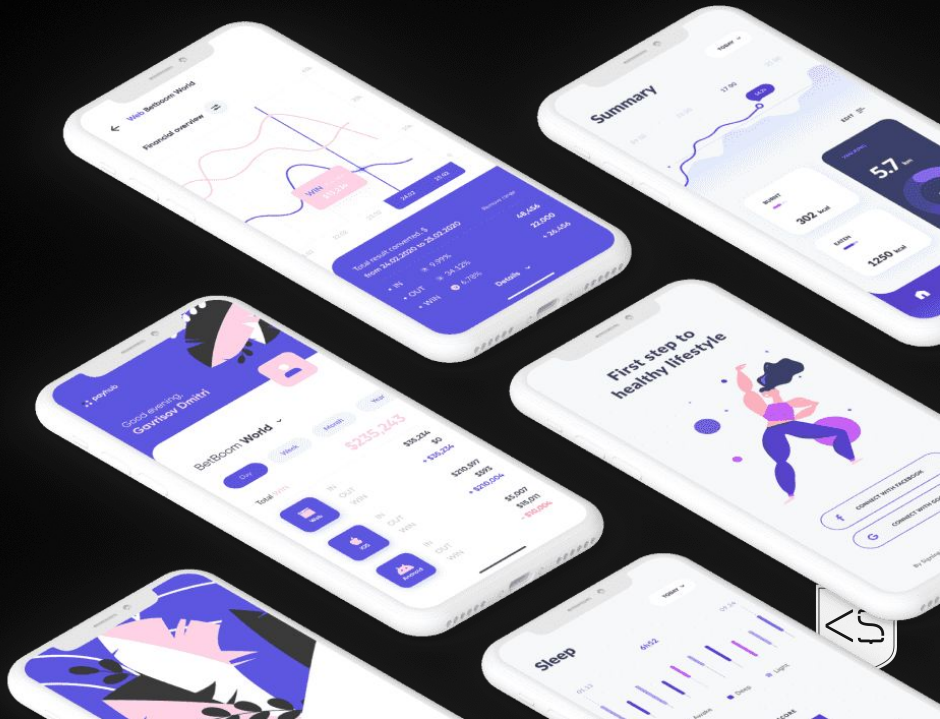


React Native Basics



INHALT

- NodeJS
- React.js Grundlagen
- React Native Grundlagen
- Expo, Builds & Veröffentlichung
- Praxis




node
JS

NodeJS

Was ist Node.js?

- Eine JavaScript-Laufzeitumgebung
- Läuft nicht im Browser, sondern direkt auf dem Server
- Baut auf der V8 Engine von Google Chrome auf

Warum Node.js?

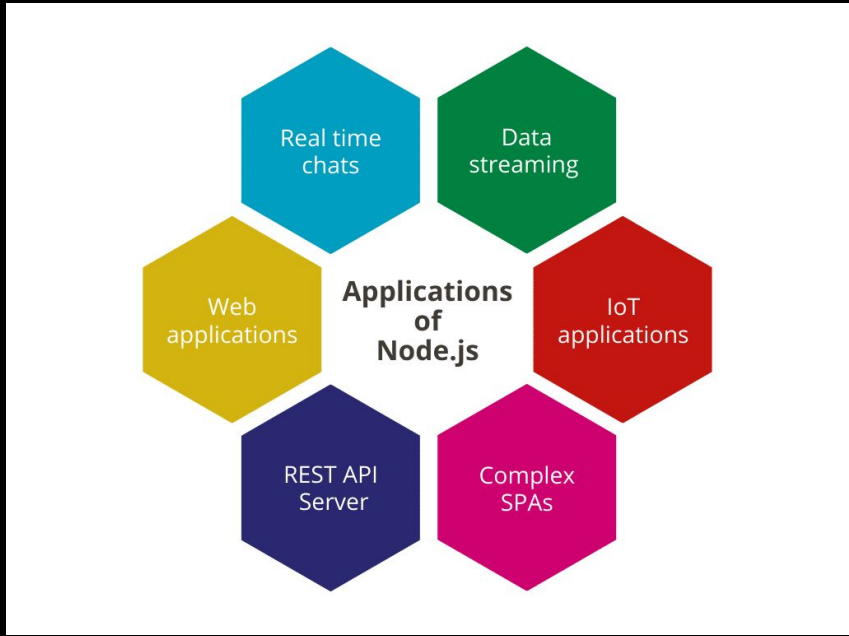
- Erlaubt es, JavaScript auch serverseitig zu nutzen
- Sehr schnell & effizient durch Event-Loop & Non-Blocking I/O
- Riesiges Ökosystem an Paketen über npm (später mehr dazu)

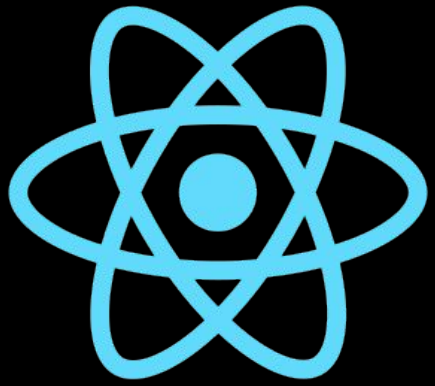
💡 **Merke: Ohne Node.js könnten wir React & npm in der Entwicklung nicht so nutzen.**

NodeJS

Typische Anwendungsfälle:

- Webserver & APIs
- Realtime-Apps (z. B. Chats, Spiele)
- Build-Tools (z. B. Webpack, Vite)
- Command Line Tools
- IoT
- Streaming & Datenverarbeitung





React JS

ReactJS

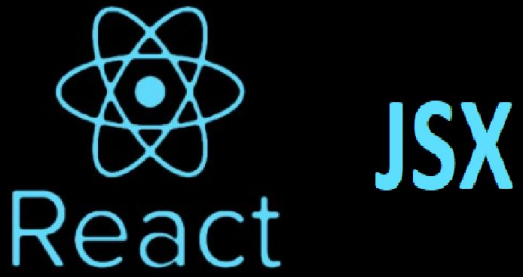
- UI-Bibliothek (kein komplettes Framework) für das Web
- Entwickelt von Facebook, Open Source seit 2013
- Komponentenbasiert
- Einwegiger Datenfluss (Unidirectional Data Flow)
- Virtual DOM für effizientes Rendering

ReactJS

Um das User-Interface (UI) einer Web-Applikation zu programmieren, nutzt man die folgenden drei Technologien:

- **HTML:** Gibt die Struktur der UI vor.
- **CSS:** Fügt den HTML-Elementen Style (Farben, Abstände etc.) hinzu
- **JavaScript:** Reagiert auf die User-Interaktion mit HTML-Elementen und löst weitere Prozesse aus.

JSX



Was ist JSX?

- Eine Erweiterung von JavaScript-Syntax
  Erlaubt es, HTML-ähnlichen Code direkt in JavaScript zu schreiben

Warum JSX?

- Macht den Code lesbarer und strukturierter
- Komponenten sehen aus wie HTML, sind aber JavaScript
- Ermöglicht die einfache Kombination von Logik + UI

```
function Welcome() {
  return <h1>Hello, React!</h1>;
}
```

JSX

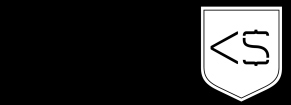
Was ist JSX?

- Eine Erweiterung von JavaScript-Syntax
- Erlaubt es, HTML-ähnlichen Code direkt in JavaScript zu schreiben

Warum JSX?

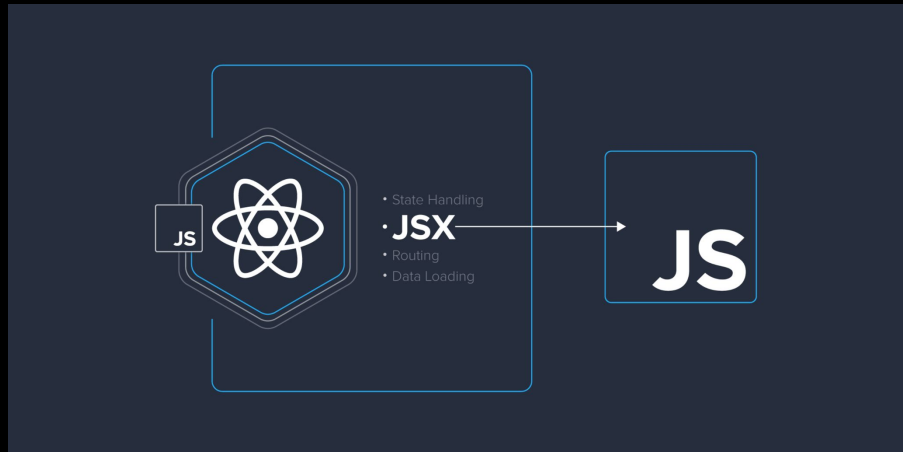
- Macht den Code lesbarer und strukturierter
- Komponenten sehen aus wie HTML, sind aber JavaScript
- Ermöglicht die einfache Kombination von Logik + UI

```
function Welcome() {
  return <h1>Hello, React!</h1>;
}
```



JSX wird zu HTML, CSS & JS

- Browser verstehen kein JSX direkt.
- JSX wird vom Compiler in reines JavaScript übersetzt.
- Der Browser rendert dann daraus normales HTML & CSS.



TailwindCSS

Was ist Tailwind CSS?

- Ein Utility-First CSS Framework
- Statt eigene Klassen zu schreiben, nutzt man vorgefertigte Utility-Klassen

Vorteile:

- Schnelles Prototyping
- Einheitliches Design ohne viel Custom-CSS
- Sehr flexibel & anpassbar (Farben, Abstände, Breakpoints usw.)

TailwindCSS vs. Vanilla CSS

-

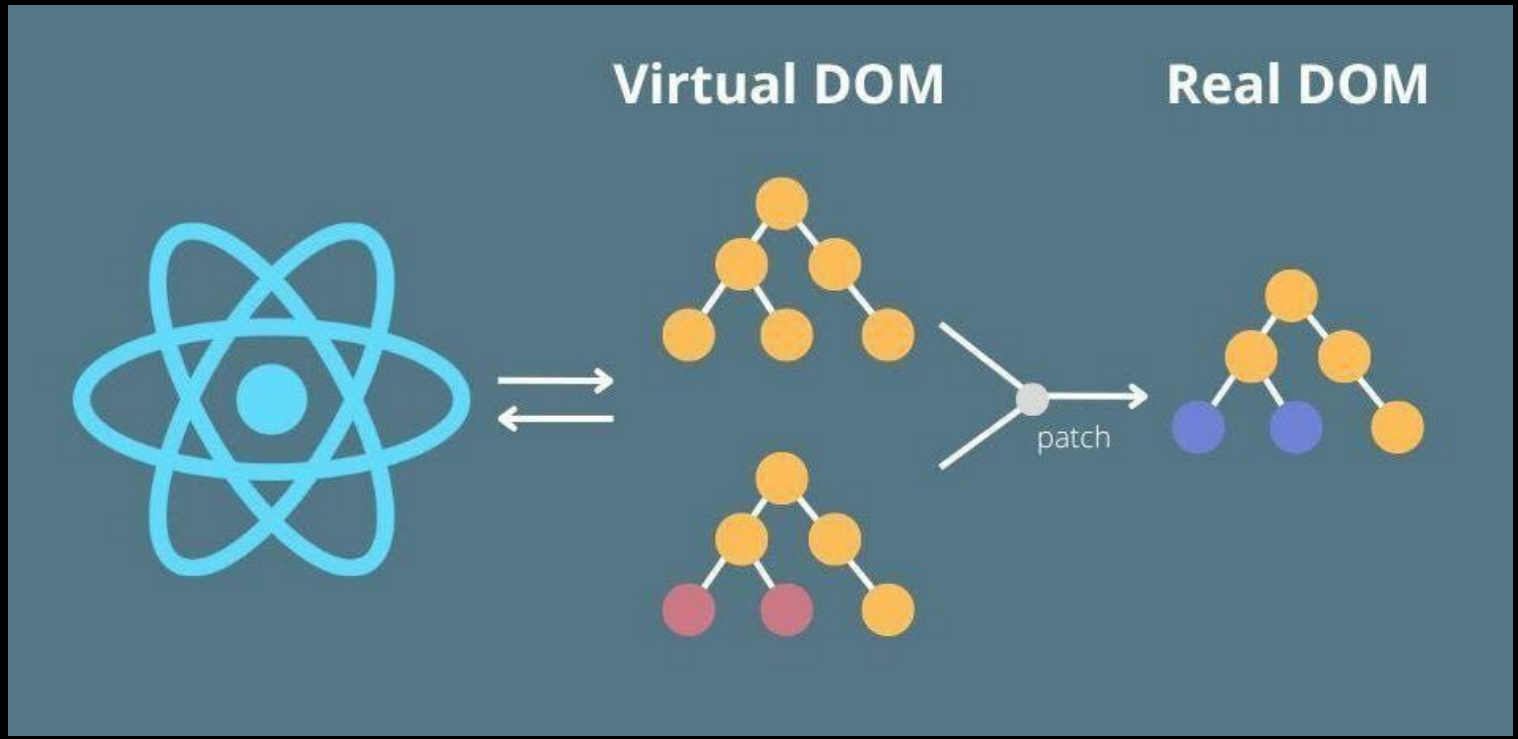
```
<button className="bg-blue-500 text-white font-bold py-2 px-4 rounded">
  Klick mich
</button>
```

