

Отчет по лабораторной работе № 8 по курсу
"Разработка Интернет-Приложений"

Выполнила:
Студентка группы
ИУ5-55Б
Зубарева Антонина Михайловна

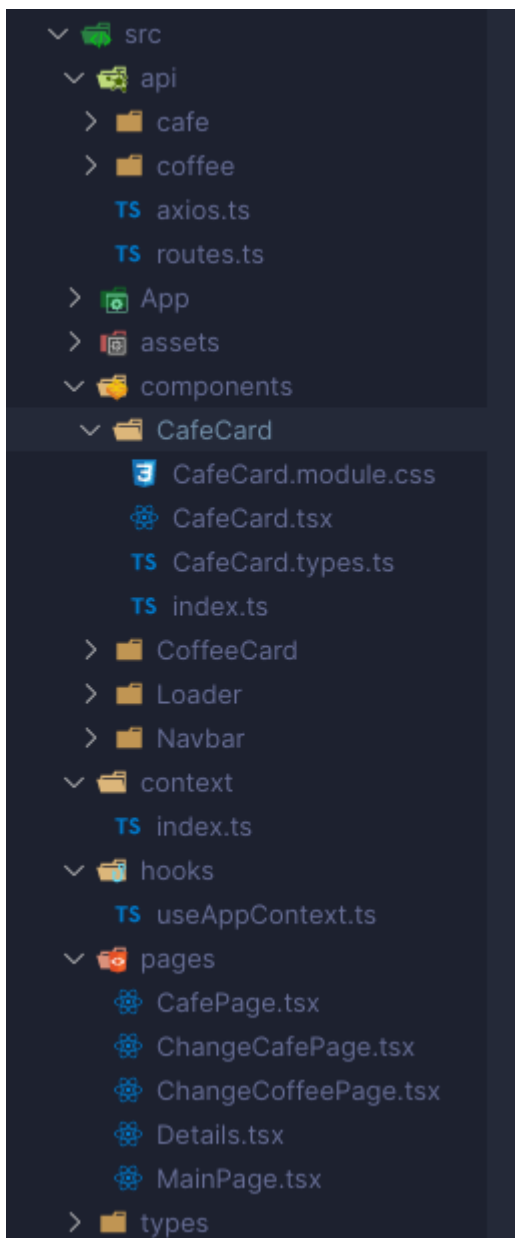
Москва, МГТУ – 2021

Задание:

На основе методических указаний разработайте React-приложение. Для создания приложения необходимо решить следующие задачи:

1. Создать стартовый React-проект. Удалить неиспользуемый код. Организовать директории для страниц, компонентов, утилит и работы с сетью.
2. Организовать роутинг в веб-приложении.
3. Разработать базовые страницы, на которых будут отображаться сущности из выбранной вами предметной области:
 - a. Стартовая страница.
 - b. Страница просмотра списка объектов.
 - c. Страница просмотра конкретного объекта.
4. Вынести переиспользуемые компоненты в отдельные файлы:
 - a. Для навигации по приложению можно добавить header.
 - b. Для отображения дополнительной информации (данные о студенте и предметной области) можно использовать footer.
 - c. Источники ввода-вывода (поля ввода (inputs)/формы/текстовые блоки).
 - d. Переиспользуемые таблицы/гриды.
5. Добавить асинхронные запросы в разработанный API, чтобы страница получала данные с сервера.
6. Если в Вашем проекте реализована сложная логика работы с состоянием приложения, то рекомендуется добавить пользовательские хуки.
7. Страницы приложения должны хорошо отображаться как на больших, так и на маленьких экранах.

Структура папок в src



Текст программы:

src/api/axios.ts

```
PROВOДНИК
IU5WEB-FALL-2021-TA...
lab4
  client
    node_modules
    public
    src
      api
        cafe
          TS fetchAllCafe.ts
          TS fetchCreateCafe.ts
          TS fetchDeleteCafe.ts
          TS fetchUpdateCafe.ts
        coffee
          TS fetchAllCoffee.ts
          TS fetchCreateCoffee.ts
          TS fetchDeleteCoffee.ts
          TS fetchUpdateCoffee.ts
          TS axios.ts
          TS routes.ts

TS axios.ts
lab4 > client > src > api > TS axios.ts > ...
You, 3 weeks ago | 1 author (You)
1 import axios from 'axios';
2
3 import { BACKEND_URL } from 'config';
4
5 const instance = axios.create({
6   baseURL: BACKEND_URL,
7   withCredentials: true,
8 });
9
10 export default instance;
11
```

