

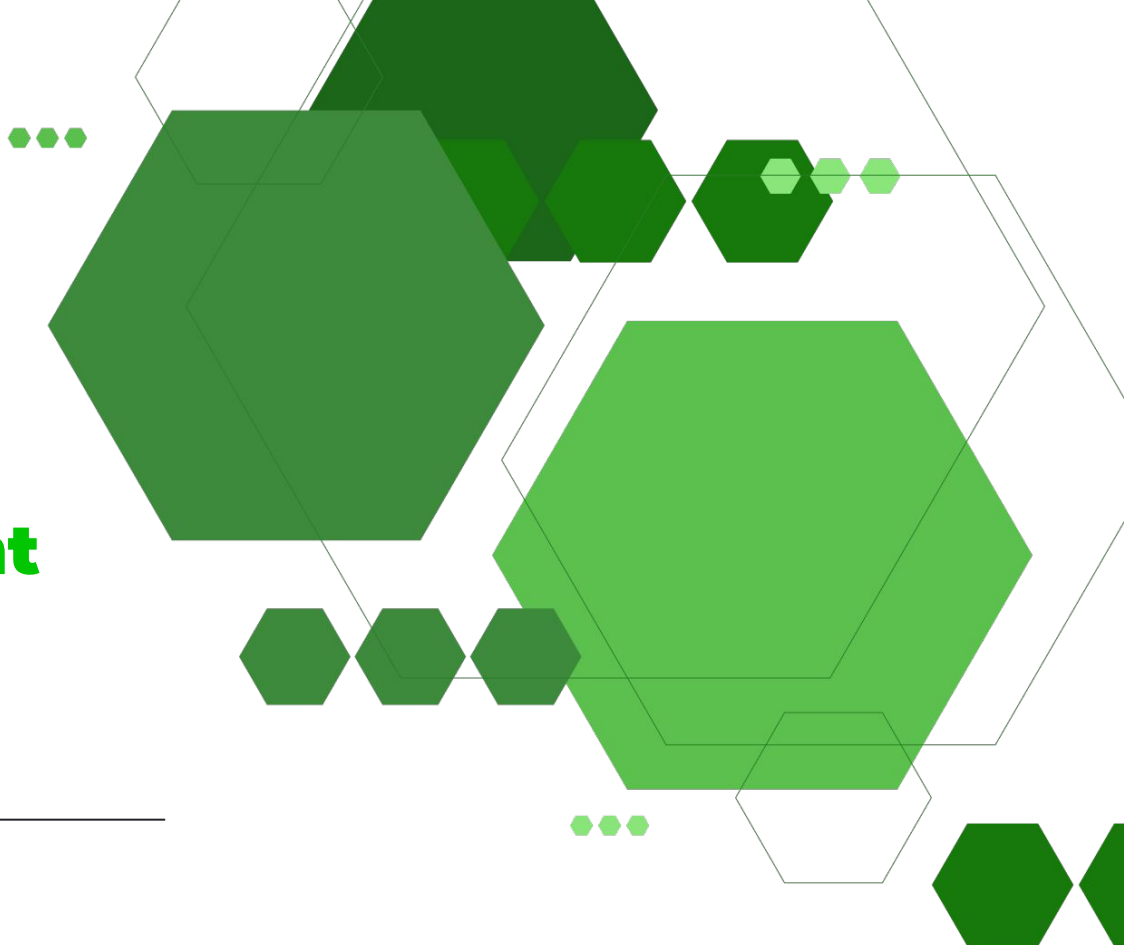
SKILLFACTORY

React State Management

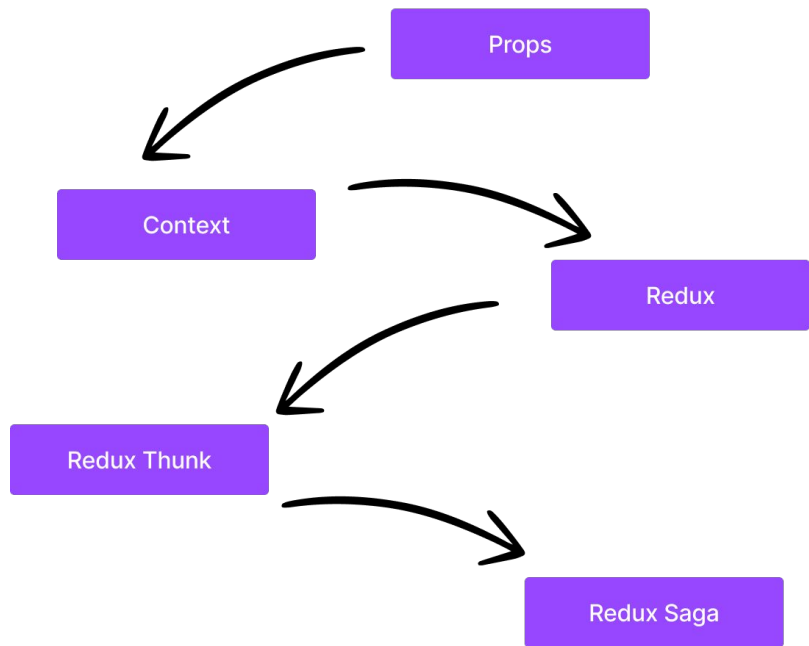
Сергей Кошельников

team lead / tech lead / developer (front/back)