tailwindcss

```
:root {
  --textFarbe: #da3ace;
}
#div1 {
  background-color: var(--textFarbe);
}
#div2 {
  background-color: #22aadd;
}
#div3 {
  background-color: var(--textFarbe);
}
h1 {
   background-color: var(--textFarbe);
}
div {
  width: 300px;
  height: 200px;
  border: 2px solid purple;
}
```

CSS

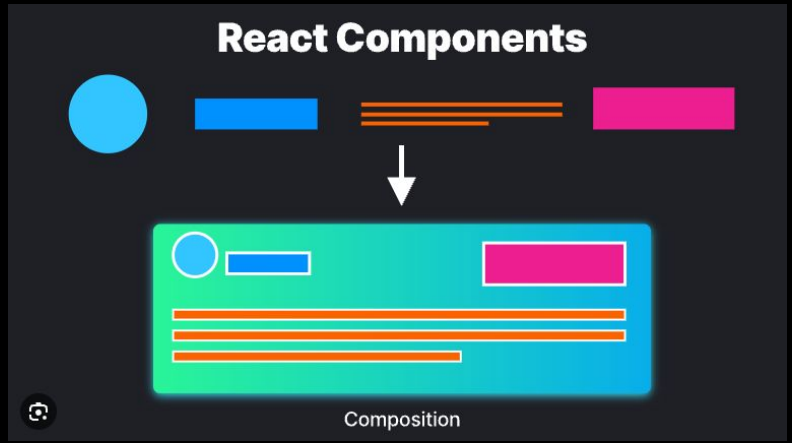
React ist schnell - dank des virtuellen DOM



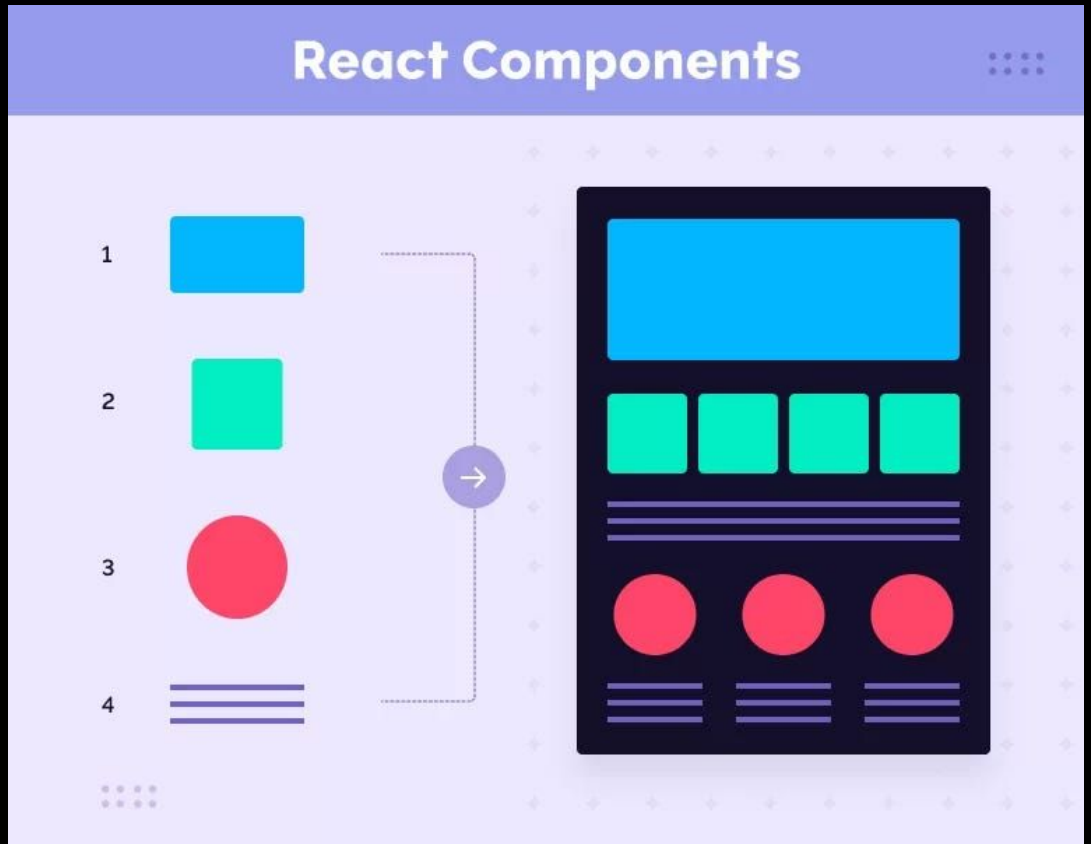
Komponenten

- Komponenten = wiederverwendbare Bausteine
- Funktionale Komponenten sind Standard

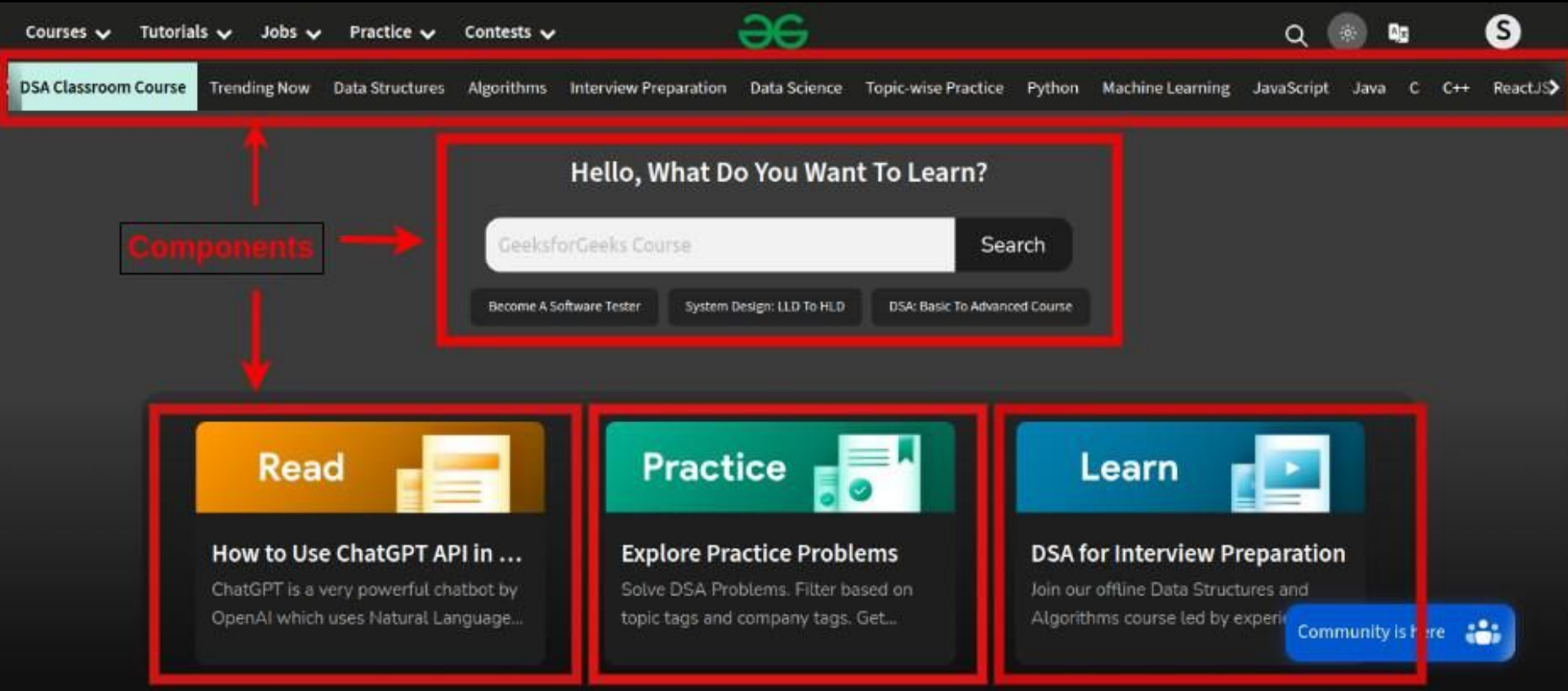
```jsx
function Hello({ name }) {
  return <h1>Hello, {name}!</h1>;
}
```



