

Programowanie Aplikacyjne (PAP)

Co nowego w programowaniu Web

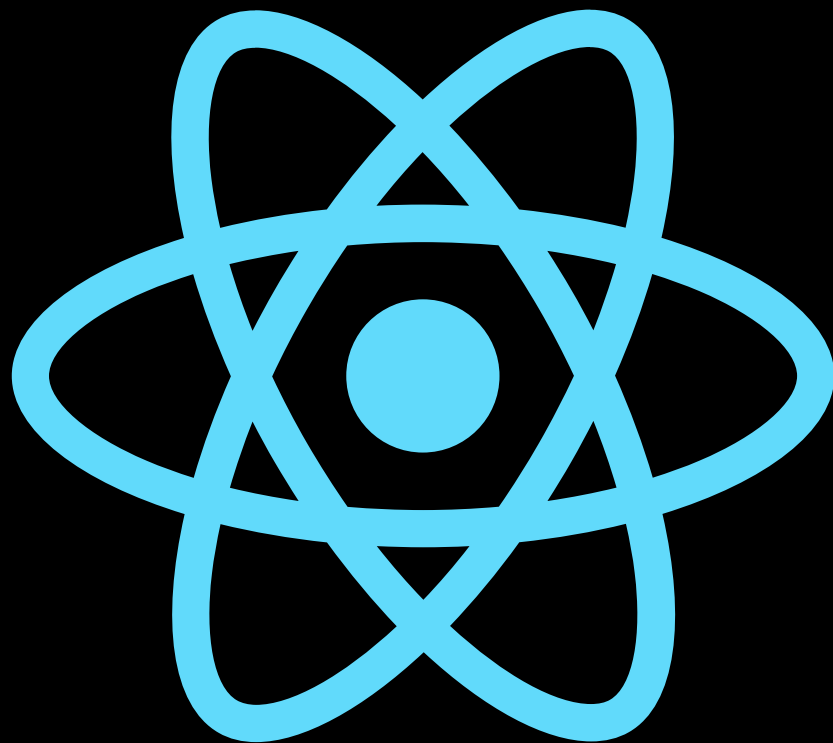
Julian Myrcha

Institute of Computer Science

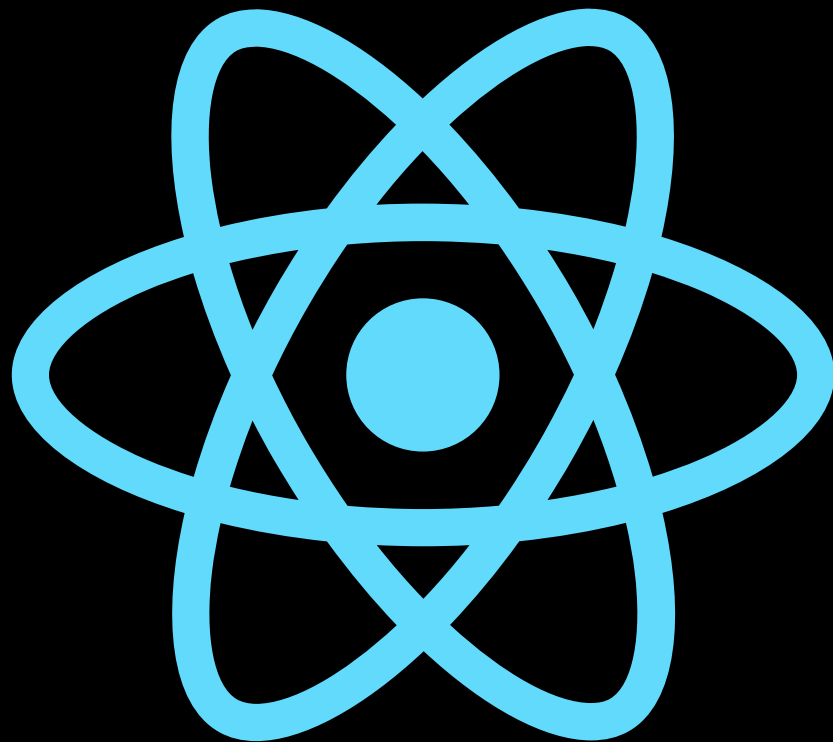
26.02.2025



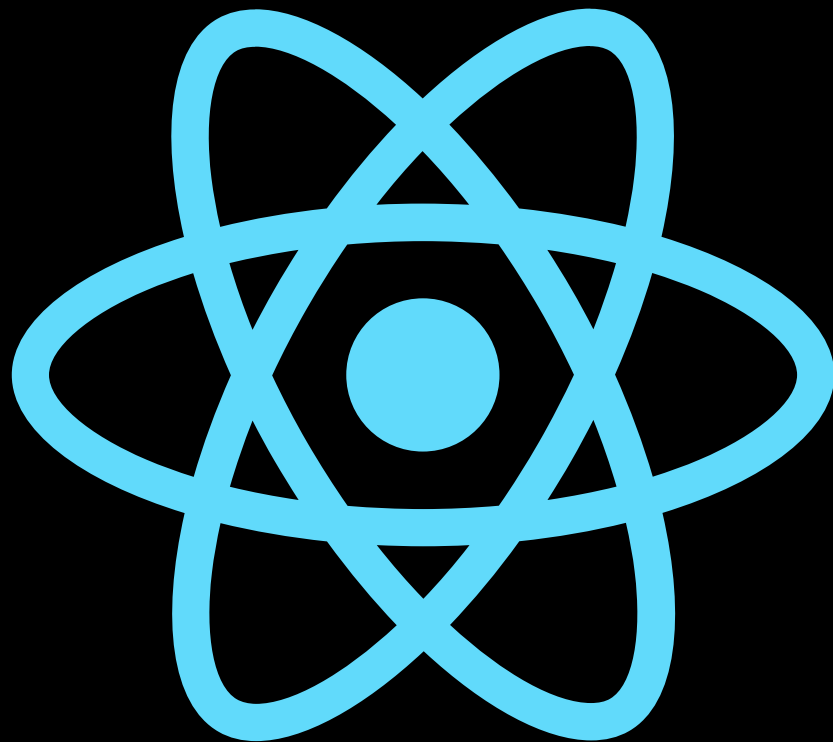
- React to biblioteka do tworzenia webowych interfejsów użytkownika, stanowi widok dla aplikacji webowych



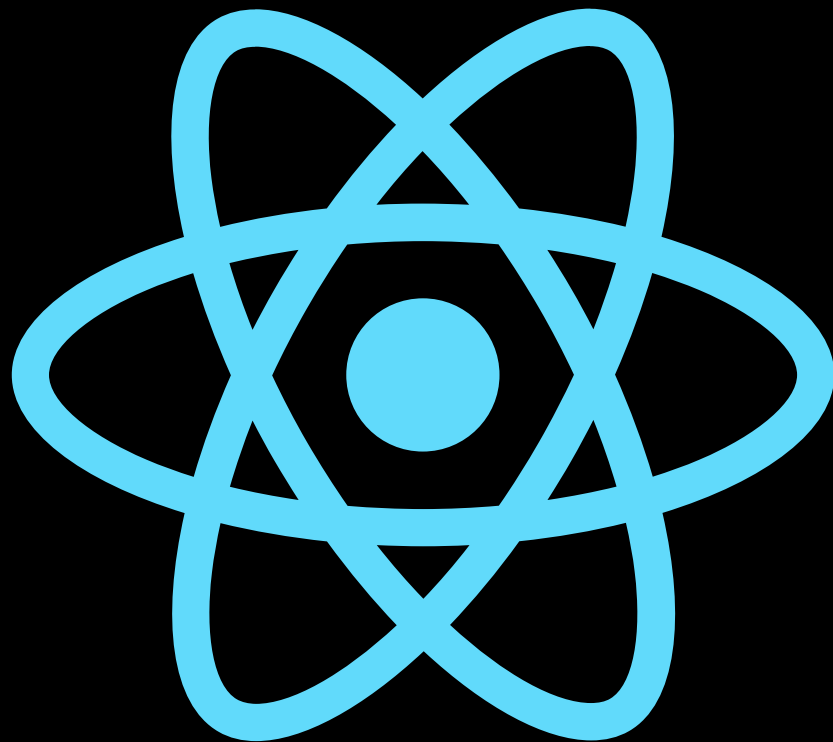
- React to biblioteka do tworzenia webowych interfejsów użytkownika, stanowi widok dla aplikacji webowych
- Budowa interfejsu polega na składaniu go z tworzonych przez programistę komponentów



- React to biblioteka do tworzenia webowych interfejsów użytkownika, stanowi widok dla aplikacji webowych
- Budowa interfejsu polega na składaniu go z tworzonych przez programistę komponentów
- Operuje na wirtualnym DOM, przechowywanym w pamięci.



- React to biblioteka do tworzenia webowych interfejsów użytkownika, stanowi widok dla aplikacji webowych
- Budowa interfejsu polega na składaniu go z tworzonych przez programistę komponentów
- Operuje na wirtualnym DOM, przechowywanym w pamięci.
- Po każdej zmianie tego wirtualnego DOM, React ustala jakie zmiany należy dokonać w rzeczywistym DOM i dokonuje ich.



```
1 <html>
2 <head>
3   <meta charset="utf-8">
4   <title>Hello world</title>
5   <script src="https://cdnjs.cloudflare.com/ajax/libs/react/16.10.2/umd/react.production.min.js"></script>
6   <script src="https://cdnjs.cloudflare.com/ajax/libs/react-dom/16.10.2/umd/react-dom.production.min.js"></script>
7   <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-standalone/6.26.0/babel.min.js"></script>
8 </head>
9 <body>
10   <div id="app"></div>
11   <script type="text/babel">
12     ReactDOM.render(<h1>Hello world</h1>,document.querySelector('#app'));
13   </script>
14 </body>
15 </html>
```

- najprostszy przykład w React

```
1 <html>
2 <head>
3   <meta charset="utf-8">
4   <title>Hello world</title>
5   <script src="https://cdnjs.cloudflare.com/ajax/libs/react/16.10.2/umd/react.production.min.js"></script>
6   <script src="https://cdnjs.cloudflare.com/ajax/libs/react-dom/16.10.2/umd/react-dom.production.min.js"></script>
7   <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-standalone/6.26.0/babel.min.js"></script>
8 </head>
9 <body>
10   <div id="app"></div>
11   <script type="text/babel">
12     ReactDOM.render(<h1>Hello world</h1>,document.querySelector('#app'));
13   </script>
14 </body>
15 </html>
```

- najprostszy przykład w React
- to nie JavaScript: "text/babel" oraz h1Hello world/h1

```
1 <html>
2 <head>
3   <meta charset="utf-8">
4   <title>Hello world</title>
5   <script src="https://cdnjs.cloudflare.com/ajax/libs/react/16.10.2/umd/react.production.min.js"></script>
6   <script src="https://cdnjs.cloudflare.com/ajax/libs/react-dom/16.10.2/umd/react-dom.production.min.js"></script>
7   <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-standalone/6.26.0/babel.min.js"></script>
8 </head>
9 <body>
10   <div id="app"></div>
11   <script type="text/babel">
12     ReactDOM.render(<h1>Hello world</h1>, document.querySelector('#app'));
13   </script>
14 </body>
15 </html>
```

- najprostszy przykład w React
- to nie JavaScript: `"text/babel"` oraz `h1Hello world/h1`
- `ReactDOM.render(what, where)` zastępuje `elementDOM` `where` przez zawartość stworzoną w `what`

JSX - JavaScript eXtension - ułatwia tworzenie HTML-a za pomocą JavaScriptu

```
1 class HelloWorld extends React.Component {
2   render() {
3     return (
4       <h1 className='large'>Hello World</h1>
5     );
6   }
7 }
```

```
1 class HelloWorld extends React.Component {
2   render() {
3     return (
4       React.createElement(
5         'h1',
6         {className: 'large'},
7         'Hello World'
8       )
9     );
10  }
11 }
```

JSX - JavaScript eXtension - ułatwia tworzenie HTML-a za pomocą JavaScriptu

ES5 -wersja ECMAScriptu od 2009

```
1 class HelloWorld extends React.Component {
2   render() {
3     return (
4       <h1 className='large'>Hello World</h1>
5     );
6   }
7 }
```

```
1 class HelloWorld extends React.Component {
2   render() {
3     return (
4       React.createElement(
5         'h1',
6         {className: 'large'},
7         'Hello World'
8       )
9     );
10  }
11 }
```

JSX - JavaScript eXtension - ułatwia tworzenie HTML-a za pomocą JavaScriptu

ES5 -wersja ECMAScriptu od 2009

ES6 -wersja ECMAScriptu od 2015 (teraz już wspierana przez przeglądarki)

```
1 class HelloWorld extends React.Component {
2   render() {
3     return (
4       <h1 className='large'>Hello World</h1>
5     );
6   }
7 }
```

```
1 class HelloWorld extends React.Component {
2   render() {
3     return (
4       React.createElement(
5         'h1',
6         {className: 'large'},
7         'Hello World'
8       )
9     );
10  }
11 }
```

JSX - JavaScript eXtension - ułatwia tworzenie HTML-a za pomocą JavaScriptu

ES5 -wersja **ECMAScriptu** od 2009

ES6 -wersja **ECMAScriptu** od 2015 (teraz już wspierana przez przeglądarki)

```
1 class HelloWorld extends React.Component {
2   render() {
3     return (
4       <h1 className='large'>Hello World</h1>
5     );
6   }
7 }
```

```
1 class HelloWorld extends React.Component {
2   render() {
3     return (
4       React.createElement(
5         'h1',
6         {className: 'large'},
7         'Hello World'
8       )
9     );
10  }
11 }
```

- kompilacja czegoś (JSX?) na JavaScript w odpowiedniej wersji następuje po załadowaniu strony do przeglądarki

JSX - JavaScript eXtension - ułatwia tworzenie HTML-a za pomocą JavaScriptu

ES5 - wersja ECMAScriptu od 2009

ES6 - wersja ECMAScriptu od 2015 (teraz już wspierana przez przeglądarki)

```
1 class HelloWorld extends React.Component {
2   render() {
3     return (
4       <h1 className='large'>Hello World</h1>
5     );
6   }
7 }
```

```
1 class HelloWorld extends React.Component {
2   render() {
3     return (
4       React.createElement(
5         'h1',
6         {className: 'large'},
7         'Hello World'
8       )
9     );
10  }
11 }
```

- kompilacja czegoś (JSX?) na JavaScript w odpowiedniej wersji następuje po załadowaniu strony do przeglądarki
- czyli źródło strony ma wersję źródłową, a nie HTML wyświetlony przez przeglądarkę

JSX - JavaScript eXtension - ułatwia tworzenie HTML-a za pomocą JavaScriptu

ES5 - wersja ECMAScriptu od 2009

ES6 - wersja ECMAScriptu od 2015 (teraz już wspierana przez przeglądarki)

```
1 class HelloWorld extends React.Component {
2   render() {
3     return (
4       React.createElement(
5         'h1',
6         {className: 'large'},
7         'Hello World'
8       )
9     );
10  }
11 }
```

```
1 var HelloWorld = function() {}
2 Object.extend(HelloWorld, React.Component)
3 HelloWorld.prototype.render = function() {
4   return (
5     React.createElement(
6       'h1',
7       {className: 'large'},
8       'Hello World'
9     )
10  );
11 }
```

Babel transluje ES6 do ES5

`class` -wykorzystywane w ES6, dlatego React używa: `className`

`class` -wykorzystywane w `ES6`, dlatego React używa: `className`

`for` -wykorzystywane w `JavaScript`, dlatego React używa: `htmlFor`

class -wykorzystywane w **ES6**, dlatego React używa: `className`

for -wykorzystywane w **JavaScript**, dlatego React używa: `htmlFor`

przykład tworzenia HTML-a w JSX:

```
1 <div>
2   
3   <h1>Welcome back Ari</h1>
4 </div>
```

class -wykorzystywane w **ES6**, dlatego React używa: `className`

for -wykorzystywane w **JavaScript**, dlatego React używa: `htmlFor`

przykład tworzenia HTML-a w JSX:

```
1 <div>
2   
3   <h1>Welcome back Ari</h1>
4 </div>
```

bez JSX:

```
1 React.createElement("div", null,
2   React.createElement("img", {src: "profile.jpg", alt: "Profile photo"}),
3   React.createElement("h1", null, "Welcome back Ari")
4 );
```

- używamy `id` tagu (unikalnego z definicji) do określenia punktu montowania

```
1 <body>
2   <div id="app"></div>
3   <script type="text/babel">
4     var app = <h1>Hello world</h1>
5     var replacedComponent = document.querySelector('#app');
6     ReactDOM.render(app, replacedComponent);
7   </script>
8 </body>
```

- używamy `id` tagu (unikalnego z definicji) do określenia punktu montowania
- wybrany tag jest zastępowany przez HTML utworzony w przeglądarce przez JavaScript

```
1 <body>
2   <div id="app"></div>
3   <script type="text/babel">
4     var app = <h1>Hello world</h1>
5     var replacedComponent = document.querySelector('#app');
6     ReactDOM.render(app, replacedComponent);
7   </script>
8 </body>
```

```
1 const students = [  
2   { text: "bart simpson",  
3     avatar: "bart-simpson-256.png" },  
4   { text: "homer simpson",  
5     avatar: "homer-simpson-256.png" },  
6   { text: "lisa simpson",  
7     avatar: "lisa-simpson-256.png" },  
8   { text: "maggie simpson",  
9     avatar: "maggie-simpson-256.png"},  
10  { text: "marge simpson",  
11    avatar: "marge-simpson-256.png" },  
12 ]
```



bart simpson



homer simpson



lisa simpson



maggie simpson



marge simpson

- **html**
- App
- Content
- Item



```
1 <html>
2   <head> <meta charset="utf-8" />
3     <script src="..."></script>
4     ...
5     <style>
6       .content * {
7         text-align: center;
8       }
9     </style>
10  </head>
11  <body>
12    <div id="app" class="demo"/>
13    <script type="text/babel">
14      ...
15    </script>
16  </body>
17 </html>
```

- html
- App
- Content
- Item



```
1 const App = (props) => {
2   const students = [
3     { text: "bart simpson", avatar: "bart-256.png" },
4     { text: "homer simpson", avatar: "homer-256.png" },
5     { text: "lisa simpson", avatar: "lisa-256.png" },
6     { text: "maggie simpson", avatar: "maggie-256.png" },
7     { text: "marge simpson", avatar: "marge-256.png" },
8   ];
9   return (
10    <div className="content">
11      <Content students={students} />
12    </div>
13  );
14 };
15 ReactDOM.render(<App />, document.querySelector("#app"));
```

- html
- App
- **Content**
- Item



```
1 const Content = (props) => {  
2   return (  
3     <table>  
4       {props.students.map((s) => (  
5         <Item item={s} />  
6       ))}  
7     </table>  
8   );  
9 };
```

- html
- App
- Content
- **Item**



```
1 const Item = (props) => {  
2   return (  
3     <tr>  
4       <td>  
5         <img  
6           alt={props.item.text}  
7           src={props.item.avatar}  
8           height="64px"  
9         />  
10      </td>  
11      <td> {props.item.text} </td>  
12    </tr>  
13  );  
14  };
```

czysta funkcja – funkcja, która zależy tylko od parametrów

-function which depends only on parameters

- Nie zmienia żadnych obiektów ani zmiennych, które istniały przed jego wywołaniem.
- Biorąc pod uwagę te same dane wejściowe, funkcja czysta powinna zawsze zwracać ten sam wynik.

Proces renderowania Reacta zawsze musi być czysty – komponenty powinny zwracać tylko swój JSX i nie z

- Components should only return their JSX, and not change any objects or variables that existed before rendering

- pure
- impure

czysta funkcja – funkcja, która zależy tylko od parametrów

-function which depends only on parameters

- Nie zmienia żadnych obiektów ani zmiennych, które istniały przed jego wywołaniem.
- Biorąc pod uwagę te same dane wejściowe, funkcja czysta powinna zawsze zwracać ten sam wynik.

Proces renderowania Reacta zawsze musi być czysty – komponenty powinny zwracać tylko swój JSX i nie

- Components should only return their JSX, and not change any objects or variables that existed before rendering

- pure
- impure

czysta funkcja – funkcja, która zależy tylko od parametrów

-function which depends only on parameters

- Nie zmienia żadnych obiektów ani zmiennych, które istniały przed jego wywołaniem.
- Biorąc pod uwagę te same dane wejściowe, funkcja czysta powinna zawsze zwracać ten sam wynik.

Proces renderowania Reacta zawsze musi być czysty – komponenty powinny zwracać tylko swój JSX i nie

- Components should only return their JSX, and not change any objects or variables that existed before rendering

- pure
- impure

czysta funkcja – funkcja, która zależy tylko od parametrów

-function which depends only on parameters

- Nie zmienia żadnych obiektów ani zmiennych, które istniały przed jego wywołaniem.
- Biorąc pod uwagę te same dane wejściowe, funkcja czysta powinna zawsze zwracać ten sam wynik.

Proces renderowania Reacta zawsze musi być czysty – komponenty powinny zwracać tylko swój JSX i nie z

- Components should only return their JSX, and not change any objects or variables that existed before rendering

- pure
- impure

czysta funkcja – funkcja, która zależy tylko od parametrów

-function which depends only on parameters

- Nie zmienia żadnych obiektów ani zmiennych, które istniały przed jego wywołaniem.
- Biorąc pod uwagę te same dane wejściowe, funkcja czysta powinna zawsze zwracać ten sam wynik.

```
1 function Cup({ guest }: {guest:number}) {
2   return <h2>Tea cup for guest #{guest}</h2>
3 }
4 export default function PureTeaSet() {
5   return (
6     <>
7       <Cup guest={1} />
8       <Cup guest={2} />
9       <Cup guest={3} />
10    </>
11  );
12 }
```

Proces renderowania Reacta zawsze musi być czysty

- Components should only return their JSX, and not change any objects or variables that existed before rendering

- pure
- impure

```
1 Tea cup for guest #1
2 Tea cup for guest #2
3 Tea cup for guest #3
```

X i nie z

czysta funkcja – funkcja, która zależy tylko od parametrów

-function which depends only on parameters

- Nie zmienia żadnych obiektów ani zmiennych, które istniały przed jego wywołaniem.
- Biorąc pod uwagę te same dane wejściowe, funkcja czysta powinna zawsze zwracać ten sam wynik.

```
1 let guest = 0;
2 function Cup() {
3   guest = guest + 1;
4   return <h2>Tea cup for guest #{guest}</h2>;
5 }
6 export default function TeaSet() {
7   return (
8     <>
9     <Cup />
10    <Cup />
11    <Cup />
12    </>
13  );
14 }
```

Proces renderowania Reacta zawsze musi być czysty — komponenty powinny zwracać tylko swój JSX i nie z

- Components should only return their JSX, and not change any objects or variables that existed before rendering

- pure
- impure

```
1 Tea cup for guest #2
2 Tea cup for guest #4
3 Tea cup for guest #6
```

czysta funkcja – funkcja, która zależy tylko od parametrów

-function which depends only on parameters

- Nie zmienia żadnych obiektów ani zmiennych, które istniały przed jego wywołaniem.
- Biorąc pod uwagę te same dane wejściowe, funkcja czysta powinna zawsze zwracać ten sam wynik.

```
1 let guest = 0;
2 function Cup() {
3   guest = guest + 1;
4   return <h2>Tea cup for guest #{guest}</h2>;
5 }
6 export default function TeaSet() {
7   return (
8     <>
9     <Cup />
10    <Cup />
11    <Cup />
12  </>
13 );
14 }
```

Proces renderowania Reacta zawsze musi być czysty — komponenty powinny zwracać tylko swój JSX i nie z

- Components should only return their JSX, and not change any objects or variables that existed before rendering

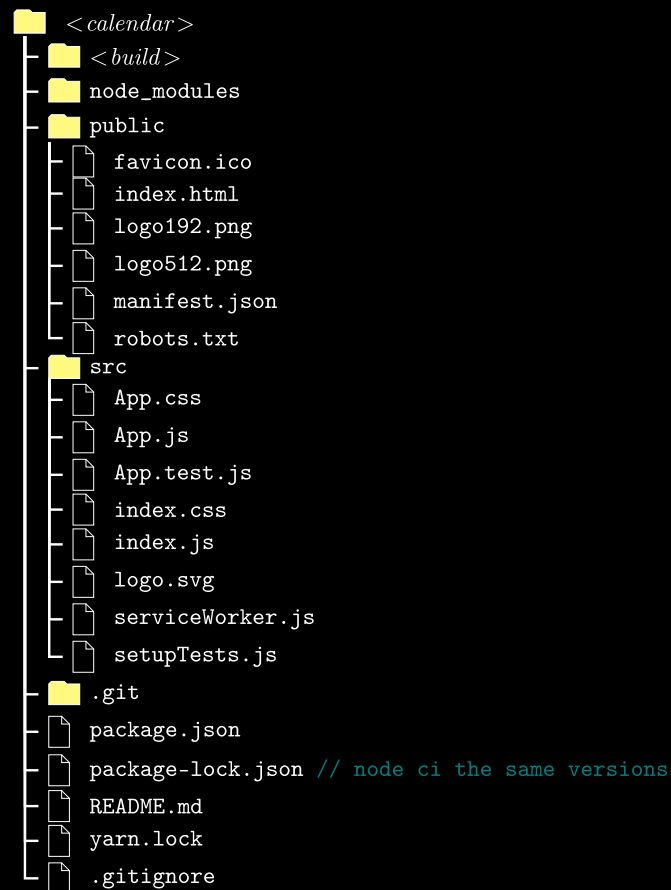
- pure
- impure

```
1 Tea cup for guest #2
2 Tea cup for guest #4
3 Tea cup for guest #6
```