Telegram: @koshelnikov



План вебинара

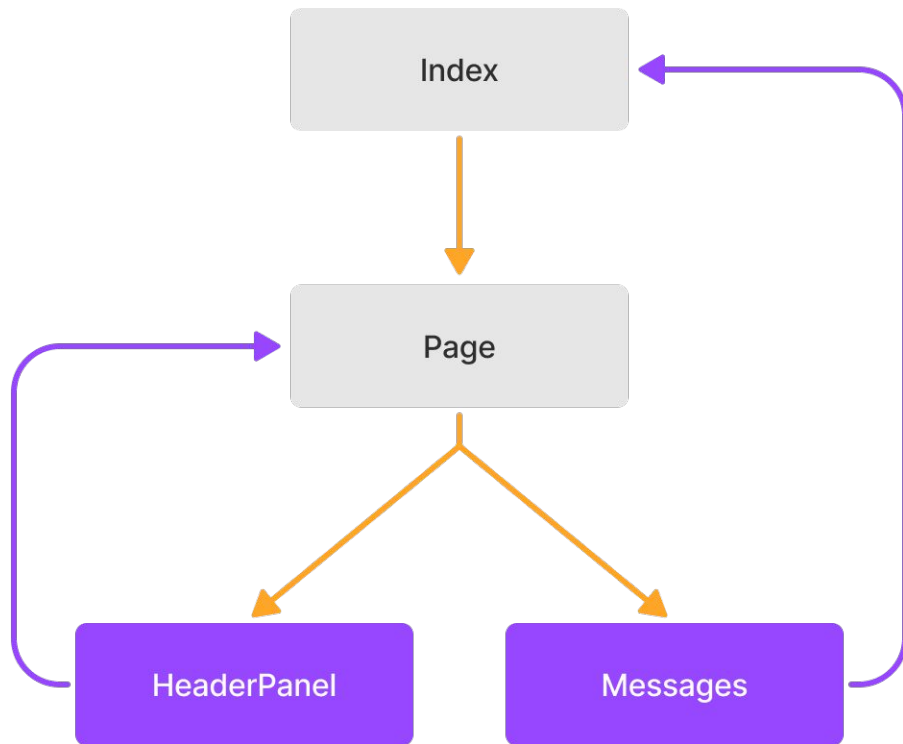


Props

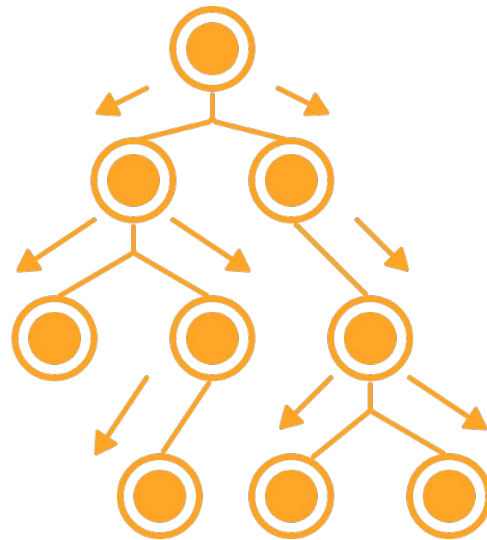
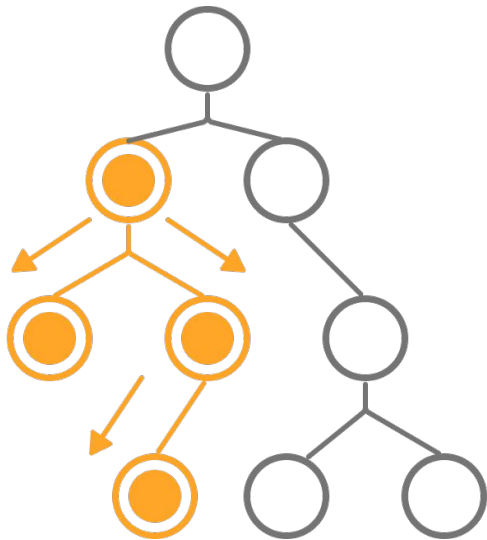
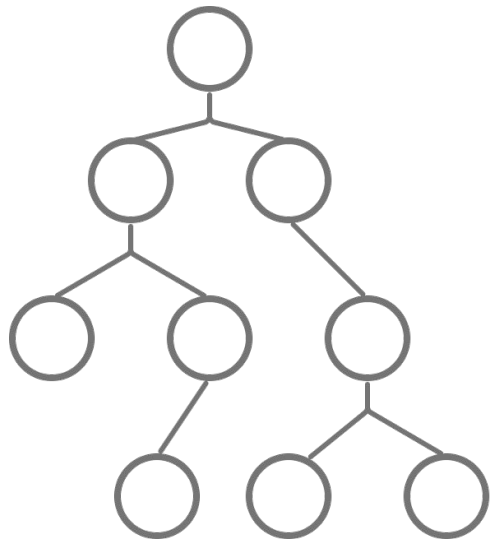
Односторонний поток данных от родителя к потомку