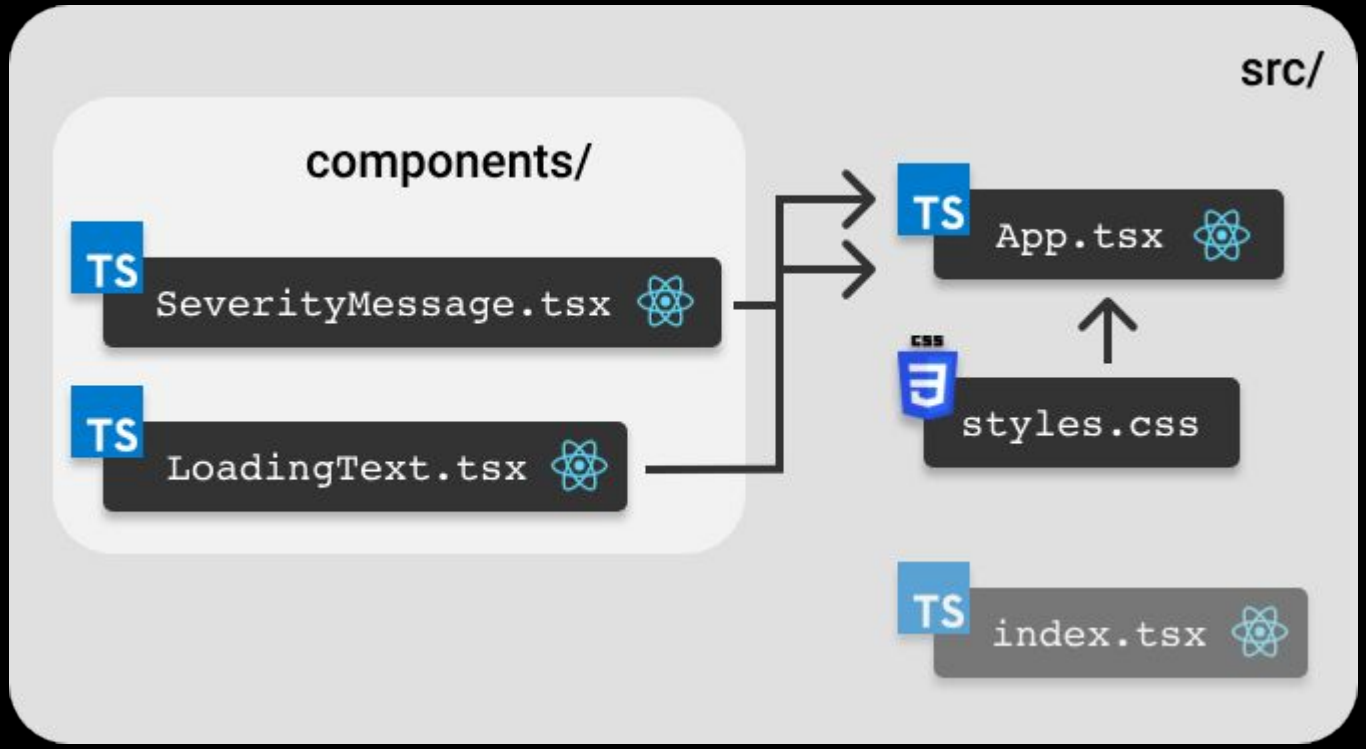
Komponenten



Komponenten



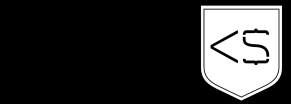
ReactJS Aufbau



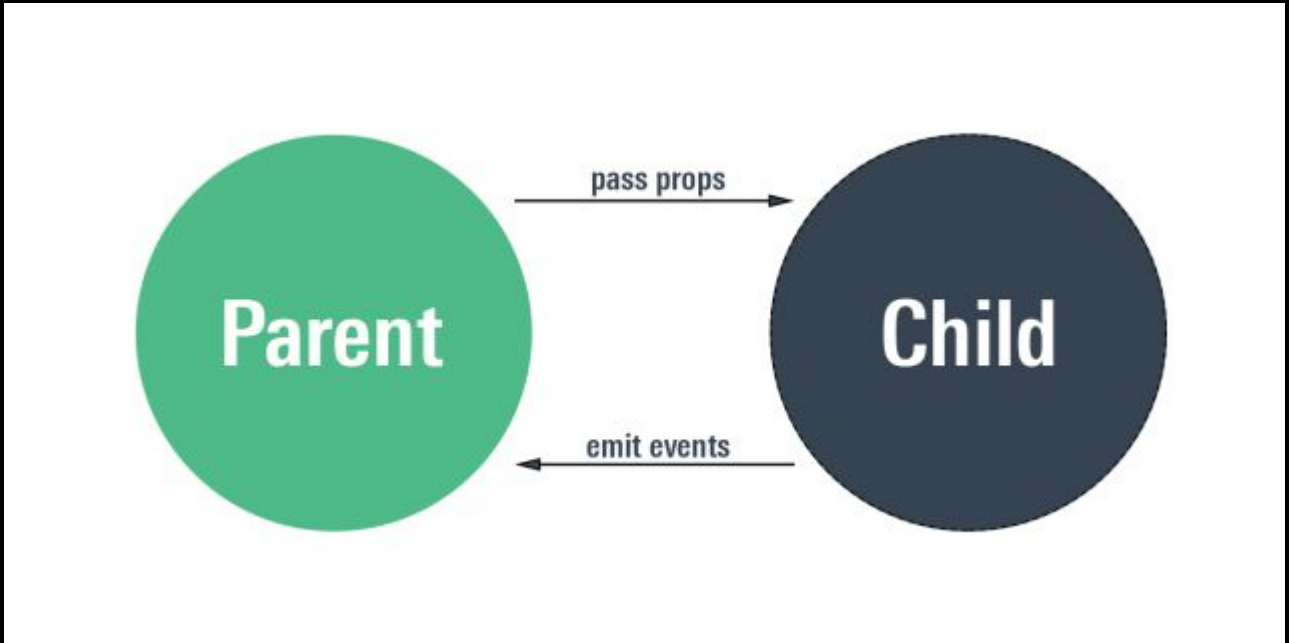
Props (Properties)

- Daten, die von außen in eine Komponente übergeben werden
- Sind read-only und unveränderlich in der Komponente
- Ermöglichen Wiederverwendung von Komponenten

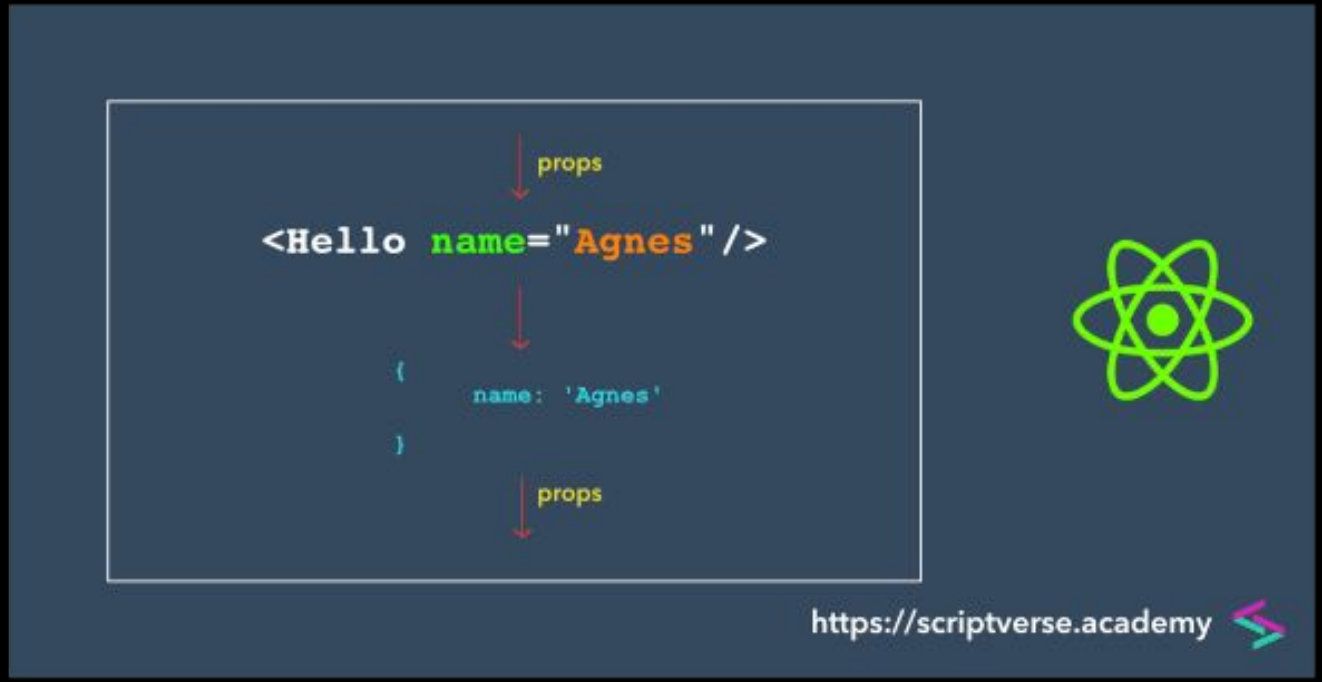
- Beispiel: Button mit Text als Prop



Props (Properties)



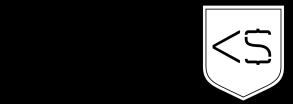
Props (Properties)



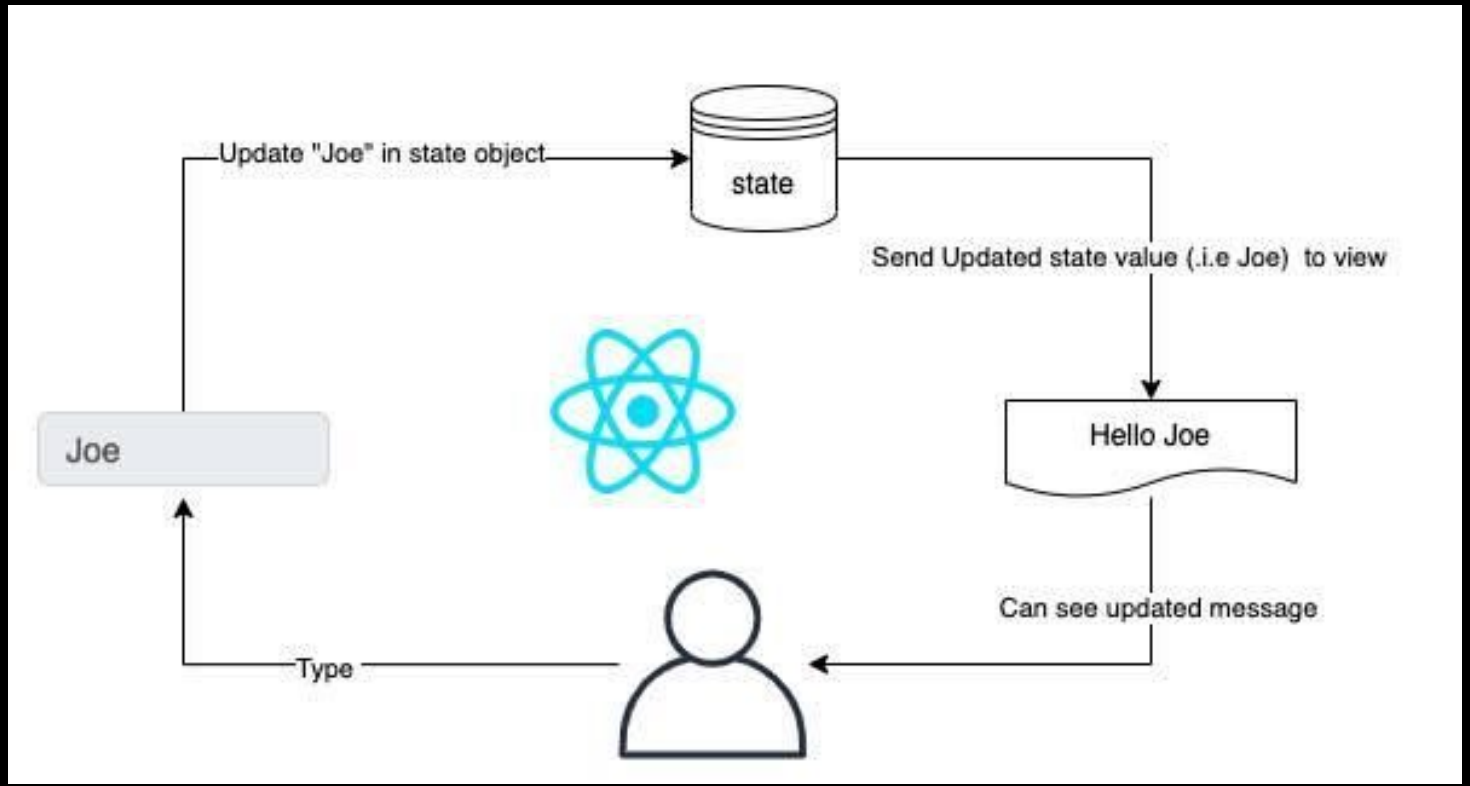
State (Zustand)

- Daten, die von außen in eine Komponente übergeben werden
- Verwaltet dynamische Daten innerhalb einer Komponente
- Ändert sich der State → UI rendert neu
- Nutzung über useState Hook

```
const [count, setCount] = useState(0);
<button onClick={() => setCount(count + 1)}>+1</button>
```