Aby pomóc w wyłapywaniu tego typu błędów, w trybie programistycznym komponenty są wykonywane dwukrotnie (jeden wynik jest ignorowany)

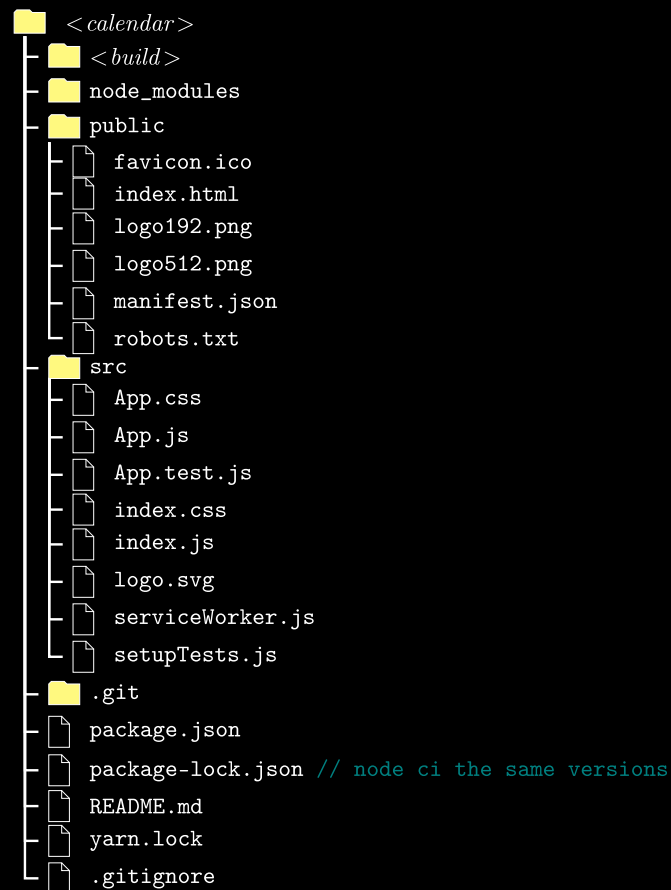
utwórz projekt

```
1 npx create-react-app calendar
2 cd calendar
3 npm install
```



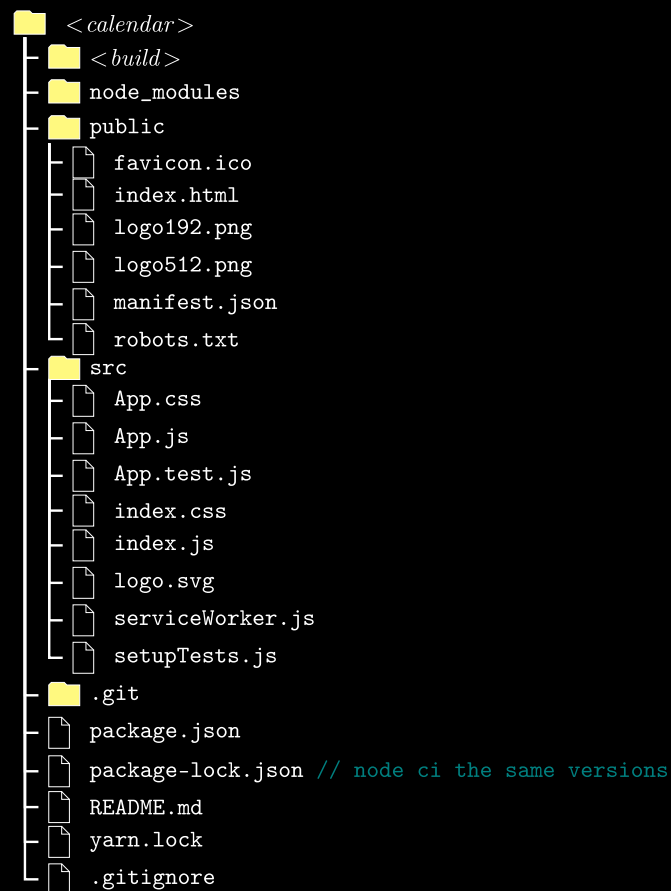
index.html:

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
6     <meta name="viewport" content="width=device-width,
7       ↪ initial-scale=1" />
8     <meta name="theme-color" content="#000000" />
9     <meta
10       name="description"
11       content="Web site created using create-react-app"
12     />
13     <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png"
14       ↪ />
15     <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
16     <title>React App</title>
17   </head>
18   <body>
19     <noscript>You need to enable JavaScript to run this
20       ↪ app.</noscript>
21     <div id="root"></div>
22   </body>
23 </html>
```



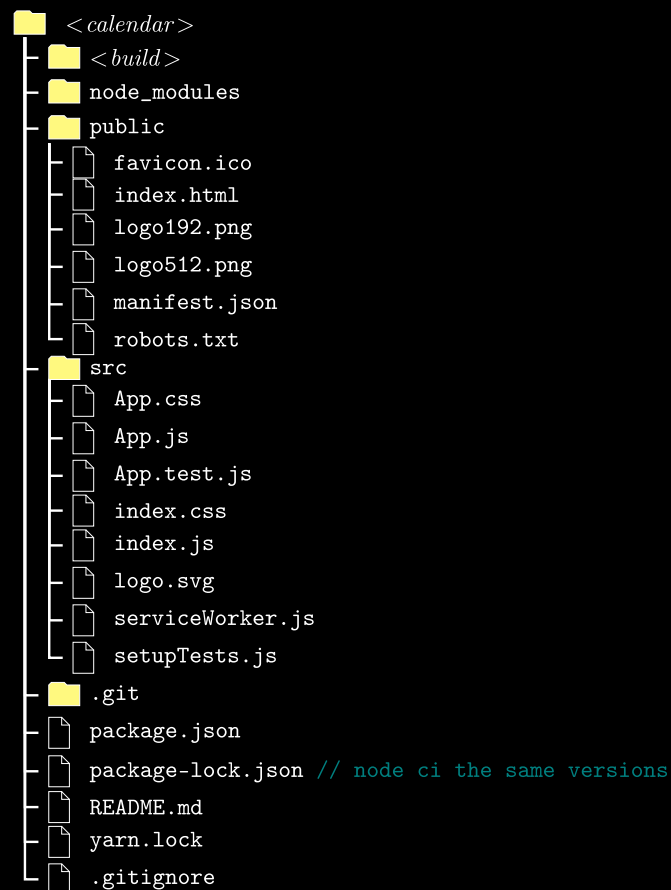
index.js:

```
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4 import App from './App';
5 import * as serviceWorker from './serviceWorker';
6 ReactDOM.render(
7   <React.StrictMode>
8     <App />
9   </React.StrictMode>,
10  document.getElementById('root')
11 );
12 // If you want your app to work offline
13 // and load faster, you can change
14 // unregister() to register() below.
15 // Note this comes with some pitfalls.
16 // Learn more about service
17 // workers: https://bit.ly/CRA-PWA
18 serviceWorker.unregister();
```



app.js:

```
1 import React from 'react';
2 import logo from './logo.svg';
3 import './App.css';
4 function App() {
5   return (
6     <div className="App">
7       <header className="App-header">
8         <img src={logo} className="App-logo"
9           alt="logo" />
10        <p> Edit <code>src/App.js</code>
11          and save to reload. </p>
12        <a className="App-link"
13          href="https://reactjs.org"
14          target="_blank"
15          rel="noopener noreferrer">
16          Learn React </a>
17      </header>
18    </div>
19  );
20 }
21 export default App;
```

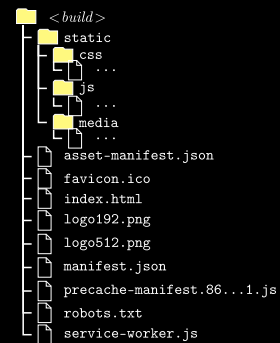


budowa paczki produkcyjnej:

```
1 npm run-script build
```

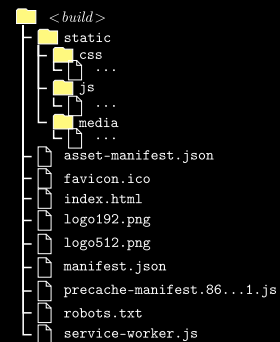
instalacja serwera webowego:

```
1 npm install -g serve
```



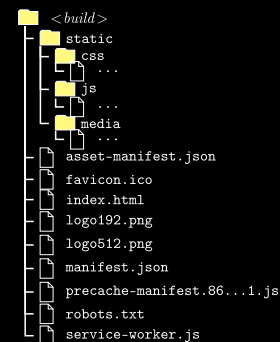
Następnie możemy udostępnić nasz statyczny folder kompilacji za pomocą serwera WWW:

```
1  serve -s build
2  -----
3  |
4  |   Serving!
5  |
6  |   - Local:           http://localhost:5000
7  |   - On Your Network: http://192.168.x.xxx:5000
8  |
9  |   Copied local address to clipboard!
10 | -----
11
```



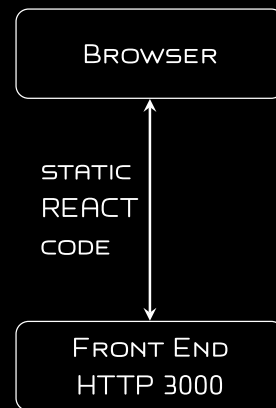
Następnie możemy udostępnić nasz statyczny folder kompilacji za pomocą serwera WWW:

```
1  serve -s build
2
3  |
4  |   Serving!
5  |
6  |   - Local:           http://localhost:5000
7  |   - On Your Network: http://192.168.x.xxx:5000
8  |
9  |   Copied local address to clipboard!
10 |
11 |
```



Dowolny server (JavaScript, C#, Java, PHP, Python, ...) umożliwiający serwowanie statycznych plików może zostać użyty do serwowania aplikacji REACT!

Nie komunikujemy się z żadnym serwerem

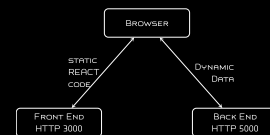


Jako źródło naszych danych będziemy używać serwera ekspresowego zbudowanego w JavaScript i działającego na node.js.

```
1 npm install express
```

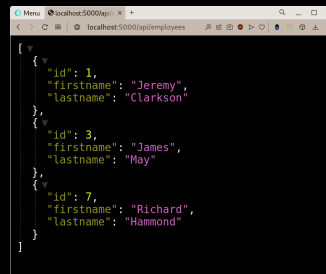
i tworzymy kod serwera webowego `server.js`:

```
1 const express = require('express');
2 const app = express();
3 const data = [
4   { "id": 1, "firstname": "Jeremy", "lastname": "Clarkson" },
5   { "id": 3, "firstname": "James", "lastname": "May" },
6   { "id": 7, "firstname": "Richard", "lastname": "Hammond" }
7 ]
8
9 app.get('/api/employees', (req, res) => res.json(data))
10 app.listen(5000)
```



i tworzymy kod serwera webowego `server.js`:

```
1 const express = require('express');
2 const app = express();
3 const data = [
4   { "id": 1, "firstname": "Jeremy", "lastname": "Clarkson" },
5   { "id": 3, "firstname": "James", "lastname": "May" },
6   { "id": 7, "firstname": "Richard", "lastname": "Hammond" }
7 ]
8
9 app.get('/api/employees', (req, res) => res.json(data))
10 app.listen(5000)
```



The screenshot shows a web browser window with the address bar displaying 'localhost:5000/api/employees'. The page content shows a JSON array of three employee objects, each with an 'id', 'firstname', and 'lastname' property. The data is: [{ "id": 1, "firstname": "Jeremy", "lastname": "Clarkson" }, { "id": 3, "firstname": "James", "lastname": "May" }, { "id": 7, "firstname": "Richard", "lastname": "Hammond" }].

czytanie danych z pliku :

db.json:

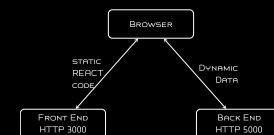
```
1 {  
2   "employees": [  
3     {"id": 1, "firstname": "Jeremy", "lastname": "Clarkson"},  
4     {"id": 3, "firstname": "James", "lastname": "May"},  
5     {"id": 7, "firstname": "Richard", "lastname": "Hammond"}  
6   ]  
7 }
```

server.js:

```
1 const express = require('express');  
2 const fs = require('fs');  
3 const app = express();  
4 const data =  
5   ↳ JSON.parse(fs.readFileSync('db.json'))  
6 app.get('/api/employees', (req, res) =>  
7   ↳ res.json(data.employees))  
8 app.listen(5000)
```

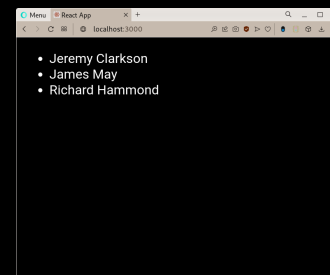
tworzymy bardzo prosty komponent REACT `App.js`:

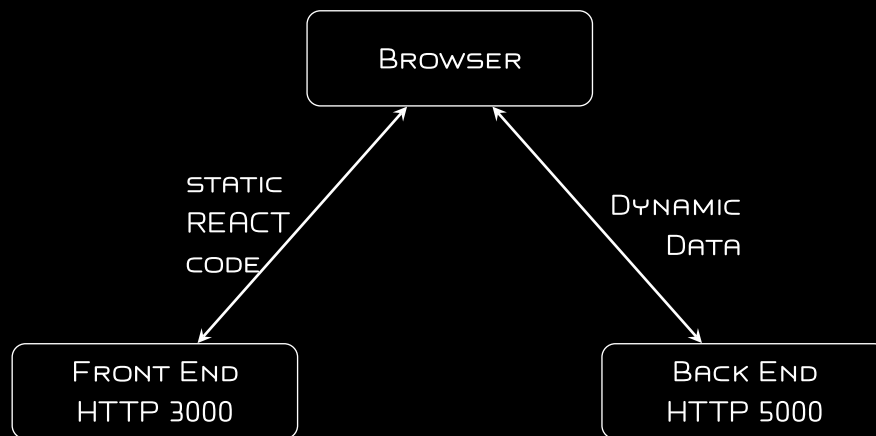
```
1 import React, { useState, useEffect } from 'react';
2 import axios from 'axios';
3 import './App.css';
4 function App() {
5   const [data, setData] = useState([]);
6   useEffect(() => {
7     const fetchData = async () => {
8       const result = await axios(
9         '/api/employees',
10       );
11       setData(result.data);
12     };
13     fetchData();
14   }, []);
15   return (
16     <ul>
17       {data.map(item => (
18         <li key={item.id}>
19           {item.firstname} {item.lastname}
20         </li>
21       ))}
22     </ul>
23   );
24 }
25 export default App;
```



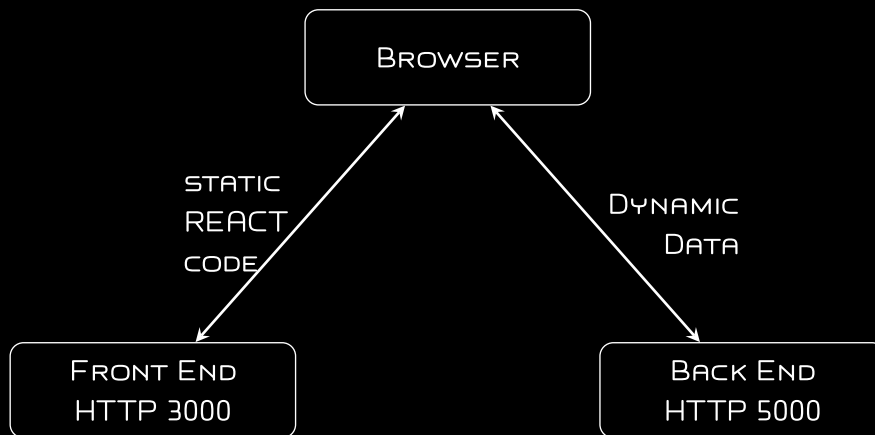
tworzymy bardzo prosty komponent REACT `App.js`:

```
1 import React, { useState, useEffect } from 'react';
2 import axios from 'axios';
3 import './App.css';
4 function App() {
5   const [data, setData] = useState([]);
6   useEffect(() => {
7     const fetchData = async () => {
8       const result = await axios(
9         '/api/employees',
10       );
11       setData(result.data);
12     };
13     fetchData();
14   }, []);
15   return (
16     <ul>
17       {data.map(item => (
18         <li key={item.id}>
19           {item.firstname} {item.lastname}
20         </li>
21       ))}
22     </ul>
23   );
24 }
25 export default App;
```

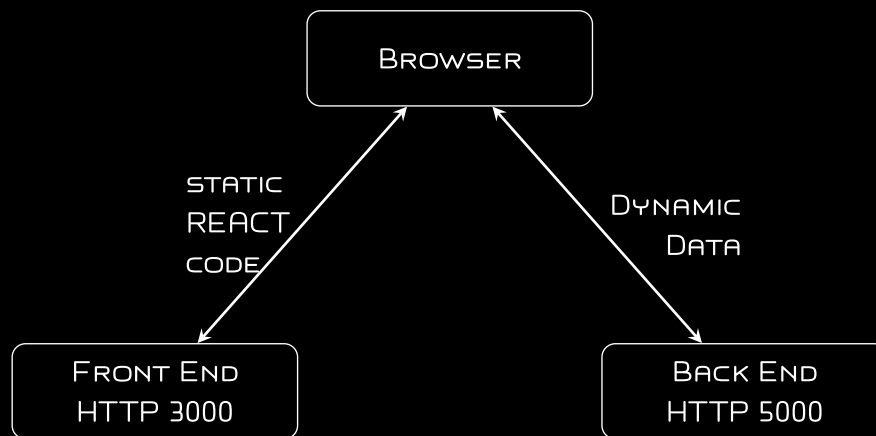




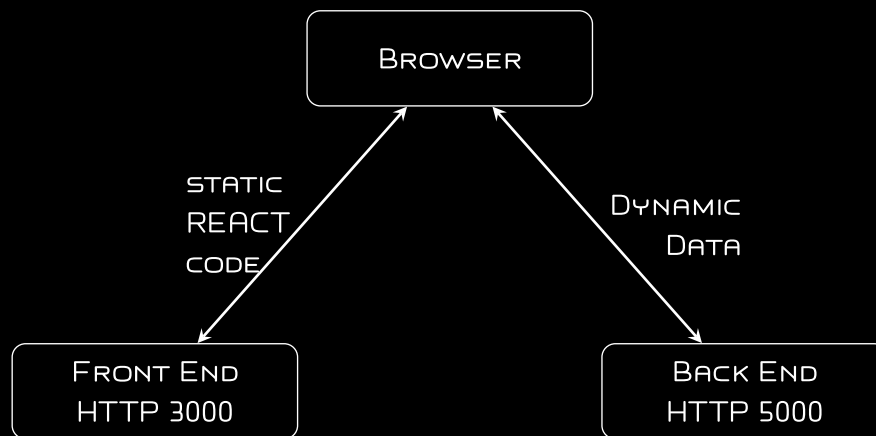
- aby uruchomić taką konfigurację musimy uruchomić dwa serwery WWW:
 - jeden do serwowania (statycznych) plików z reactem



- aby uruchomić taką konfigurację musimy uruchomić dwa serwery WWW:
 - jeden do serwowania (statycznych) plików z reactem
 - drugi dla komunikacji dynamicznej

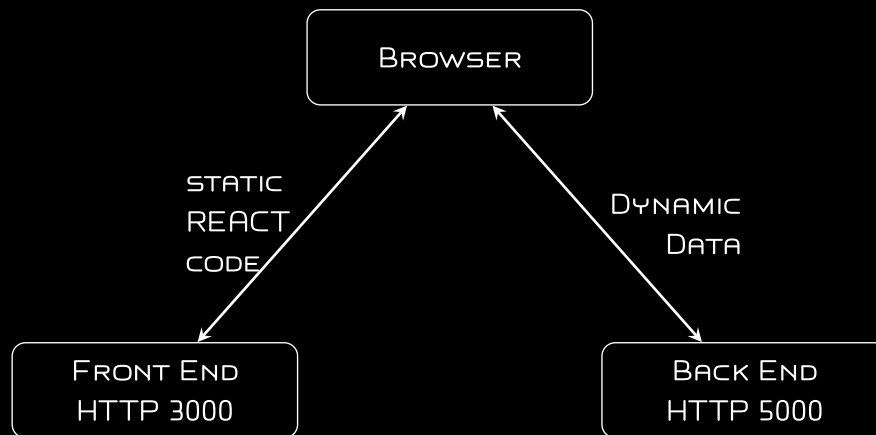


- aby uruchomić taką konfigurację musimy uruchomić dwa serwery WWW:
 - jeden do serwowania (statycznych) plików z reactem
 - drugi dla komunikacji dynamicznej



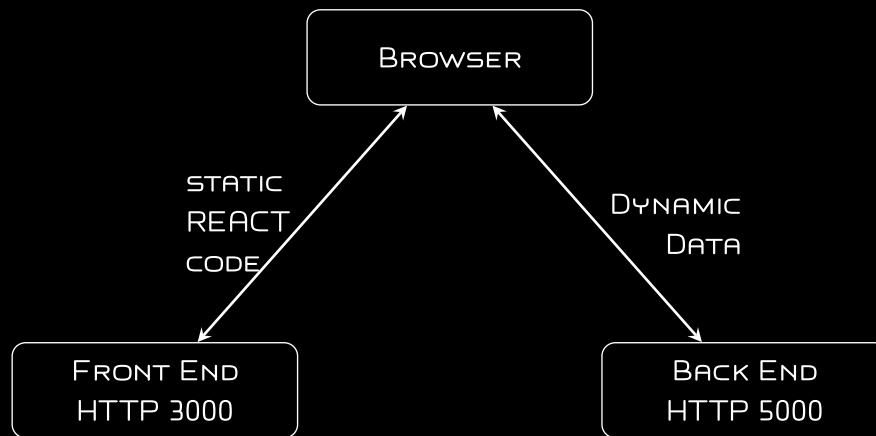
- mamy 2 problemy:

- aby uruchomić taką konfigurację musimy uruchomić dwa serwery WWW:
 - jeden do serwowania (statycznych) plików z reactem
 - drugi dla komunikacji dynamicznej



- mamy 2 problemy:
 - CORS

- aby uruchomić taką konfigurację musimy uruchomić dwa serwery WWW:
 - jeden do serwowania (statycznych) plików z reactem
 - drugi dla komunikacji dynamicznej

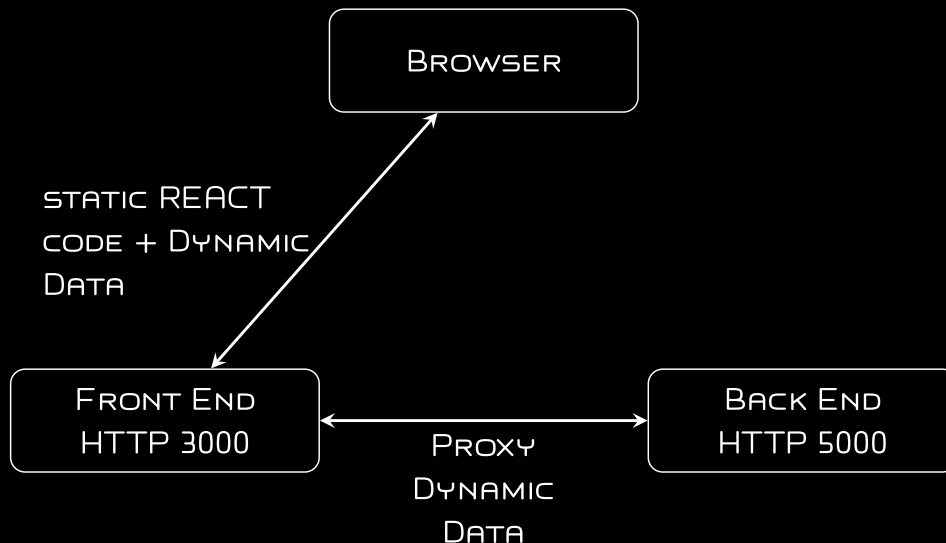


- mamy 2 problemy:
 - CORS
 - dwa serwery zamiast jednego

Uruchamianie wymaga uruchomienia REACT z ich serwera deweloperskiego.

W trakcie uruchamiania możemy tunelować komunikację z serwerem - ustawienia proxy w pliku `package.json`

```
1 "proxy": "http://localhost:5000"
```



Informuje to React o żądaniach API proxy do serwera Node.js zbudowanego z Express.

- W produkcji nie powinniśmy używać wbudowanego serwera deweloperskiego, który jest instalowany z create-react-app

- W produkcji nie powinniśmy używać wbudowanego serwera deweloperskiego, który jest instalowany z **create-react-app**
- **Powinniśmy stworzyć pakiet produkcyjny:**

