

React-router-dom

29 Kasım 2021 Pazartesi 03:31

Client

App.jsx

* Router isn't exported react-router-dom
hatalı olursa, v6 versiyonu kur.
npm : react-router-dom@v6

npm : react-router-dom

import {

BrowserRouter as Router, Routes, Route, Link } from 'react-router-dom'

<Router>

<Routes>

<Route path="/" element={<Home />} />

<Route path="/movies" element={<Home type="movies"/>} />

<Route path="/series" element={<Home type="series"/>} />

<Route path="/watch" element={<Watch />} />

<Route path="/login" element={<Login />} />

<Route path="/register" element={<Register />} />

</Routes>

</Router>

Movies.jsx

<Link to="/movies" />Movies</Link>

<Link to="/series" />Series</Link>

Featured.jsx

const Featured = ({type}) => {

{type === "movies" ? "Movies" : "Series"}

Home.jsx

```

const Home = ({ type }) => {
  return (
    <div class="home" >
      <Header />
      <Featured type={type} />
      <List />
      <hr />
      <Lang />
    </div>
  )
}

```

App.js

```

const user = true
<Router>
  <Routes>
    <Route path="/" element={user ? <Home /> : <Register />} /> // Açılmıyor!
    <Route path="/register" element={user ? <Home /> : <Navigate replace to="/register" />} /> // Açılmıyor!
    <Route path="/login" element={!user ? <Login /> : <Navigate replace to="/" />} />
    <Route path="/movies" element={<Home type="movies" />} />
    <Route path="/series" element={<Home type="series" />} />
    <Route path="/watch" element={<Watch />} />
  </Routes>

```

</Router>

Fetch Data from Api

29 Kasım 2021 Pazartesi 04:53

Home.jsx

```
const [lists, setLists] = useState([ ])
const [genre, setGenre] = useState(null)

useEffect(() => {
  const getRandomLists = async () => {
    + try {
      const res = await axios.get(
        `lists?type?${genre ? `&type=${genre}` : ''}`)
      if (res.data)
        setLists(res.data)
    } catch (err) {
      console.log(err)
    }
  }
  getRandomLists()
}, [type, genre])

```

package.json

```
]}
,
"proxy": "http://localhost:8800/api"
}
{
  headers: {
    token: "Bearer " + ...
  }
}
<Header/>
<Footer/>
{
  lists.map((list) => (
    <List list={list} />
  ))
}
```

y)

List - JSX

```
<div className="conten" or>
  {list.map((item, i) => {
    <ListItem index={i} item={item}>/>
  })
}
```

Displaying Movie From Api

29 Kasım 2021 Pazartesi 19:55

* clef(ssm)
65000r2TSZ ---
this is an id

List Item.jsx

```
const ListItem = ({index, item}) => {  
  useEffect(() => {  
    const getMovie = async () => {  
      try {  
        const res = await axios.get("/movies/find/" + item.id,  
          {  
            headers: {  
              token:  
                "Bearer -----"  
            }  
          }  
        })  
        setMovie(res.data)  
      } catch (err) {  
        console.log(err)  
      }  
    }  
    getMovie()  
  }, [item])  
  
const [movie, setMovie] = useState({})
```

* Gelmadi! api'ye gidip if(req.isAdmin)
kapatma kaldırımları, Güçlü!

Watch Page

30 Kasım 2021 Salı 03:49

List Item.jsx

```
<Link to={{ pathname: '/watch', movie: movie }}>
  <div className="listItem">
    :
    <div>
  </div>
</Link>
```

Watch.jsx

```
const location = useLocation()
const {location}
const movie = location.movie
:
<video src={movie.video} />
```

Gelmedi!

movie

boş geljer!

```
<Link to="/">
  Home
</Link>
```

Featured Movie

30 Kasım 2021 Salı 05:51

Featured.jsx

```
const [content, setContent] = useState({})
```

```
useEffect(() => {
```

```
    const getRandomContent = async () => {
```

```
        try {
```

```
            const res = await axios.get(`movies/random?type=${type}`)
```

```
            setContent(res.data[0])
```

```
        } catch (err) {
```

↳ Aşağı gelir.