State (Zustand)

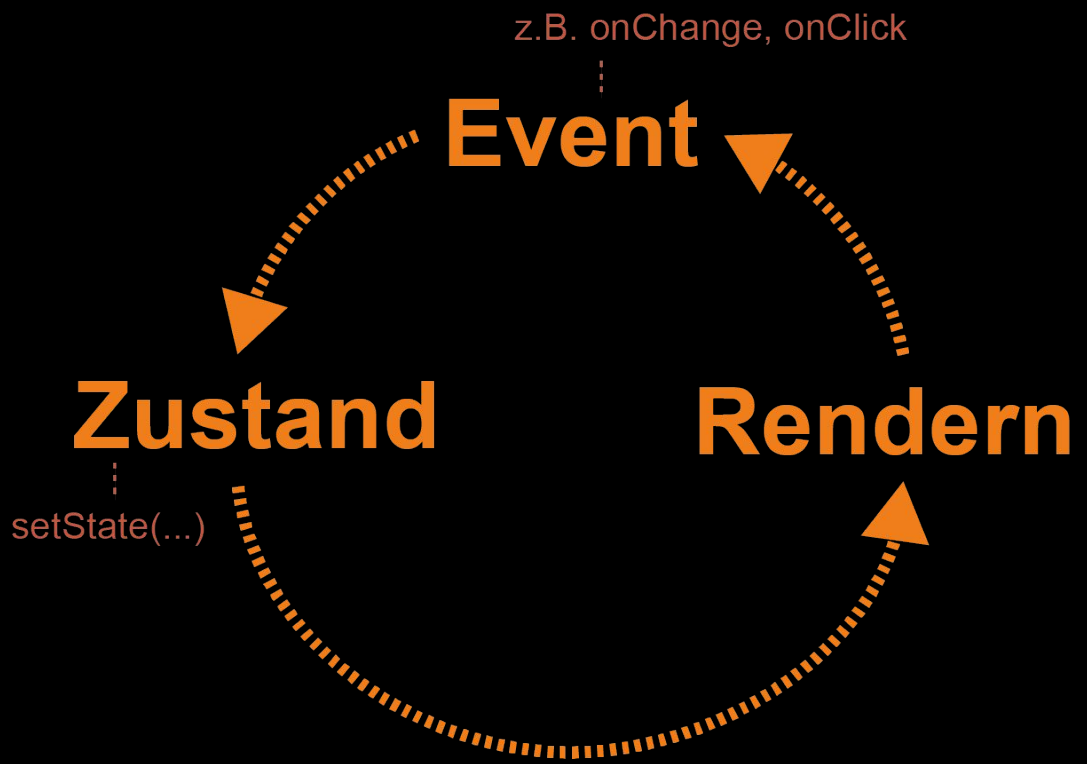


Der Render-Zyklus

- React erstellt ein virtuelles Abbild des DOM (Virtual DOM)
- Vergleicht neues mit altem Zustand (Diffing)
- Nur notwendige Änderungen werden ins echte DOM übertragen (Reconciliation)
- **Vorteil:** hohe Performance

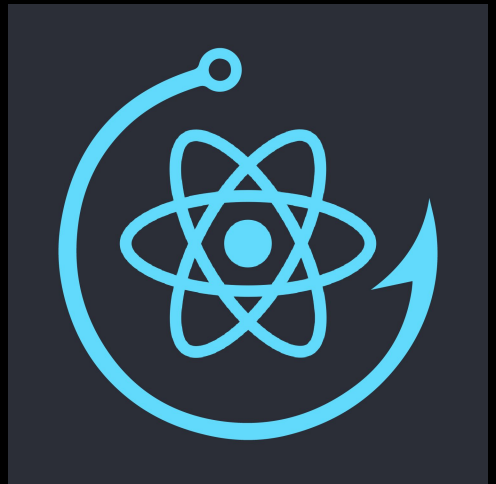
Der Render-Zyklus

-



Hooks – erste Einführung

- **useState:** für State-Management in Funktionen
- **useEffect:** für Nebenwirkungen (API Calls, Timer, Events)
- **Weitere Hooks:** useMemo, useCallback, useContext
- Vereinfachen komplexes State-Handling ohne Klassen



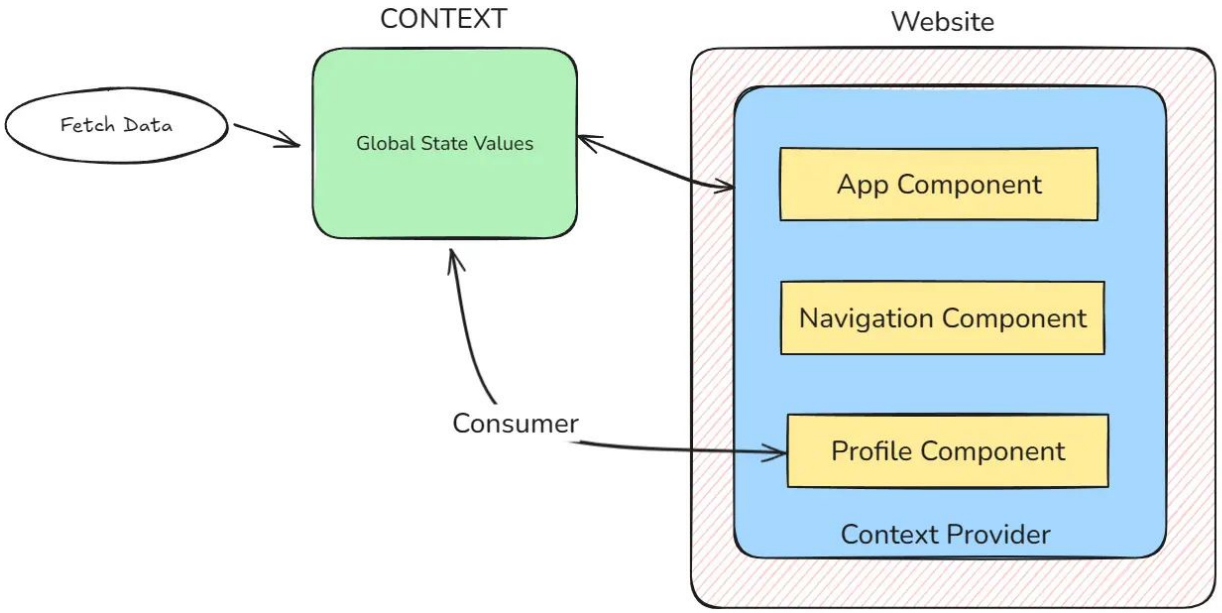
Controlled vs. Uncontrolled Components

- **Controlled:** Wert wird vom State gesteuert (Formulare, Inputs)
- **Uncontrolled:** Zugriff über Ref, kein direkter State-Zugriff
- **Vorteil Controlled:** bessere Kontrolle & Validierung

Context & Prop Drilling

- **Context API:** globale Werte wie Theme oder Auth teilen
- Vermeidet „Prop Drilling“ (Props durch viele Ebenen weiterreichen)
- useContext für Zugriff

Context & Prop Drilling



NPM

Was ist npm?

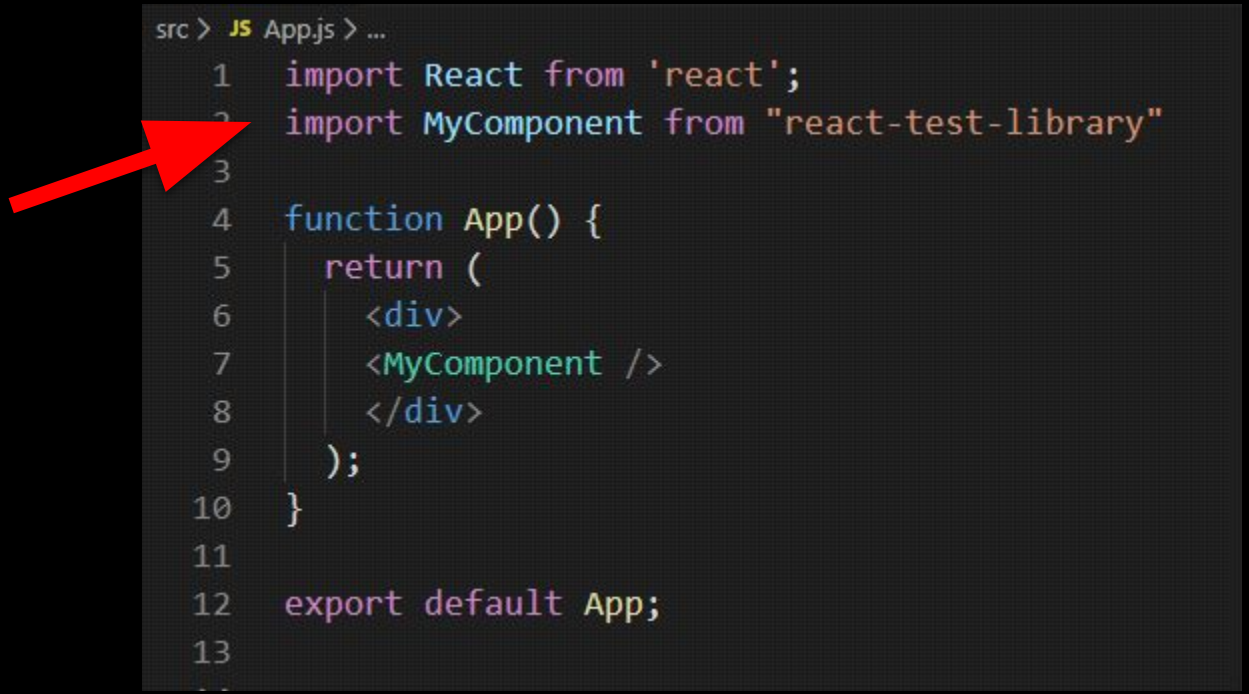
- Der größte Paketmanager für JavaScript
- Wird mit Node.js automatisch installiert

```
npm install react
```

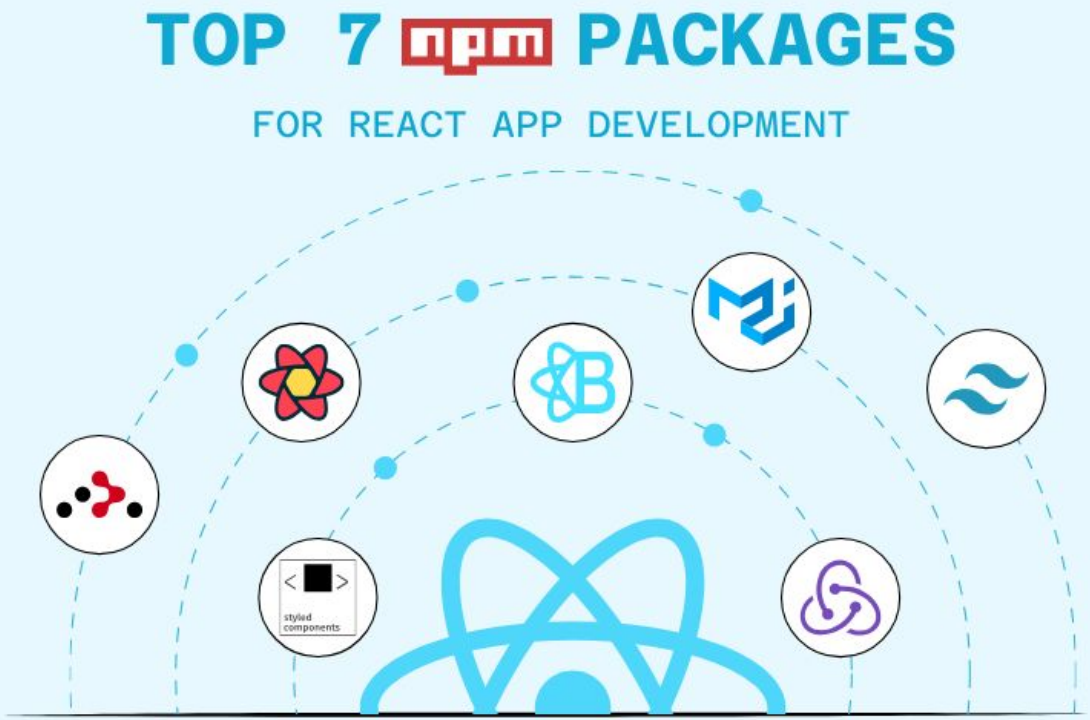
Wofür wird npm genutzt?