```
1 npm run build
```

- W produkcji nie powinniśmy używać wbudowanego serwera deweloperskiego, który jest instalowany z **create-react-app**
- Powinniśmy stworzyć pakiet produkcyjny:

```
1 npm run build
```

- i dodać obsługę plików statycznych:

```
1 const express = require('express')
2 const path = require('path')
3 const fs = require('fs');
4 const app = express()
5 const data = JSON.parse(fs.readFileSync('db.json'))
6 app.use(express.static(path.join(__dirname, 'build')))
7 app.get('/api/employees', (req, res) => res.json(data.employees))
8 app.get('/', (req, res) => {
9   res.sendFile(path.join(__dirname, 'build', 'index.html'))
10 })
11 app.listen(5000)
```

- Powinniśmy stworzyć pakiet produkcyjny:

```
1 npm run build
```

- Powinniśmy stworzyć pakiet produkcyjny:
- Plik z danymi `db.json`

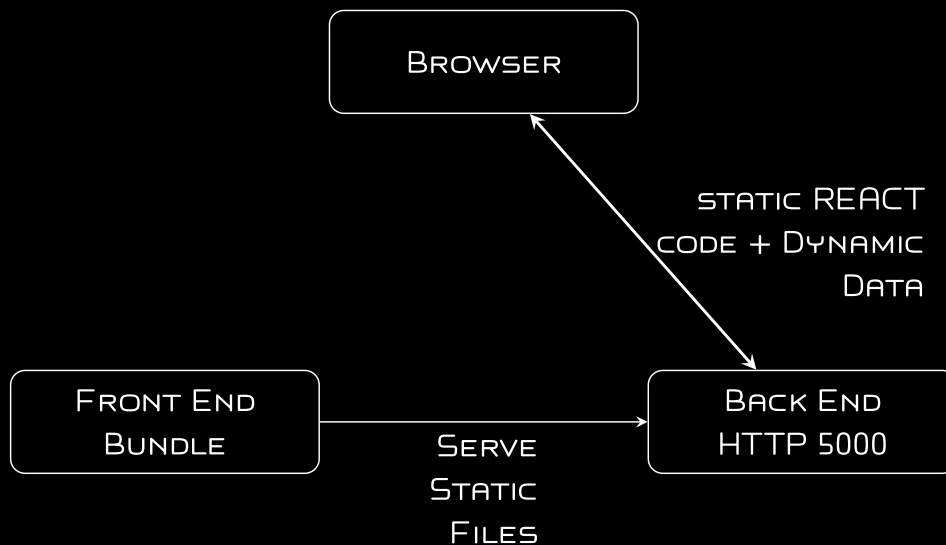
```
1 {  
2   "employees": [  
3     {  
4       "id": 1,  
5       "firstname": "Jeremy",  
6       "lastname": "Clarkson"  
7     },  
8     {  
9       "id": 3,  
10      "firstname": "James",  
11      "lastname": "May"  
12    }  
13  ],  
14  "profile": {  
15    "name": "Top Gear",  
16    "status": {  
17      "broadcast": "BBC"  
18    }  
19  }  
20 }
```

- Powinniśmy stworzyć pakiet produkcyjny:
- Plik z danymi `db.json`
- **Plik konfiguracyjny** `routes.json`

```
1 {  
2   "/api/*": "$1"  
3 }
```

- Powinniśmy stworzyć pakiet produkcyjny:
- Plik z danymi `db.json`
- Plik konfiguracyjny `routes.json`
- **uruchomienie całości** `json-server`

```
1 json-server db.json --routes routes.json --static ./build
```



W końcu mamy **jeden serwer** obsługujący zarówno **statyczną**, jak i **dynamiczną** zawartość!

- **axios**

- wersja z obsługą oczekiwania i błędu
- kod pobierania przez axiosa
- wyświetlane w czasie ładowania danych
- wykorzystanie fetch
- wykorzystanie react-query

```
1 npm install axios
```

```
1 import React, { useState, useEffect } from 'react';
2 import axios from 'axios';
3 import './App.css';
4 function App() {
5   const [data, setData] = useState([]);
6   useEffect(() => {
7     const fetchData = async () => {
8       const result = await axios(
9         '/api/employees',
10       );
11       setData(result.data);
12     };
13     fetchData();
14   }, []);
15   return (
16     <ul>
17       {data.map(item => (
18         <li key={item.id}>
19           {item.firstname} {item.lastname}
20         </li>
21       ))}
22     </ul>
23   );
24 }
25 export default App;
```

- axios
 - wersja z obsługą oczekiwania i błędu
 - kod pobierania przez axiosa
 - wyświetlane w czasie ładowania danych
- wykorzystanie fetch
- wykorzystanie react-query

```
1 npm install axios
```

```
1 import React, { useState, useEffect } from 'react';
2 import axios from 'axios';
3 import './App.css';
4 function App() {
5   const [data, setData] = useState([]);
6   const [loading, setLoading] = useState(true);
7   const [error, setError] = useState(null);
8
9   // tu kod pobierający liste to data
10
11   if(loading) return "Loading ..."
12   if(error) return "Error!";
13   return (
14     <ul>
15       {data.map(item => (
16         <li key={item.id}>
17           {item.firstname} {item.lastname}
18         </li>
19       ))}
20     </ul>
21   );
22 }
23 export default App;
```

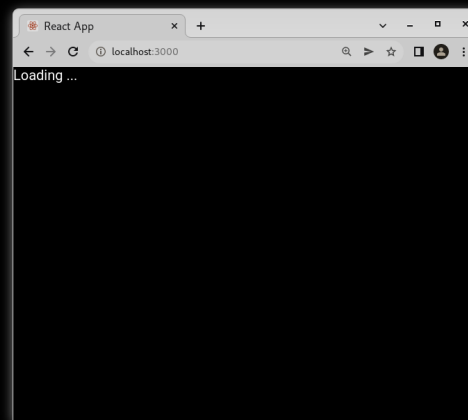
- axios
 - wersja z obsługą oczekiwania i błędu
 - kod pobierania przez axiosa
 - wyświetlane w czasie ładowania danych
- wykorzystanie fetch
- wykorzystanie react-query

```
1 npm install axios
```

```
1  async function getData() {  
2    await axios('/api/employees')  
3    .then(response=>{  
4      setData(response.data);  
5    })  
6    .catch(error=>{  
7      console.error("Error fetching data", error);  
8      setError(error)  
9    })  
10   .finally(()=>{  
11     setLoading(false)  
12   })  
13 }  
14  
15 useEffect(() => {  
16   getData();  
17 }, []);
```

- axios
 - wersja z obsługą oczekiwania i błędu
 - kod pobierania przez axiosa
 - wyświetlane w czasie ładowania danych
- wykorzystanie `fetch`
- wykorzystanie `react-query`

```
1 npm install axios
```



- axios
 - wersja z obsługą oczekiwania i błędu
 - kod pobierania przez axiosa
 - wyświetlane w czasie ładowania danych
- wykorzystanie `fetch`
- wykorzystanie `react-query`

```
1  useEffect(() => {
2    fetch('/api/employees')
3    .then(response=>{
4      if(response.ok) {
5        return response.json()
6      }
7      throw response;
8    })
9    .then(data=>{
10     setData(data);
11   })
12   .catch(error=>{
13     console.error("Error fetching data", error);
14     setError(error)
15   })
16   .finally(()=>{
17     setLoading(false)
18   })
19 }, []);
```

- axios
 - wersja z obsługą oczekiwania i błędu
 - kod pobierania przez axiosa
 - wyświetlane w czasie ładowania danych
- wykorzystanie `fetch`
- wykorzystanie `react-query`

```
1 npm install react-query
```

```
1 import React from "react";
2 import {QueryClient, QueryClientProvider, useQuery} from
  ↳ "react-query";
3 const queryClient = new QueryClient();
4 function App(props) {
5   return (
6     <QueryClientProvider client={queryClient}>
7       <Example />
8     </QueryClientProvider>
9   );
10 }
11 function Example() {
12   const { isLoading, error, data, isFetching }
13     = useQuery("repoData", () =>
14       fetch(
15         '/api/employees'
16       ).then((res) => res.json())
17   );
18   if (isLoading) return "Loading...";
19   if (error) return "error occurred: "+error.message;
20   return (
21     <ul>
22       {data.map(item => (
23         <li key={item.id}>
24           {item.firstname} {item.lastname}
25         </li>
26       ))}
27     </ul>
28   );
29 }
30 export default App;
```

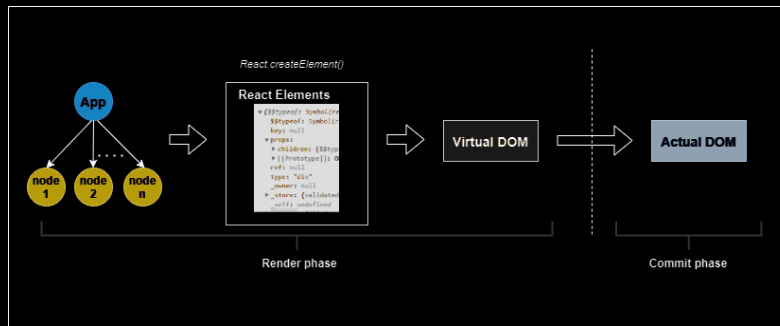
render - wywoływanie funkcji renderowania komponentu(ów) (lub wywoływanie funkcji, jeśli jest to komponent funkcji).

```
1 export default function Clock({ time }) {  
2   return (  
3     <>  
4       <h1>{time}</h1>  
5       <input />  
6     </>  
7   );  
8 }
```

- jeżeli komponent `clock` jest renderowany co sekundę
- ... treść wprowadzona przez użytkownika do pola `input` jest zachowywana

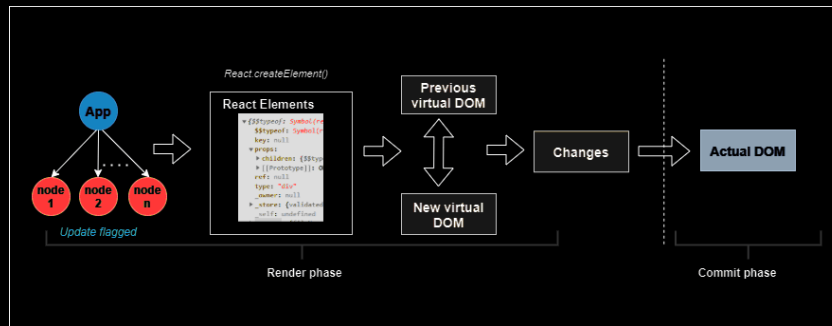
render - wywoływanie funkcji renderowania komponentu(ów) (lub wywoływanie funkcji, jeśli jest to komponent funkcji).

- **Początkowe renderowanie komponentu.**
- Stan komponentu (lub jednego z jego przodków) został zaktualizowany.



render - wywoływanie funkcji renderowania komponentu(ów) (lub wywoływanie funkcji, jeśli jest to komponent funkcji).

- Początkowe renderowanie komponentu.
- Stan komponentu (lub jednego z jego przodków) został zaktualizowany.



render - wywoływanie funkcji renderowania komponentu(ów) (lub wywoływanie funkcji, jeśli jest to komponent funkcji).

- Początkowe renderowanie komponentu.
- Stan komponentu (lub jednego z jego przodków) został zaktualizowany.
 - renderowanie komponentu, gdy zmienił się element nadrzędny, może być nieefektywne - gdy trzeba należy to poprawić (`useCallback()` lub `useMemo()`)

render - wywoływanie funkcji renderowania komponentu(ów) (lub wywoływanie funkcji, jeśli jest to komponent funkcji).

- Początkowe renderowanie komponentu.
- Stan komponentu (lub jednego z jego przodków) został zaktualizowany.
 - renderowanie komponentu, gdy zmienił się element nadrzędny, może być nieefektywne - gdy trzeba należy to poprawić (`useCallback()` lub `useMemo()`)

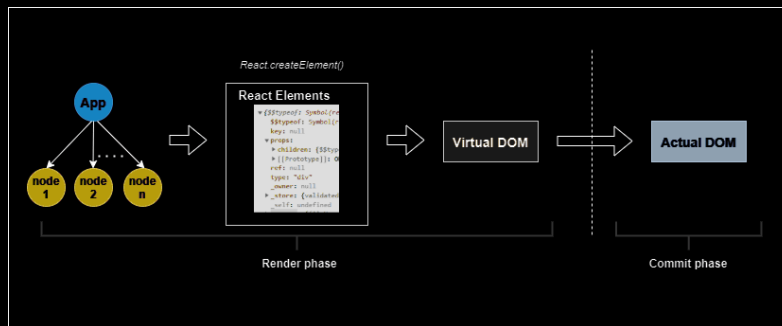
commit - przepisanie uzyskanego w wirtualnym DOM drzewa tagów HTML do DOM przeglądarki

render - wywoływanie funkcji renderowania komponentu(ów) (lub wywoływanie funkcji, jeśli jest to komponent funkcji).

- Początkowe renderowanie komponentu.
- Stan komponentu (lub jednego z jego przodków) został zaktualizowany.
 - renderowanie komponentu, gdy zmienił się element nadrzędny, może być nieefektywne - gdy trzeba należy to poprawić (`useCallback()` lub `useMemo()`)

commit - przepisanie uzyskanego w wirtualnym DOM drzewa tagów HTML do DOM przeglądarki

- Przy początkowym renderowaniu React użyje interfejsu API DOM `appendChild()`, aby wyświetlić na ekranie wszystkie utworzone węzły DOM.

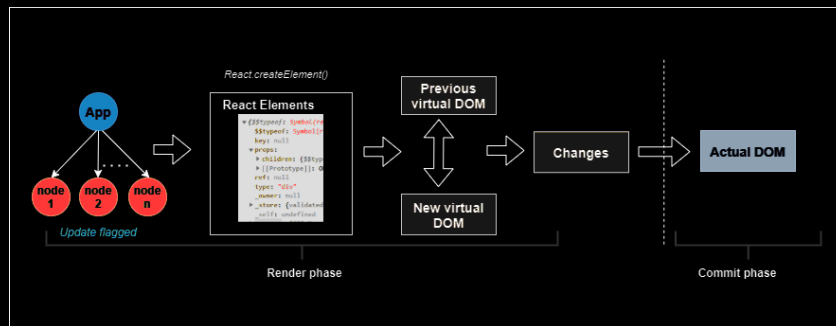


render - wywoływanie funkcji renderowania komponentu(ów) (lub wywoływanie funkcji, jeśli jest to komponent funkcji).

- Początkowe renderowanie komponentu.
- Stan komponentu (lub jednego z jego przodków) został zaktualizowany.
 - renderowanie komponentu, gdy zmienił się element nadrzędny, może być nieefektywne - gdy trzeba należy to poprawić (`useCallback()` lub `useMemo()`)

commit - przepisanie uzyskanego w wirtualnym DOM drzewa tagów HTML do DOM przeglądarki

- Przy początkowym renderowaniu React użyje interfejsu API DOM `appendChild()`, aby wyświetlić na ekranie wszystkie utworzone węzły DOM.
- W przypadku ponownego renderowania React zastosuje minimalne niezbędne operacje (obliczane podczas renderowania!), aby dopasować DOM do najnowszego wyniku renderowania.



W przypadku ponownego renderowania React zastosuje minimalne niezbędne operacje (obliczane podczas renderowania!), aby dopasować DOM do najnowszego wyniku renderowania.

```
1 export default function Clock({ time }) {  
2   return (  
3     <>  
4       <h1>{time}</h1>  
5       <input />  
6     </>  
7   );  
8 }
```

- jeżeli komponent `clock` jest renderowany co sekundę
- ... treść wprowadzona przez użytkownika do pola `input` jest zachowywana

W przypadku ponownego renderowania React zastosuje minimalne niezbędne operacje (obliczane podczas renderowania!), aby dopasować DOM do najnowszego wyniku renderowania.

```
1 export default function Clock({ time }) {  
2   return (  
3     <>  
4       <h1>{time}</h1>  
5       <input value={"balbinka"}/>  
6     </>  
7   );  
8 }
```

- jeżeli komponent `clock` jest renderowany co sekundę
- ... treść wprowadzona przez użytkownika do pola `input` jest zachowywana

W przypadku ponownego renderowania React zastosuje minimalne niezbędne operacje (obliczane podczas renderowania!), aby dopasować DOM do najnowszego wyniku renderowania.

```
1 export default function Clock({ time }) {  
2   return (  
3     <>  
4       <h1>{time}</h1>  
5       <input value={"balbinka"}/>  
6     </>  
7   );  
8 }
```

- jeżeli komponent `clock` jest renderowany co sekundę
- ... treść wprowadzona przez użytkownika do pola `input` jest zachowywana

Hooks to funkcje umożliwiające „podczepienie się” do funkcji stanu i cyklu życia Reacta z poziomu komponentów funkcyjnych

- Hook `useState` przyjmuje początkową wartość stanu i zwraca parę stałych: bieżący stan i funkcję modyfikującą bieżący stan

```
1 import React, { useState } from 'react';
2 function Recipes() {
3   const [ name, setName ] = useState('')
4   function handleIt (evt) {
5     setName(evt.target.value)
6   }
7   return (
8     <div>
9       <h1>My Name is: {name}</h1>
10      <input type="text" value={name}
11        onChange={handleIt}/>
12    </div>
13  )
14 }
15 export default Recipes;
```

- Hook `useState` przyjmuje początkową wartość stanu i zwraca parę stałych: bieżący stan i funkcję modyfikującą bieżący stan
- modyfikacja stanu powoduje ponowne renderowanie komponentu (i wszystkich komponentów podrzędnych)

```
1 import React, { useState } from 'react';
2 function Recipes() {
3   const [ name, setName ] = useState('')
4   function handleIt (evt) {
5     setName(evt.target.value)
6   }
7   return (
8     <div>
9       <h1>My Name is: {name}</h1>
10      <input type="text" value={name}
11        onChange={handleIt}/>
12    </div>
13  )
14 }
15 export default Recipes;
```

- Hook `useState` przyjmuje początkową wartość stanu i zwraca parę stałych: bieżący stan i funkcję modyfikującą bieżący stan
- modyfikacja stanu powoduje ponowne renderowanie komponentu (i wszystkich komponentów podrzędnych)
 - `onChange` may specify a function taking one argument (`event`)

```
1 import React, { useState } from 'react';
2 function Recipes() {
3   const [ name, setName ] = useState('')
4   function handleIt (evt) {
5     setName(evt.target.value)
6   }
7   return (
8     <div>
9       <h1>My Name is: {name}</h1>
10      <input type="text" value={name}
11        onChange={handleIt}/>
12    </div>
13  )
14 }
15 export default Recipes;
```

- Hook `useState` przyjmuje początkową wartość stanu i zwraca parę stałych: bieżący stan i funkcję modyfikującą bieżący stan
- modyfikacja stanu powoduje ponowne renderowanie komponentu (i wszystkich komponentów podrzędnych)
 - `onChange` may specify a function taking one argument (`event`)
 - property `target` of argument (`event`) refers to the HTML tag which was the origin of the event

```
1 import React, { useState } from 'react';
2 function Recipes() {
3   const [ name, setName ] = useState('')
4   function handleIt (evt) {
5     setName(evt.target.value)
6   }
7   return (
8     <div>
9       <h1>My Name is: {name}</h1>
10      <input type="text" value={name}
11              onChange={handleIt}/>
12    </div>
13  )
14 }
15 export default Recipes;
```

- Hook `useState` przyjmuje początkową wartość stanu i zwraca parę stałych: bieżący stan i funkcję modyfikującą bieżący stan
- modyfikacja stanu powoduje ponowne renderowanie komponentu (i wszystkich komponentów podrzędnych)
 - `onChange` may specify a function taking one argument (`event`)
 - property `target` of argument (`event`) refers to the HTML tag which was the origin of the event
 - property `value` of the HTML tag contains current text in the input tag

```
1 import React, { useState } from 'react';
2 function Recipes() {
3   const [ name, setName ] = useState('')
4   function handleIt (evt) {
5     setName(evt.target.value)
6   }
7   return (
8     <div>
9       <h1>My Name is: {name}</h1>
10      <input type="text" value={name}
11              onChange={handleIt}/>
12     </div>
13   )
14 }
15 export default Recipes;
```

- Hook `useState` przyjmuje początkową wartość stanu i zwraca parę stałych: bieżący stan i funkcję modyfikującą bieżący stan
- modyfikacja stanu powoduje ponowne renderowanie komponentu (i wszystkich komponentów podrzędnych)
 - `onChange` may specify a function taking one argument (`event`)
 - property `target` of argument (`event`) refers to the HTML tag which was the origin of the event
 - property `value` of the HTML tag contains current text in the input tag
 - calling `setName` triggers re-rendering of the component with updated `name` constant

```
1 import React, { useState } from 'react';
2 function Recipes() {
3   const [ name, setName ] = useState('')
4   function handleIt (evt) {
5     setName(evt.target.value)
6   }
7   return (
8     <div>
9       <h1>My Name is: {name}</h1>
10      <input type="text" value={name}
11        onChange={handleIt}/>
12    </div>
13  )
14 }
15 export default Recipes;
```

- useRef to React Hook, który pozwala odwoływać się do wartości, która nie jest potrzebna do renderowania.
- obliczana za pierwszym razem
- modyfikacje przez current

- `ref` attribute can be used to store reference to the DOM node
- this reference may be used in subsequent handlers to get access to that DOM node
 - in the handler of the button click we can access DOM nodes
 - this component refreshes after button click, not after every character entered to the input field

```
1 import { useRef, useState } from 'react';
2 function RefRecipes() {
3   const inputRef = useRef<HTMLInputElement>(null);
4   const [name, setName] = useState<string>("")
5   function handleIt () {
6     setName(inputRef.current!.value)
7   }
8   return (
9     <div>
10      <h1>My Name is: {name}</h1>
11      <input type="text" ref = {inputRef}/>
12      <button onClick={handleIt}>copy</button>
13    </div>
14  )
15 }
16 export default RefRecipes;
```

- `ref` attribute can be used to store reference to the DOM node
- this reference may be used in subsequent handlers to get access to that DOM node
 - in the handler of the button click we can access DOM nodes
 - this component refreshes after button click, not after every character entered to the input field

```
1 import { useRef, useState } from 'react';
2 function RefRecipes() {
3   const inputRef = useRef<HTMLInputElement>(null);
4   const [name, setName] = useState<string>("")
5   function handleIt () {
6     setName(inputRef.current!.value)
7   }
8   return (
9     <div>
10       <h1>My Name is: {name}</h1>
11       <input type="text" ref = {inputRef}/>
12       <button onClick={handleIt}>copy</button>
13     </div>
14   )
15 }
16 export default RefRecipes;
```

- `ref` attribute can be used to store reference to the DOM node
- this reference may be used in subsequent handlers to get access to that DOM node
 - in the handler of the button click we can access DOM nodes
 - this component refreshes after button click, not after every character entered to the input field

```
1 import { useRef, useState } from 'react';
2 function RefRecipes() {
3   const inputRef = useRef<HTMLInputElement>(null);
4   const [name, setName] = useState<string>("")
5   function handleIt () {
6     setName(inputRef.current!.value)
7   }
8   return (
9     <div>
10       <h1>My Name is: {name}</h1>
11       <input type="text" ref = {inputRef}/>
12       <button onClick={handleIt}>copy</button>
13     </div>
14   )
15 }
16 export default RefRecipes;
```

- `ref` attribute can be used to store reference to the DOM node
- this reference may be used in subsequent handlers to get access to that DOM node
 - in the handler of the button click we can access DOM nodes
 - this component refreshes after button click, not after every character entered to the input field

```
1 import { useRef, useState } from 'react';
2 function RefRecipes() {
3   const inputRef = useRef<HTMLInputElement>(null);
4   const [name, setName] = useState<string>("")
5   function handleIt () {
6     setName(inputRef.current!.value)
7   }
8   return (
9     <div>
10       <h1>My Name is: {name}</h1>
11       <input type="text" ref = {inputRef}/>
12       <button onClick={handleIt}>copy</button>
13     </div>
14   )
15 }
16 export default RefRecipes;
```

- `useState` - render after any change

```
1 import { useState } from 'react';
2 function Calc() {
3   const [ a1, setA1 ] = useState('0')
4   const [ a2, setA2 ] = useState('0')
5   const [sum, setSum] = useState<number>(0);
6   return (
7     <div>
8       <input type="number" value={a1} id="a1" onChange={e=>setA1(e.target.value)} />+
9       <input type="number" value={a2} id="a2" onChange={e=>setA2(e.target.value)} />
10      <button onClick={ ()=>setSum(Number(a1)+Number(a2)) }>
11        =
12      </button>
13      <input type="number" value={sum} readOnly />
14    </div>
15  )
16 }
17 export default Calc;
```

- `useRef` - render only after pressing a button, but we need a `useState` to display sum value

