5.Pizza

1.Качаем фейковую DB с гитхаба db.json:

{

    "pizzas": [

      {

        "id": 0,

        "imageUrl": "https://dodopizza.azureedge.net/static/Img/Products/f035c7f46c0844069722f2bb3ee9f113\_584x584.jpeg",

        "name": "Пепперони Фреш с перцем",

        "types": [0, 1],

        "sizes": [26, 30, 40],

        "price": 803,

        "category": 0,

        "rating": 4

      },

      {

        "id": 1,

        "imageUrl": "https://dodopizza.azureedge.net/static/Img/Products/Pizza/ru-RU/2ffc31bb-132c-4c99-b894-53f7107a1441.jpg",

        "name": "Сырная",

        "types": [0],

        "sizes": [26, 40],

        "price": 245,

        "category": 1,

        "rating": 6

      },

      {

        "id": 2,

        "imageUrl": "https://dodopizza.azureedge.net/static/Img/Products/Pizza/ru-RU/6652fec1-04df-49d8-8744-232f1032c44b.jpg",

        "name": "Цыпленок барбекю",

        "types": [0],

        "sizes": [26, 40],

        "price": 295,

        "category": 1,

        "rating": 4

      },

      {

        "id": 3,

        "imageUrl": "https://dodopizza.azureedge.net/static/Img/Products/Pizza/ru-RU/af553bf5-3887-4501-b88e-8f0f55229429.jpg",

        "name": "Кисло-сладкий цыпленок",

        "types": [1],

        "sizes": [26, 30, 40],

        "price": 275,

        "category": 2,

        "rating": 2

      },

      {

        "id": 4,

        "imageUrl": "https://dodopizza.azureedge.net/static/Img/Products/Pizza/ru-RU/b750f576-4a83-48e6-a283-5a8efb68c35d.jpg",

        "name": "Чизбургер-пицца",

        "types": [0, 1],

        "sizes": [26, 30, 40],

        "price": 415,

        "category": 3,

        "rating": 8

      },

      {

        "id": 5,

        "imageUrl": "https://dodopizza.azureedge.net/static/Img/Products/Pizza/ru-RU/1e1a6e80-b3ba-4a44-b6b9-beae5b1fbf27.jpg",

        "name": "Крэйзи пепперони",

        "types": [0],

        "sizes": [30, 40],

        "price": 580,

        "category": 2,

        "rating": 2

      },

      {

        "id": 6,

        "imageUrl": "https://dodopizza.azureedge.net/static/Img/Products/Pizza/ru-RU/d2e337e9-e07a-4199-9cc1-501cc44cb8f8.jpg",

        "name": "Пепперони",

        "types": [0, 1],

        "sizes": [26, 30, 40],

        "price": 675,

        "category": 1,

        "rating": 9

      },

      {

        "id": 7,

        "imageUrl": "https://dodopizza.azureedge.net/static/Img/Products/Pizza/ru-RU/d48003cd-902c-420d-9f28-92d9dc5f73b4.jpg",

        "name": "Маргарита",

        "types": [0, 1],

        "sizes": [26, 30, 40],

        "price": 450,

        "category": 4,

        "rating": 10

      },

      {

        "id": 8,

        "imageUrl": "https://dodopizza.azureedge.net/static/Img/Products/Pizza/ru-RU/ec29465e-606b-4a04-a03e-da3940d37e0e.jpg",

        "name": "Четыре сезона",

        "types": [0, 1],

        "sizes": [26, 30, 40],

        "price": 395,

        "category": 5,

        "rating": 10

      },

      {

        "id": 9,

        "imageUrl": "https://dodopizza.azureedge.net/static/Img/Products/Pizza/ru-RU/30367198-f3bd-44ed-9314-6f717960da07.jpg",

        "name": "Овощи и грибы 🌱",

        "types": [0, 1],

        "sizes": [26, 30, 40],

        "price": 285,

        "category": 5,

        "rating": 7

      }

    ]

  }

2.Учимся делать fetch запросы.Идем в App.js и пришем useEffect:

useEffect(() => {

}, []

)

В таком виде это аналог componentDidMount,выполнится один раз ,когдакогда компонент вмонтируется.

3.Пробуем отправлять запросы при помощи fetch:

fetch("http://localhost:3000/db.json")

.then((response)=>console.log(response.json())).then(response=>console.log(response))

Метод вытягивает json() тело из респонса.Мы получим обьект json.