Component(props) - поток данных направлен сверху вниз

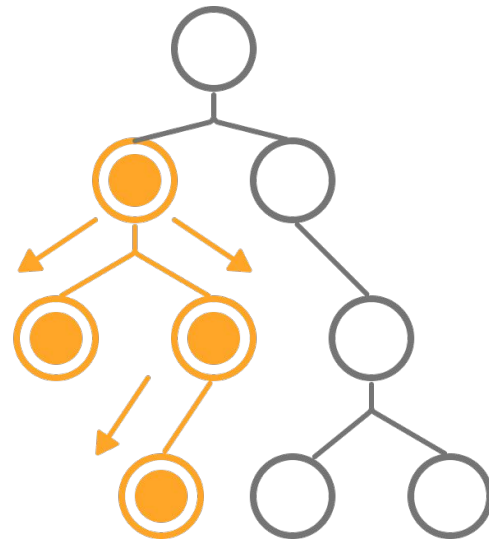
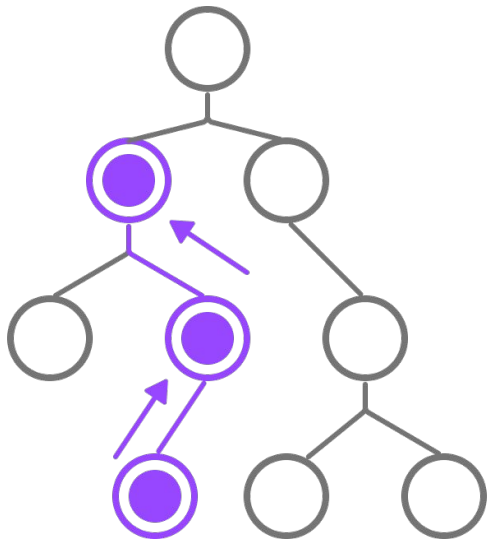
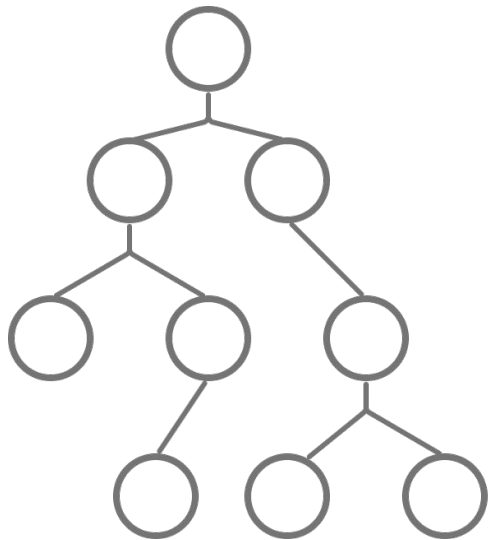
props.onChange - инициализация смены состояния снизу вверх

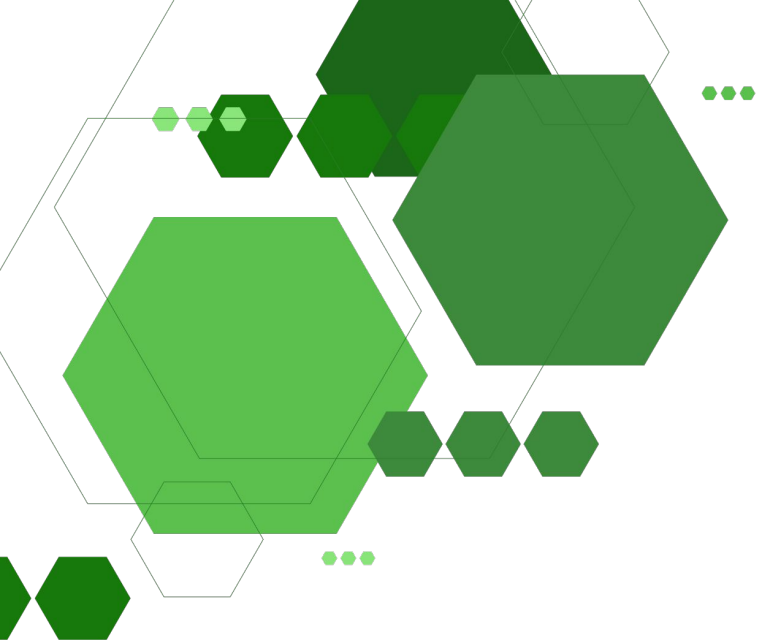


Односторонний поток данных от родителя к потомку



Смена состояния





Вопросы?



Context

Предоставляет возможность дереву React компонентов иметь общий доступ к одним данным.

createContext() - создаёт контекст

Context.Provider - JSX элемент, который позволяет дочерним React компонентам подписываться на Context