```
1 import { useRef, useState } from 'react';
2 function CalcRef() {
3   const a1R = useRef<HTMLInputElement>(null);
4   const a2R = useRef<HTMLInputElement>(null);
5   const [sum, setSum] = useState<number>(0);
6   return (
7     <div>
8       <input type="number" ref={a1R}/>+
9       <input type="number" ref={a2R}/>
10      <button onClick={() => setSum(Number(a1R.current!.value) + Number(a2R.current!.value))}>
11        =
12      </button>
13      <input type="number" value={sum} readOnly/>
14    </div>
15  )
16 }
17 export default CalcRef;
```

- we duplicate a code in lines 8 and 9

```
1 import { useState } from 'react';
2 function Calc() {
3   const [ a1, setA1 ] = useState('0')
4   const [ a2, setA2 ] = useState('0')
5   const [sum, setSum] = useState<number>(0);
6   return (
7     <div>
8       <input type="number" value={a1} id="a1" onChange={e=>setA1(e.target.value)} />+
9       <input type="number" value={a2} id="a2" onChange={e=>setA2(e.target.value)} />
10      <button onClick={()=>setSum(Number(a1)+Number(a2))}>=</button>
11      <input type="number" value={sum} readOnly/>
12    </div>
13  )
14 }
15 export default Calc;
```

- function `update` returns function which is called as a event handler

```
1 import { useState } from 'react';
2 function Calc() {
3   const [ a1, setA1 ] = useState('0')
4   const [ a2, setA2 ] = useState('0')
5   const [sum, setSum] = useState<number>(0);
6   const update=(setA:React.Dispatch<React.SetStateAction<string>>) => {
7     return (e: React.ChangeEvent<HTMLInputElement>)=>setA(e.target.value)
8   }
9   return (
10     <div>
11       <input type="number" value={a1} id="a1" onChange={update(setA1)}/>+
12       <input type="number" value={a2} id="a2" onChange={update(setA2)}/>
13       <button onClick={()=>setSum(Number(a1)+Number(a2))}>=</button>
14       <input type="number" value={sum} readOnly/>
15     </div>
16   )
17 }
18 export default Calc;
```

- Hook `useReducer` jest alternatywą dla hooka `useState` i jest lepszy, gdy logika stanu jest złożona lub gdy następny stan zależy od poprzedniego stanu
- Hook `useReducer` potrzebuje funkcji `(state, action)=>newState` oraz wartości początkowej stanu i zwraca aktualny obiekt stanu sparowany z metodą `dispatch`, podobnie jak w Redux.

- Hook `useReducer` jest alternatywą dla hooka `useState` i jest lepszy, gdy logika stanu jest złożona lub gdy następny stan zależy od poprzedniego stanu
- Hook `useReducer` potrzebuje funkcji `(state, action)=>newState` oraz wartości początkowej stanu i zwraca aktualny obiekt stanu sparowany z metodą `dispatch`, podobnie jak w Redux.

- **typy**
 - funkcja redukująca
 - użycie

```
1 // możliwe akcje
2 enum Kind {
3   INCREASE = 'INCREASE',
4   DECREASE = 'DECREASE',
5 }
6
7 // interfejs dla akcji
8 interface CountAction {
9   type: Kind;
10  payload: number;
11 }
12
13 // Interfejs dla stanu
14 interface CountState {
15   count: number;
16 }
```

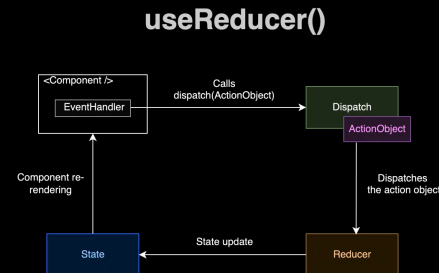
- Hook `useReducer` jest alternatywą dla hooka `useState` i jest lepszy, gdy logika stanu jest złożona lub gdy następny stan zależy od poprzedniego stanu
- Hook `useReducer` potrzebuje funkcji `(state, action) => newState` oraz wartości początkowej stanu i zwraca aktualny obiekt stanu sparowany z metodą `dispatch`, podobnie jak w Redux.
 - typy
 - funkcja redukująca
 - użycie

```
1 function reFun(state: CountState,  
2           action: CountAction) {  
3   const { type, payload } = action;  
4   switch (type) {  
5     case Kind.INCREASE:  
6       return {  
7         ...state,  
8         value: state.count + payload,  
9       };  
10    case Kind.DECREASE:  
11      return {  
12        ...state,  
13        value: state.count - payload,  
14      };  
15    default: return state;  
16  }  
17 }
```

- Hook `useReducer` jest alternatywą dla hooka `useState` i jest lepszy, gdy logika stanu jest złożona lub gdy następny stan zależy od poprzedniego stanu
- Hook `useReducer` potrzebuje funkcji `(state, action) => newState` oraz wartości początkowej stanu i zwraca aktualny obiekt stanu sparowany z metodą `dispatch`, podobnie jak w Redux.
 - typy
 - funkcja redukująca
 - użycie

```
1 const Counter: FunctionComponent = () => {  
2   const [state, dispatch] = useReducer(reFun,  
3                                     { count: 0 });  
4   return (  
5     <div>  
6       Count: {state.count}  
7       {/* Przyciski wywołujące akcje */}  
8       <button  
9         onClick={() => dispatch(  
10           { type: Kind.INCREASE, payload: 5 }}}  
11       >-</button>  
12       <button onClick={() => dispatch(  
13         { type: Kind.DECREASE, payload: 5 }}}  
14       >+</button>  
15     </div>  
16   );  
17 };
```

- Hook `useReducer` jest alternatywą dla hooka `useState` i jest lepszy, gdy logika stanu jest złożona lub gdy następny stan zależy od poprzedniego stanu
- Hook `useReducer` potrzebuje funkcji `(state, action) = newState` oraz wartości początkowej stanu i zwraca aktualny obiekt stanu sparowany z metodą `dispatch`, podobnie jak w Redux.
 - typy
 - funkcja redukująca
 - użycie



- Hooki można używać tylko w komponentach funkcyjnych. Nie można ich używać w komponentach klasowych!

```
1 import React, { useState } from 'react';
2 function Recipes() {
3   const [ name, setName ] = useState('')
4   function handleChange (evt) {
5     setName(evt.target.value)
6   }
7   return (
8     <div>
9       <h1>My Name is: {name}</h1>
10      <input type="text" value={name} onChange={handleChange} />
11    </div>
12  )
13 }
14 export default Recipes;
```

- Hooki można używać tylko w komponentach funkcyjnych. Nie można ich używać w komponentach klasowych!
- Ponieważ nie mamy teraz do czynienia z klasami, nie ma już potrzeby robienia `bind`!

```
1 import React, { useState } from 'react';
2 function Recipes() {
3   const [ name, setName ] = useState('')
4   function handleChange (evt) {
5     setName(evt.target.value)
6   }
7   return (
8     <div>
9       <h1>My Name is: {name}</h1>
10      <input type="text" value={name} onChange={handleChange} />
11    </div>
12  )
13 }
14 export default Recipes;
```

- Hooki można używać tylko w komponentach funkcyjnych. Nie można ich używać w komponentach klasowych!
- Ponieważ nie mamy teraz do czynienia z klasami, nie ma już potrzeby robienia `bind`!
- Kolejność definicji hooków ma znaczenie, dlatego nie możemy wstawiać hooków w przypadku funkcji warunkowych, pętli lub zagnieżdżonych

```
1 import React, { useState } from 'react';
2 function Recipes() {
3   const [ name, setName ] = useState('')
4   function handleChange (evt) {
5     setName(evt.target.value)
6   }
7   return (
8     <div>
9       <h1>My Name is: {name}</h1>
10      <input type="text" value={name} onChange={handleChange} />
11    </div>
12  )
13 }
14 export default Recipes;
```

Podstawowe hooki zapewniają najczęściej potrzebną funkcjonalność aplikacji React

- `useState`

-

-

Hook `useState` zastępuje `this.state` i `this.setState()` i zwraca:

- stateful value (`state`)
- setter function(`setState`)

```
1 import { useState } from 'react'
2 const [ state, setState ] = useState(initialState)
```

Podstawowe hooki zapewniają najczęściej potrzebną funkcjonalność aplikacji React

- useState
- [useEffect](#)
-

Hook `useEffect` służy do obsługi zdarzeń, takich jak liczniki czasu, subskrypcje, żądania i tak dalej. Możemy go użyć w następujący sposób:

```
1 import { useEffect } from 'react'
2
3 useEffect (didUpdate)
```

Hook `useEffect` zastępuje metody `componentDidMount`, `componentDidUpdate` i `componentWillUnmount`

Podstawowe hooki zapewniają najczęściej potrzebną funkcjonalność aplikacji React

- useState
- useEffect
- useContext

Hook useContext jest używany do obsługi kontekstu w Reakcie. Możemy go użyć w następujący sposób:

```
1 import { useContext } from 'react'
2
3 const value = useContext(MyContext)
```

Hook useContext zastępuje konsumentów kontekstu.

- Przechowywanie danych
 - database
 - ręczna implementacja (node.js, springboot, c#)
 - ORM (Sequelize, JPA, Entity Framework)
 - MongoDB?
 - pliki *.json
 - json-server - rozwiązanie z półki (node.js)
 - ręczna implementacja (node.js, springboot, c#)

- Przechowywanie danych
 - database
 - ręczna implementacja (node.js, springboot, c#)
 - ORM (Sequelize, JPA, Entity Framework)
 - MongoDB?
 - pliki *.json
 - json-server - rozwiązanie z półki (node.js)
 - ręczna implementacja (node.js, springboot, c#)

- Przechowywanie danych
 - database
 - ręczna implementacja (node.js, springboot, c#)
 - ORM (Sequelize, JPA, Entity Framework)
 - MongoDB?
 - pliki *.json
 - json-server - rozwiązanie z półki (node.js)
 - ręczna implementacja (node.js, springboot, c#)

- Przechowywanie danych
 - database
 - ręczna implementacja ([node.js](#), [springboot](#), [c#](#))
 - **ORM** ([Sequelize](#), [JPA](#), [Entity Framework](#))
 - MongoDB?
 - pliki *.json
 - json-server - rozwiązanie z pólki ([node.js](#))
 - ręczna implementacja ([node.js](#), [springboot](#), [c#](#))

- Przechowywanie danych
 - database
 - ręczna implementacja ([node.js](#), [springboot](#), [c#](#))
 - ORM ([Sequelize](#), [JPA](#), [Entity Framework](#))
 - [MongoDB?](#)
 - pliki *.json
 - json-server - rozwiązanie z półki ([node.js](#))
 - ręczna implementacja ([node.js](#), [springboot](#), [c#](#))

- Przechowywanie danych
 - database
 - ręczna implementacja ([node.js](#), [springboot](#), [c#](#))
 - ORM ([Sequelize](#), [JPA](#), [Entity Framework](#))
 - MongoDB?
 - pliki *.json
 - json-server - rozwiązanie z półki ([node.js](#))
 - ręczna implementacja ([node.js](#), [springboot](#), [c#](#))

- Przechowywanie danych
 - database
 - ręczna implementacja ([node.js](#), [springboot](#), [c#](#))
 - ORM ([Sequelize](#), [JPA](#), [Entity Framework](#))
 - MongoDB?
 - pliki *.json
 - [json-server](#) - rozwiązanie z półki ([node.js](#))
 - ręczna implementacja ([node.js](#), [springboot](#), [c#](#))

- Przechowywanie danych
 - database
 - ręczna implementacja ([node.js](#), [springboot](#), [c#](#))
 - ORM ([Sequelize](#), [JPA](#), [Entity Framework](#))
 - MongoDB?
 - pliki *.json
 - json-server - rozwiązanie z półki ([node.js](#))
 - ręczna implementacja ([node.js](#), [springboot](#), [c#](#))

- **install**

```
i npm install -g json-server
```

- datafile db.json
- uruchomienie
 - odwołanie po identyfikatorze a nie po pozycji w tablicy
- odwołania do tablic
 - filtrowanie
 - stronicowanie
- odwołania do pojedynczego obiektu

- install
- datafile db.json
- uruchomienie
 - odwołanie po identyfikatorze a nie po pozycji w tablicy
- odwołania do tablic
 - filtrowanie
 - stronicowanie
- odwołania do pojedynczego obiektu

```
1 {  
2   "employees": [  
3     { "id": 1, "firstname": "Jeremy", "lastname": "Clarkson" },  
4     { "id": 3, "firstname": "James", "lastname": "May" },  
5     { "id": 7, "firstname": "Richard", "lastname": "Hammond" }  
6   ],  
7   "profile": {  
8     "name": "Top Gear",  
9     "status": {  
10      "broadcast": "BBC"  
11    }  
12  }  
13 }
```

- install
- datafile db.json
- uruchomienie
 - odwołanie po identyfikatorze a nie po pozycji w tablicy
- odwołania do tablic
 - filtrowanie
 - stronicowanie
- odwołania do pojedynczego obiektu

routes.json

```
1 {  
2   "/api/*": "$1"  
3 }
```

run:

```
1 json-server --watch db.json --port 5000 --routes routes.json
```

- install
- datafile db.json
- uruchomienie
 - odwołanie po identyfikatorze a nie po pozycji w tablicy
- odwołania do tablic
 - filtrowanie
 - stronicowanie
- odwołania do pojedynczego obiektu

```
1 http://localhost:5000/employees/3
```

daje:

```
1 {  
2   "id": 3,  
3   "firstname": "James",  
4   "lastname": "May"  
5 }
```

- install
- datafile db.json
- uruchomienie
 - odwołanie po identyfikatorze a nie po pozycji w tablicy
- **odwołania do tablic**
 - filtrowanie
 - stronicowanie
- odwołania do pojedynczego obiektu

```
1 GET      /employees
2 GET      /employees/1
3 POST     /employees
4 PUT      /employees/1
5 PATCH    /employees/1
6 DELETE   /employees/1
```

- install
- datafile db.json
- uruchomienie
 - odwołanie po identyfikatorze a nie po pozycji w tablicy
- odwołania do tablic
 - filtrowanie
 - stronicowanie
- odwołania do pojedynczego obiektu

```
1 http://localhost:5000/employees?lastname=May&lastname=Clarkson
```

daje:

```
1 [  
2   {  
3     "id": 1,  
4     "firstname": "Jeremy",  
5     "lastname": "Clarkson"  
6   },  
7   {  
8     "id": 3,  
9     "firstname": "James",  
10    "lastname": "May"  
11  }  
12 ]
```

- install
- datafile db.json
- uruchomienie
 - odwołanie po identyfikatorze a nie po pozycji w tablicy
- odwołania do tablic
 - filtrowanie
 - **stronicowanie**
- odwołania do pojedynczego obiektu

1

daje:

```
1 [
2   {
3     "id": 7,
4     "firstname": "Richard",
5     "lastname": "Hammond"
6   }
7 ]
```

- install
- datafile db.json
- uruchomienie
 - odwołanie po identyfikatorze a nie po pozycji w tablicy
- odwołania do tablic
 - filtrowanie
 - stronicowanie
- odwołania do pojedynczego obiektu

```
1 GET    /profile
2 POST   /profile
3 PUT    /profile
4 PATCH  /profile
```

- **id wstawianego obiektu (int)**
 - brak id - wstawia max+1
 - id == 0 - wstawia max+1
 - id == n - wstawia n
 - błąd gdy
 powtórzenie
- zwraca wstaw-
iony/zmodyfikowany
obiekt

- id wstawianego obiektu (int)
 - brak id - wstawia max+1
 - id == 0 - wstawia max+1
 - id == n - wstawia n
 - błąd gdy
 powtórzenie
- zwraca wstaw-
iony/zmodyfikowany
obiekt

- id wstawianego obiektu (int)
 - brak id - wstawia max+1
 - id == 0 - wstawia max+1
 - id == n - wstawia n
 - błąd gdy powtórzenie
- zwraca wstawiony/zmodyfikowany obiekt

- id wstawianego obiektu (int)
 - brak id - wstawia max+1
 - id == 0 - wstawia max+1
 - id == n - wstawia n
 - błąd gdy
 powtórzenie
- zwraca wstaw-
iony/zmodyfikowany
obiekt

- id wstawianego obiektu (int)
 - brak id - wstawia max+1
 - id == 0 - wstawia max+1
 - id == n - wstawia n
 - błąd gdy
powtórzenie
- zwraca wstaw-
iony/zmodyfikowany
obiekt

- id wstawianego obiektu (int)
 - brak id - wstawia max+1
 - id == 0 - wstawia max+1
 - id == n - wstawia n
 - błąd gdy
powtórzenie
- zwraca wstaw-
iony/zmodyfikowany
obiekt

```
1 const express = require('express');
2 const app = express();
3 const data = [
4   {"id": 1, "firstname": "Jeremy", "lastname": "Clarkson"},
5   {"id": 3, "firstname": "James", "lastname": "May"},
6   {"id": 7, "firstname": "Richard", "lastname": "Hammond"}
7 ]
8
9 app.get('/employees', (req, res) => res.json(data))
10 app.listen(5000)
```

server.js:

```
1 const express = require('express');
2 const fs = require('fs');
3 const app = express();
4 const data = JSON.parse(fs.readFileSync('db.json'))
5
6 app.get('/employees', (req, res) => res.json(data.employees))
7 app.listen(5000)
```

db.json:

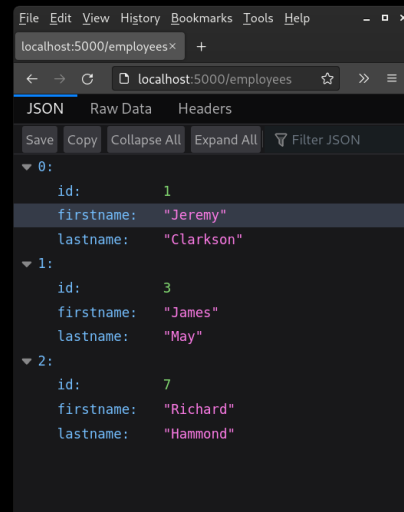
```
1 {
2   "employees": [
3     {"id": 1, "firstname": "Jeremy", "lastname": "Clarkson"},
4     {"id": 3, "firstname": "James", "lastname": "May"},
5     {"id": 7, "firstname": "Richard", "lastname": "Hammond"}
6   ]
7 }
```

./routes/employeeRouter.js:

```
1 var express = require('express');
2 var router = express.Router();
3
4 const fs = require('fs');
5 const data = JSON.parse(fs.readFileSync('db.json'))
6
7 router.get('/', (req, res) => res.json(data.employees))
8 module.exports = router;
```

server.js:

```
1 const express = require('express');
2 const app = express();
3 var employeeRouter =
4   ↪ require('./routes/employeeRouter');
5 app.use('/employees', employeeRouter);
6 app.listen(5000)
```



data.json

- plik z danymi
- odczyt wszystkich pracowników
- utworzenie nowego
- modyfikacja
- usunięcie

```
1 {  
2   "employees": [  
3     {"employeeid":1, "firstname":"Jeremy", "lastname":"Clarkson", "salary":123.45, "departmentid":1},  
4     {"employeeid":1, "firstname":"Richard", "lastname":"Hammond", "salary":123.45, "departmentid":1},  
5     {"employeeid":1, "firstname":"James", "lastname":"May", "salary":123.45, "departmentid":2}  
6   ],  
7   "departments": [  
8     {"departmentid":1, "name":"Top Gear"},  
9     {"departmentid":2, "name":"Grand Tour"}  
10  ]  
11 }
```

- plik z danymi
- odczyt wszystkich pracowników
- utworzenie nowego
- modyfikacja
- usunięcie

```
1 http://localhost:3000/api/employees
```

daje:

```
1 [{"employeeid":1,"firstname":"Jeremy","lastname":"Clarkson",
2 {"employeeid":1,"firstname":"Richard","lastname":"Hammond",
3 {"employeeid":1,"firstname":"James","lastname":"May","salary":150000}
```

app.js:

```
1  ...
2  var employeeRestRouter = require('./api/employees');
3  ...
4  app.use('/api', employeeRestRouter);
5  ...
```

/api/employees.js:

```

1 var express = require('express');
2 var router = express.Router();
3
4 /* GET employees */
5 router.get('/employees', (req, res) => {
6   const fs = require('fs')
7   let data = JSON.parse(fs.readFileSync('./data.json'))
8   res.json(data.employees);
9 });
10 module.exports = router;

```

- plik z danymi
- odczyt wszystkich pracowników
- **utworzenie nowego**
- modyfikacja
- usunięcie

```
1  ...
2  router.put('/employees', (req, res) => { // save new employee
3    const employee = req.body;
4    let data = JSON.parse(fs.readFileSync('./data.json'))
5    data.employees.push(employee);
6    fs.writeFileSync("data.json", JSON.stringify(data, null, 2), 'utf8', err=> {
7      if (err) { //in case of error when writing
8        console.log("An error occured while writing JSON to File.");
9        return console.log(err);
10     }
11     console.log("JSON file has been saved.");
12   });
13   res.json({employeeid:employee.employeeid});
14 });
15 module.exports = router;
```

- plik z danymi
- odczyt wszystkich pracowników
- utworzenie nowego
- **modyfikacja**
- **usunięcie**

```
1 router.post('/employees', (req, res) => {
2   const employee = req.body;
3   let data = JSON.parse(fs.readFileSync('./data.json'))
4   const found = data.employees.find(e=>e.employeeid === employee.employeeid)
5   if(found) {
6     for(p in employee)
7       found[p] = employee[p]
8     fs.writeFileSync("data.json", JSON.stringify(data, null, 2), 'utf8', function (err) {
9       if (err) { //in case of error when writing
10         console.log("An error occurred while writing JSON to File.");
11         return console.log(err);
12       }
13       console.log("JSON file has been saved."); //this is printed in the console
14     });
15   }
16   res.json({employeeid:employee.employeeid}); //this is a message displayed in the web
17 });
```

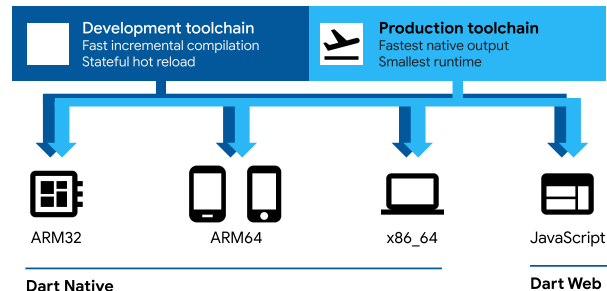
- plik z danymi
- odczyt wszystkich pracowników
- utworzenie nowego
- modyfikacja
- **usunięcie**

```
1 router.delete('/employees', (req, res) => {
2   const employee = req.body;
3   let data = JSON.parse(fs.readFileSync('./data.json'))
4   const found = data.employees.find(e=>e.employeeid === employee.employeeid)
5   if(found) {
6     data.employees = data.employees.filter(e=>e.employeeid !== employee.employeeid)
7     fs.writeFileSync("data.json", JSON.stringify(data, null, 2), 'utf8', function (err) {
8       if (err) { //in case of error when writing
9         console.log("An error occurred while writing JSON to File.");
10        return console.log(err);
11      }
12      console.log("JSON file has been saved."); //this is printed in the console
13    });
14  }
15  res.json({employeeid:employee.employeeid}); //this is a message displayed in the web
16 });
```

- plik z danymi
- odczyt wszystkich pracowników
- utworzenie nowego
- modyfikacja
- usunięcie

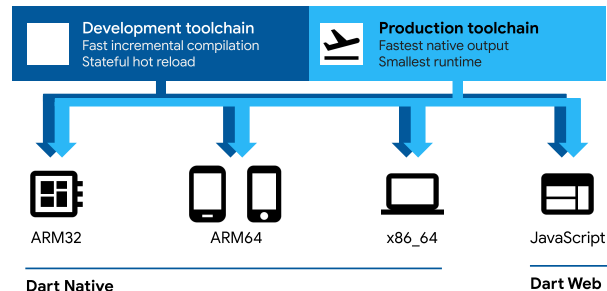
```
1 router.get('/employees/:employeeid', (req, res) => {
2   const employeeid = parseInt(req.params.employeeid);
3   let data = JSON.parse(fs.readFileSync('./data.json'))
4   const found = data.employees.find(e=>e.employeeid === employeeid)
5   if(found) {
6     res.json(found)
7   } else
8     res.json({employeeid:0}); //this is a message displayed in the web
9 })
```

- Dart jest językiem obiektowym i zawiera takie mechanizmy jak hermetyzacja, dziedziczenie, polimorfizm i interfejs
- Dart zawiera kompilator dart2js, który tłumaczy kod dart na kod JavaScript, działający we wszystkich nowoczesnych przeglądarkach.