İlk eklenen ad.

```
}
```

```
        getRandomContent()
```

```
    }, [type]);
```

```
<img src={content.img} />
```

```
<span desc>
```

```
    {content.desc}
```

```
</span>
```

* Series İndeksde

Lost List Item Feature hep: Categories

* Movies'de Sadecce Featured garantıyor.

Admin Panel

30 Kasım 2021 Salı 06:14

Repo'dan kopyala

git clone --single-branch -b react-admin
http://github

git clone --single-branch -b react-admin
http://github -

git clone --single-branch -b react-admin
http://github ---

↳ npm install → potoller: yoldanca için

*admin .env dosyası aç.

PORT=4000 → http://localhost:4000

npm start portunda çalışma başladı

Admin Dashboard Chart

1 Aralık 2021 Çarşamba 05:45

Admin → home.jsx

const MONTHS = useMemo(() => ["Jan", "Feb", "Mar", "Apr", "May", "June", "July", "Aug", "Sep", "Oct", "Nov", "Dec"], []);

const [userStats, setUserStats] = useState([]);
useEffect(() => {

const getStats = async () => {
try {

const res = await axios.get("/users/stats", {
headers: {}

token:

}
});
;

const statLists = res.data.sort(function(a, b) {
return a.id - b.id
});

statLists.map((item) =>

setUserStats((prev) => [

...prev,

{ name: MONTHS[item.id - 1],

"New User": item.total },

);
);

) catch(error) {

console.log(error)

};

getStats()

y, [MONTHS])

<chart data=[^{new state} y -- - 1]

Fetching Latest User

2 Aralık 2021 Perşembe 19:26

WidgetSm.js

const [newUsers, setNewUsers] = useState([])

useEffect(() => {

const getNewUsers = async () => {

try {

const res = await axios.get("/users?new=true") headers

setNewUsers(res.data)

} catch {

console.log(err)

}

} getNewUsers()

}, [])

* Genelde list componenti sil.

{newUsers.map(user) => {

:

{user.username}

Sidebar.js

<Link to="/movies" -->

Movies

</Link>

App.js

<Route "movies"

Context-api

3 Aralık 2021 Cuma 03:22

AuthActions.js

```
export const loginStart = () => {
    type: "LOGIN-START"
}

export const loginSuccess = (user) => {
    type: "LOGIN-SUCCESS",
    payload: user
}

export const loginFailure = () => {
    type: "LOGIN-FAILURE"
}

export const logout = () => {
    type: "LOGOUT"
}
```

AuthReducer.js

```
const AuthReducer = (state, action) => {
    switch (action.type) {
        case "LOGIN-START":
            return {
                user: null,
                isFetching: true,
                error: false
            }
        case "LOGIN-SUCCESS":
            return {
                user: action.payload,
                isFetching: false,
                error: false
            }
    }
}
```

```

case "LOGIN_FAILURE":
    return {
        user: null,
        isFetching: false,
        error: true
    }
}

case "LOGOUT":
    return {
        user: null,
        isFetching: false,
        error: false
    }
}

default:
    return { ...state }
}
}

```

export default AuthReducer

AuhtContext.js

```

import AuthReducer from './AuthReducer'
import {createContext, useState, useReducer} from 'react';

const INITIAL_STATE = {
    user: JSON.parse(localStorage.getItem("user")) || null,
    isFetching: false,
    error: false
}

export const AuthContext = createContext(INITIAL_STATE)
export const AuthContextProvider = ({children}) => {
    const [state, dispatch] = useReducer(AuthReducer, INITIAL_STATE)
    useState(() => {
        localStorage.setItem("user", JSON.stringify(state.user))
    })
}