- Installieren von Bibliotheken & Frameworks (npm install ...)
  Verwalten von Abhängigkeiten im Projekt (package.json)
  Teilen von eigenen Paketen mit der Community

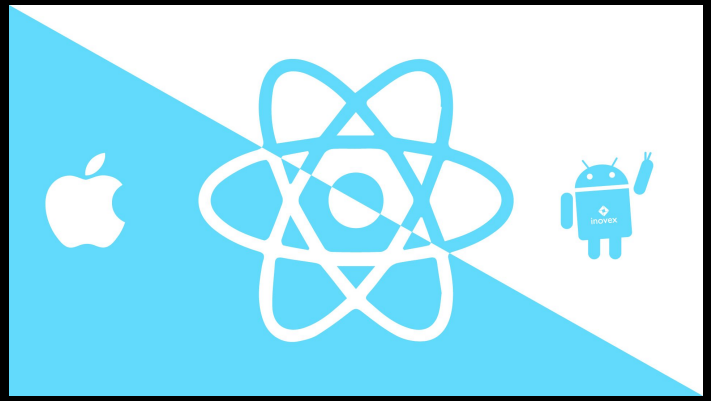
Dependency Injection



NPM

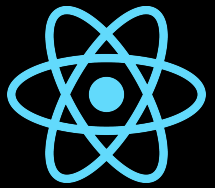


Nimblechapps



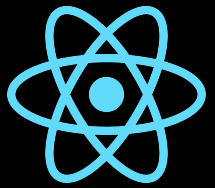
React Native Grundlagen

React Native – Entstehungsgeschichte



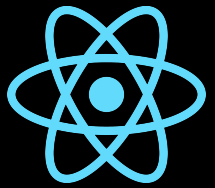
- **2013:** Hackathon bei Facebook – Idee: „React für Mobile“
- **2015:** Offizielle Open-Source-Veröffentlichung
- **Ziel:** iOS- und Android-Apps schneller entwickeln können
- **Heute:** große Community, sehr viele Firmen setzen RN produktiv ein

Grundprinzipien von React Native



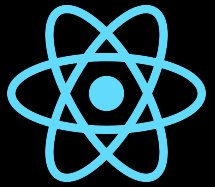
- „Learn once, write anywhere“
- Nutzt React-Konzepte (Komponenten, State, Hooks)
- Rendert native UI-Elemente, keine WebViews
- JS/TS Code wird über eine Bridge mit dem Betriebssystem verbunden

React Native vs. Hybrid-Ansätze

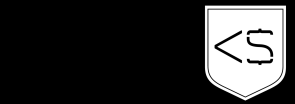


- **Hybrid (Cordova, Ionic):** läuft in einer WebView, UI ≈ HTML/CSS
- **React Native:** echte native Views, bessere Performance
- **Vorteil RN:** fühlt sich an wie eine „echte App“
- Native Module können jederzeit eingebunden werden

Die React Native Bridge

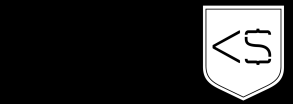


- Übersetzt JavaScript-Befehle in native API-Aufrufe
- **Umgekehrt:** native Events (z. B. Sensor-Daten) gehen zurück in JS
- **Performance:** gut, solange keine extrem vielen Calls pro Sekunde
- **Neue Architektur mit JSI:** direkter Zugriff auf native Funktionen ohne klassischen Bridge-Overhead

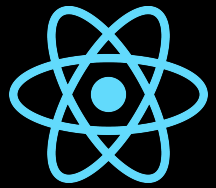


React Native – Neue Architektur

- **Fabric:** neues Rendering-System, ersetzt den alten Bridge-Ansatz → schneller & stabiler
- **TurboModules:** effizienterer Zugriff auf native APIs
- **Hermes:** performante JS-Engine optimiert für Mobile
- **Ziel:** RN noch näher an die Performance nativer Apps bringen



Unterschiede zu klassischem React (Web)



- Kein DOM → stattdessen: View, Text, Image
- Kein CSS → Styling mit JavaScript-Objekten (mittlerweile auch Tailwind Support!)
- **Standard-Flexrichtung:** column (im Web: row)
- **Navigation:** react-navigation statt React Router
- Medien & Assets anders eingebunden (require oder asset-linking)

Plattformunterschiede (iOS vs. Android)

Unterschiedliche Design Guidelines:

- iOS → Human Interface Guidelines
- Android → Material Design

Unterschiedliche Systemkomponenten (Dateipicker, Push-Handling etc.)

- Unterschiede bei Permissions (z. B. Kamera, Standort)
- Unterschiede in Deployment (App Store Review vs. Play Store)

React Native – Wichtige Bibliotheken

Navigation:

- `react-navigation` → Screens & Tab-Navigation

UI-Komponenten:

- `react-native-paper` → Material Design Komponenten
- `react-native-elements` → vorgefertigte UI-Bausteine

Animationen:

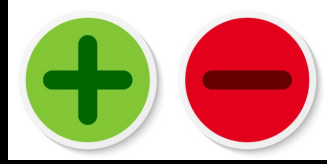
- `react-native-reanimated` → performante Animationen
- `react-native-gesture-handler` → Gestensteuerung

Daten & APIs:

- `axios` oder `fetch` → API-Anfragen
- `react-query` → State Management für Daten

Sonstiges:

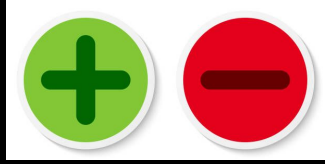
- `expo` → erleichtert Entwicklung & Deployment
- `react-native-vector-icons` → Iconsammlung



Vorteile von React Native

- Schneller Einstieg für Web-Entwickler:innen (JS/TS)
- Gemeinsamer Code für verschiedene Plattformen
- Große Community & Ökosystem
- OTA-Updates (Over the Air) möglich (über Expo/EAS)
- Viele Bibliotheken (z. B. für Navigation, Animation, Formulare)

Nachteile von React Native



- Performance bei extrem grafikintensiven Apps (z. B. High-End-Spiele) eingeschränkt
- Abhängigkeit von Drittanbieter-Bibliotheken (kann Pflegeprobleme geben)
- Manchmal notwendig, native Module selbst zu schreiben (Swift/Kotlin)
- Komplexität bei sehr plattform-spezifischen Anforderungen

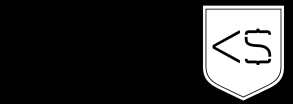
Wann eignet sich React Native?

✅ Gute Wahl:

- Business-Apps (z. B. E-Commerce, Social, Tools)
- MVPs und schnelle Prototypen
- Projekte mit Web-Teams, die Mobile-Know-how brauchen

❌ Weniger geeignet:

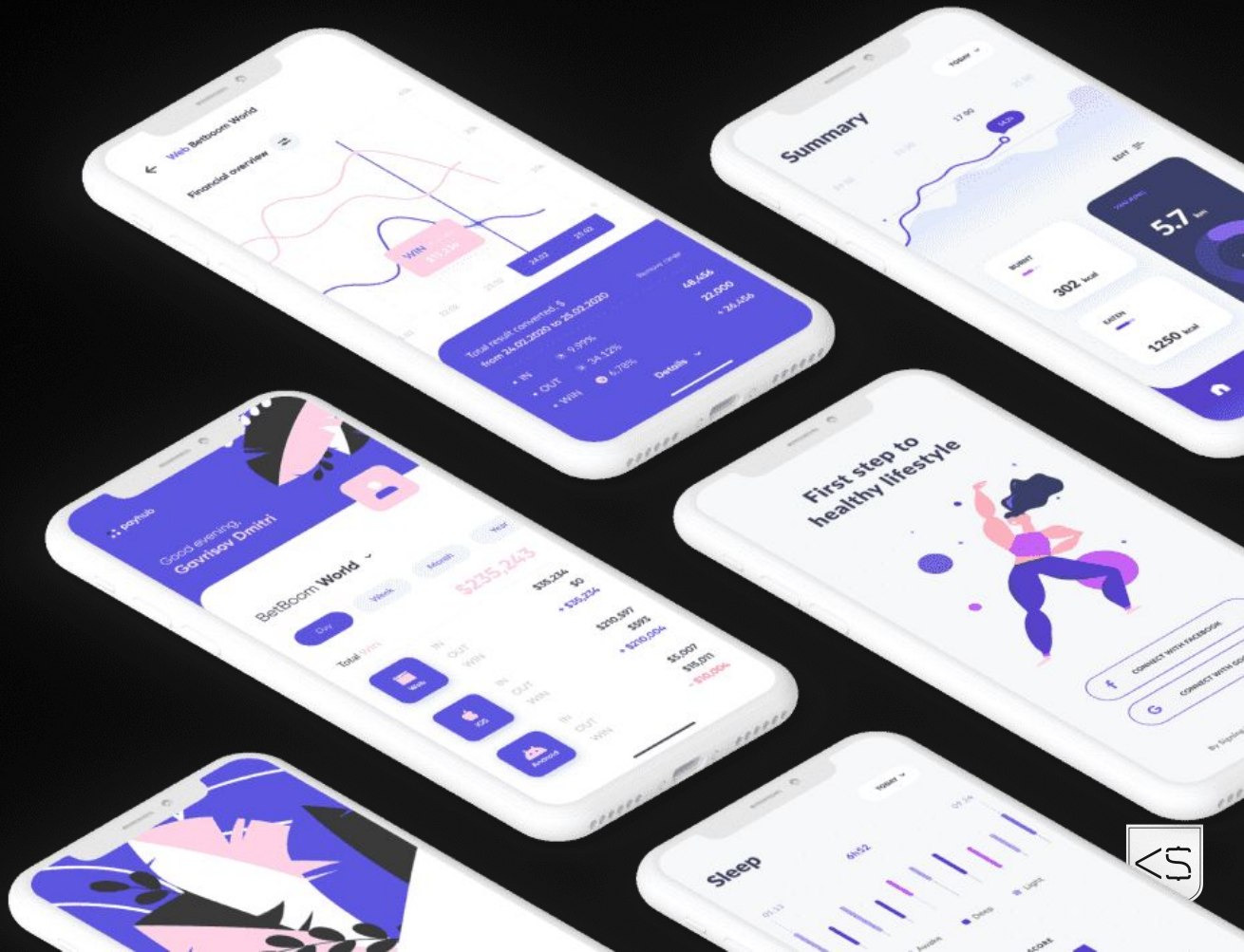
- High-End-Gaming oder AR/VR-Apps
- Apps mit extrem plattformspezifischem UI
  Hardware-nahe Anwendungen (z. B. spezielle Sensorik)