```
1 sudo apt-get install apt-transport-https
2 wget -qO- https://dl-ssl.google.com/linux/linux_signing_key.pub | sudo gpg --dearmor -o
   ↪ /usr/share/keyrings/dart.gpg
3 echo 'deb [signed-by=/usr/share/keyrings/dart.gpg arch=amd64]
   ↪ https://storage.googleapis.com/download.dartlang.org/linux/debian stable main' | sudo tee
   ↪ /etc/apt/sources.list.d/dart_stable.list
4 sudo apt-get update
5 sudo apt-get install dart
```

- Dart jest językiem obiektowym i zawiera takie mechanizmy jak hermetyzacja, dziedziczenie, polimorfizm i interfejs
- Dart zawiera kompilator `dart2js`, który tłumaczy kod dart na kod JavaScript, działający we wszystkich nowoczesnych przeglądarkach.

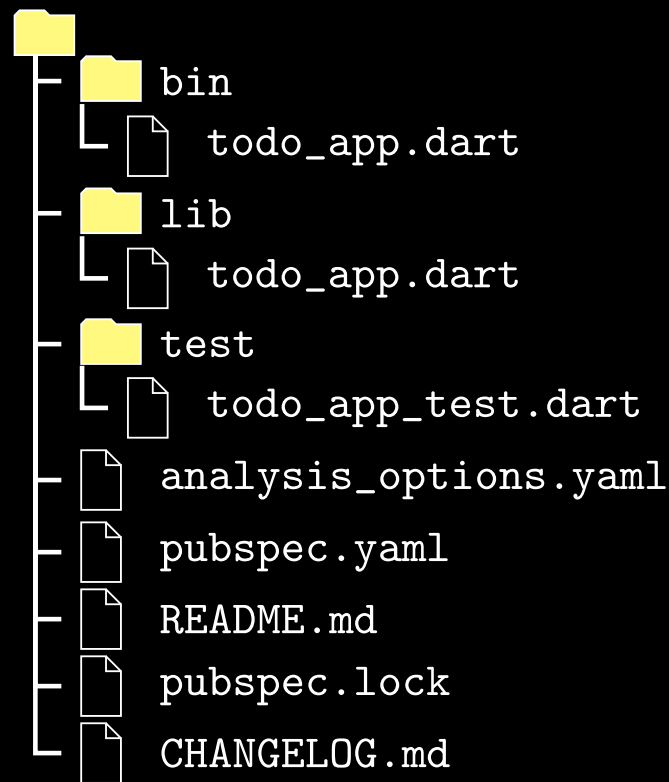


```
1 sudo apt-get install apt-transport-https
2 wget -qO- https://dl-ssl.google.com/linux/linux_signing_key.pub | sudo gpg --dearmor -o
  ↳ /usr/share/keyrings/dart.gpg
3 echo 'deb [signed-by=/usr/share/keyrings/dart.gpg arch=amd64]
  ↳ https://storage.googleapis.com/download.dartlang.org/linux/debian stable main' | sudo tee
  ↳ /etc/apt/sources.list.d/dart_stable.list
4 sudo apt-get update
5 sudo apt-get install dart
```

- **Utworzenie projektu Dart**

- start w 'bin/'
- biblioteka (kod) w 'lib/'
- przykładowe testy jednostkowe w 'test/'

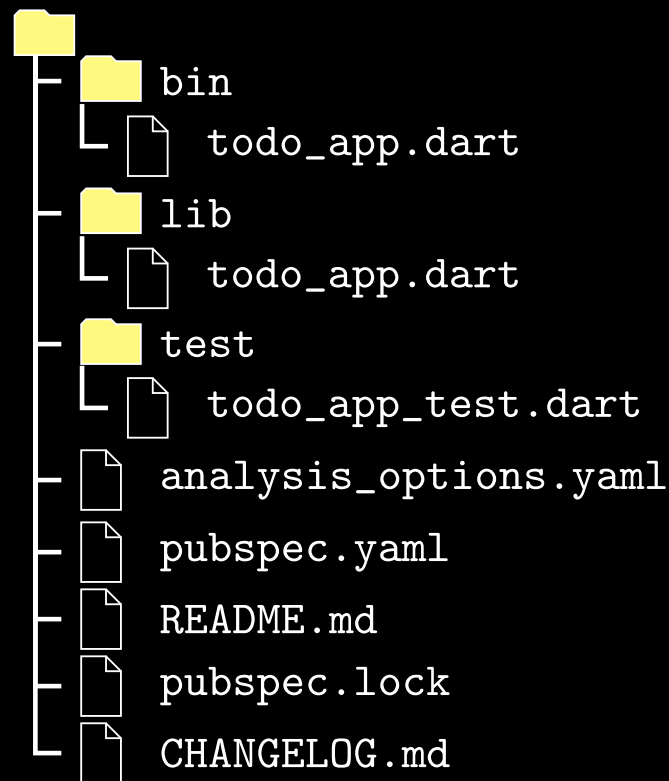
```
1 dart create todo_app
2 cd todo_app
```



- Utworzenie projektu Dart

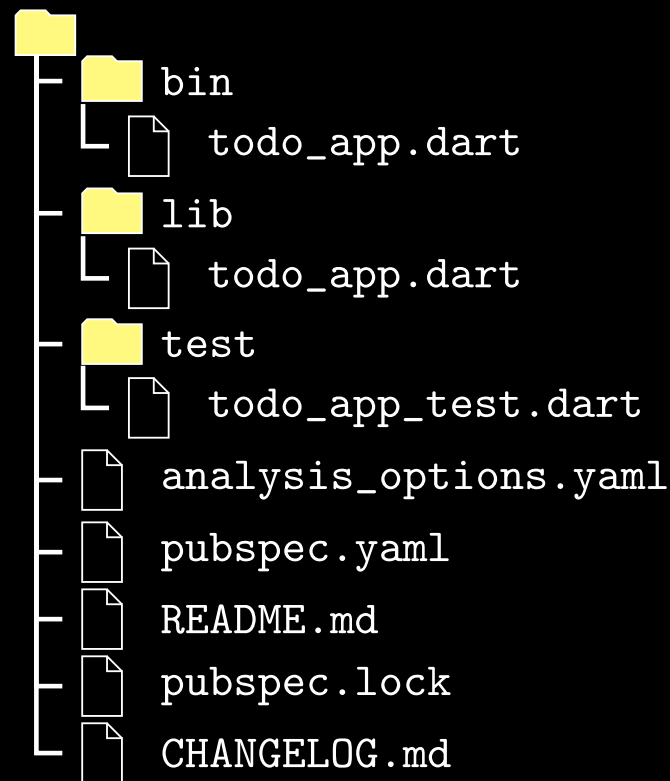
- start w 'bin/'
- biblioteka (kod) w 'lib/'
- przykładowe testy jednostkowe w 'test/'

```
1 import 'package:todo_app/todo_app.dart' as todo_app;
2
3 void main(List<String> arguments) {
4   print('Hello world: ${todo_app.calculate()}!');
5 }
```



- Utworzenie projektu Dart
 - start w 'bin/'
 - **biblioteka (kod) w 'lib/'**
 - przykładowe testy jednostkowe w 'test/'

```
1 int calculate() {
2   return 6 * 7;
3 }
4
```

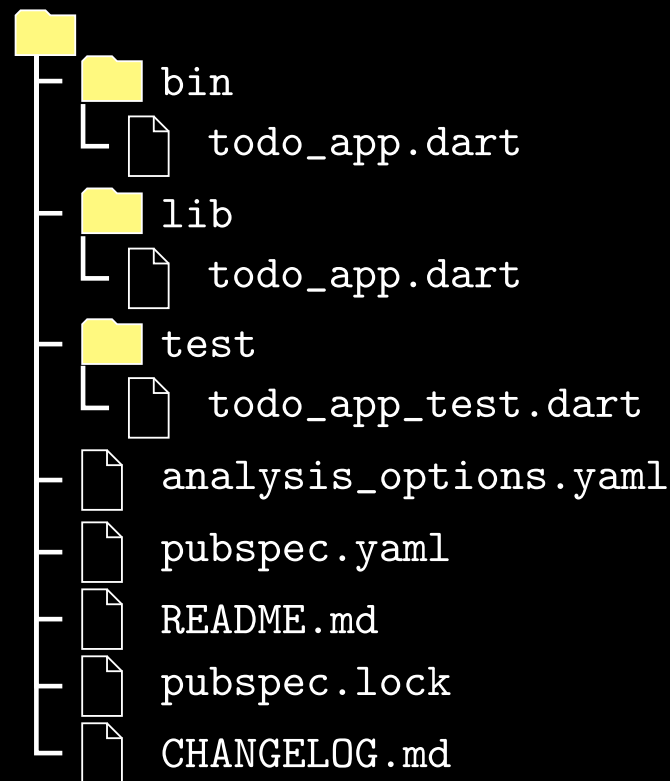


- Utworzenie projektu Dart
 - start w 'bin/'
 - biblioteka (kod) w 'lib/'
 - przykładowe testy jednostkowe w 'test/'

```

1 import 'package:todo_app/todo_app.dart';
2 import 'package:test/test.dart';
3
4 void main() {
5   test('calculate', () {
6     expect(calculate(), 42);
7   });
8 }
9

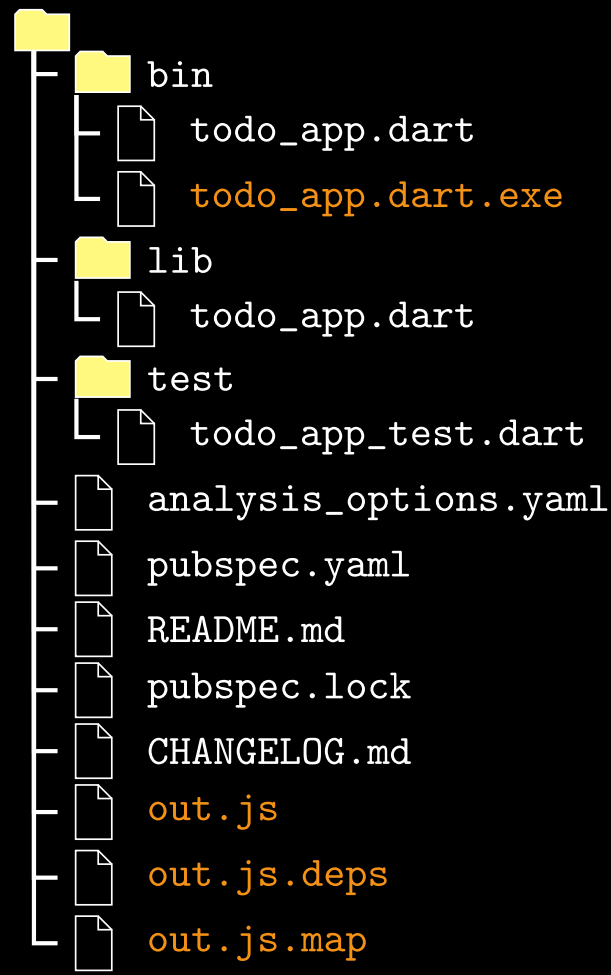
```



- uruchomienie interpretera, lub

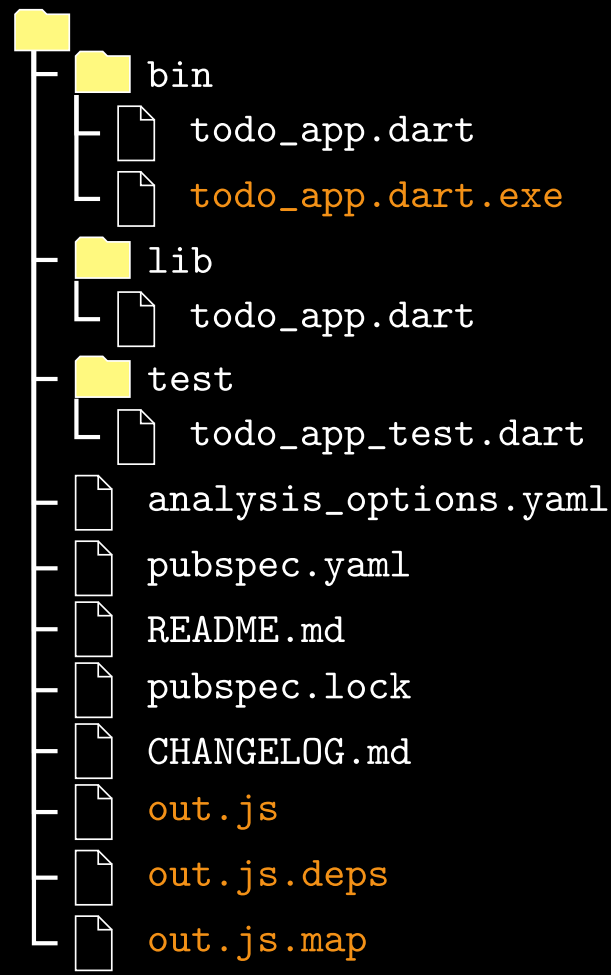
- kompilowanie do exe (tworzy dość duży plik wykonywalny zawierający wymagane biblioteki, rozszerzenie *.exe nawet w systemie Linux)
- kompilowanie do js (tworzy dość duży plik out.js w bieżącym folderze, który można uruchomić za pomocą node)

```
1 dart bin/todo_app.dart
```



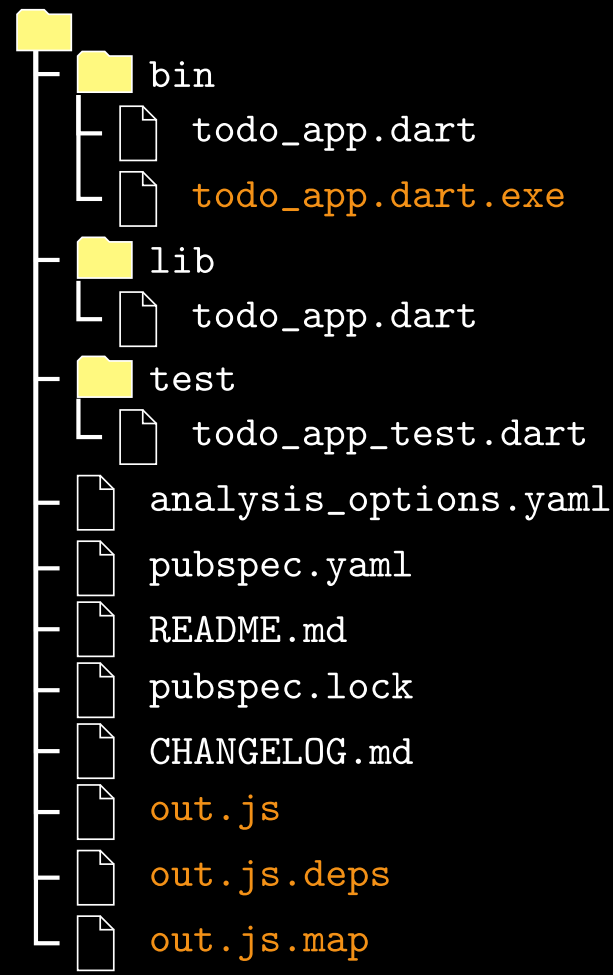
- uruchomienie interpretera, lub
 - kompilowanie do exe (tworzy dość duży plik wykonywalny zawierający wymagane biblioteki, rozszerzenie *.exe nawet w systemie Linux)
 - kompilowanie do js (tworzy dość duży plik out.js w bieżącym folderze, który można uruchomić za pomocą node)

```
1 dart compile exe bin/todo_app.dart
2 Generated: bin/todo_app.exe
```



- uruchomienie interpretera, lub
 - kompilowanie do exe (tworzy dość duży plik wykonywalny zawierający wymagane biblioteki, rozszerzenie *.exe nawet w systemie Linux)
 - kompilowanie do js (tworzy dość duży plik `out.js` w bieżącym folderze, który można uruchomić za pomocą `node`)

```
1 dart compile js bin/todo_app.dart
2 Compiled 10,141,708 input bytes (5,140,315 characters source) to
  ↳ 97,927 characters JavaScript in 0.46 seconds using 0.000 MB of
  ↳ memory
```



- lowerCamelCase

```
1 import 'package:todo_app/todo_app.dart' as todo_app;
2
3 void main(List<String> arguments) {
4   String name = "Jeremy Clarkson";
5   String address = "Cotswolds";
6   num age = 70; // int age 70;
7   num height = 6.5; // num for both int and double
8   bool isMarried = true;
9   var status = "Married" // type from value
10
11   print("Name is $name");
12   print("Address is $address");
13   print("Age is $age");
14   print("Height is $height");
15   print("Married Status is $isMarried");
16   print('Hello world: ${todo_app.calculate()}!');
17 }
```

Typ	Keyword	Opis
Liczby	int, double, num	reprezentuje wartości liczbowe
Stringi	String	Reprezentuje ciąg znaków
Booleans	bool	Reprezentuje wartości boolowskie: <code>true</code> i <code>false</code>
Listy	List	Jest to uporządkowana grupa elementów
Słowniki	Map	Reprezentuje zbiór wartości jako pary klucz-wartość
Zbiory	Set	Jest to nieuporządkowana lista unikalnych wartości tego samego typu
Elementy stringu	runes	Reprezentuje wartości Unicode ciągu
Wartość pusta	null	Reprezentuje wartość null

```
1 void main() {  
2   double price = 1234.5678901234;  
3   print(price.toStringAsFixed(2));  
4 }
```

Typ	Keyword	Opis
Liczby	int, double, num	reprezentuje wartości liczbowe
Stringi	String	Reprezentuje ciąg znaków
Booleans	bool	Reprezentuje wartości boolowskie: <code>true</code> i <code>false</code>
Listy	List	Jest to uporządkowana grupa elementów
Słowniki	Map	Reprezentuje zbiór wartości jako pary klucz-wartość
Zbiory	Set	Jest to nieuporządkowana lista unikalnych wartości tego samego typu
Elementy stringu	runes	Reprezentuje wartości Unicode ciągu
Wartość pusta	null	Reprezentuje wartość null

```

1 void main() {
2     var line = "Line";
3     String multiLine = ''' First $line of Text
4     Second $line of Text
5     '''; // you can also use """
6     print(multiLine);
7 }

```

Typ	Keyword	Opis
Liczby	int, double, num	reprezentuje wartości liczbowe
Stringi	String	Reprezentuje ciąg znaków
Booleans	bool	Reprezentuje wartości boolowskie: <code>true</code> i <code>false</code>
Listy	List	Jest to uporządkowana grupa elementów
Słowniki	Map	Reprezentuje zbiór wartości jako pary klucz-wartość
Zbiory	Set	Jest to nieuporządkowana lista unikalnych wartości tego samego typu
Elementy stringu	runes	Reprezentuje wartości Unicode ciągu
Wartość pusta	null	Reprezentuje wartość null

```

1 void main() {
2     var line = "Line";
3     String multiLine = r''' First $line of Text
4     Second $line of Text
5     '''; // raw string => $line not interpreted
6     print(multiLine);
7 }

```

Typ	Keyword	Opis
Liczby	int, double, num	reprezentuje wartości liczbowe
Stringi	String	Reprezentuje ciąg znaków
Booleans	bool	Reprezentuje wartości boolowskie: <code>true</code> i <code>false</code>
Listy	List	Jest to uporządkowana grupa elementów
Słowniki	Map	Reprezentuje zbiór wartości jako pary klucz-wartość
Zbiory	Set	Jest to nieuporządkowana lista unikalnych wartości tego samego typu
Elementy stringu	runes	Reprezentuje wartości Unicode ciągu
Wartość pusta	null	Reprezentuje wartość null

```
1 void main() {  
2   var svalue = "123.456";  
3   int ivalue = int.parse(svalue); // exception  
4   print(ivalue);  
5 }
```

Typ	Keyword	Opis
Liczby	int, double, num	reprezentuje wartości liczbowe
Stringi	String	Reprezentuje ciąg znaków
Booleans	bool	Reprezentuje wartości boolowskie: <code>true</code> i <code>false</code>
Listy	List	Jest to uporządkowana grupa elementów
Słowniki	Map	Reprezentuje zbiór wartości jako pary klucz-wartość
Zbiory	Set	Jest to nieuporządkowana lista unikalnych wartości tego samego typu
Elementy stringu	runes	Reprezentuje wartości Unicode ciągu
Wartość pusta	null	Reprezentuje wartość null

```

1 void main() {
2   var svalue = "123.456";
3   num ivalue = num.parse(svalue); // OK
4   print(ivalue);                 // 123.456
5   print(ivalue.runtimeType);     // double
6 }
```

Typ	Keyword	Opis
Liczby	int, double, num	reprezentuje wartości liczbowe
Stringi	String	Reprezentuje ciąg znaków
Booleans	bool	Reprezentuje wartości boolowskie: <code>true</code> i <code>false</code>
Listy	List	Jest to uporządkowana grupa elementów
Słowniki	Map	Reprezentuje zbiór wartości jako pary klucz-wartość
Zbiory	Set	Jest to nieuporządkowana lista unikalnych wartości tego samego typu
Elementy stringu	runes	Reprezentuje wartości Unicode ciągu
Wartość pusta	null	Reprezentuje wartość null

```

1 void main() {
2   var svalue = "123";
3   num ivalue = num.parse(svalue); // OK
4   print(ivalue);                  // 123
5   print(ivalue.runtimeType);      // int
6 }

```

Typ	Keyword	Opis
Liczby	int, double, num	reprezentuje wartości liczbowe
Stringi	String	Reprezentuje ciąg znaków
Booleans	bool	Reprezentuje wartości boolowskie: <code>true</code> i <code>false</code>
Listy	List	Jest to uporządkowana grupa elementów
Słowniki	Map	Reprezentuje zbiór wartości jako pary klucz-wartość
Zbiory	Set	Jest to nieuporządkowana lista unikalnych wartości tego samego typu
Elementy stringu	runes	Reprezentuje wartości Unicode ciągu
Wartość pusta	null	Reprezentuje wartość null

```
1 void main() {  
2   List<String> names = ["Jeremy", "James", "Richard"];  
3   print("names = $names"); // names = ["Jeremy", "James", "Richard"]  
4 }
```

Typ	Keyword	Opis
Liczby	int, double, num	reprezentuje wartości liczbowe
Stringi	String	Reprezentuje ciąg znaków
Booleans	bool	Reprezentuje wartości boolowskie: <code>true</code> i <code>false</code>
Listy	List	Jest to uporządkowana grupa elementów
Słowniki	Map	Reprezentuje zbiór wartości jako pary klucz-wartość
Zbiory	Set	Jest to nieuporządkowana lista unikalnych wartości tego samego typu
Elementy stringu	runes	Reprezentuje wartości Unicode ciągu
Wartość pusta	null	Reprezentuje wartość null

```

1 void main() {
2     Set<String> names = {"Jeremy", "James", "Richard", "James"};
3     print("names =
4 }
```

Typ	Keyword	Opis
Liczby	int, double, num	reprezentuje wartości liczbowe
Stringi	String	Reprezentuje ciąg znaków
Booleans	bool	Reprezentuje wartości boolowskie: <code>true</code> i <code>false</code>
Listy	List	Jest to uporządkowana grupa elementów
Słowniki	Map	Reprezentuje zbiór wartości jako pary klucz-wartość
Zbiory	Set	Jest to nieuporządkowana lista unikalnych wartości tego samego typu
Elementy stringu	runes	Reprezentuje wartości Unicode ciągu
Wartość pusta	null	Reprezentuje wartość null

```

1 void main() {
2   Map<String, int> ages = {'Jeremy': 70, 'Richard': 50};
3   int? age = ages["James"]; // nullable type
4   print(age);               // null
5   print(age.runtimeType);    // Null
6 }

```

Typ	Keyword	Opis
Liczby	int, double, num	reprezentuje wartości liczbowe
Stringi	String	Reprezentuje ciąg znaków
Booleans	bool	Reprezentuje wartości boolowskie: <code>true</code> i <code>false</code>
Listy	List	Jest to uporządkowana grupa elementów
Słowniki	Map	Reprezentuje zbiór wartości jako pary klucz-wartość
Zbiory	Set	Jest to nieuporządkowana lista unikalnych wartości tego samego typu
Elementy stringu	runes	Reprezentuje wartości Unicode ciągu
Wartość pusta	null	Reprezentuje wartość null

```

1 void main() {
2   Map<String, int> ages = {'Jeremy': 70, 'Richard': 50};
3   int age = ages["James"]!; // exception if null
4   print(age);
5   print(age.runtimeType);
6 }

```

Typ	Keyword	Opis
Liczby	int, double, num	reprezentuje wartości liczbowe
Stringi	String	Reprezentuje ciąg znaków
Booleans	bool	Reprezentuje wartości boolowskie: <code>true</code> i <code>false</code>
Listy	List	Jest to uporządkowana grupa elementów
Słowniki	Map	Reprezentuje zbiór wartości jako pary klucz-wartość
Zbiory	Set	Jest to nieuporządkowana lista unikalnych wartości tego samego typu
Elementy stringu	runes	Reprezentuje wartości Unicode ciągu
Wartość pusta	null	Reprezentuje wartość null

```

1 void main() {
2   String polish = "ACEace";
3   Runes l = polish.runes;
4   print(l); // (260, 262, 280, 261, 263, 281)
5 }

```

- może być typowany statycznie
- może być typowany dynamicznie

```
1 void main() {
2     // nie drukuje z powodu bledow
3     print("start");
4     var myVariable = 50; // int or num also possible
5     // to spowoduje to blad kompilacji
6     myVariable = "Hello";
7     print(myVariable);
8 }
```

```
1 dart bin/todo_app.dart
```

- może być typowany statycznie
- może być typowany dynamicznie