useContext - hook для доступа к Context

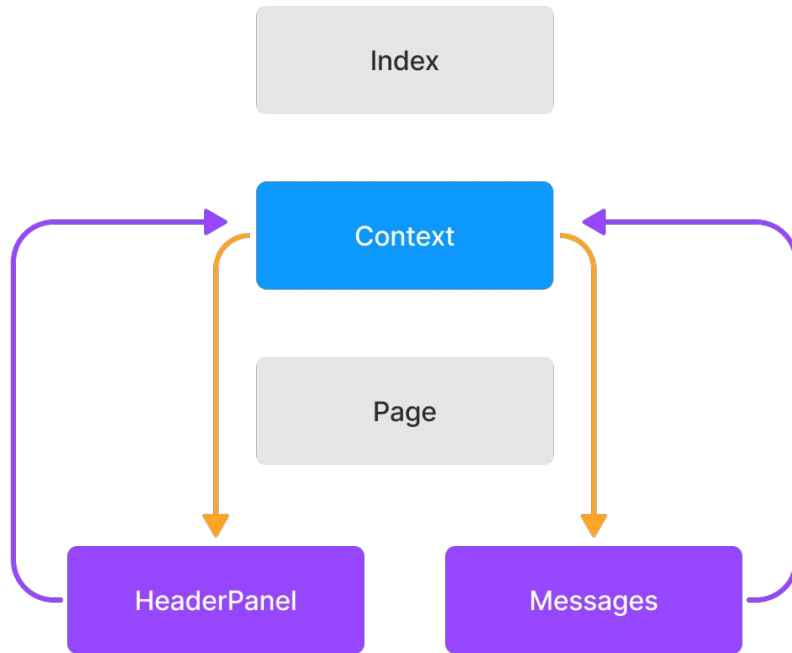
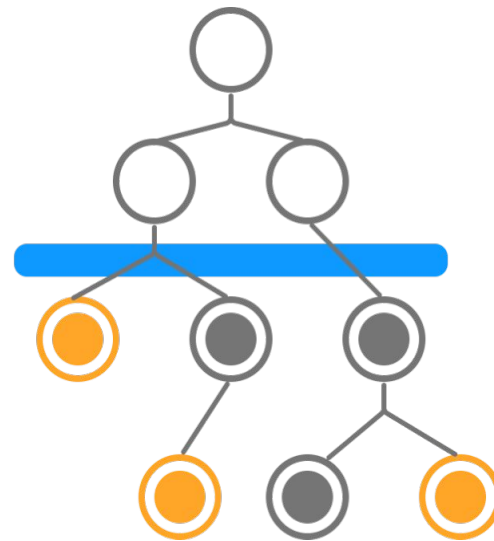
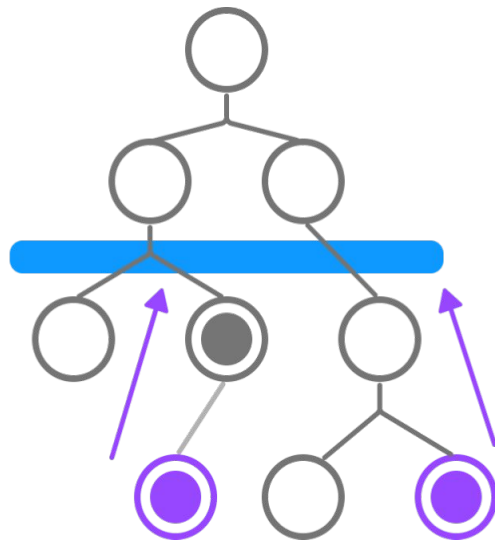
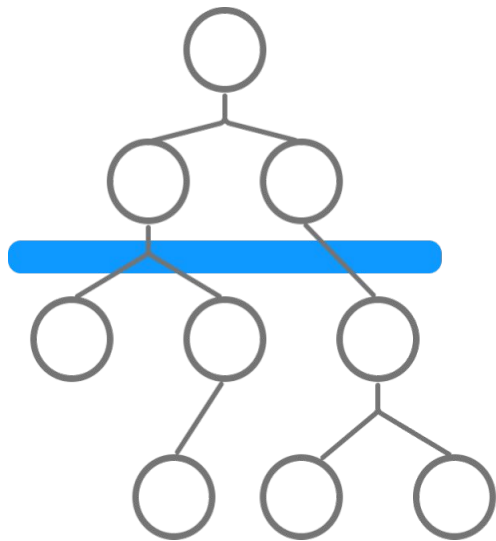
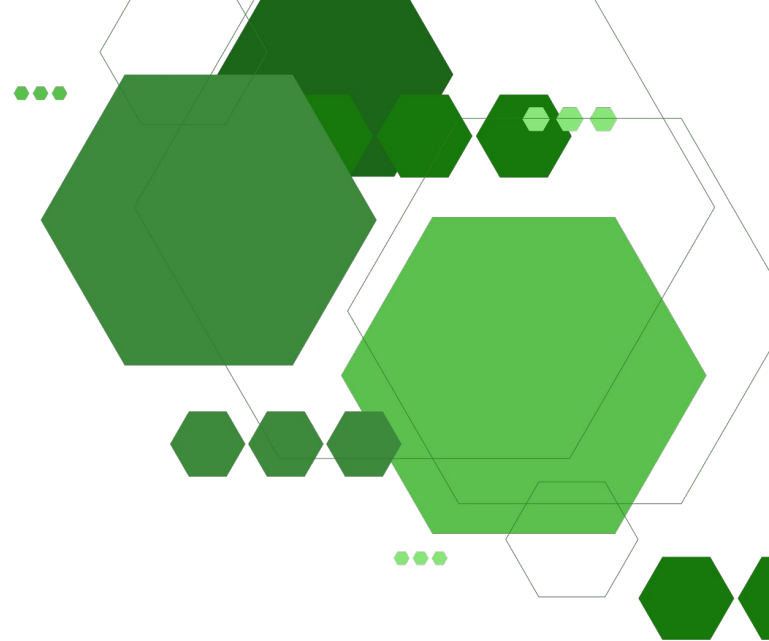


Схема потока данных



Вопросы?



Redux

Централизованное хранилище данных

createStore() - создаёт хранилище

Action - действие для изменения состояния хранилища данных, представленное в виде объекта.

Reducer - чистая функция с двумя аргументами state и action. State - это текущее состояние хранилища.