routes.ts

```
TS routes.ts
lab4 > client > src > api > TS routes.ts > ...
You, 3 weeks ago | 1 author (You)
1 const COFFEE_PREFIX = '/coffee';
2 const CAFE_PREFIX = '/cafe';
3
4 export const COFFEE_ROUTES = {
5   create: (prefix = COFFEE_PREFIX) => `${prefix}/`,
6   update: (id: number, prefix = COFFEE_PREFIX) => `${prefix}/${id}/`,
7   delete: (id: number, prefix = COFFEE_PREFIX) => `${prefix}/${id}/`,
8   getAll: (prefix = COFFEE_PREFIX) => `${prefix}/`,
9 };
10
11 export const CAFE_ROUTES = {
12   create: (prefix = CAFE_PREFIX) => `${prefix}/`,
13   update: (id: number, prefix = CAFE_PREFIX) => `${prefix}/${id}/`,
14   delete: (id: number, prefix = CAFE_PREFIX) => `${prefix}/${id}/`,
15   getAll: (prefix = CAFE_PREFIX) => `${prefix}/`,
16 };
17
```

```
TS fetchAllCafe.ts
lab4 > client > src > api > cafe > TS fetchAllCafe.ts > [e] fetchAllCafe
You, 3 weeks ago | 1 author (You)
1 import axios from 'api/axios';
2 import { CAFE_ROUTES } from 'api/routes';
3 import { Cafe } from 'types/cafe.types';
4
5 export const fetchAllCafe = async (): Promise<Cafe[]> => {
6   return await axios.get<Cafe[]>({ url: `${CAFE_ROUTES.getAll()}`).then( onfulfilled: (response) => response.data);
7 };
8
```

```
TS fetchCreateCafe.ts x
lab4 > client > src > api > cafe > TS fetchCreateCafe.ts > fetchCreateCafe
You, 2 weeks ago | 1 author (You)
1 import axios from 'api/axios';
2 import { CAFE_ROUTES } from 'api/routes';
3 import { AxiosResponse } from 'axios';
4 import { Cafe } from 'types/cafe.types';
5
6 export const fetchCreateCafe = async (cafe: Cafe): Promise<Cafe> => { You, 2 weeks ago • final fixes
7   return await axios
8     .post<Cafe, AxiosResponse<Cafe>>(`url: `${CAFE_ROUTES.create()}`, data: cafe)
9     .then( onfulfilled: (response) => response.data);
10 };
11
```

```
TS fetchDeleteCafe.ts x
lab4 > client > src > api > cafe > TS fetchDeleteCafe.ts > fetchDeleteCafe
You, 3 weeks ago | 1 author (You)
1 import axios from 'api/axios';
2 import { CAFE_ROUTES } from 'api/routes';
3 import { AxiosResponse } from 'axios';
4 import { Cafe } from 'types/cafe.types';
5
6 export const fetchDeleteCafe = async (cafe: Cafe): Promise<Cafe> => { You, 3 weeks ago • frontend created
7   return await axios
8     .delete<Cafe, AxiosResponse<Cafe>>(`url: `${CAFE_ROUTES.update(cafe.pk)}`)
9     .then( onfulfilled: (response) => response?.data);
10 };
11
```

```
TS fetchUpdateCafe.ts x
lab4 > client > src > api > cafe > TS fetchUpdateCafe.ts > fetchUpdateCafe
You, 3 weeks ago | 1 author (You)
1 import axios from 'api/axios';
2 import { CAFE_ROUTES } from 'api/routes';
3 import { AxiosResponse } from 'axios';
4 import { Cafe } from 'types/cafe.types';
5
6 export const fetchUpdateCafe = async (cafe: Cafe): Promise<Cafe> => {
7   return await axios
8     .put<Cafe, AxiosResponse<Cafe>>(`url: `${CAFE_ROUTES.update(cafe.pk)}`, data: cafe)
9     .then( onfulfilled: (response) => response.data);
10 };
11
```