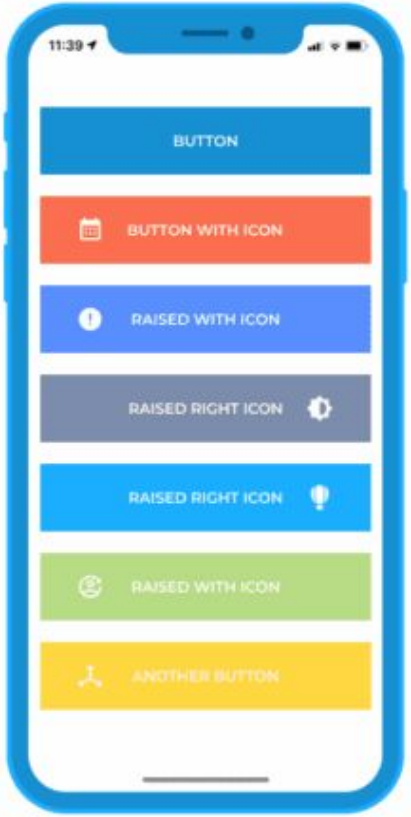
React Native im Vergleich zu nativ (iOS/Android)

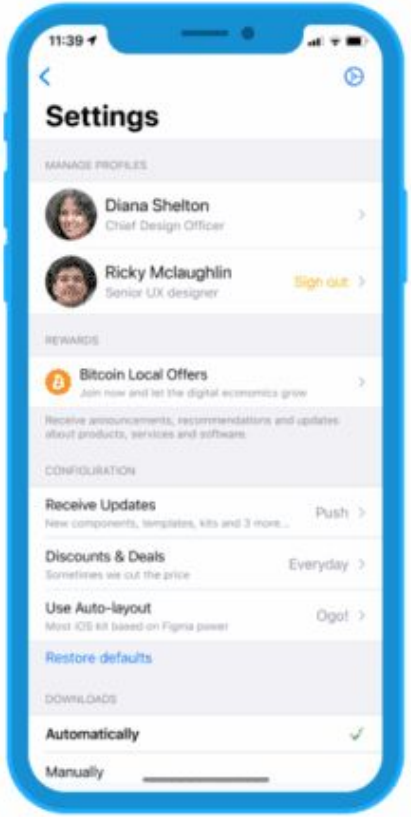
- **Entwicklungssprache:**
  - iOS: Swift/Objective-C
  - Android: Kotlin/Java
  - RN: JavaScript/TypeScript
- **UI:**
  - Nativ: UIKit/SwiftUI, Jetpack Compose
  - RN: React-Komponenten → native Views
- **Performance:**
  - Nativ: Beste Performance
  - RN: Nahe an nativ, ausreichend für die meisten Apps
- **Team-Skills:**
  - RN: Web-Know-how nutzbar
  - Nativ: tieferes Plattformwissen nötig

UI Libraries



REACT NATIVE ELEMENTS






Material UI

paper

Need a customized UI kit for Android & iOS? Get started

Making your React Native apps look and feel native

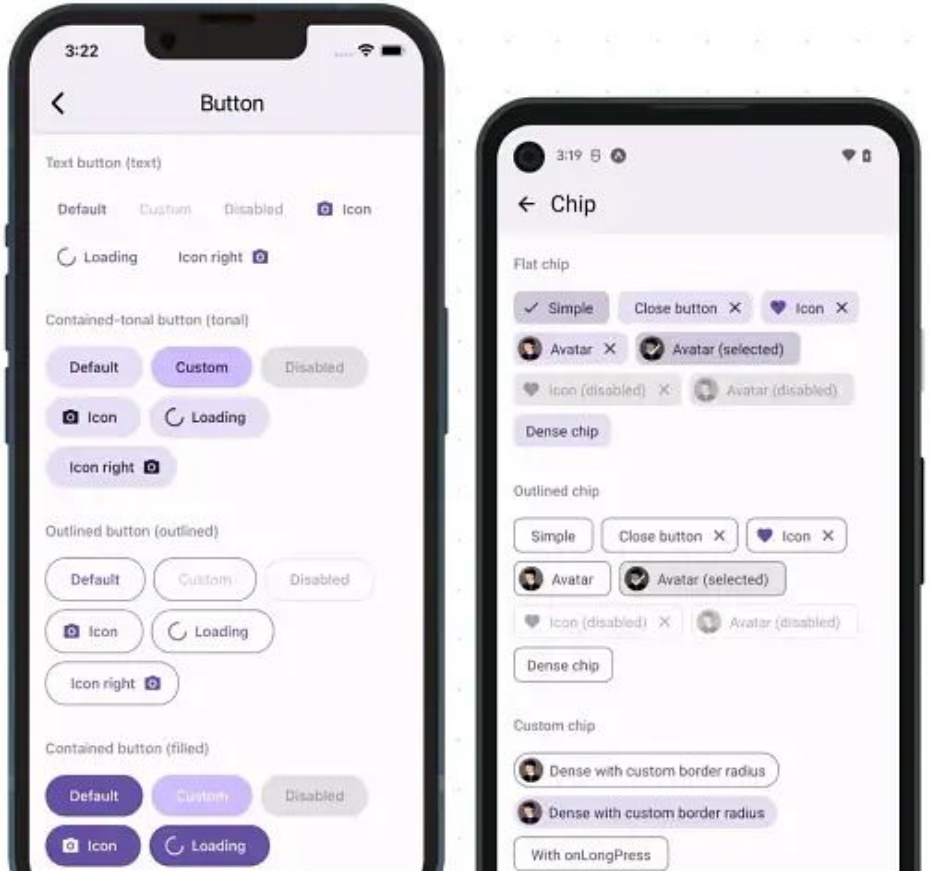
★ Star 12720

React Native Paper is a high-quality, standard-compliant Material Design library that has you covered in all major use-cases.

Learn more | Docs

Try out components in our demo apps:

GET IT ON Google Play | Download on the App Store

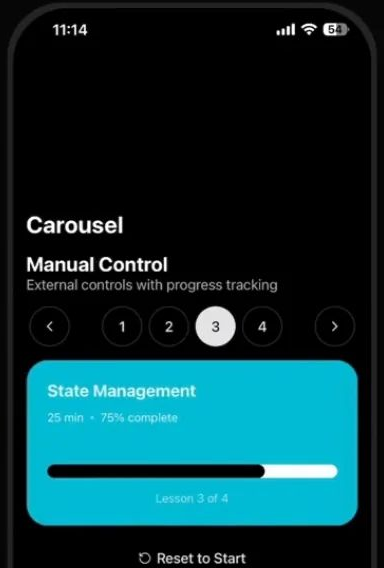


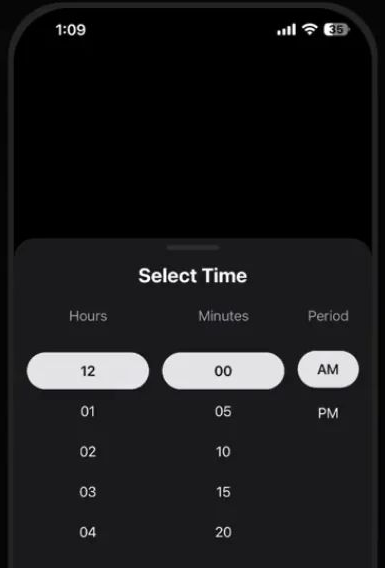
BNA UI

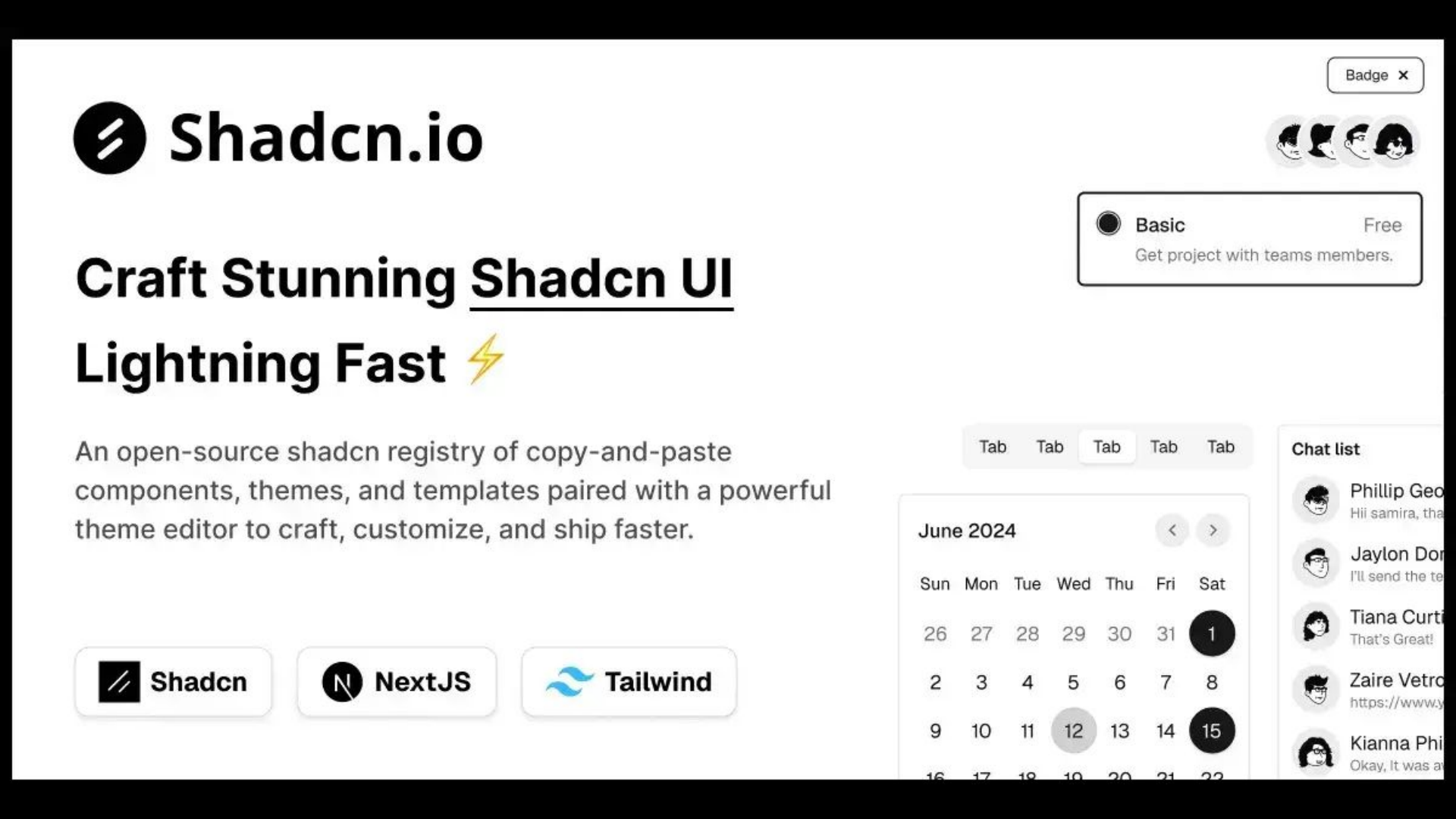
Build your Component Library

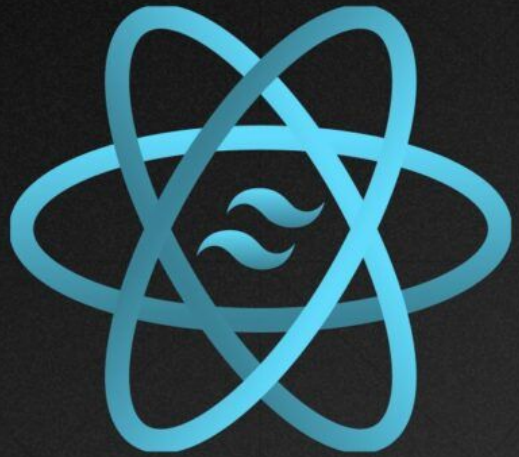
Beautifully-designed, accessible React Native components.
Built for Expo and React Native. Open Source. Open Code.

