Лабораторна робота №8

1. Створіть нову папку і склонуйте в неї проект з лабораторної №7 за допомогою команди git clone
2. Створіть Redux сховище
3. Винесіть шлях до картинки з компоненту Logo до сховища
4. Винесіть масив з елеменати меню з компоненту меню до сховища
5. Винесіть обєкт дати з компоненту годинника до сховища та оновлюйте його за допомогою редюсера

index.jsx

import { appStore } from "./redux/store";

const root = ReactDOM.createRoot(document.getElementById("root"));

root.render(

  <React.StrictMode>

    <Provider store={appStore}>

      <App />

    </Provider>

  </React.StrictMode>

);

store.js

import { applyMiddleware, createStore } from "redux";

import { thunk } from "redux-thunk";

import { actionsTypes } from "./actions";

import { initialState } from "./initialState";

export const appStore = createStore(appReducer, applyMiddleware(thunk));

function appReducer(state = initialState, action) {

  switch (action.type) {

    case actionsTypes.updateDate:

      return {

        ...state,

        date: action.payload,

      };

    case actionsTypes.updateFacts:

      return {

        ...state,

        facts: action.payload,

      };

    default:

      console.error("unknown action", action.type);

      return state;

  }

}

Для того, щоб використовувати Redux Thunks, окрім Redux та React-Redux, встановлено ще й Redux-Thunk

initialState.js

export const initialState = {

  logo: "./logo.png",

  date: new Date(),

  facts: [],

};

actions.js

export const actionsTypes = {

  updateDate: "update-date",

  updateFacts: "update-facts",

};

*/\*\**

*\* @param {Date} payload*

*\*/*

export const updateDate = (payload) => ({

  type: actionsTypes.updateDate,

  payload,

});

*/\*\**

*\* @param {string[]} payload*

*\*/*

export const updateFacts = (payload) => ({

  type: actionsTypes.updateFacts,

  payload,

});

selectors.js

export const dateSelector = (data) => data.date;

export const factsSelector = (data) => data.facts;

export const logoSelector = (data) => data.logo;

DateTime.jsx

import React, { useEffect } from "react";

import { useDispatch, useSelector } from "react-redux";

import { CurrentDate } from "../currentDate/CurrentDate";

import { CurrentTime } from "../currentTime/CurrentTime";

import { dateSelector } from "../../redux/selectors";

import { updateDateConstatntly } from "../../redux/thunks";

import classes from "./dateTime.module.css";

export const DateTime = () => {

  const dateObj = useSelector(dateSelector);

  const dispatch = useDispatch();

  useEffect(() => dispatch(updateDateConstatntly()), [dispatch]);

  return (

    <div className={classes.wrapper}>

      <CurrentTime dateObj={dateObj} />

      <CurrentDate dateObj={dateObj} />

    </div>

  );

};

thunks.js

export const updateDateConstatntly = () => (dispatch) => {

  const intervalLabel = setInterval(() => {

    dispatch(updateDate(new Date()));

  }, 1000);

  return () => {

    clearInterval(intervalLabel);

  };

};

FactsList.jsx

import React, { useCallback, useEffect } from "react";

import { useDispatch, useSelector } from "react-redux";

import { FactsListItem } from "../factsListItem/FactsListItem";

import { factsSelector } from "../../redux/selectors";

import { updateFetchedFacts } from "../../redux/thunks";

import { getId } from "../../utils/helpers";

import classes from "./factsList.module.css";

export const FactsList = () => {

  const facts = useSelector(factsSelector);

  const dispatch = useDispatch();

  const updateFacts = useCallback(() => dispatch(updateFetchedFacts()), [dispatch]);

  useEffect(() => {

    updateFacts();

  }, [updateFacts]);

  return (

    <div className={classes.container}>

      <ul className={classes.list}>

        {facts.map((fact) => (

          <FactsListItem key={getId("facts-list-item")}>{fact}</FactsListItem>

        ))}

      </ul>

      <button

        className={classes.button}

        onClick={updateFacts}>

        Get new facts

      </button>

    </div>

  );

};

thunks.js

export const updateFetchedFacts = (numberOfFacts) => async (dispatch) => {

  const abortController = new AbortController();

  const facts = await getFacts({ numberOfFacts, signal: abortController.signal });

  dispatch(updateFacts(facts.data));

  return abortController.abort;

};

Logo.jsx

import React from "react";

import { useSelector } from "react-redux";

import { logoSelector } from "../../redux/selectors";

import classes from "./logo.module.css";

export const Logo = () => {

  const logoPath = useSelector(logoSelector);

  return (

    <a

      href="/"

      className={classes.logo}>

      <img

        src={logoPath}

        alt="batman logo"

      />

    </a>

  );

};

Git репозиторій: <https://github.com/NeYaroslav/react-course-task-8>