Для кофе аналогично

App.tsx

```
import React, { useEffect, useState } from 'react';
import { Route, Switch } from 'react-router';
import { BrowserRouter } from 'react-router-dom';
import { fetchAllCafe } from 'api/cafe/fetchAllCafe';
import { fetchAllCoffee } from 'api/coffee/fetchAllCoffee';
import { AppContext } from 'context';
import { CafePage } from 'pages/CafePage';
import { ChangeCafePage } from 'pages/ChangeCafePage';
import { ChangeCoffeePage } from 'pages/ChangeCoffeePage';
import { Details } from 'pages/Details';
import { MainPage } from 'pages/MainPage';
import { Cafe } from 'types/cafe.types';
```

```

import { Coffee } from 'types/coffee.types';

import { Loader } from 'components/Loader';
import { Navbar } from 'components/Navbar';

import './App.css';

export const App = () => {
  const [coffee, setCoffee] = useState<Coffee[]>([]);
  const [coffeeeLoaded, setCoffeeeLoaded] = useState(false);
  const [cafe, setCafe] = useState<Cafe[]>([]);
  const [cafeLoaded, setCafeLoaded] = useState(false);
  const [loaded, setLoaded] = useState(false);

  useEffect(() => {
    if (!loaded) {
      fetchAllCoffee()
        .then((coffee) => {
          setCoffee(coffee);
          setCoffeeeLoaded(true);
        })
        .catch((e) => console.error(e));

      fetchAllCafe()
        .then((cafe) => {
          setCafe(cafe);
          setTimeout(() => setCafeLoaded(true), 0);
        })
        .catch((e) => console.error(e));

      setLoaded(coffeeeLoaded && cafeLoaded);
    }
  }, [cafeLoaded, coffeeeLoaded, loaded]);

  return (
    <AppContext.Provider value={{ coffee, loaded, cafe, setLoaded
}}>

      <BrowserRouter>

        <Navbar />

```

```

        {coffeeLoaded && cafeLoaded ? (
            <Switch>
                <Route exact path="/">
                    <MainPage />
                </Route>
                <Route exact path="/create-coffee">
                    <ChangeCoffeePage />
                </Route>
                <Route exact path="/update-coffee/:id">
                    <ChangeCoffeePage />
                </Route>
                <Route exact path="/cafe">
                    <CafePage />
                </Route>
                <Route exact path={['/update-cafe/:id',
'/create-cafe']}>
                    <ChangeCafePage />
                </Route>
                <Route path="/:id">
                    <Details />
                </Route>
            </Switch>
        ) : (
            <Loader />
        )}
    </BrowserRouter>
</AppContext.Provider>
);
};

```

Данные доступны во всех частях приложения благодаря
ИСПОЛЬЗОВАНИЮ КОНТЕКСТА

```

import React, { createContext, SetStateAction } from 'react';
import { Cafe } from 'types/cafe.types';
import { Coffee } from 'types/coffee.types';

export type AppContext = {

```

```
    coffee: Coffee[];
    cafe: Cafe[];
    loaded: boolean;
    setLoaded: React.Dispatch<SetStateAction<boolean>>;
  };

export const AppContext = createContext<AppContext>({
  coffee: [],
  loaded: false,
  cafe: [],
  setLoaded: () => undefined,
});
```

TS useAppContext.ts ×

lab4 > client > src > hooks > TS useAppContext.ts > ...

You, 3 weeks ago | 1 author (You)

```
1 import { useContext } from 'react';
2 import { AppContext } from 'context';
3
4 export const useAppContext = () => useContext( context: AppContext);
5
```

Карточки


```

CafeCard.tsx x
lab4 > client > src > components > CafeCard > CafeCard.tsx > CafeCard
You, 2 weeks ago | 1 author (You)
1 import React, { useCallback } from 'react';
2 import { useHistory } from 'react-router';
3 import { fetchDeleteCafe } from 'api/cafe/fetchDeleteCafe';
4 import cnBind, { Argument } from 'classnames/bind';
5
6 import { CafeCardProps } from './CafeCard.types';
7
8 import styles from './CafeCard.module.css';
9
10 const cx = cnBind.bind( this: styles ) as (...args: Argument[]) => string;
11
12 export const CafeCard = ( { cafe }: CafeCardProps ): JSX.Element => {
13   const history = useHistory();
14
15   const handleDelete = useCallback( () => void>(callback: () => void: () => {
16     void fetchDeleteCafe( cafe: cafe );
17     history.push( path: '/' );
18   }, deps: [cafe, history]);
19
20   const handleEdit = useCallback( () => void>(callback: () => void: () => {
21     history.push( path: `/update-cafe/${cafe.pk}` );
22   }, deps: [cafe.pk, history]);
23
24   return (
25     <div className={cx('container')}>
26       <div className={cx('info')}>
27         <h1>{cafe.name}</h1>
28         <p>{cafe.address}</p>
29         <button type="button" onClick={handleDelete} className={cx('button')}>
30           Delete
31         </button>
32         <button type="button" onClick={handleEdit} className={cx('button')}> You, 2 weeks ago • final fix
33           Edit
34         </button>
35       </div>
36     </div>
37   );
38 };
39

