**// Импортируем метод `createSlice` из библиотеки "@reduxjs/toolkit"**

// `createSlice` помогает создавать срезы состояния Redux с автоматической генерацией экшенов (actions) и редьюсеров (reducers).

import { createSlice } from "@reduxjs/toolkit";

**// Указываем начальное состояние для темы. По умолчанию тема будет "light".**

const initialThemeState: ThemeState = {

  theme: "light",

};

**// Создаем "срез" состояния, который включает в себя:**

export const themeSlice = createSlice({

**// Указываем имя среза, чтобы идентифицировать его в хранилище Redux.**

  name: "theme",

**// Указываем начальное состояние для данного среза.**

  initialState: initialThemeState,

**// Описываем функции-редьюсеры, которые изменяют состояние.**

  reducers: {

    toggleTheme: state => {

      state.theme = state.theme === "light" ? "dark" : "light";

    },

  },

**// Селекторы (необязательно, но можно добавить для удобства извлечения данных из состояния Redux).**

  selectors: {

    // `selectTheme` — функция-селектор, которая извлекает текущую тему из состояния.

    selectTheme: theme => theme.theme,

  },

});

**// Экспортируем экшен `toggleTheme`, чтобы его можно было использовать в компонентах для изменения состояния темы.**

// `themeSlice.actions` автоматически генерирует все экшены на основе названий функций редьюсеров.

export const { toggleTheme } = themeSlice.actions;

**// Экспортируем селектор `selectTheme`, чтобы компоненты могли получать значения текущей темы из состояния.**

export const { selectTheme } = themeSlice.selectors;

**// Экспортируем редьюсер `themeReducer`. Этот редьюсер будет использоваться в настройке хранилища Redux.**

// Его задача — управлять состоянием в зависимости от вызванных экшенов.

export const themeReducer = themeSlice.reducer;