|  |  |
| --- | --- |
| **FRONTEND** | |
| **COMPONENTS** | |
| State aanmaken met lege array | const [categories, setCategories] = useState([]); |
| Destructuring | const {hasCreatorAccess} = useAccess(); |
| Hook in variabele steken (navigate) | const navigate = useNavigate(); |
| Code uitvoeren laden component | useEffect( () => { UIT TE VOEREN CODE }, []); |
| Data aan state geven | setCategories(response.data); |
| Propertie aan component geven | <TodosList todos={todos} /> |
| Shit mappen fzo idk wa da ier doe | {categories.map((c) => (            <Tabs.Item key={c.id} title={c.name}>              <TodosList                todos={todos.filter((t) => t.category.name === c.name)}              />            </Tabs.Item>          ))} |
| Object megeven via navigate | onClick={() => navigate("/add", { state: { categories } })} |
| Destructuring van uit hook | const { hasCreatorAccess, isLoading } = useAccess(); |
| Component weergeven | return <LoadingPage />; |
| Prop toevoegen | const TodoItem = ({todo}) => {  }; |
| Property meegeven van object dat gelijk is aan property van ander object | todoId: todo.id, |
| Properties ophalen van object en if statement op gebruiken | const isDone = todo.sessions.length ? todo.sessions[0].isDone : false;  const isOverDue = todo.sessions.length ? todo.sessions[0].isOverDue : false; |
| Methode oproepen bij onClick met object meegeven | onClick={() => handleTodo(todo)} |
| Property van object weergeven | {todo.title} |
| Property tonen als deze bestaat | {todo.deadline && (            <p className="font-thin text-sm">              {new Date(todo.deadline).toLocaleString()}            </p>          )} |
| Alle items uit lijst weergeven met mapper | {todos.map((t) => (          <TodoItem key={t.id} todo={t} />        ))} |
| **CONTEXTS** | |
| Verschillende states aanmaken | const [hasAccess, setHasAccess] = useState(false);  const [hasCreatorAccess, setHasCreatorAccess] = useState(false);  const [isLoading, setIsLoading] = useState(true);  const [isAccessLoading, setIsAccessLoading] = useState(true);  const [error, setError] = useState(false); |
| State aanpassen | setHasAccess(true); |
| Geen idee wa dit doe | {props.children} |
| **PAGES** | |
| Hooks en states maken / gebruiken | const [accessCode, setAccessCode] = useState("");    const { gainAccess, isLoading } = useAccess();    const navigate = useNavigate(); |
| Tekst uit tekstvak gebruiken wanneer het veranderd | onChange={(e) => setAccessCode(e.target.value)} |
| Methode oproepen en wachten op resultaat | await gainAccess(accessCode); |
| State aanmaken met object | const [todo, setTodo] = useState({      title: "",      description: "",      deadline: "",      categoryId: categories[0].id,    }); |
| Properties object meegeven | …todo |
| Navigeren naar een pad | navigate("/"); |
| Methode koppelen aan onchange property | onChange={handleChange} |
|  |  |

Yarn

Yarn dev

import { useNavigate } from "react-router-dom";

 const navigate = useNavigate();