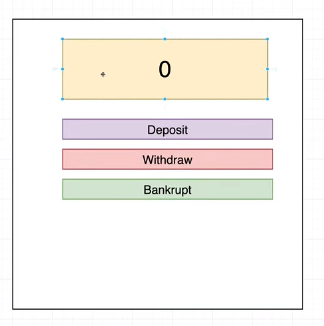
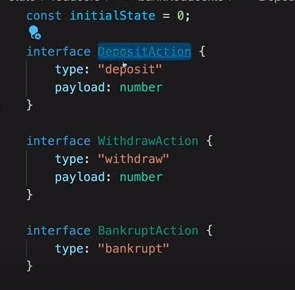
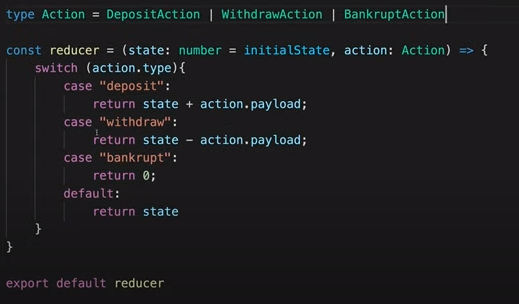
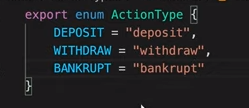
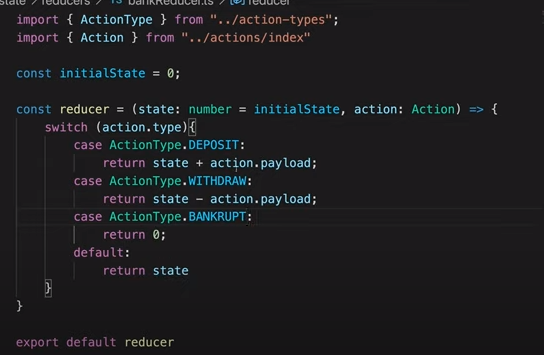
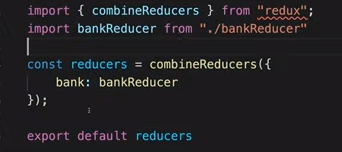
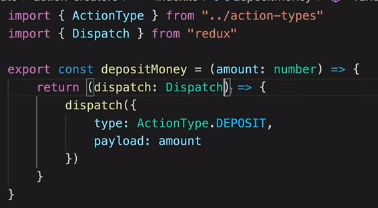
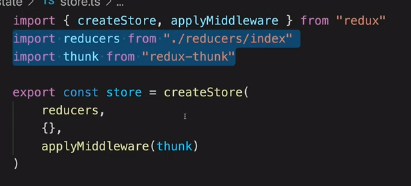
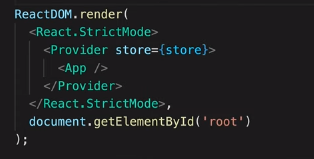
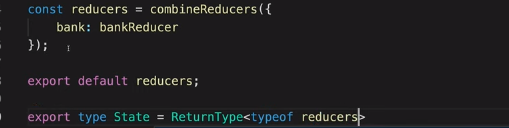
**1) Creating typescript react app.**  
  
**2) Reducers & Typescript.**  
  
Our App.  
  
  
Папка state, папка reducers, файл bankReducer.  
  
Создаём так же initial state.  
Создаём интерфейсы для всех случаев action.  
  
  
Объединяем их в один тип и типизурем этим типом action как принимаемый параметр в reducer.  
  
  
**3) File organization.**  
  
Хорошая идея отделить типизацию и логику.  
Папка actions и actions.ts.  
Перенести туда все типы и экспорт type Action, с последующим import этого типа в редьюсер.  
  
Ещё хорошая практика создать папку action-types, чтобы не надеяться на простые строки в action.  
Назначим в этом файле enum ActionType.  
  
Теперь импортируем его везде, где есть строки как Action type.  
  
  
  
Теперь лучше сделать combineReducers.  
  
Для этого нужно подключить уже сам redux.  
npm install redux react-redux redux-thunk @types/react-redux @types/redux-thunk.  
  
Продолжим работу с combineReducers.  
  
  
**4) Action Creators.**  
  
Создаем папку action-creators.  
Нам нужно создать 3 различные функции под каждый action.  
  
Типизируем dispatch с помощью Dispatch от redux.  
  
  
Но Dispatch Не будет отлавливать типизацию action.  
Тогда нужно как Generic указать action.  
  
  
  
Теперь следующий шаг – создадим файл в папке state.  
Создадим store.ts.  
Создаём в нём store.  
  
  
  
  
**5) Using Redux.**  
Создаём Prodiver в index tsx.  
  
  
  
Создаём dispatch с помощью useDispatch.  
Получаем все action Creators.  
  
  
  
С useSelector достаём наш state.  
Но тут проблема с типизацией.  
Фикс: создаём type у нашего reducer.  
  
  
Теперь импортируем этот тип и подставляем.  
  
  
  
Теперь задача назначить actionCreators на клик.