```

```

}, [state.user])
return (
  <AuthContext.Provider
    value={{
      user: state.user,
      isFetching: state.isFetching,
      error: state.error,
      dispatch
    }}
  >
  {children}
  </AuthContext.Provider>
)

```

apiCalls.js

```

import axios from 'axios';
import { loginFailure, loginStart, loginSuccess } from
  "./AuthActions";
export const login = async (user, dispatch) => {
  dispatch(loginStart())
  try {
    const res = await axios.post("auth/login", user)
    if(res.data.isAdmin) dispatch(loginSuccess(res.data))
  } catch (err) {
    dispatch(loginFailure())
  }
}

```

Src → context → authContext
 movieContext
 listContext
 userContext

authContext → AuthActions.js

AuthContext.js

AuthReducer.js

apiCalls.js

index.js

<AuthContextProvider>

<App>

</AuthContextProvider>

pages → Login → Login.jsx

login.scs

div.login

form.loginForm

input.loginInput → email

input.loginInput → password

button.loginButton

App.js

<Router>

<Switch>

<Route path="/login">

(Login /)

<Route>

<TabBar />

<div className="container">

<SideBar />

<Router exact path="/" />

↳ Home ↳
↳ routes ↳

.login {

w: 100px

h: 100px

ds

q: c

JCC

}

.loginForm {

dt

for column

}

.loginInput {

width: 10

Login with JWT

3 Aralık 2021 Cuma 05:36

Login.jsx

```
const [email, setEmail] = useState('')
const [password, setPassword] = useState('')

<input onChange={(e) => setEmail(e.target.value)} />
<input onChange={(e) => setPassword(e.target.value)} />
<button onClick={handleLogin}>Login</button>
```

apiCalls.js

```
import axios from 'axios'
import { loginFailure, loginStart, loginSuccess } from
  './AuthActions'

export const login = async (user, dispatch) => {
  dispatch(loginStart())
  try {
    const res = await axios.post('auth/login', user)
    // res.data.password -> Sonradan encode olunacak
    dispatch(loginSuccess(res.data))
  } catch (err) {
    dispatch(loginFailure())
  }
}
```

login.jsx

```
const { isLoading, dispatch } = useAuth()
const handleLogin = (e) => {
  e.preventDefault()
  login({ email, password }, dispatch)
}

<button disabled={isLoading}></button>
```

AuthContext.js

```
useEffect(() => {
  localStorage.setItem("user", JSON.stringify(state.user))
}, [state.user])

const INITIAL_STATE = {
  user: localStorage.getItem("user") || null,
  isFetching: false,
  error: false
}

user: JSON.parse(localStorage.getItem("user") || null)
```

App.js

```
const [user] = useContext(AuthContext)

<Router>
  <Switch>
    <Route path="/login">
      user ? <Redirect to="/" /> : <Login />
    </Route>
    {user ? <div>
      </div>
      :
      <div></div>
    </div>
  </Switch>
</Router>
```

MovieContext

3 Aralik 2021 Cuma 15:10

```
export const getMoviesSuccess = (movies) => {
    type: 'GET-MOVIES-SUCCESS',
    payload: movies
}

export const createMovieSuccess = (movie) => {
    type: 'CREATE-MOVIE-SUCCESS',
    payload: movie
}

export const deleteMovieSuccess = (id) => {
    type: 'DELETE-MOVIE-SUCCESS',
    payload: id
}

export const updateMovieSuccess = (movie) => {
    type: 'UPDATE-MOVIE-SUCCESS',
    payload: movie
}
```

Movie Reducer ↴

```
const MovieReducer = (state, action) => {
    switch (action.type) {
        case 'GET-MOVIES-SUCCESS':
            return {
                movies: action.payload,
                isFetching: false,
                error: false
            }
        case 'CREATE-MOVIE-SUCCESS':
            return {
                movies: [...state.movies, action.payload],
                isFetching: false,
                error: false
            }
    }
}
```

```

case 'DELETE-MOVIE-SUCCESS':
  return {
    movies: state.movies.filter(
      (movie) => movie._id !== action.payload)
    isFetching: false,
    error: false
  }
case 'UPDATE-MOVIE-SUCCESS':
  return {
    movies: state.movies.map(
      (movie) => movie._id === action.payload._id ? action.payload,
      isFetching: false,
      error: false
    )
  }
default:
  return {
    ...state
  }
}
}

export default MovieReducer;