Get Started






Shadcn.io
Badge ×
Basic
Free
Get project with teams members.
Craft Stunning Shadcn UI
Lightning Fast ⚡
An open-source shadcn registry of copy-and-paste components, themes, and templates paired with a powerful theme editor to craft, customize, and ship faster.
Tab Tab Tab Tab Tab
June 2024
Sun Mon Tue Wed Thu Fri Sat
Chat list
Phillip Geo
Jaylon Dor
Tiana Curti
That's Great!
Zaire Vetro
Kianna Phi
Okay, It was a
Shadcn
NextJS
Tailwind



NativeWind

iOS Apps by Microsoft that Use React Native

Based on Appfigures SDK Intelligence


Bing Search


Microsoft OneDrive


Microsoft Outlook


Xbox


Microsoft Word


Microsoft OneNote


Microsoft Excel


Mixer – Interactive Streaming


Microsoft SharePoint


Microsoft Teams


Cortana


Microsoft Edge


Office Delve – for Office 365


Face Swap – Automatic and Funny


Microsoft Visio Viewer


Dynamics 365 for phones


PowerApps


MS Executive Industry Summit

Popular Apps Built with React Native

Instagram

Facebook

Pinterest

Discord

Shopify

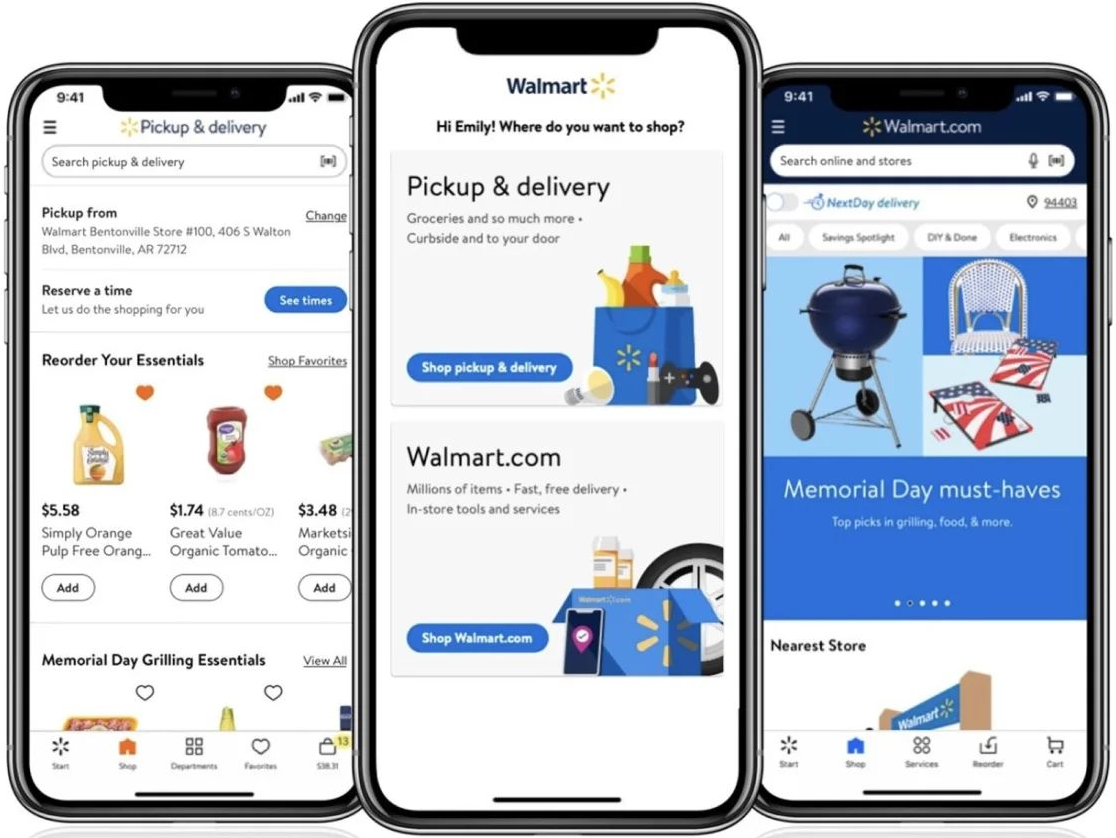
Uber Eats

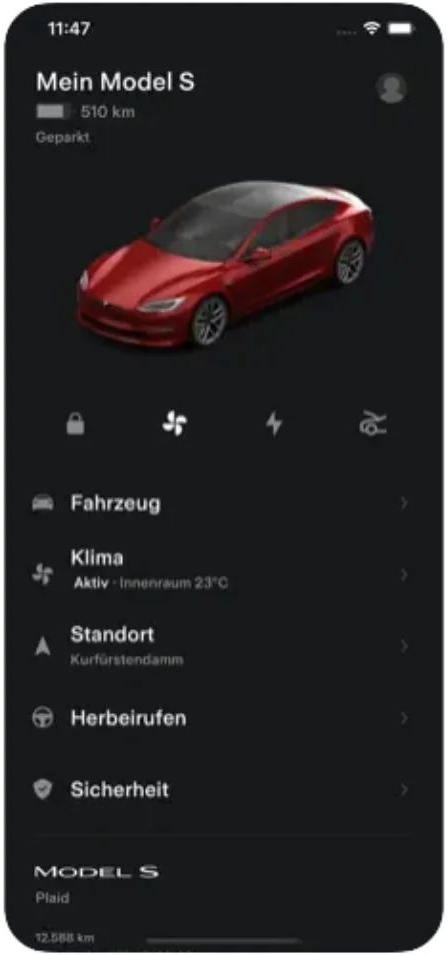
Uber Eats

Bloomberg

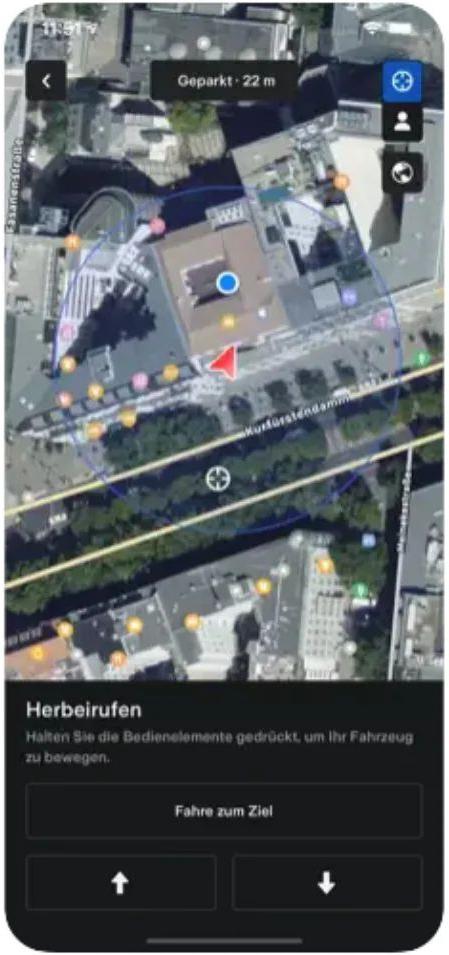
Bloomberg

BRILWORKS

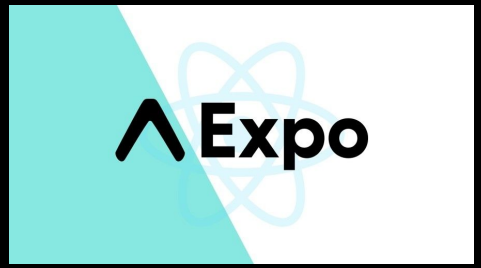

9:41
Pickup & delivery
Search pickup & delivery
Pickup from
Walmart Bentonville Store #100, 406 S Walton Blvd, Bentonville, AR 72712
Change
Reserve a time
Let us do the shopping for you
See times
Reorder Your Essentials
Shop Favorites
$5.58
Simply Orange Pulp Free Orang...
$1.74 (8.7 cents/OZ)
Great Value Organic Tomato...
$3.48
Add
Add
Add
Memorial Day Grilling Essentials
View All
Start
Shop
Departments
Favorites
$38.31
Walmart
Hi Emily! Where do you want to shop?
Pickup & delivery
Groceries and so much more •
Curbside and to your door
Shop pickup & delivery
Walmart.com
Millions of items • Fast, free delivery •
In-store tools and services
Shop Walmart.com
9:41
Walmart.com
Search online and stores
NextDay delivery
94403
All
Savings Spotlight
DIY & Done
Electronics
Memorial Day must-haves
Top picks in grilling, food, & more.
Nearest Store
Start
Shop
Services
Reorder
Cart


11:47
Mein Model S
510 km
Geparkt
Fahrzeug
Klima
Aktiv · Innenraum 23°C
Standort
Kurfürstendamm
Herbeirufen
Sicherheit
MODEL S
Plaid
12.588 km

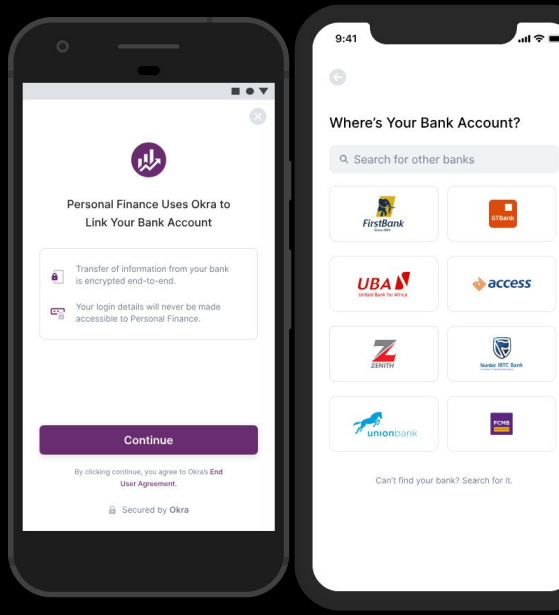

11:46
Innenraum 23°C · Außen 19°C
Aktiviert
20.0°
Spalt


Geparkt · 22 m
Herbeirufen
Halten Sie die Bedienelemente gedrückt, um Ihr Fahrzeug zu bewegen.
Fahre zum Ziel

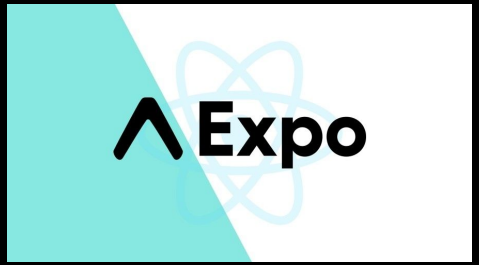

12:48
Starbase
Wird entladen
6.9 kW
Starship
5.4 kW
SOLAR
1.2 kW
HAUS
4.7 kW · 97%
POWERWALL
2.0 kW
STROMNETZ
Energie
70.9 kWh heute erzeugt
Ergebnis
Heute 98% Eigenversorgung
Vom Stromnetz trennen
Virtuelles Kraftwerk
Einstellungen


Expo

Expo

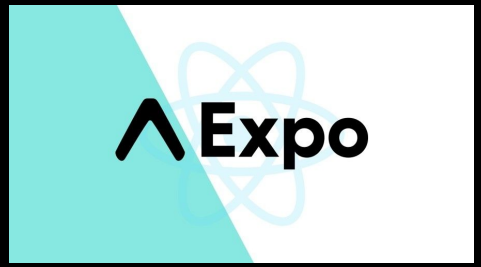

Personal Finance Uses Okra to Link Your Bank Account
Transfer of information from your bank is encrypted end-to-end.
Your login details will never be made accessible to Personal Finance.
Continue
By clicking continue, you agree to Okra's End User Agreement.
Secured by Okra
9:41
Where's Your Bank Account?
Search for other banks
FirstBank
GTBank
UBA
access
ZENITH
unionbank
Can't find your bank? Search for it.