```
1 void main() {
2   print("start");
3   dynamic myVariable = 50;
4   myVariable = "Hello"; // OK
5   print(myVariable);    // Hello
6 }
```

Symbol	Name	Opis
		jest wiele podobnych do javy, c++
/	dzielenie całkowite	Do dzielenia dwóch operandów i podawania wyniku w postaci liczby całkowitej
is	jest	Zwraca wartość logiczną true, jeśli obiekt ma określony typ
is!	nie jest	Zwraca wartość logiczną false, jeśli obiekt ma określony typ

- porównanie stringów
- split, join
- inne funkcje

```
1 import 'dart:io';
2
3 void main() {
4   String p1 = "Jeremy";
5   String p2 = stdin.readLineSync()!;
6   var plp2 = p1 == p2;
7   print("p1 == p2: $plp2"); // true/false
8   print("p1 == p2: ${p1 == p2}"); // true/false
9   print("p1 == p2: ${p1.compareTo(p2)}"); // -1/0/1
10 }
```

- porównanie stringów
- **split, join**
- inne funkcje

```
1 void main() {  
2   var s = "Jeremy, James, Richard";  
3   var l = s.split(","); // [Jeremy, James, Richard]  
4   var r = l.join(":"); // "Jeremy: James: Richard"  
5   print(r);  
6 }
```

- porównanie stringów
- split, join
- inne funkcje

```
1 void main() {  
2   print(" "+"Jeremy"+" "+"Clarkson"+" ");  
3   print("  Jeremy Clarkson ".codeUnitAt(2)); // 74  
4   print("  Jeremy Clarkson ".trim()); // "Jeremy Clarkson"  
5   print("  Jeremy Clarkson ".toUpperCase()); // "  JEREMY CLARKSON "  
6   print("  Jeremy Clarkson ".substring(2, 8)); // "Jeremy"  
7   print("  Jeremy Clarkson ".replaceAll(" ", "+")); // "+Jeremy Clarkson+"  
8   print(20.toStringAsExponential()); // "2e+1"  
9 }
```

- Aplikacja Flutter używa języka `Dart`
- Interfejs użytkownika jest zbudowany z widżetów. Widżety opisują, jak powinien wyglądać ich widok, biorąc pod uwagę ich bieżącą konfigurację i stan
- Gdy stan widżetu ulega zmianie, widżet odbudowuje swój opis
 - Liczona jest różnica w celu minimalizacji zakresu zmian w drzewie
- Jeden kod do tworzenia aplikacji Desktop Windows, Linux, Apple, aplikacji mobilnych Android i Iphone oraz aplikacji webowych

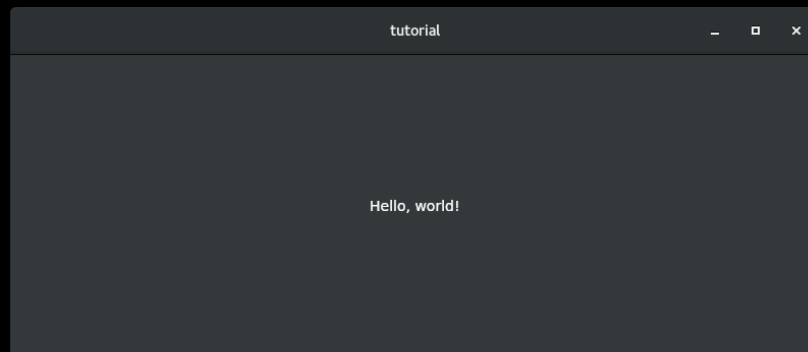


- **Aplikacja wyświetlająca napis**

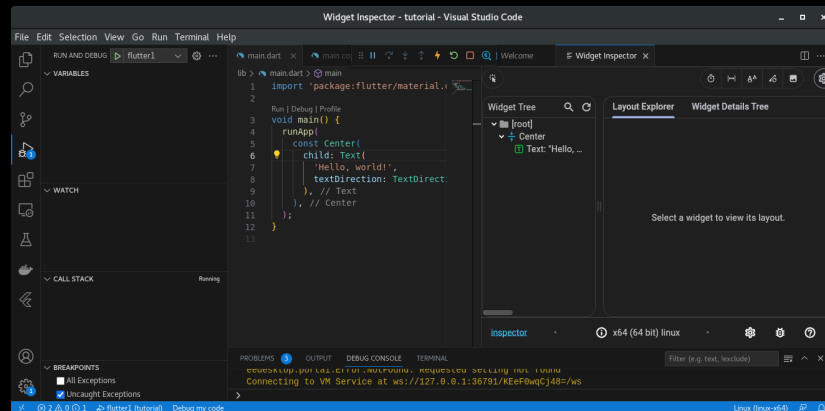
- aplikacja
- widget-inspector
- Funkcja `runApp()` przyjmuje podany `Widget` i ustawia go jako korzeń drzewa widżetów
 - Struktura wymusza zakrycie ekranu przez widżet główny
- Piszemy nowe widżety, które są podklasami `StatelessWidget` lub `StatefulWidget`
 - Głównym zadaniem widżetu jest implementacja funkcji `build()`, która opisuje widżet w odniesieniu do innych widżetów niższego poziomu

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(
5     const Center(
6       child: Text(
7         'Hello, world!',
8         textDirection: TextDirection.ltr,
9       ),
10    ),
11  );
12 }
```

- Aplikacja wyświetlająca napis
 - aplikacja
 - widget-inspector
- Funkcja `runApp()` przyjmuje podany `Widget` i ustawia go jako korzeń drzewa widżetów
 - Struktura wymusza zakrycie ekranu przez widżet główny
- Piszemy nowe widżety, które są podklasami `StatelessWidget` lub `StatefulWidget`
 - Głównym zadaniem widżetu jest implementacja funkcji `build()`, która opisuje widżet w odniesieniu do innych widżetów niższego poziomu



- Aplikacja wyświetlająca napis
 - aplikacja
 - **widget-inspector**
- Funkcja `runApp()` przyjmuje podany Widget i ustawia go jako korzeń drzewa widżetów
 - Struktura wymusza zakrycie ekranu przez widżet główny
- Piszemy nowe widżety, które są podklasami `StatelessWidget` lub `StatefulWidget`
 - Głównym zadaniem widżetu jest implementacja funkcji `build()`, która opisuje widżet w odniesieniu do innych widżetów niższego poziomu



- Aplikacja wyświetlająca napis
 - aplikacja
 - widget-inspector
- Funkcja `runApp()` przyjmuje podany `Widget` i ustawia go jako korzeń drzewa widgetów
 - Struktura wymusza zakrycie ekranu przez widget główny
- Piszemy nowe widżety, które są podklasami `StatelessWidget` lub `StatefulWidget`
 - Głównym zadaniem widżetu jest implementacja funkcji `build()`, która opisuje widżet w odniesieniu do innych widżetów niższego poziomu

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(
5     const Center(
6       child: Text(
7         'Hello, world!',
8         textDirection: TextDirection.ltr,
9       ),
10    ),
11  );
12 }
```

- Aplikacja wyświetlająca napis
 - aplikacja
 - widget-inspector
- Funkcja `runApp()` przyjmuje podany `Widget` i ustawia go jako korzeń drzewa widżetów
 - **Struktura wymusza zakrycie ekranu przez widżet główny**
- Piszemy nowe widżety, które są podklasami `StatelessWidget` lub `StatefulWidget`
 - Głównym zadaniem widżetu jest implementacja funkcji `build()`, która opisuje widżet w odniesieniu do innych widżetów niższego poziomu

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(
5     const Center(
6       child: Text(
7         'Hello, world!',
8         textDirection: TextDirection.ltr,
9       ),
10    ),
11  );
12 }
```

- Aplikacja wyświetlająca napis
 - aplikacja
 - widget-inspector
- Funkcja `runApp()` przyjmuje podany `Widget` i ustawia go jako korzeń drzewa widżetów
 - Struktura wymusza zakrycie ekranu przez widżet główny
- **Piszemy nowe widżety, które są podklasami `StatelessWidget` lub `StatefulWidget`**
 - Głównym zadaniem widżetu jest implementacja funkcji `build()`, która opisuje widżet w odniesieniu do innych widżetów niższego poziomu

```
1 import 'package:flutter/material.dart';
2
3 class MyWidget extends StatelessWidget {
4   const MyWidget({super.key});
5
6   @override
7   Widget build(BuildContext context) {
8     return const Center(
9       child: Text(
10         'Hello, world!!',
11         textDirection: TextDirection.ltr,
12       ),
13     );
14   }
15 }
16
17 void main() {
18   runApp(const MyWidget());
19 }
```

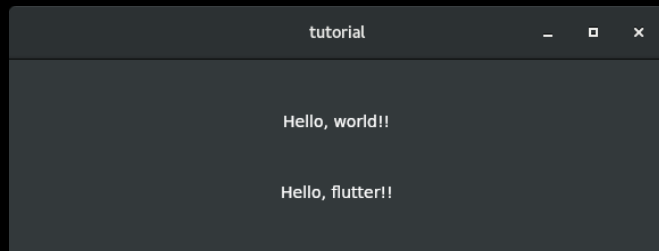
- Aplikacja wyświetlająca napis
 - aplikacja
 - widget-inspector
- Funkcja `runApp()` przyjmuje podany `Widget` i ustawia go jako korzeń drzewa widżetów
 - Struktura wymusza zakrycie ekranu przez widżet główny
- Piszemy nowe widżety, które są podklasami `StatelessWidget` lub `StatefulWidget`
 - Głównym zadaniem widżetu jest implementacja funkcji `build()`, która opisuje widżet w odniesieniu do innych widżetów niższego poziomu

```
1 import 'package:flutter/material.dart';
2
3 class MyWidget extends StatelessWidget {
4   const MyWidget({super.key});
5
6   @override
7   Widget build(BuildContext context) {
8     return const Center(
9       child: Text(
10         'Hello, world!!',
11         textDirection: TextDirection.ltr,
12       ),
13     );
14   }
15 }
16
17 void main() {
18   runApp(const MyWidget());
19 }
```

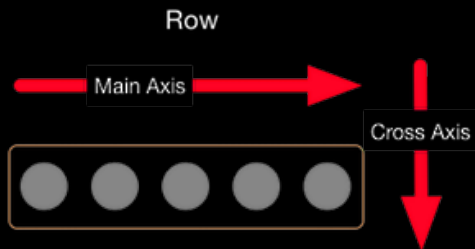
- Widżet `Text` tworzy stylizowany tekst
- `Row`, `Column` - tworzą elastyczne układy w obu kierunkach
- `container` - tworzy prostokąt

```
1 class MyWidget extends StatelessWidget {  
2   const MyWidget({super.key});  
3  
4   @override  
5   Widget build(BuildContext context) {  
6     return const Center(  
7       child: Text(  
8         'Hello, world!!',  
9         textDirection: TextDirection.ltr,  
10      ),  
11    );  
12  }  
13 }
```

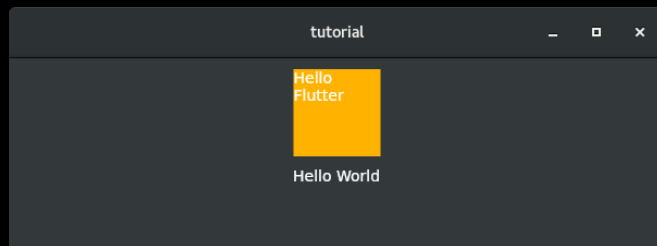
- Widżet `Text` tworzy stylizowany tekst
- `Row`, `Column` - tworzą elastyczne układy w obu kierunkach
- `container` - tworzy prostokąt



```
1 return const Column(  
2   mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
3   children: [  
4     Text(  
5       'Hello, world!!',  
6       textDirection: TextDirection.ltr,  
7     ),  
8     Text(  
9       'Hello, flutter!!',  
10      textDirection: TextDirection.ltr,  
11    ),  
12  ],  
13 );
```

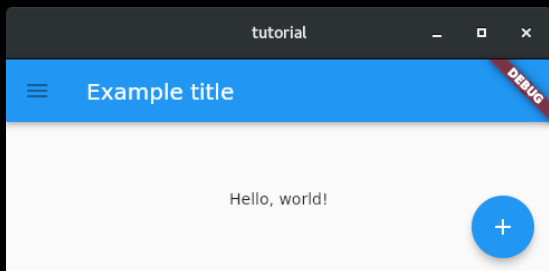


- Widżet `Text` tworzy stylizowany tekst
- `Row`, `Column` - tworzą elastyczne układy w obu kierunkach
- `container` - tworzy prostokąt



```
1 import 'package:flutter/material.dart';
2 class MyWidget extends StatelessWidget {
3   const MyWidget({super.key});
4   @override
5   Widget build(BuildContext context) {
6     return Container(
7       margin: const EdgeInsets.all(10.0),
8       color: Colors.amber[600],
9       width: 80.0,
10      height: 80.0,
11      child: const Text(
12        'Hello Flutter',
13        textDirection: TextDirection.ltr,
14      ),
15    );
16  }
17 }
18 void main() {
19   runApp(
20     const Column(children: [
21       MyWidget(),
22       Text(
23         'Hello World',
24         textDirection: TextDirection.ltr,
25       ),
26     ]),
27   );
28 }
```

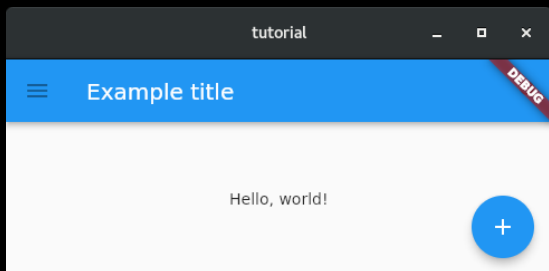
- Flutter udostępnia szereg widżetów, które pomagają tworzyć aplikacje zgodne z Material Design
- Aplikacja Material rozpoczyna się od widżetu MaterialApp
 - Posiada wbudowany Navigator do przechodzenia między ekranami
 - Wiele widżetów Material Design musi znajdować się w MaterialApp, aby wyświetlać się prawidłowo



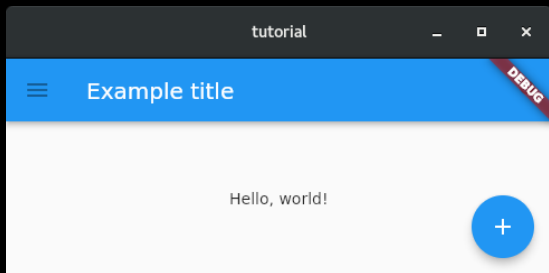
```
1 import 'package:flutter/material.dart';
2 void main() {
3   runApp(
4     const MaterialApp(
5       title: 'Flutter Tutorial',
6       home: TutorialHome(),
7     ),
8   );
9 }
10 class TutorialHome extends StatelessWidget {
11   const TutorialHome({super.key});
12   @override
13   Widget build(BuildContext context) {
14     // Scaffold is a layout for the Material.
15     return Scaffold(
16       appBar: AppBar(
17         leading: const IconButton(
18           icon: Icon(Icons.menu),
19           tooltip: 'Navigation menu',
20           onPressed: null,
21         ),
22       title: const Text('Example title'),
23       actions: const [
24         IconButton(
25           icon: Icon(Icons.search),
26           tooltip: 'Search',
27           onPressed: null,
28         ),
29       ],
30     );
31 }
```

- Flutter udostępnia szereg widżetów, które pomagają tworzyć aplikacje zgodne z Material Design
- Aplikacja Material rozpoczyna się od widżetu MaterialApp
 - Posiada wbudowany Navigator do przechodzenia między ekranami
 - Wiele widżetów Material Design musi znajdować się w MaterialApp, aby wyświetlać się prawidłowo

```
30 // body is the majority of the screen.
31 body: const Center(
32   child: Text('Hello, world!'),
33 ),
34 floatingActionButton:
35   const FloatingActionButton(
36     tooltip: 'Add',
37     onPressed: null,
38     child: Icon(Icons.add),
39   ),
40 );
41 }
42 }
```



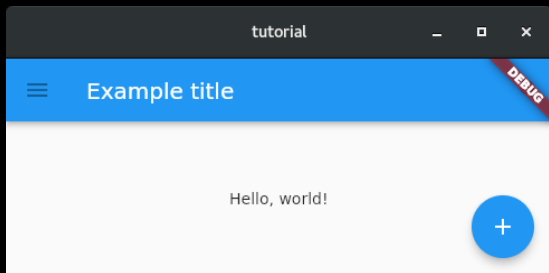
- Flutter udostępnia szereg widżetów, które pomagają tworzyć aplikacje zgodne z Material Design
- Aplikacja Material rozpoczyna się od widżetu MaterialApp
 - Posiada wbudowany Navigator do przechodzenia między ekranami
 - Wiele widżetów Material Design musi znajdować się w MaterialApp, aby wyświetlać się prawidłowo



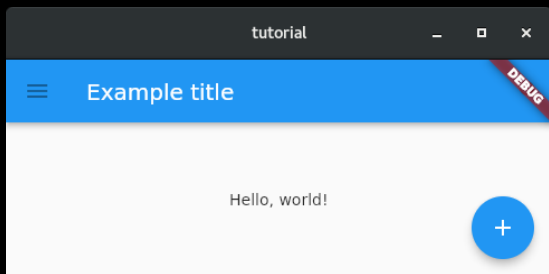
```
1 import 'package:flutter/material.dart';
2 void main() {
3   runApp(
4     const MaterialApp(
5       title: 'Flutter Tutorial',
6       home: TutorialHome(),
7     ),
8   );
9 }
10 class TutorialHome extends StatelessWidget {
11   const TutorialHome({super.key});
12   @override
13   Widget build(BuildContext context) {
14     // Scaffold is a layout for the Material.
15     return Scaffold(
16       appBar: AppBar(
17         leading: const IconButton(
18           icon: Icon(Icons.menu),
19           tooltip: 'Navigation menu',
20           onPressed: null,
21         ),
22       title: const Text('Example title'),
23       actions: const [
24         IconButton(
25           icon: Icon(Icons.search),
26           tooltip: 'Search',
27           onPressed: null,
28         ),
29       ],
30     ),
31   );
32 }
```

- Flutter udostępnia szereg widżetów, które pomagają tworzyć aplikacje zgodne z Material Design
- **Aplikacja Material rozpoczyna się od widżetu MaterialApp**
 - Posiada wbudowany Navigator do przechodzenia między ekranami
 - Wiele widżetów Material Design musi znajdować się w MaterialApp, aby wyświetlać się prawidłowo

```
30 // body is the majority of the screen.  
31 body: const Center(  
32   child: Text('Hello, world!'),  
33 ),  
34 floatingActionButton:  
35   const FloatingActionButton(  
36     tooltip: 'Add',  
37     onPressed: null,  
38     child: Icon(Icons.add),  
39   ),  
40 );  
41 }  
42 }
```



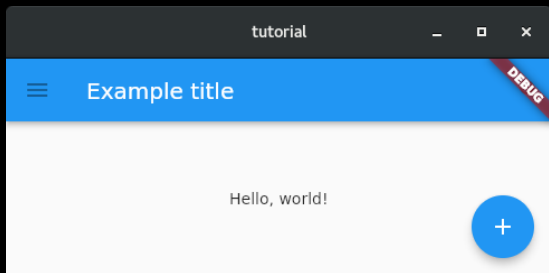
- Flutter udostępnia szereg widżetów, które pomagają tworzyć aplikacje zgodne z Material Design
- Aplikacja Material rozpoczyna się od widżetu MaterialApp
 - Posiada wbudowany Navigator do przechodzenia między ekranami
 - Wiele widżetów Material Design musi znajdować się w MaterialApp, aby wyświetlać się prawidłowo



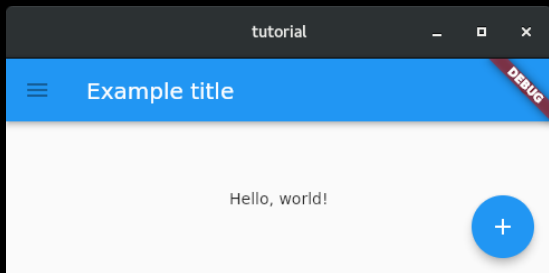
```
1 import 'package:flutter/material.dart';
2 void main() {
3   runApp(
4     const MaterialApp(
5       title: 'Flutter Tutorial',
6       home: TutorialHome(),
7     ),
8   );
9 }
10 class TutorialHome extends StatelessWidget {
11   const TutorialHome({super.key});
12   @override
13   Widget build(BuildContext context) {
14     // Scaffold is a layout for the Material.
15     return Scaffold(
16       appBar: AppBar(
17         leading: const IconButton(
18           icon: Icon(Icons.menu),
19           tooltip: 'Navigation menu',
20           onPressed: null,
21         ),
22       title: const Text('Example title'),
23       actions: const [
24         IconButton(
25           icon: Icon(Icons.search),
26           tooltip: 'Search',
27           onPressed: null,
28         ),
29       ],
30     ),
31   );
32 }
```

- Flutter udostępnia szereg widżetów, które pomagają tworzyć aplikacje zgodne z Material Design
- Aplikacja Material rozpoczyna się od widżetu MaterialApp
 - Posiada wbudowany Navigator do przechodzenia między ekranami
 - Wiele widżetów Material Design musi znajdować się w MaterialApp, aby wyświetlać się prawidłowo

```
30 // body is the majority of the screen.  
31 body: const Center(  
32   child: Text('Hello, world!'),  
33 ),  
34 floatingActionButton:  
35   const FloatingActionButton(  
36     tooltip: 'Add',  
37     onPressed: null,  
38     child: Icon(Icons.add),  
39   ),  
40 );  
41 }  
42 }
```



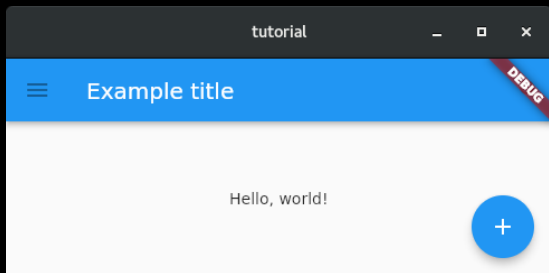
- Flutter udostępnia szereg widżetów, które pomagają tworzyć aplikacje zgodne z Material Design
- Aplikacja Material rozpoczyna się od widżetu MaterialApp
 - Posiada wbudowany **Navigator** do przechodzenia między ekranami
 - Wiele widżetów Material Design musi znajdować się w MaterialApp, aby wyświetlać się prawidłowo

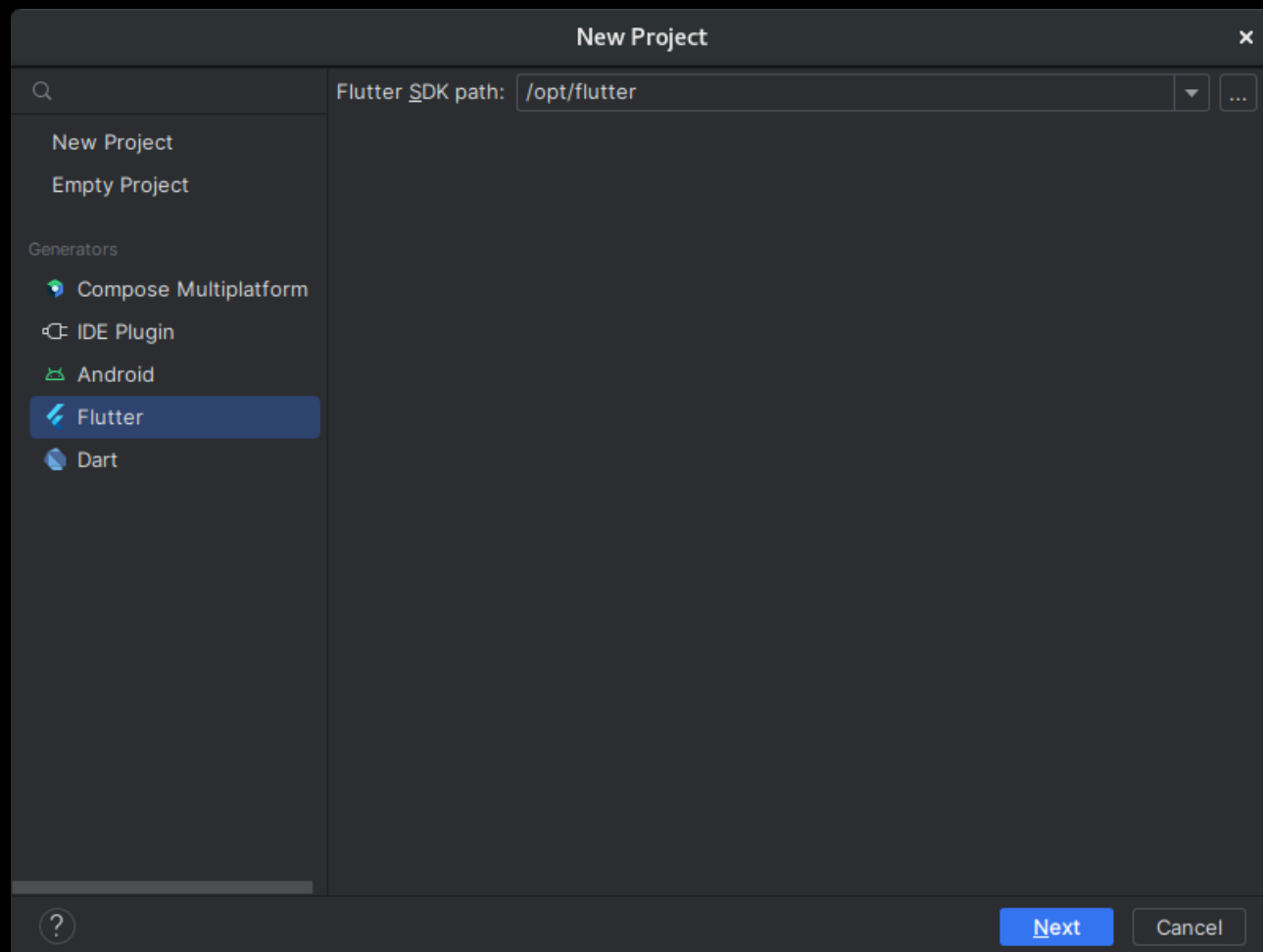


```
1 import 'package:flutter/material.dart';
2 void main() {
3   runApp(
4     const MaterialApp(
5       title: 'Flutter Tutorial',
6       home: TutorialHome(),
7     ),
8   );
9 }
10 class TutorialHome extends StatelessWidget {
11   const TutorialHome({super.key});
12   @override
13   Widget build(BuildContext context) {
14     // Scaffold is a layout for the Material.
15     return Scaffold(
16       appBar: AppBar(
17         leading: const IconButton(
18           icon: Icon(Icons.menu),
19           tooltip: 'Navigation menu',
20           onPressed: null,
21         ),
22       title: const Text('Example title'),
23       actions: const [
24         IconButton(
25           icon: Icon(Icons.search),
26           tooltip: 'Search',
27           onPressed: null,
28         ),
29       ],
30     );
31 }
```

- Flutter udostępnia szereg widżetów, które pomagają tworzyć aplikacje zgodne z Material Design
- Aplikacja Material rozpoczyna się od widżetu MaterialApp
 - Posiada wbudowany `Navigator` do przechodzenia między ekranami
 - Wiele widżetów Material Design musi znajdować się w `MaterialApp`, aby wyświetlać się prawidłowo

```
30 // body is the majority of the screen.
31 body: const Center(
32   child: Text('Hello, world!'),
33 ),
34 floatingActionButton:
35   const FloatingActionButton(
36     tooltip: 'Add',
37     onPressed: null,
38     child: Icon(Icons.add),
39   ),
40 );
41 }
42 }
```





New Project

Project name:

first_flutter

Project location:

~/AndroidStudioProjects/first_flutter

...