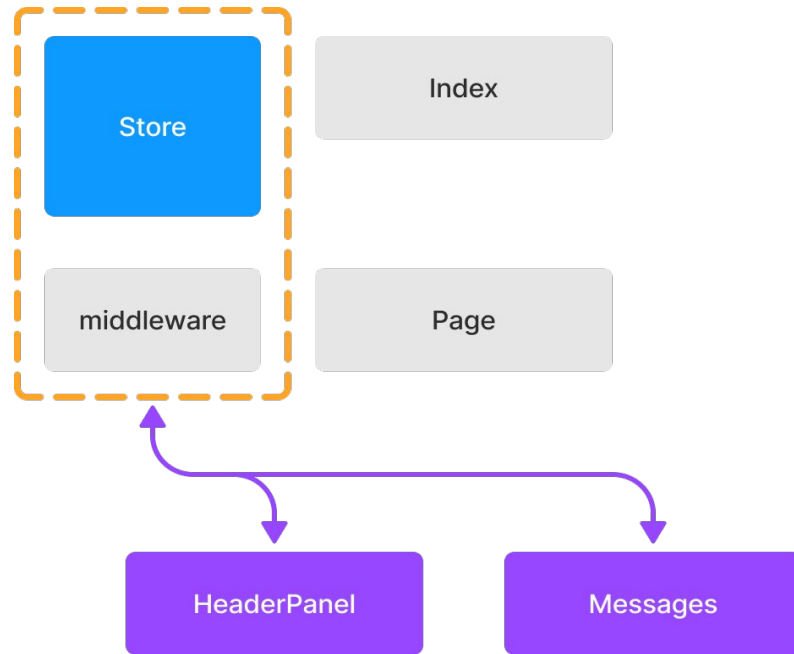
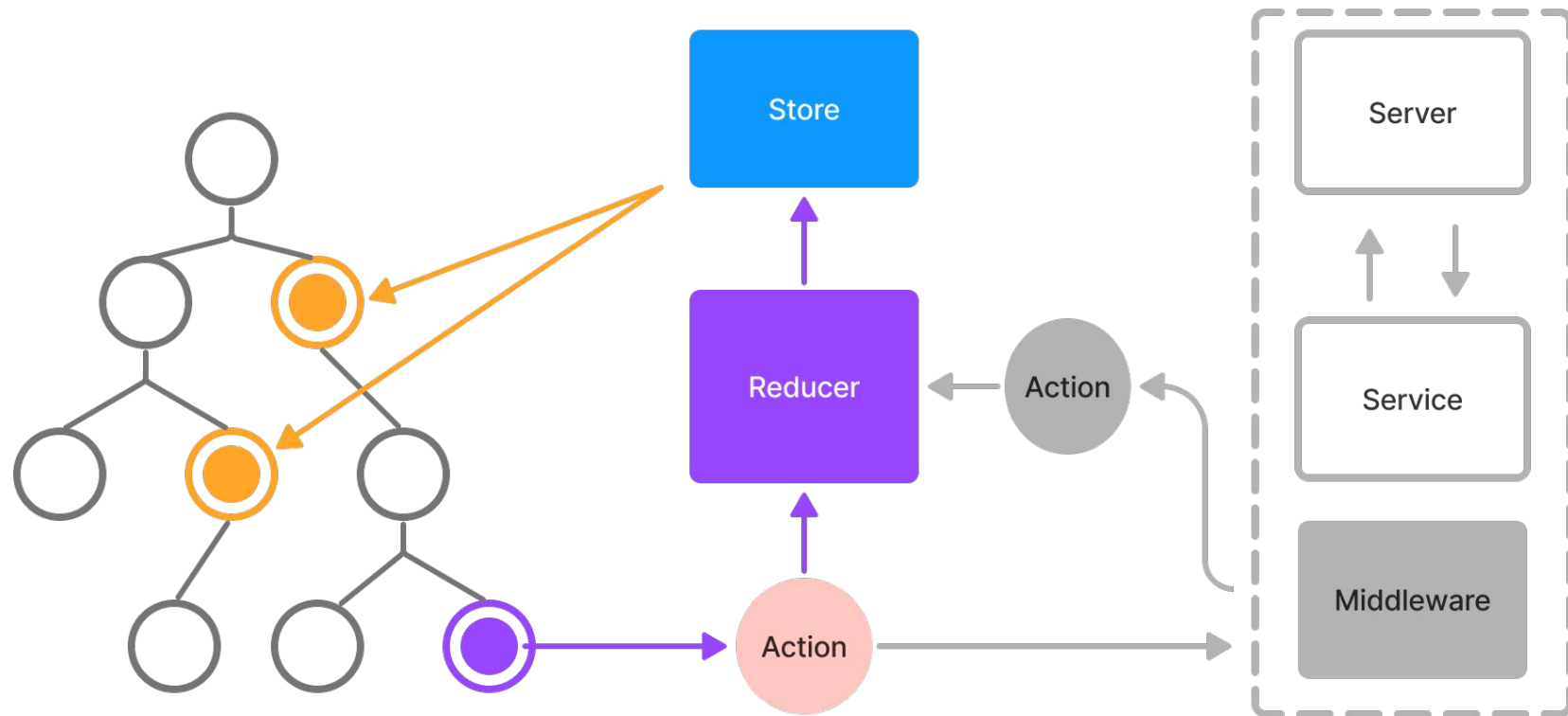
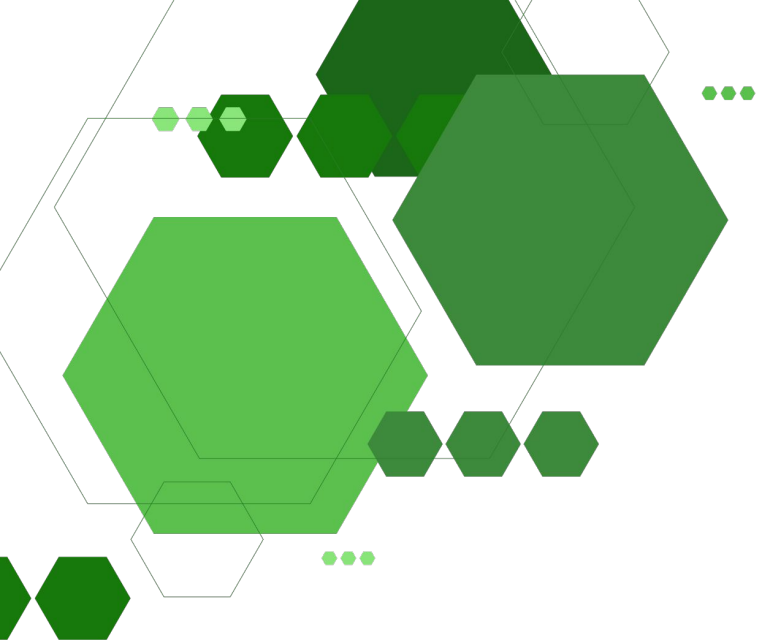


Схема потока данных





Вопросы?



Redux Thunk

Middleware Thunk исполняет Action функции с асинхронным кодом. Компоненты не знают ничего об асинхронности.