Was ist Expo?



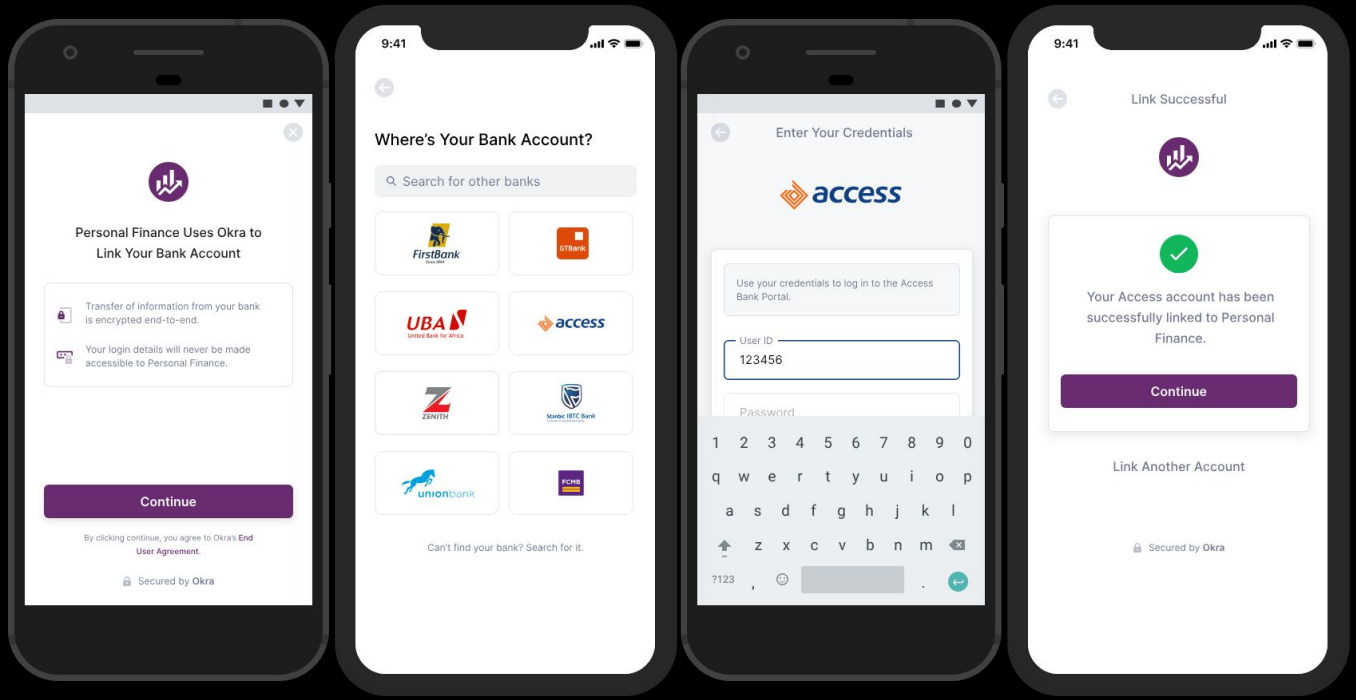
- Open-Source-Framework & Toolchain für React Native
- Baut auf React Native auf, bietet aber viele vorkonfigurierte Funktionen
- Enthält SDK mit nativen APIs: Kamera, Push-Notifications, Location, Haptics u. v. m.
- Expo Go App: Code auf echtem Gerät testen – ohne Xcode/Android Studio
- EAS (Expo Application Services): Cloud-Builds, Updates „Over the Air“, einfache Veröffentlichung

Warum Expo?



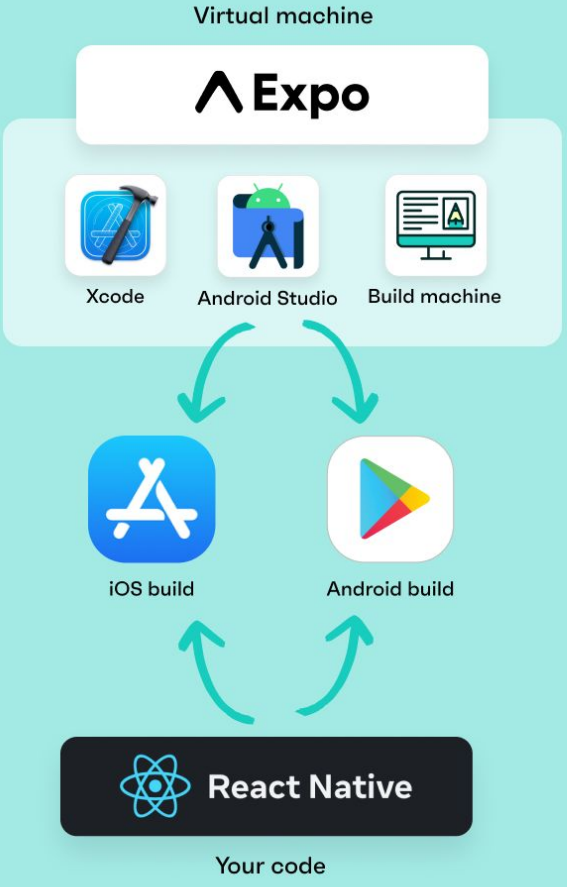
- Open-Source-Framework & Toolchain für React Native
- Schneller Start: kein kompliziertes Setup mit Xcode/Android Studio nötig
- Einfaches Testing: QR-Code scannen → App läuft sofort auf Smartphone
- Großes SDK: viele Funktionen sofort nutzbar, ohne eigene native Module
- Updates ohne App Store: kleine Änderungen direkt per OTA-Update ausliefern
- Ideal für Workshops, Prototypen und MVPs → Fokus liegt auf Lernen & App-Logik, nicht auf komplizierter Konfiguration

Expo

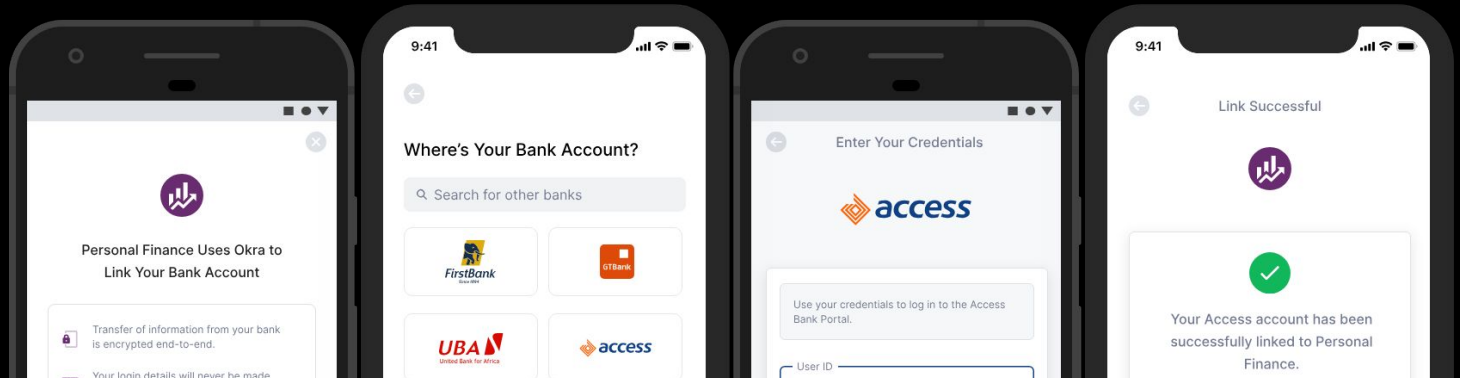

Personal Finance Uses Okra to Link Your Bank Account
Transfer of information from your bank is encrypted end-to-end.
Your login details will never be made accessible to Personal Finance.
Continue
By clicking continue, you agree to Okra's End User Agreement.
Secured by Okra
9:41
Where's Your Bank Account?
Search for other banks
FirstBank
GTBank
UBA
access
ZENITH
Stanbic IBTC Bank
unionbank
FCMB
Can't find your bank? Search for it.
Enter Your Credentials
access
Use your credentials to log in to the Access Bank Portal.
User ID
123456
Password
9:41
Link Successful
Your Access account has been successfully linked to Personal Finance.
Continue
Link Another Account
Secured by Okra

BUILD THE APP ON YOUR MACHINE USING REACT NATIVE

USING EXPO TO BUILD THE APP



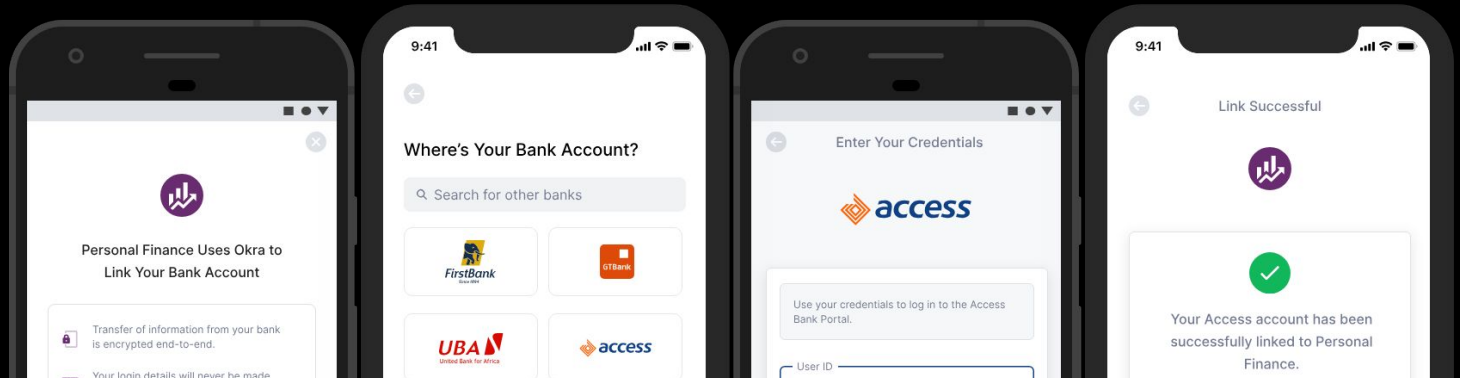
React Native Installation



React Native Installation

```
npx create-expo-app@latest —-template
```

Praxis


Personal Finance Uses Okra to Link Your Bank Account
Transfer of information from your bank is encrypted end-to-end.
9:41
Where's Your Bank Account?
Search for other banks
FirstBank
GTBank
UBA
access
Enter Your Credentials
access
Use your credentials to log in to the Access Bank Portal.
User ID
9:41
Link Successful
Your Access account has been successfully linked to Personal Finance.

Praxis Aufgaben

Erste Komponente anlegen:
Eine Welcome-Komponente schreiben, die “Hallo React Native” zurückgibt.

Props nutzen:
Eine Greeting-Komponente, die den Namen als Prop annimmt → Hello Christoph.

State einbauen:
Button, der bei jedem Klick den Zähler erhöht (useState).

Styles anwenden:
Text in Blau darstellen oder einen Button mit Hintergrundfarbe stylen.

Flexbox-Layout:
Zwei Boxen nebeneinander oder untereinander anordnen.

Input-Feld & Text:
Ein TextInput, in das man etwas eintippt → darunter erscheint der eingegebene Text.

Liste anzeigen:
Mit FlatList eine kleine Liste (z. B. Namen oder To-dos) darstellen.

Wichtige Links

- https://legacy.reactjs.org/
- https://reactnative.dev/
- https://nodejs.org/en
- https://expo.dev/
- https://www.npmjs.com/
- https://www.nativewind.dev/