Description:

A new Flutter project.

Project type:

Application

Organization:

pl.edu.pw

Android language:

☒ Java

☐ Kotlin

iOS language:

☐ Objective-C

☒ Swift

Platforms:

☒ Android ☒ iOS ☒ Linux ☒ MacOS ☒ Web ☒ Windows

When created, the new project will run on the selected platforms (others can be added later).

☐ Create project offline

More Settings

Module name:

first_flutter

Content root:

/home/jmy/AndroidStudioProjects/first_flutter

📁

Module file location:

/home/jmy/AndroidStudioProjects/first_flutter

📁

Project format:

.idea (directory-based)

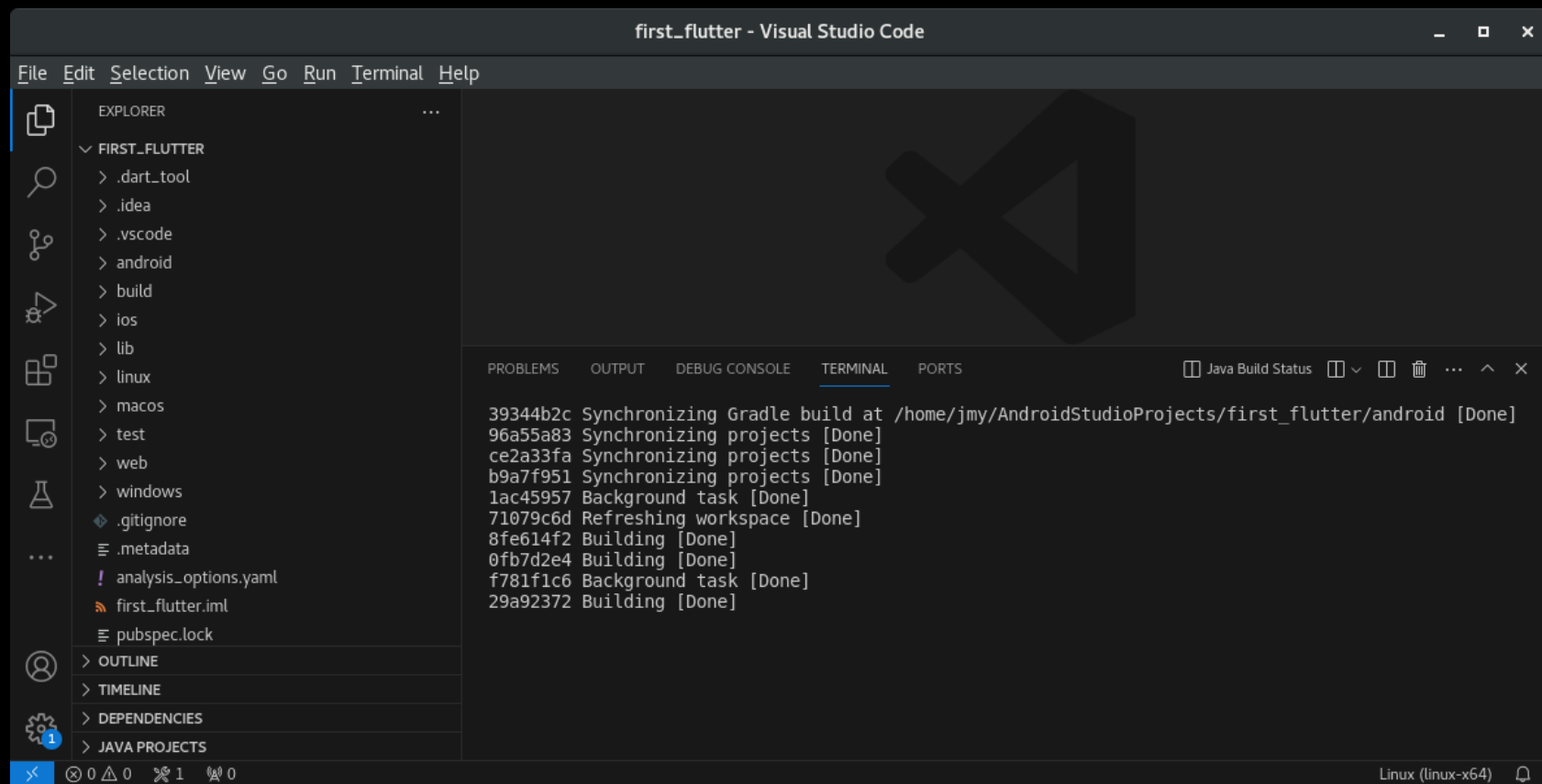
▼

?

Previous

Create

Cancel



Extension: Awesome Flutter Snippets - tutorial - Visual Studio Code

File Edit Selection View Go Run Terminal Help

main.dart x main copy.dart {} launch.json ...

lib > main.dart > ...

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(
5     const MaterialApp(
6       title: 'Flutter Tutorial',
7       home: TutorialHome(),
8     ),
9   );
10 }
```

Run | Debug | Profile

Awesome Flutter Snippets

Neevash Ramdial (Nash) [neevash.dev](#)

Awesome Flutter Snippets is a collection sn...

Disable Uninstall

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

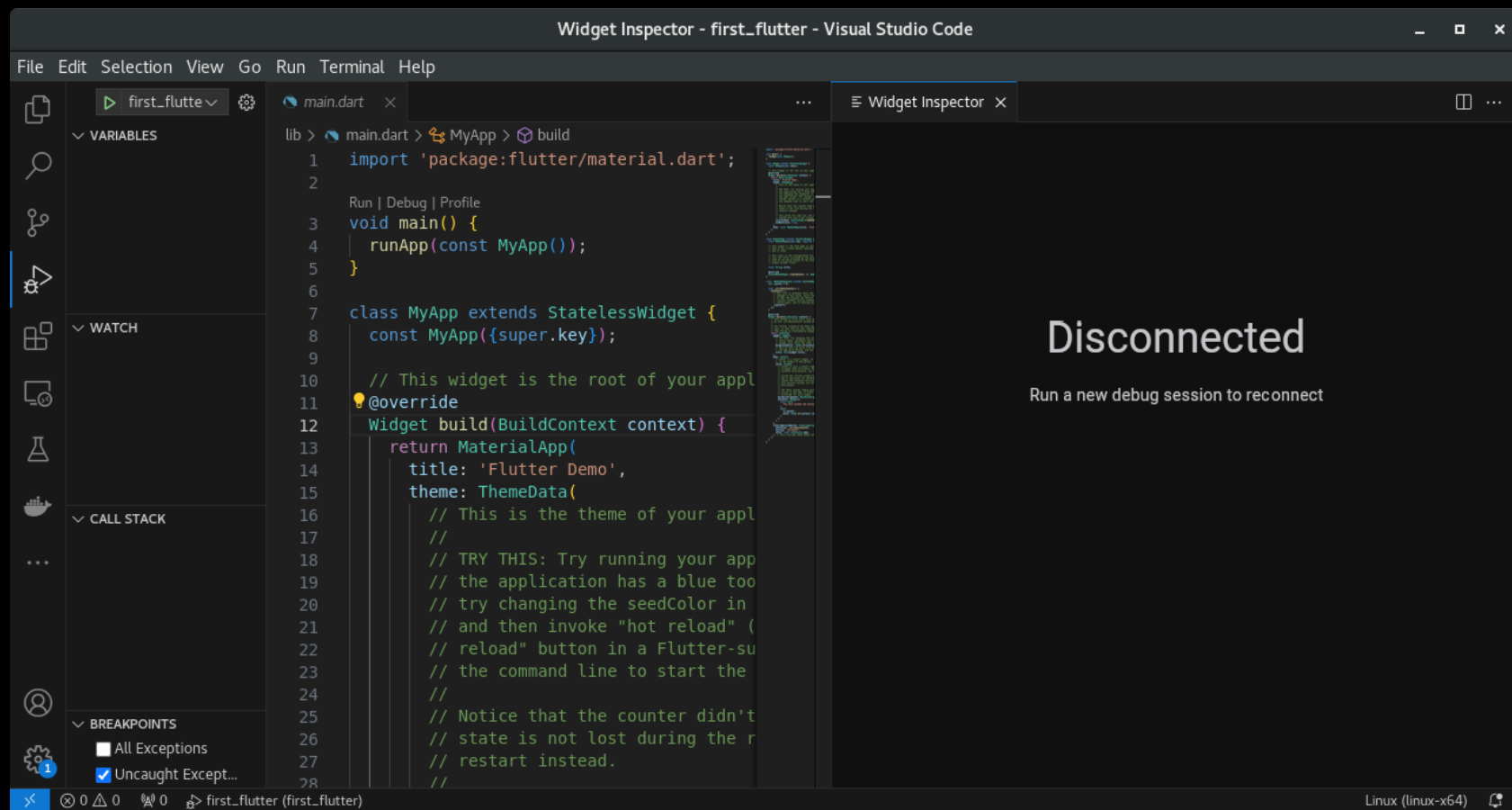
jmy@alma560x ~/FlutterProjects/tutorial \$ emulator -list-avds
Pixel_3a_API_34_extension_level_7_x86_64
Pixel_6a_API_29

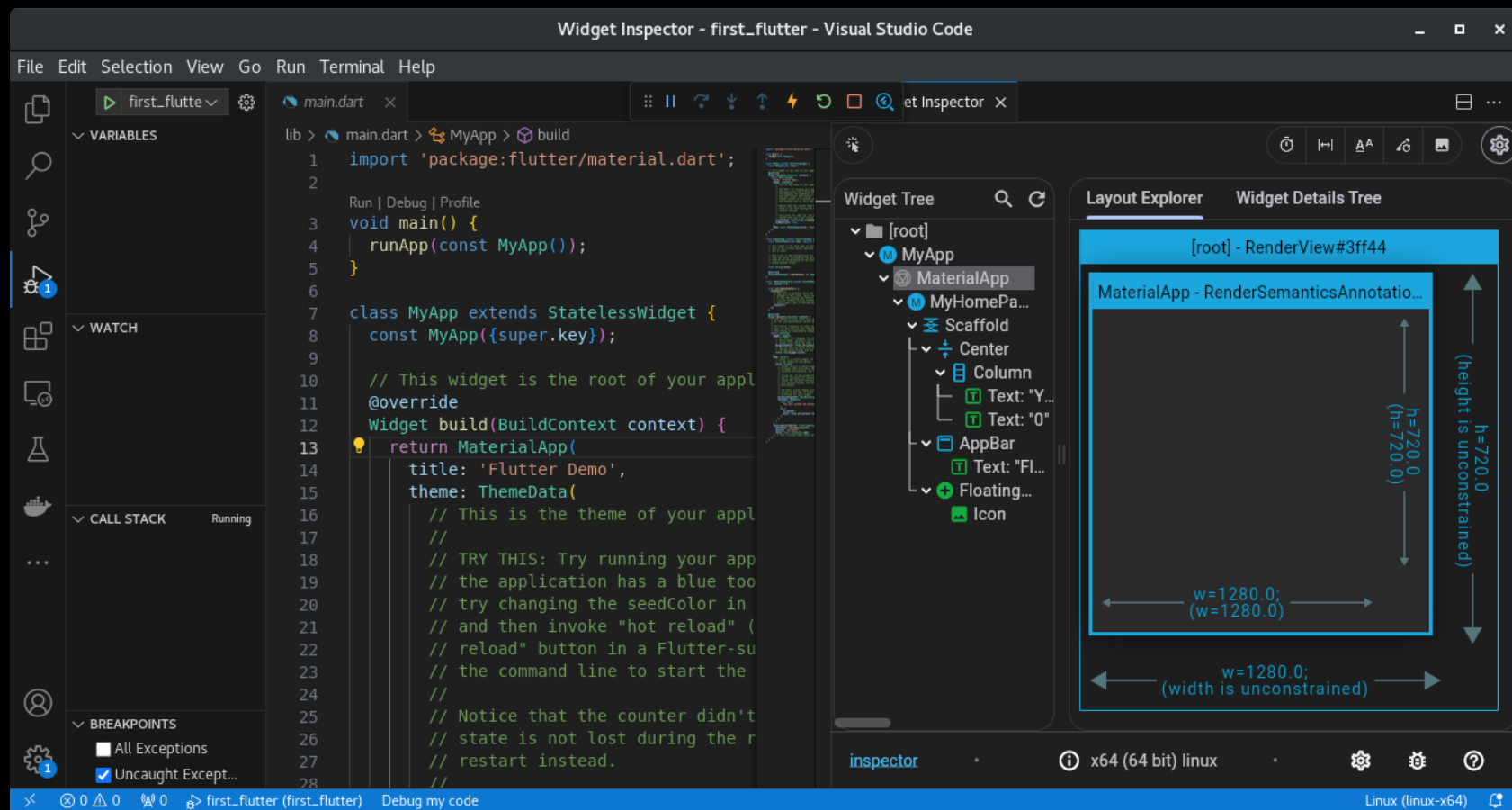
jmy@alma560x ~/FlutterProjects/tutorial \$ emulator @Pixel_6a_API_29

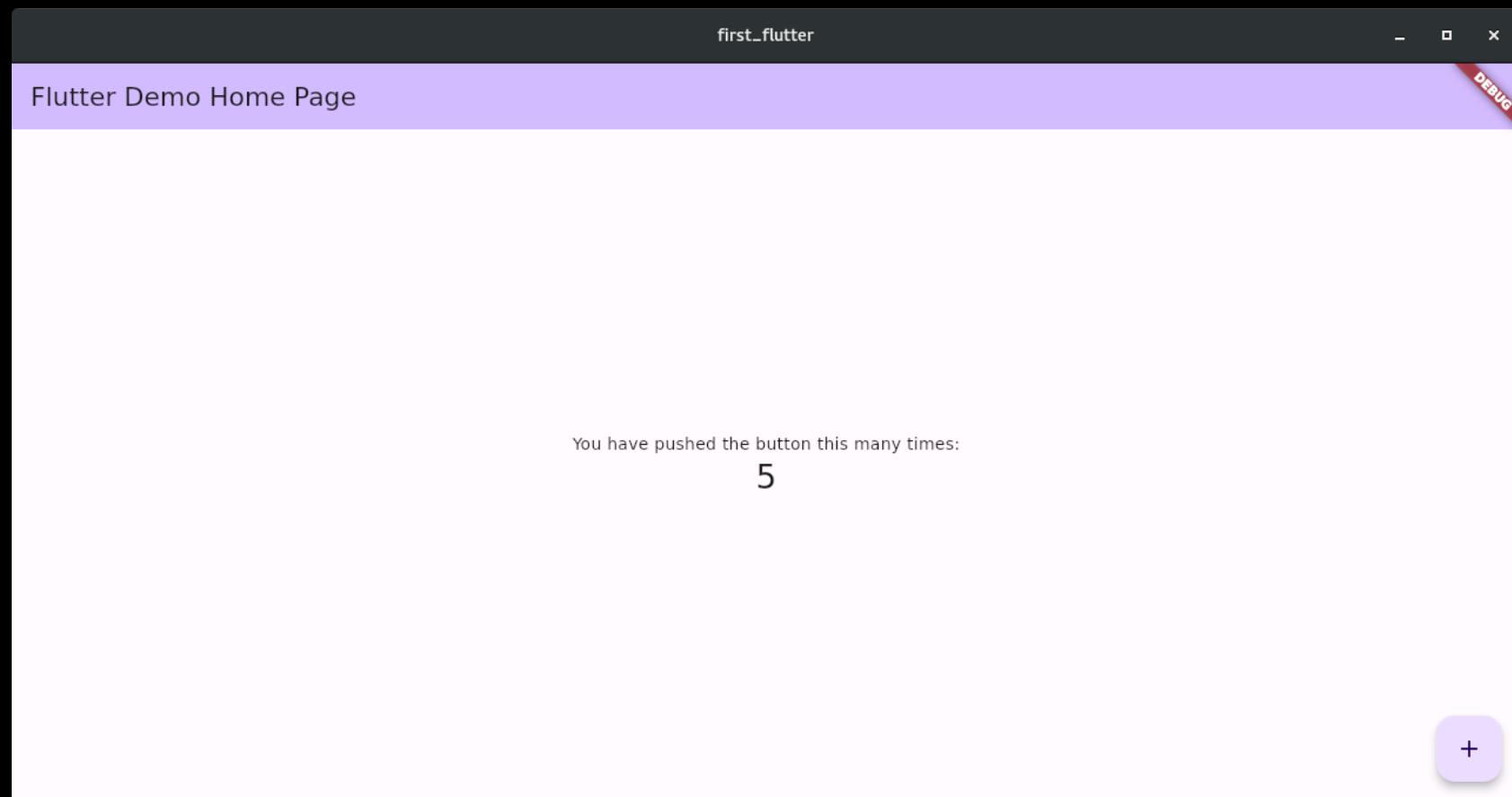
```
INFO | Android emulator version 32.1.15.0 (build_id 10696886) (CL:N/A)
INFO | Found systemPath /home/jmy/Android/Sdk/system-images/android-29/google_apis_playstore/x86/
INFO | Storing crashdata in: /tmp/android-jmy/emu-crash.db, detection is enabled
INFO | Duplicate loglines will be removed, if you wish to see each individual line launch with the -log-nofilter flag.
WARNING | cannot add library /home/jmy/Android/Sdk/emulator/qemu/linux-x86_64/lib64/vulkan/libvulkan.so: failed
INFO | added library /home/jmy/Android/Sdk/emulator/lib64/vulkan/libvulkan.so
WARNING | *** No gRPC protection active, consider launching with the -grpc-use-jwt flag.***
INFO | Started GRPC server at 127.0.0.1:8554, security: Local, auth: none
INFO | Advertising in: /run/user/1000/avd/running/pid_14164.ini
INFO | Setting display: 0 configuration to: 1080x2400, dpi: 420x420
```

bash
qemu-syste...

Pixel 6a API 29 (android-x86 emulator)







Widget Inspector - first_flutter - Visual Studio Code

File Edit Selection View Go Run Terminal Help

first_flutter main.dart

lib > main.dart > MyApp > build

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({super.key});
9
10  // This widget is the root of your application
11  @override
12  Widget build(BuildContext context) {
13    return MaterialApp(
14      title: 'Flutter Demo',
15      theme: ThemeData(
16        // This is the theme of your application
17        // TRY THIS: Try running your application with the
18        // the application has a blue toolbar at the bottom.
19        // try changing the seedColor in the
20        // and then invoke "hot reload" (hold ⌘ and press R)
21        // reload" button in a Flutter-supported IDE, or
22        // the command line to start the application.
23        // Notice that the counter didn't reset to zero.
24        // state is not lost during the reload.
25        // restart instead.
26      ),
27    );
28  }
```

Run | Debug | Profile

void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({super.key});

// This widget is the root of your application

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

// This is the theme of your application

// TRY THIS: Try running your application with the

the application has a blue toolbar at the bottom.

try changing the seedColor in the

and then invoke "hot reload" (hold ⌘ and press R)

reload" button in a Flutter-supported IDE, or

the command line to start the application.

Notice that the counter didn't reset to zero.

state is not lost during the reload.

restart instead.

}

Disconnected

Run a new debug session to reconnect

Chrome (web-javascript)

Widget Inspector - first_flutter - Visual Studio Code

File Edit Selection View Go Run Terminal Help

first_flutter main.dart

et Inspector

VARIABLES

WATCH

CALL STACK Running

BREAKPOINTS

- ☐ All Exceptions
- ☒ Uncaught Except...

```
lib > main.dart > MyApp > build
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({super.key});
9
10  // This widget is the root of your appl
11  @override
12  Widget build(BuildContext context) {
13    return MaterialApp(
14      title: 'Flutter Demo',
15      theme: ThemeData(
16        // This is the theme of your appl
17        //
18        // TRY THIS: Try running your app
19        // the application has a blue too
20        // try changing the seedColor in
21        // and then invoke "hot reload" (
22        // reload" button in a Flutter-su
23        // the command line to start the
24        //
25        // Notice that the counter didn't
26        // state is not lost during the r
27        // restart instead.
28        //
```

Widget Tree

- [root]
- MyApp

Layout Explorer

[root] - RenderView#d45ed

Widget Details Tree

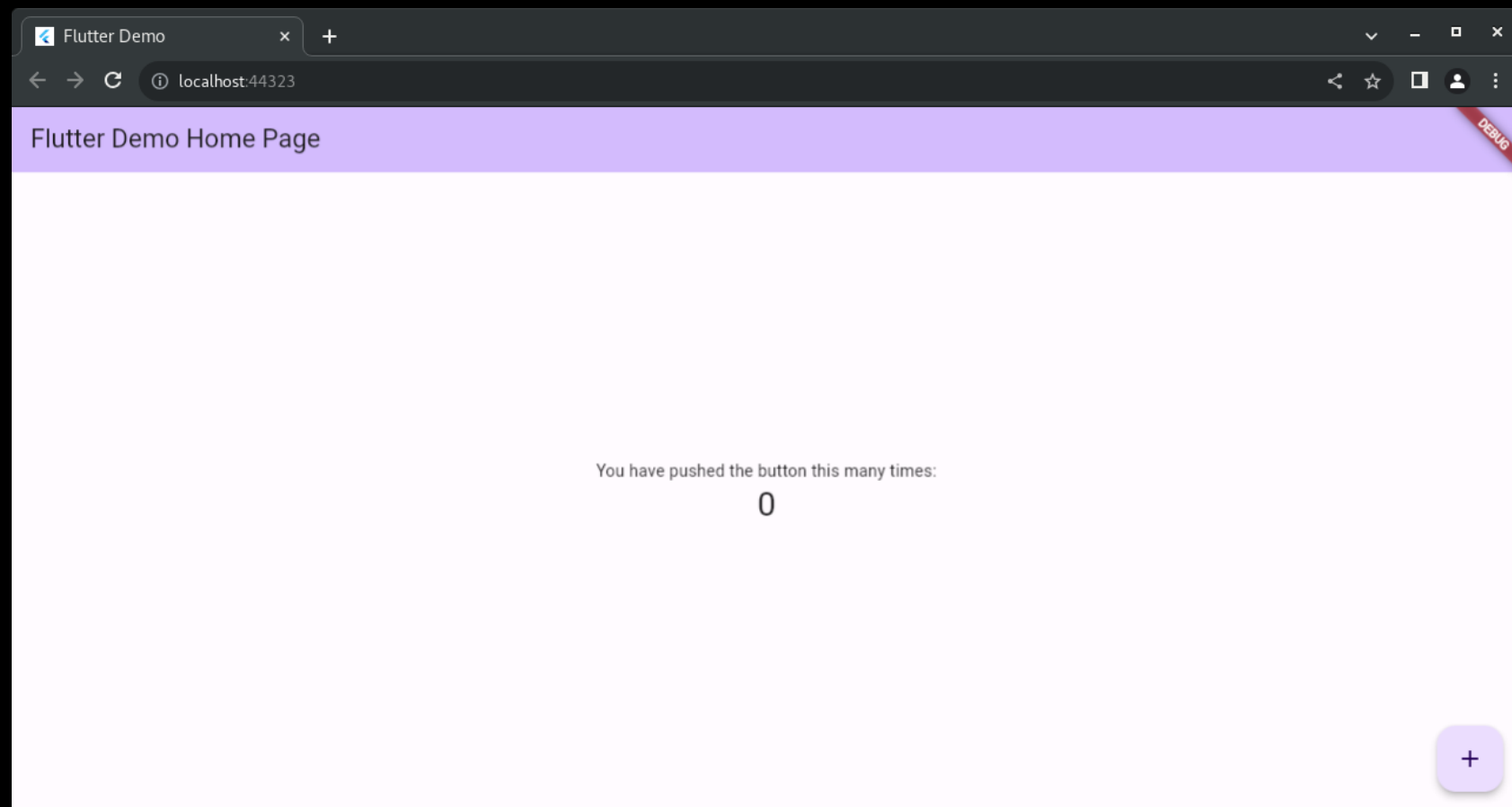
Device Info

- CPU / OS:** Web linux
- Dart Version:** 3.1.1
- Flutter Version:** 3.13.3 / stable
- Framework / Engine:** 2524052335 / b8d35810e9
- Connected app type:** Flutter web (debug build)
- VM Service Connection:** ws://127.0.0.1:43681/ttPSSrZWzn4=/ws

CONNECT TO A NEW APP **CLOSE**

inspector web app

Chrome (web-javascript)



Widget Inspector - first_flutter - Visual Studio Code

File Edit Selection View Go Run Terminal Help

first_flutter main.dart

lib > main.dart > MyApp > build

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({super.key});
9
10  // This widget is the root of your application
11  @override
12  Widget build(BuildContext context) {
13    return MaterialApp(
14      title: 'Flutter Demo',
15      theme: ThemeData(
16        // This is the theme of your application
17        // TRY THIS: Try running your application with the
18        // the application has a blue toolbar at the bottom.
19        // try changing the seedColor in the
20        // and then invoke "hot reload" (hold ⌘ and press R)
21        // reload" button in a Flutter-su
22        // the command line to start the
23        //
24        // Notice that the counter didn't
25        // state is not lost during the
26        // restart instead.
27      ),
28    );
29  }
```

Run | Debug | Profile

void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({super.key});

// This widget is the root of your application

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

// This is the theme of your application

// TRY THIS: Try running your application with the

the application has a blue toolbar at the bottom.

try changing the seedColor in the

and then invoke "hot reload" (hold ⌘ and press R)

reload" button in a Flutter-su

the command line to start the

// Notice that the counter didn't

state is not lost during the

restart instead.

}

Disconnected

Run a new debug session to reconnect

Pixel 6a API 29 (android-x86 emulator)

Widget Inspector - first_flutter - Visual Studio Code

File Edit Selection View Go Run Terminal Help

first_flutter main.dart

et Inspector

VARIABLES

WATCH

CALL STACK Running

BREAKPOINTS

- ☐ All Exceptions
- ☒ Uncaught Except...