Thunk Action

```
(dispatch, store) => {  
  fetch rest api  
}
```

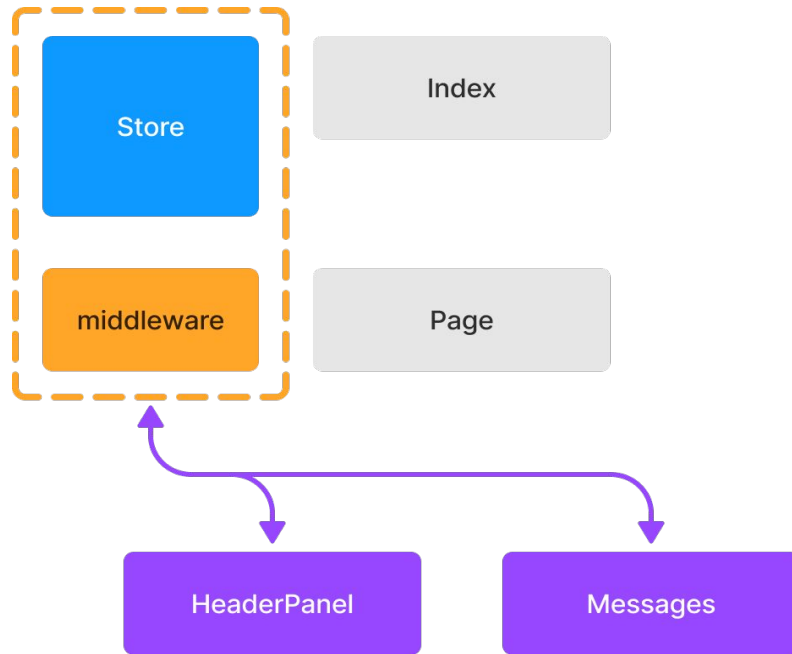
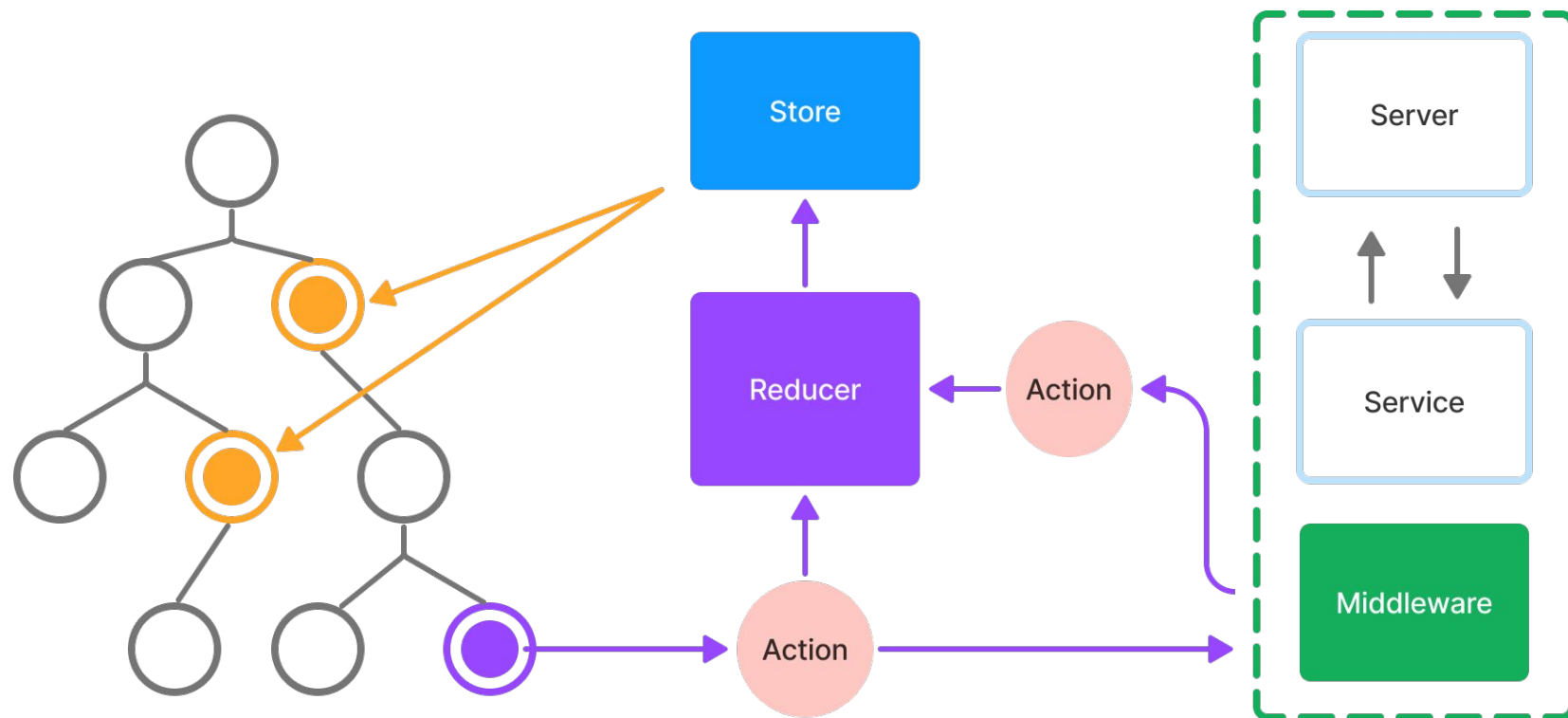
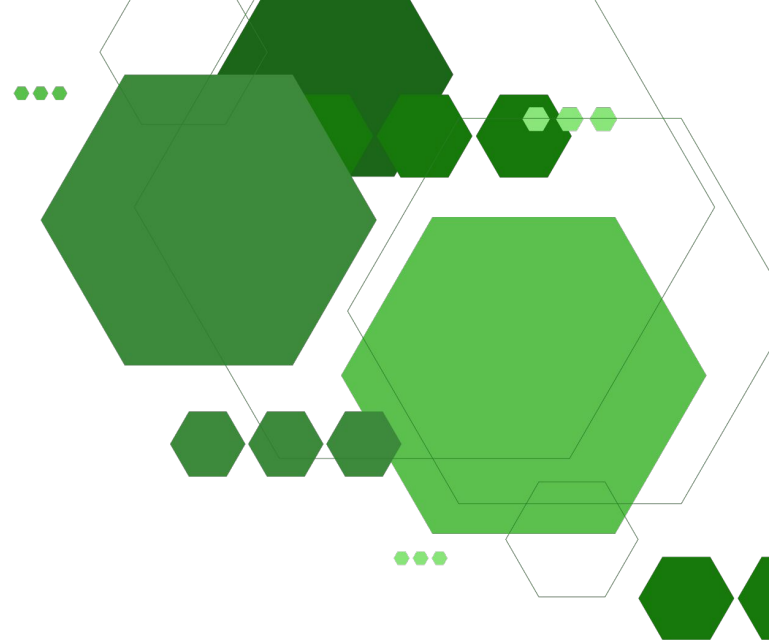


Схема потока данных



Вопросы?