```

MovieContext.js

```

import MovieReducer from './MovieReducer'
import createContext from "react"

const INITIAL_STATE = {
  movies: [],
  isFetching: false,
  error: false
}

export const MovieContext = createContext(INITIAL_STATE)

export const MovieContextProvider = ({ children }) => {
  const [state, dispatch] = useReducer(MovieReducer, INITIAL_STATE)
  return (
    <MovieContext.Provider
      value={{
        movies: state.movies,
        isFetching: state.isFetching,
        error: state.error,
      }}
    >{children}</MovieContext.Provider>
  )
}

```

```
    } dispatch  
    > { children }  
< 'MovieContext.Provider'>  
    }  
  }
```

apicalls.js

```
export const getMovies = async (dispatch) => {
  dispatch(getMoviesStart())
  try {
    const res = await axios.get('/movies', {
      headers: {
        token: "Bearer " + JSON.parse(localStorage.getItem("user")).accessToken
      }
    })
    dispatch(getMoviesSuccess(res.data))
  } catch (err) {
    dispatch(getMoviesFailure())
  }
}

export const createMovie = async (movie, dispatch) => {
  dispatch(createMovieStart())
  try {
    const res = await axios.post('/movies', movie, {
      headers: {
        token: "Bearer " + JSON.parse(localStorage.getItem("user")).accessToken
      }
    })
    dispatch(createMovieSuccess(res.data))
  } catch (err) {
    dispatch(createMovieFailure())
  }
}

export const deleteMovie = async (id, dispatch) => {
  dispatch(deleteMovieStart())
  try {
    await axios.delete('/movies/' + id, {
      headers: {
        token: "Bearer " + JSON.parse(localStorage.getItem("user")).accessToken
      }
    })
  } catch (err) {
  }
}
```

```

    token : "Bearer " + JSON.parse(localStorage.getItem("user")).accessToken
  }
  try {
    dispatch(deleteMovieSuccess(id))
  } catch (err) {
    dispatch(deleteMovieFailure())
  }
}

export const updateMovie = async (id, movie, dispatch) => {
  dispatch(updateMovieStart())
  try {
    const res = await axios.put('/movies/' + id, movie, {
      headers: {
        token: "Bearer " + JSON.parse(localStorage.getItem('user')).accessToken
      }
    })
    dispatch(updateMovieSuccess(res.data))
  } catch (err) {
    dispatch(updateMovieFailure())
  }
}

```

Admin Crud Operations

3 Aralık 2021 Cuma 15:55

ProductList.jsx

```
const {movies, dispatch} = useContext(MovieContext)  
useEffect(() => {  
    getMovies(dispatch)  
}, [dispatch])
```

index.js

```
<AuthContextProvider>  
<MovieContextProvider>  
<App>  
</MovieContextProvider>  
</AuthContextProvider>
```

ProductList.jsx

```
<Datagrid  
rows={movies}>  
:  
getRowId={(r) => r.id}  
<  
fields  
id -- id  
movie -- params.row.img params.row.title  
genre --  
year  
limit  
series
```

Actions

```
<Link to={{ pathname: '/movie/' + params.row.id,  
          movie: params.row }}>
```

```
<button onClick={() => handleDelete({params.movie.id})}>
```

```
const handleDelete = (id) => {
```

```
    deleteMovie(id, d.patch)
```

Product.jsx

```
const location = useLocation()
```

```
const movie = location.movie
```

form.productForm

```
input title }  
placeholder year } text  
            genre }  
            limit }  
            trailer } file  
            video }
```

```
<img src={movie.img}/>
```