```
lib > main.dart > MyApp > build
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({super.key});
9
10  // This widget is the root of your appl
11  @override
12  Widget build(BuildContext context) {
13    return MaterialApp(
14      title: 'Flutter Demo',
15      theme: ThemeData(
16        // This is the theme of your appl
17        //
18        // TRY THIS: Try running your app
19        // the application has a blue too
20        // try changing the seedColor in
21        // and then invoke "hot reload" (
22        // reload" button in a Flutter-su
23        // the command line to start the
24        //
25        // Notice that the counter didn't
26        // state is not lost during the r
27        // restart instead.
28
```

Widget Tree

- [root]
- MyApp

Layout Explorer

Widget Details Tree

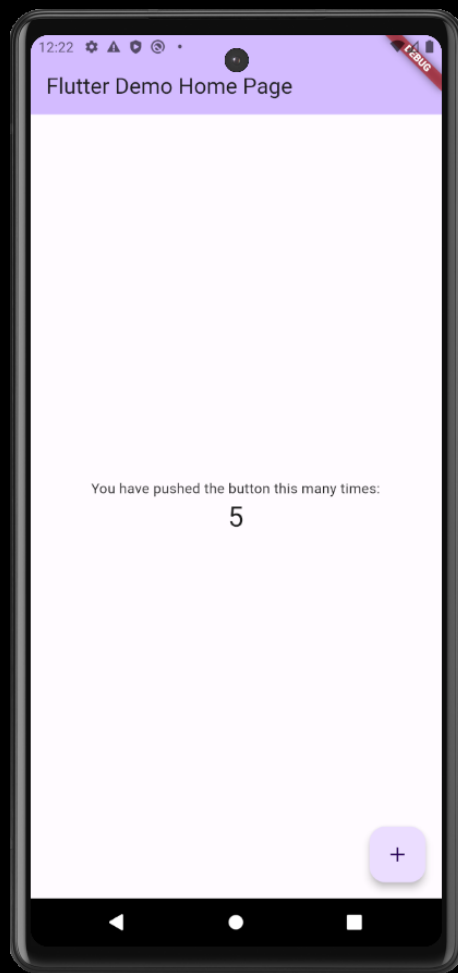
[root] - RenderView#4cbb3

Device Info

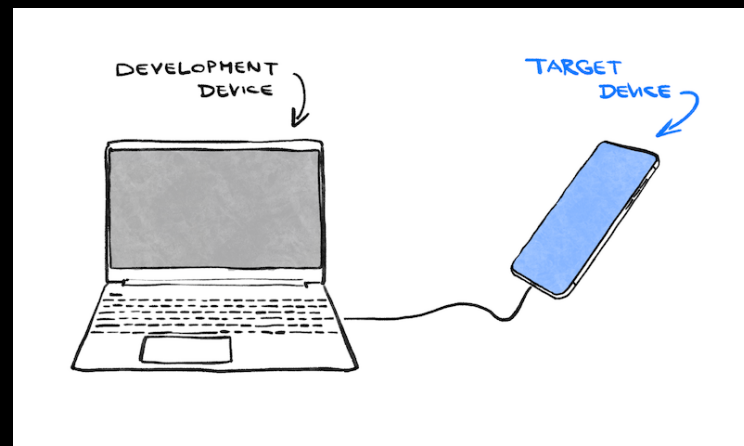
- CPU / OS:** ia32 (32 bit) android
- Dart Version:** 3.1.1
- Flutter Version:** 3.13.3 / stable
- Framework / Engine:** 2524052335 / b8d35810e9
- Connected app type:** Flutter native (debug build)
- VM Service Connection:** ws://127.0.0.1:40455/BnmvY5emYJM=ws

CONNECT TO A NEW APP **VIEW VM FLAGS** **CLOSE**

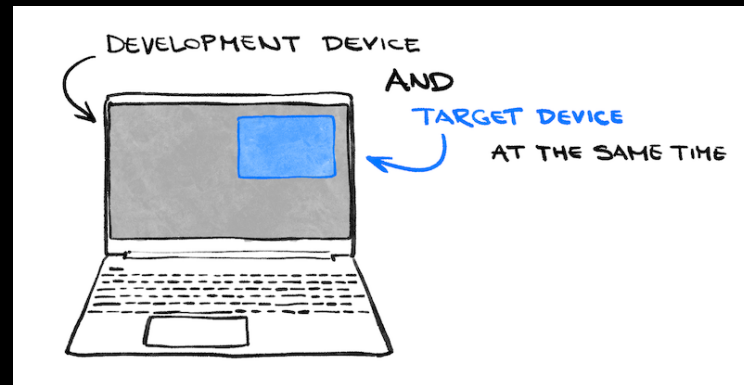
inspector ia32 (32 bit) android Pixel 6a API 29 (android-x86 emulator)



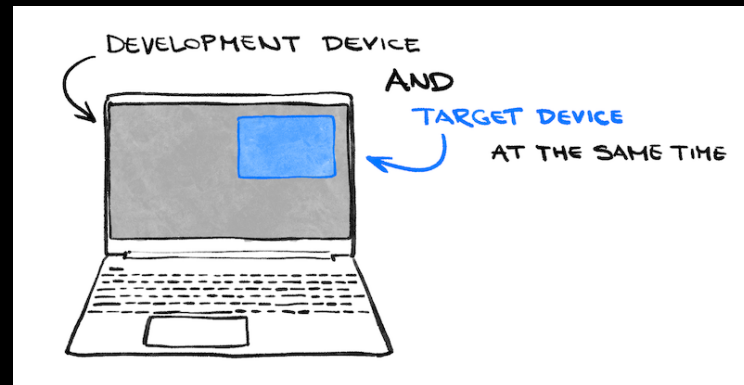
- możemy tworzyć aplikacje, które będą działać na innych typach urządzeń
- jeśli to możliwe powinniśmy skupić się na urządzeniu na którym pracujemy
- aplikacje internetowe są możliwe, ale niektóre funkcje (gorące przeładowywanie podczas rozwoju) nie są jeszcze obsługiwane



- możemy tworzyć aplikacje, które będą działać na innych typach urządzeń
- **jeśli to możliwe powinniśmy skupić się na urządzeniu na którym pracujemy**
- aplikacje internetowe są możliwe, ale niektóre funkcje (gorące przeładowywanie podczas rozwoju) nie są jeszcze obsługiwane

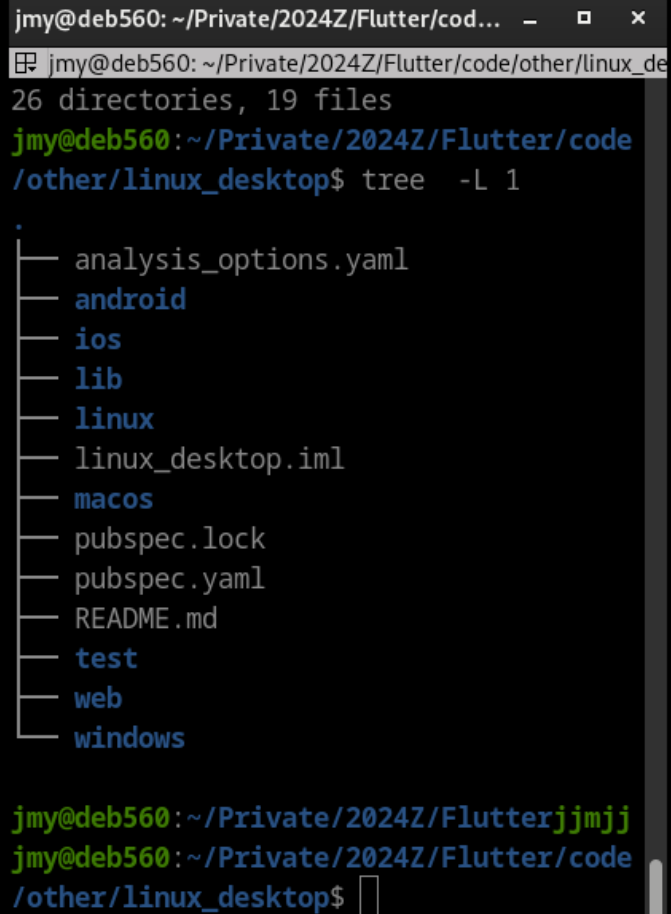


- możemy tworzyć aplikacje, które będą działać na innych typach urządzeń
- jeśli to możliwe powinniśmy skupić się na urządzeniu na którym pracujemy
- aplikacje internetowe są możliwe, ale niektóre funkcje (gorące przeładowywanie podczas rozwoju) nie są jeszcze obsługiwane



- W programie Visual Studio Code wybieramy `Widok-Paleta poleceń` (`Ctrl-Shift-P`)

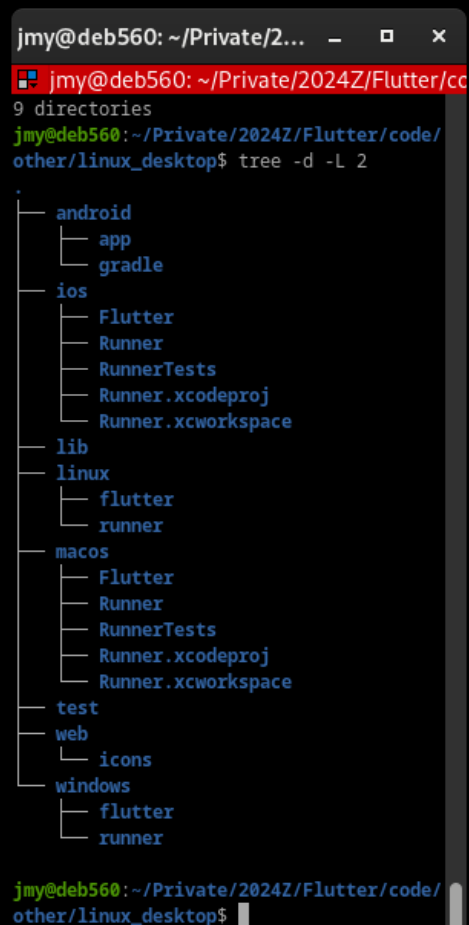
- W programie Visual Studio Code wybieramy Widok-Paleta poleceń (Ctrl-Shift-P)
- Wybieramy folder i nazwę tworzonej aplikacji



```
jmy@deb560: ~/Private/2024Z/Flutter/cod... - □ ×
jmy@deb560: ~/Private/2024Z/Flutter/code/other/linux_de
26 directories, 19 files
jmy@deb560:~/Private/2024Z/Flutter/code
/other/linux_desktop$ tree -L 1
.
├── analysis_options.yaml
├── android
├── ios
├── lib
├── linux
├── linux_desktop.iml
├── macos
├── pubspec.lock
├── pubspec.yaml
├── README.md
├── test
├── web
└── windows

jmy@deb560:~/Private/2024Z/Flutterjmyjj
jmy@deb560:~/Private/2024Z/Flutter/code
/other/linux_desktop$
```

- W programie Visual Studio Code wybieramy Widok-Paleta poleceń (Ctrl-Shift-P)
- Wybieramy folder i nazwę tworzonej aplikacji



```
jmy@deb560: ~/Private/2... - □ x
jmy@deb560: ~/Private/2024Z/Flutter/code/
9 directories
jmy@deb560:~/Private/2024Z/Flutter/code/
other/linux_desktop$ tree -d -L 2
.
├── android
│   ├── app
│   └── gradle
├── ios
│   ├── Flutter
│   ├── Runner
│   ├── RunnerTests
│   ├── Runner.xcodeproj
│   └── Runner.xcworkspace
├── lib
├── linux
│   ├── flutter
│   └── runner
├── macos
│   ├── Flutter
│   ├── Runner
│   ├── RunnerTests
│   ├── Runner.xcodeproj
│   └── Runner.xcworkspace
├── test
├── web
│   └── icons
└── windows
    ├── flutter
    └── runner

jmy@deb560:~/Private/2024Z/Flutter/code/
other/linux_desktop$
```

- W programie Visual Studio Code wybieramy Widok-Paleta poleceń (Ctrl-Shift-P)
- Wybieramy folder i nazwę tworzonej aplikacji

```

jmy@deb560: ~/Private/2024Z/Flutter/code/other/linux_d...
jmy@deb560: ~/Private/2024Z/Flutter/code/other/linux_desktop$ tree -d
.
├── android
│   ├── app
│   │   ├── src
│   │   │   ├── debug
│   │   │   ├── main
│   │   │   │   ├── java
│   │   │   │   │   ├── io
│   │   │   │   │   │   ├── flutter
│   │   │   │   │   │   └── plugins
│   │   │   │   ├── kotlin
│   │   │   │   │   ├── com
│   │   │   │   │   └── example
│   │   │   │   │       └── linux_desktop
│   │   │   └── res
│   │   │       ├── drawable
│   │   │       ├── drawable-v21
│   │   │       ├── mipmap-hdpi
│   │   │       ├── mipmap-mdpi
│   │   │       ├── mipmap-xhdpi
│   │   │       ├── mipmap-xxhdpi
│   │   │       ├── mipmap-xxxhdpi
│   │   │       ├── values
│   │   │       └── values-night
│   │   └── profile
│   └── gradle
│       └── wrapper
├── ios
│   ├── Flutter
│   ├── Runner
│   │   ├── Assets.xcassets
│   │   ├── AppIcon.appiconset
│   │   ├── LaunchImage.imageset
│   │   └── Base.lproj
│   ├── RunnerTests
│   ├── Runner.xcodeproj
│   │   ├── project.xcworkspace
│   │   │   ├── xcshareddata
│   │   │   └── xcschemas
│   │   └── Runner.xcworkspace
│   │       └── xcshareddata
├── lib
├── linux
│   ├── flutter
│   └── runner
├── macos
│   ├── Flutter
│   │   ├── ephemeral
│   │   └── Runner
│   │       ├── Assets.xcassets
│   │       ├── AppIcon.appiconset
│   │       ├── Base.lproj
│   │       ├── Configs
│   │       ├── RunnerTests
│   │       ├── Runner.xcodeproj
│   │       │   ├── project.xcworkspace
│   │       │   │   ├── xcshareddata
│   │       │   │   └── xcschemas
│   │       │   └── Runner.xcworkspace
│   │       │       └── xcshareddata
├── test
├── web
│   └── icons
├── windows
│   ├── flutter
│   └── runner
│       └── resources

```

69 directories
jmy@deb560: ~/Private/2024Z/Flutter/code/other/linux_desktop\$

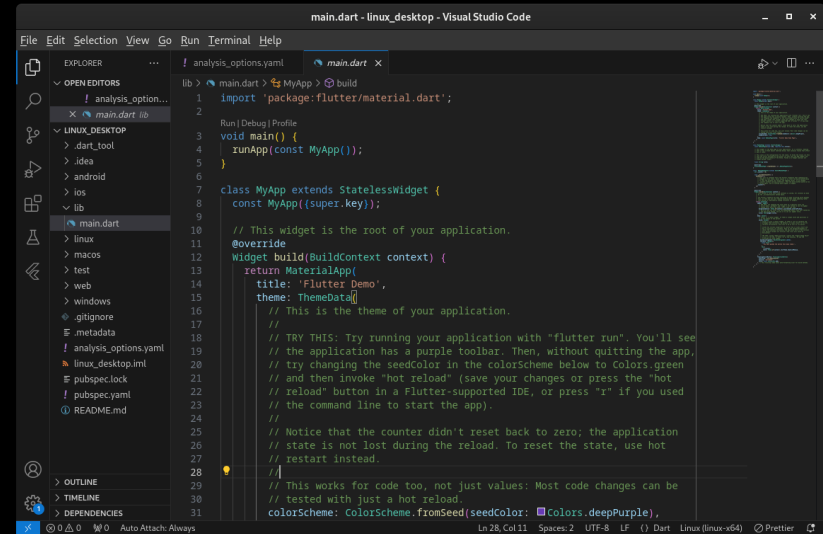
- W programie Visual Studio Code wybieramy Widok-Paleta poleceń (Ctrl-Shift-P)
- Wybieramy folder i nazwę tworzonej aplikacji
 - Plik `pubspec.yaml` określa podstawowe informacje o aplikacji, takie jak jej bieżąca wersja, jej zależności i zasoby, które zostaną dostarczone.

```
1 name: linux_desktop
2 description: "A new Flutter project."
3 publish_to: 'none'
4 version: 1.0.0+1
5 environment:
6   sdk: ^3.6.1
7 dependencies:
8   flutter:
9     sdk: flutter
10   cupertino_icons: ^1.0.8
11 dev_dependencies:
12   flutter_test:
13     sdk: flutter
14 flutter:
15   uses-material-design: true
```

- W programie Visual Studio Code wybieramy Widok-Paleta poleceń (Ctrl-Shift-P)
- Wybieramy folder i nazwę tworzonej aplikacji
 - Plik `pubspec.yaml` określa podstawowe informacje o aplikacji, takie jak jej bieżąca wersja, jej zależności i zasoby, które zostaną dostarczone.
 - `analysis_options.yaml` Ten plik określa, jak rygorystyczny powinien być Flutter podczas analizowania kodu (zmieniono na odczep się)

```
1 include: package:flutter_lints/flutter.yaml
2
3 linter:
4   rules:
5     avoid_print: false
6     prefer_const_constructors_in_immutables: false
7     prefer_const_constructors: false
8     prefer_const_literals_to_create_immutables: false
9     prefer_final_fields: false
10    unnecessary_breaks: true
11    use_key_in_widget_constructors: false
```

- W programie Visual Studio Code wybieramy Widok-Paleta poleceń (Ctrl-Shift-P)
- Wybieramy folder i nazwę tworzonej aplikacji
 - Plik `pubspec.yaml` określa podstawowe informacje o aplikacji, takie jak jej bieżąca wersja, jej zależności i zasoby, które zostaną dostarczone.
 - `analysis_options.yaml` Ten plik określa, jak rygorystyczny powinien być Flutter podczas analizowania kodu (zmieniono na odczep się)
 - `lib/main.dart` - kod źródłowy naszej aplikacji
- uruchomienie aplikacji
 - wybranie docelowego urządzenia i uruchomienie aplikacji

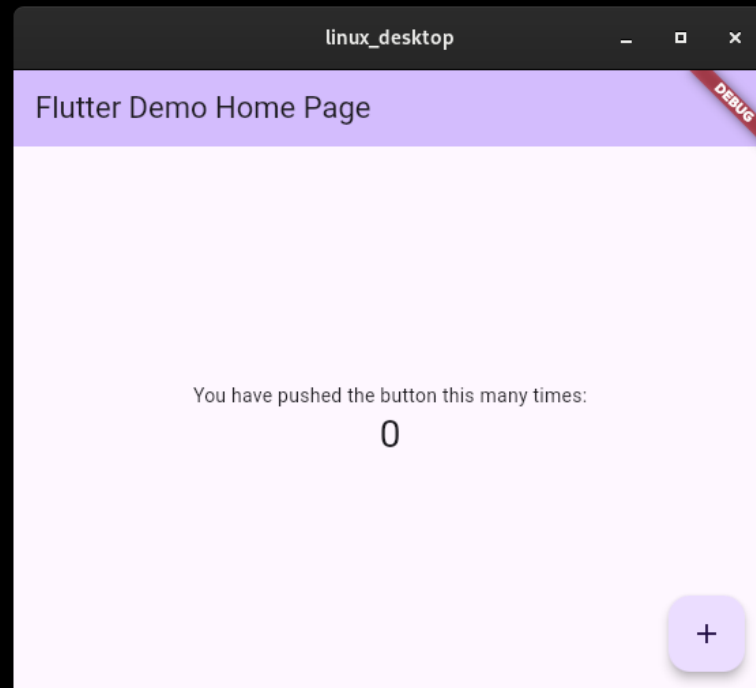


The screenshot shows the Visual Studio Code editor with the file `main.dart` open. The code is for a Flutter desktop application. The Explorer sidebar on the left shows the project structure with folders for `linux`, `macos`, `test`, `web`, `windows`, `.gitignore`, `.metadata`, `analysis_options.yaml`, `linux_desktop.iml`, `pubspec.lock`, `pubspec.yaml`, and `README.md`. The main editor area displays the following Dart code:

```
lib > main.dart > MyApp > build
1  import 'package:flutter/material.dart';
2
3  void main() {
4    runApp(const MyApp());
5  }
6
7  class MyApp extends StatelessWidget {
8    const MyApp({super.key});
9
10   // This widget is the root of your application.
11   @override
12   Widget build(BuildContext context) {
13     return MaterialApp(
14       title: 'Flutter Demo',
15       theme: ThemeData(
16         // This is the theme of your application.
17         // TRY THIS: Try running your application with "flutter run". You'll see
18         // the application has a purple toolbar. Then, without quitting the app,
19         // try changing the seedColor in the colorScheme below to Colors.green
20         // and then invoke "hot reload" (save your changes or press the "hot
21         // reload" button in a Flutter-supported IDE, or press "r" if you used
22         // the command line to start the app).
23         // Notice that the counter didn't reset back to zero; the application
24         // state is not lost during the reload. To reset the state, use hot
25         // restart instead.
26         // This works for code too, not just values: Most code changes can be
27         // tested with just a hot reload.
28         colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
29       ),
30     );
31   }
32 }
```

- W programie Visual Studio Code wybieramy Widok-Paleta poleceń (Ctrl-Shift-P)
- Wybieramy folder i nazwę tworzonej aplikacji
 - Plik `pubspec.yaml` określa podstawowe informacje o aplikacji, takie jak jej bieżąca wersja, jej zależności i zasoby, które zostaną dostarczone.
 - `analysis_options.yaml` Ten plik określa, jak rygorystyczny powinien być Flutter podczas analizowania kodu (zmieniono na odczep się)
 - `lib/main.dart` - kod źródłowy naszej aplikacji
- uruchomienie aplikacji
 - wybranie docelowego urządzenia i uruchomienie aplikacji
 - aplikacja jest uruchomiona

```
../build/linux/x64/debug/bundle/linux_desktop
```



- to nowoczesne narzędzie do tworzenia interfejsu graficznego dla wielu platform
- używa miłego, ale niszowego języka

- to nowoczesne narzędzie do tworzenia interfejsu graficznego dla wielu platform
- używa miłego, ale niszowego języka