Redux Saga

Асинхронные Action реализуется при помощи генератора из ES6. Saga Middleware выполняет асинхронные action.

Saga Action

```
function* someAction() {  
  yield put({action})  
  yield call(() => fetch rest api)  
}
```

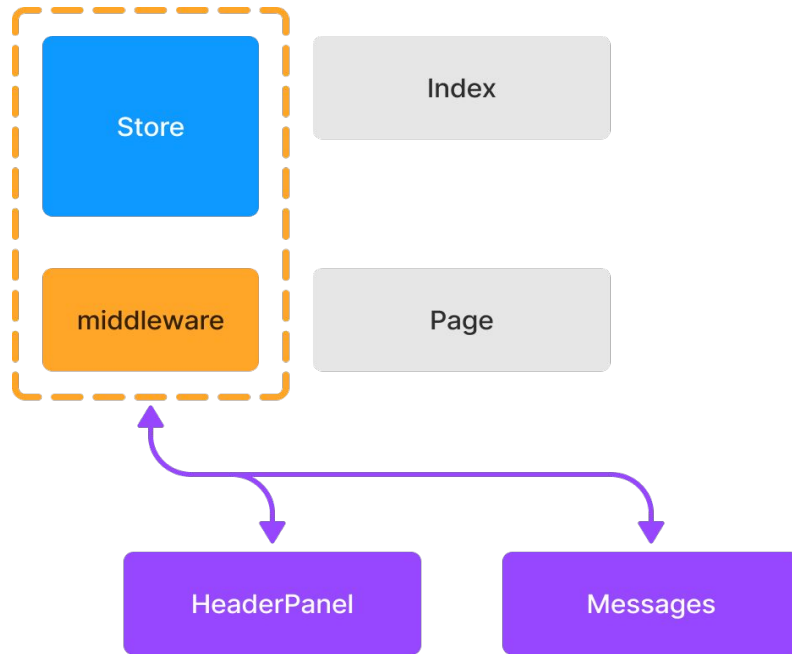
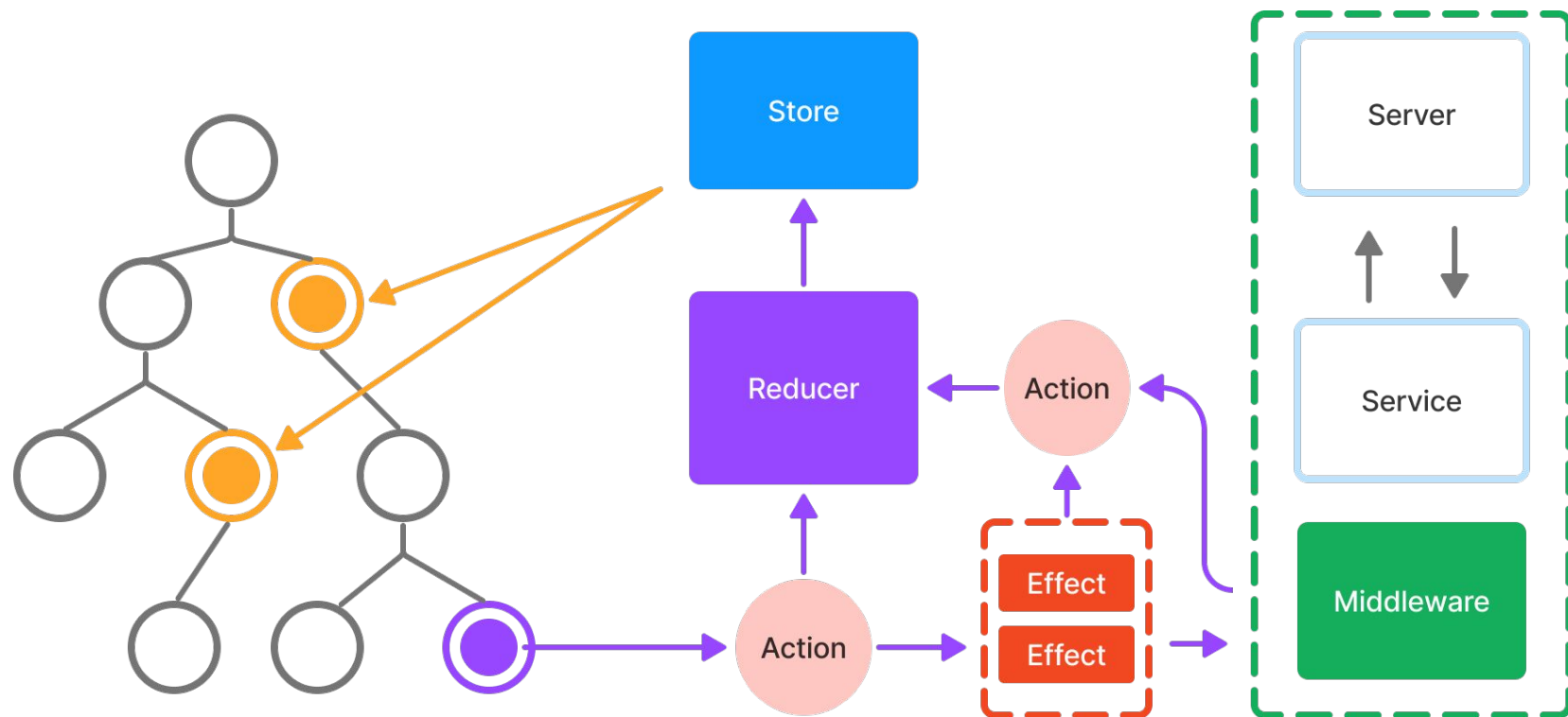
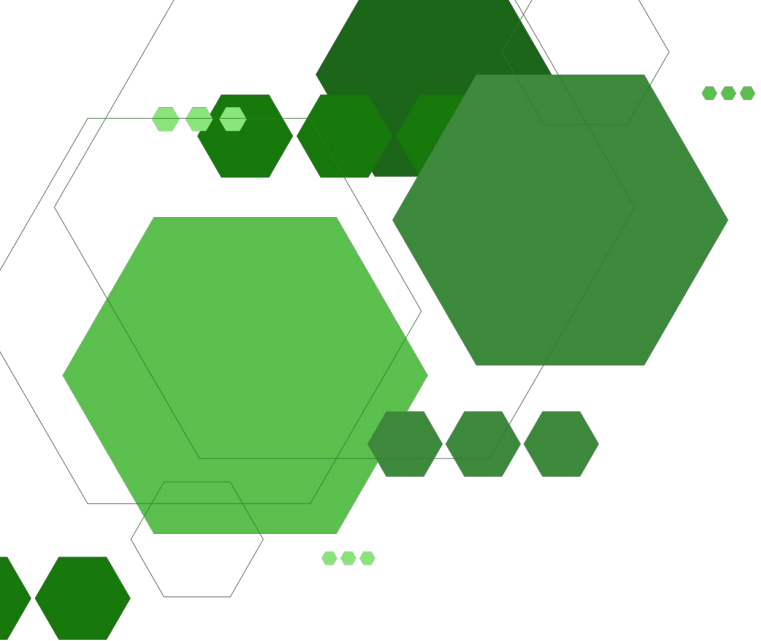