New Product.jsx

New Movie
Input file img id=img

Title

Description

Year

Limit

Genre

Select Is Series?

input file Trailer

input file video

newProduct.css

.addProductForm {

 display:

 flex-wrap: wrap

.addProductItem {

 margin: 20px

}

.button { height: 30px }

 align-self: center

const [movie, setMovie] = useState(null)

const [img, setImg] = useState(null)

const [trailer, setTrailer] = useState(null)

const [video, setVideo] = useState(null)

text inputs \Rightarrow on change = { handleChange }

select \Rightarrow on change = { handleChange }

(Is series?)



const handleChange = (e) \Rightarrow {

 const value = e.target.value

 setMovie({ ...movie, [e.target.name]: value })

}

Firebase Uploading Files

4 Aralık 2021 Cumartesi 04:37

New Product .jsx

```
<input type="file" onChange={e=>setImg(e.target.files[0])}>/>
<input type="file" onChange={e=>setText(e.target.files[0])}>/>
<input type="file" onChange={e=>setVideo(e.target.files[0])}>/>
```

firebase → Go to console → project name

npm i firebase

storage → get started

src → firebase.js

..... if true

firebase.js

```
import firebase from 'firebase';
```

```
const firebaseConfig = {
```

```
    :
```

```
}
```

```
firebase.initializeApp(firebaseConfig)
```

```
const storage = firebase.storage()
```

```
export default storage
```

NewProduct.jsx

```
const [uploaded, setUploaded] = useState(0)
```

```
{ uploaded == 3 ?
```

```
    (<button onClick={handleSubmit}>Create</button>) :
```

```
    (<button onClick={handleUpload}>Upload</button>)
```

```
}
```

```
const handleUpload = (e) => {
  e.preventDefault()
  upload([
    {file: img, label: "img"}, 
    {file: trailer, label: "trailer"}, 
    {file: video, label: "video"}])
}
```

```
const upload = (items) => {
  items.forEach(item => {
    const fileName = new Date().getTime() + item.label + item.file.name
    const uploadTask = storage.ref(`items/${fileName}`).put(item.file)
    uploadTask.on(
      "state-changed",
      (snapshot) => {
        const progress =
          (snapshot.bytesTransferred / snapshot.totalBytes) * 100
        console.log(`Upload is ${progress}% done`)
      },
      (error) => {
        console.log(error)
      },
      () => {
        uploadTask.snapshot.ref.getDownloadURL().then(url => {
          setMovie((prev) => {
            return {...prev, [item.label]: url}
          })
        })
        setUploaded((prev) => prev + 1)
      }
    )
  })
}
```

```
    3)  
        setUpload((prev) => prev + 1)  
    }  
    3)  
}  
  
const handleSubmit = (e) => {  
    e.preventDefault()  
    createMovie(movie, dispatch)  
}  
  
const { dispatch } = useContext(MovieContext)
```

List Crud

5 Aralık 2021 Pazar 04:44

Sidebar.jsx

Linklerin degerlendirme.

```
<Link to="/lists" -->  
  <List -->  
</Link>
```

listList → ListList.jsx
listList.css

newList → newList.jsx
newList.css

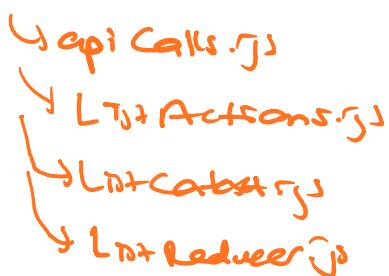
list → List.jsx
list.css

index.js

Cover my app with ListContext Provider

```
<ListContextProvider>  
  <App/>  
</ListContextProvider>
```

ListContext



title
type
genre
content

} List Model

ListList.jsx

```
const {lists, dispatch} = useContext(ListContext)
<TableGrid
  rows={lists}
  ...
  getRowId={r=>r.id}
/>
```