4.Помещаем наш запрос в useEffect и вытягиваем из респонса свойство pizzas:

useEffect(() => {

  fetch("http://localhost:3000/db.json")

.then((response)=>response.json()).then(json=>{

  setPizzas(json.pizzas)

})

}, []

)

5.Создаем хук:

const[pizzas,setPizzas]=useState([])

6.Создаем HomeCopy.jsx и удаляем все блоки пиц кроме одного.Используем Map и раскидываем свойства по компоненте.Облагораживаем наш Home.jsx:

import React from 'react'

import CategoriesWithHooks from '../CategoriesWithHooks'

import SortPopupWithHooks from '../SortPopupWithHooks'

function Home(props) {

    return (

        <div>

                  <div className="container">

          <div className="content\_\_top">

          <CategoriesWithHooks onClickItem={(elem)=>console.log(elem)}  items={["Мясные","Вегетарианская","Гриль","Острые","Закрытые"]}/>

           <SortPopupWithHooks varies={["популярности","цене","алфавиту"]}/>

          </div>

          <h2 className="content\_\_title">Все пиццы</h2>

          <div className="content\_\_items">

      {props.pizzas.map((elem,index)=>{

    return    <div key={elem.id}className="pizza-block">

  <img

    className="pizza-block\_\_image"

    src={elem.imageUrl}/\*"https://dodopizza-a.akamaihd.net/static/Img/Products/Pizza/ru-RU/b750f576-4a83-48e6-a283-5a8efb68c35d.jpg"\*/

    alt="Pizza"

  />

  <h4 className="pizza-block\_\_title">{/\*Чизбургер-пицца\*/}{elem.name}</h4>

  <div className="pizza-block\_\_selector">

    <ul>

     {elem.types.map((item,index)=> {

     return <li key={index}className="active">{/\*тонкое\*/}{item}</li>}

     )}

    </ul>

    <ul>

      {elem.sizes.map((item,index)=>{

      return <li key={index} className="active">{/\*26 см.\*/}{item}</li>}

      )}

     {/\* <li>30 см.</li>

      <li>40 см.</li>\*/}

    </ul>

  </div>

  <div className="pizza-block\_\_bottom">

    <div className="pizza-block\_\_price">{/\*от 395 ₽\*/}{"от "+elem.price+" ₽\*"}</div>

    <div className="button button--outline button--add">

      <svg

        width="12"

        height="12"

        viewBox="0 0 12 12"

        fill="none"

        xmlns="http://www.w3.org/2000/svg"

      >

        <path

          d="M10.8 4.8H7.2V1.2C7.2 0.5373 6.6627 0 6 0C5.3373 0 4.8 0.5373 4.8 1.2V4.8H1.2C0.5373 4.8 0 5.3373 0 6C0 6.6627 0.5373 7.2 1.2 7.2H4.8V10.8C4.8 11.4627 5.3373 12 6 12C6.6627 12 7.2 11.4627 7.2 10.8V7.2H10.8C11.4627 7.2 12 6.6627 12 6C12 5.3373 11.4627 4.8 10.8 4.8Z"

          fill="white"

        />

      </svg>

      <span>Добавить</span>

      <i>{/\*2\*/}{elem.rating}</i>

    </div>

  </div>

</div>

      })

}

         </div>

        </div>

        </div>

    )

}

export default Home

Промежуточное значение.

7.Создаем Pizza Block и прокидываем туда пропсы:

import React from 'react'

function PizzaBlock(props) {

    return (

        <div>

    <div className="pizza-block">

  <img

    className="pizza-block\_\_image"

    src={props.imageUrl}

    alt="Pizza"

  />

  <h4 className="pizza-block\_\_title">Чизбургер-пицца</h4>

  <div className="pizza-block\_\_selector">

    <ul>

      {props.types.map((elem,index)=>{

        return <li key={index} className="active">{elem===0?"тонкое":"традиционное"}</li>

})

     }

    </ul>

    <ul>

      {props.sizes.map((elem,index)=>{

      return <li key={index} className="active">{elem}</li>})

      }

    </ul>

  </div>

  <div className="pizza-block\_\_bottom">

    <div className="pizza-block\_\_price">{`от ${props.price} ₽`}</div>

    <div className="button button--outline button--add">

      <svg

        width="12"

        height="12"

        viewBox="0 0 12 12"

        fill="none"

        xmlns="http://www.w3.org/2000/svg"

      >

        <path

          d="M10.8 4.8H7.2V1.2C7.2 0.5373 6.6627 0 6 0C5.3373 0 4.8 0.5373 4.8 1.2V4.8H1.2C0.5373 4.8 0 5.3373 0 6C0 6.6627 0.5373 7.2 1.2 7.2H4.8V10.8C4.8 11.4627 5.3373 12 6 12C6.6627 12 7.2 11.4627 7.2 10.8V7.2H10.8C11.4627 7.2 12 6.6627 12 6C12 5.3373 11.4627 4.8 10.8 4.8Z"

          fill="white"