Схема потока данных



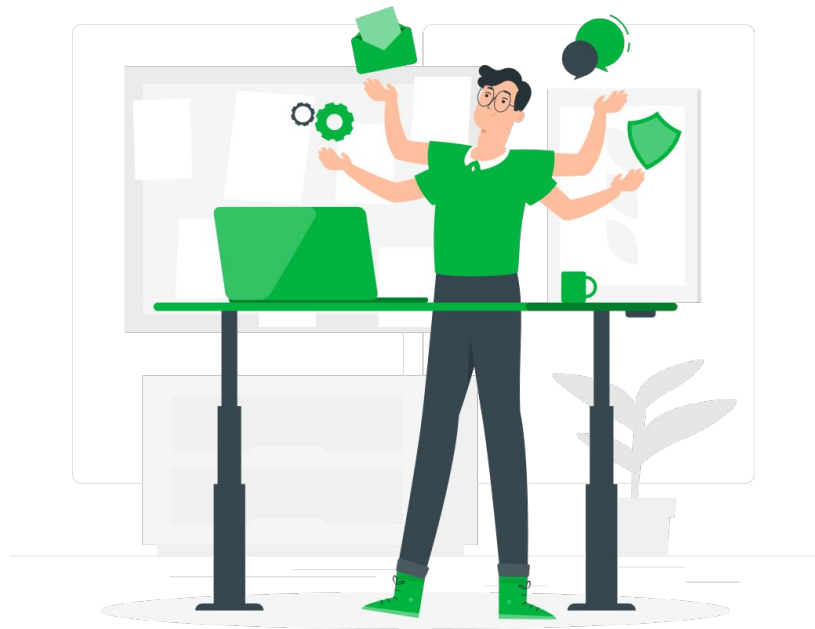


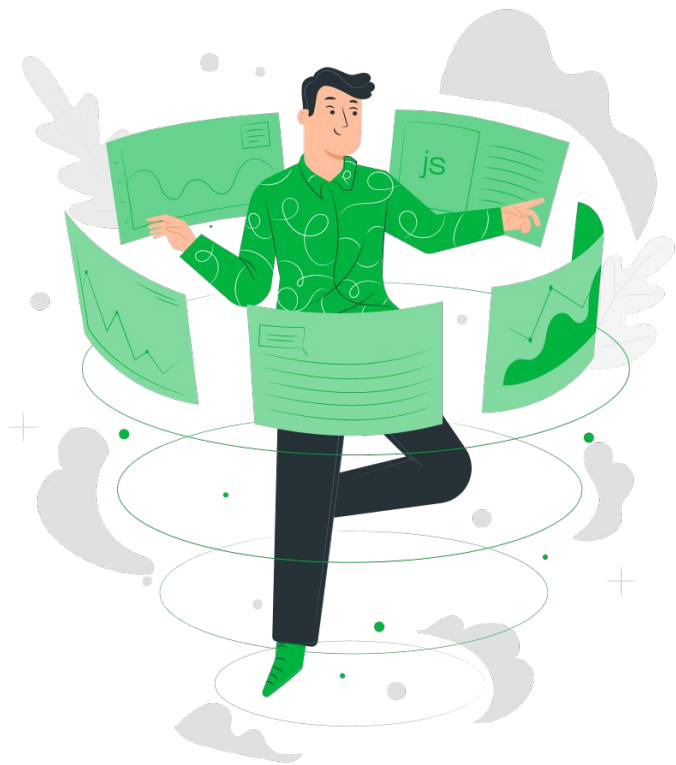
Вопросы?



Подведем итоги

1. Вспомнили one way data flow
2. Разобрали Context подход доступа к данным из разных компонентов
3. Узнали как работать с Redux
4. Добавили асинхронность с помощью Redux Thunk и Redux Saga





Полезные ссылки

- [React Props](#)
- [Thinking in React](#)
- [React Context](#)
- [Redux Toolkit](#)
- [Redux Thunk](#)
- [Redux Saga](#)
- [Github проект](#)

Спасибо за внимание Успехов ❤️



Сергей Кошельников

team lead / tech lead / developer (front/back)

Telegram: @koshelnikov