App.jsx

```
<Route path="/lists" element={ListList}/>
<Route path="/newList" element={NewList}/>
<Route path="/list/:id" element={List}/>
```

List.jsx

```
<Link to={`/list${person.row.id}`} state={person.row}>
  <button>Edit</button>
</Link>
```

List.jsx

```
const location = useLocation()
const list = location.state
```

NewList.jsx

```
const [list, setList] = useState(null)
const {dispatch} = useContext(ListContext)
const {movies, dispatch: dispatchMovie} = useContext(MovieContext)
const handleChange = (e)=>{
  const value = e.target.value
```

```
const handleChange = (e) => {
    const value = e.target.value
    setList({ ...list, [e.target.name]: value })
}
```

```
<select multiple name="content" onChange={handleSelect}>
    {movies.map((movie) => (
        <option key={movie.id} value={movie.id}>{movie.title}</option>
    )));
}
```

```
</select>
```

```
* movie'ler: getMovie :> ~
useEffect(() => {
    GetMovies(dispatchMovie)
}, [dispatchMovie]);
```

* select → html elementi 반환.

```
clf(e.target.selectedOptions) → (num)
```

```
const handleSelect = (e) => {
```

```
    let value = Array.from(e.target.selectedOptions, option) = option.value;
    setList({ ...list, [e.target.name]: value });
}
```

Fixing Bugs

8 Aralık 2021 Çarşamba 19:21

App.js

<Route path="/movies" element={<Home type="movie"/>} />

Home.js

<Featured type={type} setGenre={setGenre} />

Featured.js

export default function Featured({type, setGenre})

<select onChange={e=>setGenre(e.target.value)} />

Client Side Authentication

8 Aralık 2021 Çarşamba 19:32

Admin' deki Auth Context'i tamamen al.

apiCalls.js

~~req.user.isAdmin~~ dd dispatch(loginSuccess(res.data))

dispatch(loginSuccess(res.data))

App.jsx

const [user] = useAuthContext(AuthContext)

index.jsx

{AuthContext.Provider}

<App>

{/AuthContext.Provider}

Register.jsx

const handleFinish = async () => {

setPassword(passwordRef.current.value)

await axios.post('/auth/register', {email, username, password})
} navigate('/login')

const navigate = useNavigate()

const handleFinish = async () => {

e.preventDefault()

setPassword(passwordRef.current.value)

try {

await axios.post('/auth/register', {email, username, password})

navigate('/login')

} catch(err) {

console.log(err)

}

}

const [username, setUsername] = useState("")

```
<input type="text" name="username" id="username"/>
```

```
const handleFinish = (e) => {
```

```
    setUsername(usernameRef.current.value)
```

Login.jsx

```
const [email, setEmail] = useState("")
```

```
const [password, setPassword] = useState("")
```

```
<input type="email" ... onChange={e => setEmail(e.target.value)} />
```

```
<input type="password" ... onChange={e => setPassword(e.target.value)} />
```

```
<button className="loginButton" onClick={handleLogin}> Login </button>
```

```
const handleLogin = (e) => {
```

```
    e.preventDefault();
```

```
    login({email, username, password}, dispatch)
```

```
}
```

```
const {dispatch} = useContext(AuthContext)
```

Logout.jsx → Layout

```
<span onClick={() => dispatch(logout)}> Logout </span>
```

```
const {dispatch} = useContext(AuthContext)
```

Making App Responsive

8 Aralık 2021 Çarşamba 20:09

app.scss

@ mixin tablet {

 @ media (max-width: 768px) {

 @ content

}
}

@ mixin mobile {

 @ media (max-width: 480px) {

 @ content

}

}
}

navbar.scss

@import "../.../app.scss";

span {

 margin-right: 20px;

 cursor: pointer

 @ include tablet {

 display: none;

}

img {

 width: 200px

 @ include mobile {

 display: none;

}

.desc {

margin: 20px 0px;

@include mobile {

width: 60vw;

}

o — o o — o o — o — o

featured.scss

@import ".../.../app.scss";