        />

      </svg>

      <span>Добавить</span>

      <i>{props.rating}</i>

    </div>

  </div>

</div>

        </div>

    )

}

export default PizzaBlock

8.Идем дальше и создаем хуки useState:

Вроде более менее готово.Еще один промежуточный:

import React,{useEffect, useRef, useState} from 'react'

function PizzaBlock(props) {

const[activePizzaSize,setActivePizzaSize]=useState(0)

const[activePizzaType,setActivePizzaType]=useState(0)

let[count,setCount]=useState(props.rating)

const pizzaRef=useRef()

/\*console.log(pizzaRef)\*/

console.log(count)

function func(){

setCount(count++)

console.log(pizzaRef)

}

useEffect(()=>{

pizzaRef.current.addEventListener("click",func)},

[]

)

const chooseSize=(index)=>{

setActivePizzaSize(index)

}

const chooseType=(index)=>{

setActivePizzaType(index)

}

console.log(activePizzaSize)

    return (

        <div>

    <div className="pizza-block">

  <img

    className="pizza-block\_\_image"

    src={props.imageUrl}

    alt="Pizza"

  />

  <h4 className="pizza-block\_\_title">Чизбургер-пицца</h4>

  <div className="pizza-block\_\_selector">

    <ul>

      {props.types.map((elem,index)=>{

        return <li key={index} onClick={()=>chooseType(index)} className={activePizzaType===index?"active":""}>{elem===0?"тонкое":"традиционное"}</li>

})

     }

    </ul>

    <ul>

      {props.sizes.map((elem,index)=>{

      return <li key={index} onClick={()=>chooseSize(index)}

      className={activePizzaSize===index?"active":""}>{elem}</li>})

      }

    </ul>

  </div>

  <div className="pizza-block\_\_bottom">

    <div className="pizza-block\_\_price">{`от ${props.price} ₽`}</div>

    <div className="button button--outline button--add" ref={pizzaRef} /\*onClick={func}\*/ >

      <svg

        width="12"

        height="12"

        viewBox="0 0 12 12"

        fill="none"

        xmlns="http://www.w3.org/2000/svg"

      >

        <path

          d="M10.8 4.8H7.2V1.2C7.2 0.5373 6.6627 0 6 0C5.3373 0 4.8 0.5373 4.8 1.2V4.8H1.2C0.5373 4.8 0 5.3373 0 6C0 6.6627 0.5373 7.2 1.2 7.2H4.8V10.8C4.8 11.4627 5.3373 12 6 12C6.6627 12 7.2 11.4627 7.2 10.8V7.2H10.8C11.4627 7.2 12 6.6627 12 6C12 5.3373 11.4627 4.8 10.8 4.8Z"

          fill="white"

        />

      </svg>

      <span >Добавить</span>

      <i>{/\*pizzaRating\*/}{count}</i>

    </div>

  </div>

</div>

        </div>

    )

}

export default PizzaBlock

9.Устанавливаем библиотеку Axios:

npm install axios.Axios конвертирует response в обьект json.Вытягивает его из body в отличие от fetch.Благодаря Axios мы из:

useEffect(() => {

  fetch("http://localhost:3000/db.json")

.then((response)=>response.json()).then(json=>{

  setPizzas(json.pizzas)

})

}, []

)

Делаем:

useEffect(() => {

  axios.get("http://localhost:3000/db.json").

then(data=>{

  setPizzas(data.pizzas)

})

}, []

)

10.Импортируем prop-types из PropTypes:

import PropTypes from 'prop-types';

11.Теперь прописываем :

PizzaBlock.propTypes={

name:PropTypes.string.isRequired

}

Заменяет в reactJs некоторые вещи из тапскрипта.Строгую типизацию.

Пишем так:

PizzaBlock.propTypes={

name:PropTypes.string.isRequired,

rating:PropTypes.number.isRequired,

price:PropTypes.number.isRequired,

types:PropTypes.arrayOf(PropTypes.number).isRequired,

sizes:PropTypes.arrayOf(PropTypes.number).isRequired

}

12.Значения по умолчанию.Если вдруг какие то свойства без которых приложение ломается не будут переданы,мы передаем их по умолчанию:

PizzaBlock.defaultProps={

price:0,

types:[],

sizes:[]

}

13.Это все добавляет более менее нормальную типизацию в компоненту без использования тайпскрипта.