```

```

CoffeeCard.tsx x
lab4 > client > src > components > CoffeeCard > CoffeeCard.tsx > CoffeeCard
16 const { cafe } = useAppContext();
17
18 const itemCafeName = useMemo( factory: () => cafe.find( predicate: (item) => item.pk === coffee.cafe)?.name, deps: [
19
20 const onClick = useCallback( () => void>(callback: () => void: () => history.push( path: `/${coffee.pk}` ), deps: [coffee
21
22 const rating = useMemo( factory: () => '♥♥♥♥♥♥♥♥'.slice( start: 0, end: coffee.score ), deps: [coffee.score]);
23
24 const handleDelete = useCallback( () => void>(callback: () => void: () => {
25   void fetchDeleteCoffee( coffee: coffee );
26   history.push( path: '/' );
27 }, deps: [coffee, history]);
28
29 const handleEdit = useCallback( () => void>(callback: () => void: () => {
30   history.push( path: `/update-coffee/${coffee.pk}` );
31 }, deps: [coffee.pk, history]);
32
33 return (
34   <div className={cx('container', { large })}>
35     <img className={cx('image')} src={coffee.main_image || CoffeeDefault} alt={coffee.name} onClick={onClick}
36     <div className={cx('info')}>
37       <h1>{coffee.name}</h1>
38       <p>{itemCafeName}</p>
39       <p>{coffee.country}</p>
40       <p className={cx('description')}>{coffee.description}</p>
41       <p>{coffee.price} RUB</p>
42       <p className={cx('score')}>{rating}</p>
43       {large && (
44         <div>
45           <button type="button" onClick={handleDelete} className={cx('button')}>
46             Delete
47           </button>
48           <button type="button" onClick={handleEdit} className={cx('button')}> You, 3 weeks ago • f
49             Edit
50           </button>
51         </div>
52       )}
53     </div>
54   </div>
55 );
56 };
57

```

Лoader

```
Loader.tsx X
lab4 > client > src > components > Loader > Loader.tsx > ...
You, 3 weeks ago | 1 author (You)
1 import React from 'react';
2 import { HeartIcon } from 'assets';
3 import cnBind, { Argument } from 'classnames/bind';
4
5 import styles from './Loader.module.css';
6
7 const cx = cnBind.bind( this: styles) as (...args: Argument[]) => string;
8
9 export const Loader = () => {
10   return (
11     <div className={cx('spinner')}>
12       <HeartIcon />
13       <HeartIcon />
14       <HeartIcon />
15     </div>
16   );
17 };
18
```

Навбар

```
Navbar.tsx X
lab4 > client > src > components > Navbar > Navbar.tsx > [e] Navbar
You, 2 weeks ago | 1 author (You)
1 import React from 'react';
2 import { Link } from 'react-router-dom';
3
4 export const Navbar: React.FC = () => (
5   <div className='navbar'>
6     <Link to={'/'}>Home</Link>
7     <Link to={'/cafe'}>Cafe List</Link>
8     <Link to="/create-coffee">Create coffee</Link>
9     <Link to="/create-cafe">Create cafe</Link> You, 2 weeks ago
10   </div>
11 );
12
```

Результат работы программы:



INDONESIA FRINSA MANIS

Brrrew!

Indonesia

Лот Маниса - по вкусу напоминает черри колу, красное яблоко, шоколадный стаут, а так же ноты лесных ягод.

800 RUB



COLOMBIA EL PARAISO Y_Y

Brrrew!

Colombia

Сочный фруктовый микс из личи, абрикосового джема, персиков и вишни, молочного улуна, сбалансированная кислотность, сладкое, продолжительное послевкусие, шелковистое тело.

950 RUB



INDONESIA FRINSA MANIS

Brrrew!

Ethiopia

Лот Маниса - по вкусу напоминает черри колу, красное яблоко, шоколадный стаут, а так же ноты лесных ягод.

800 RUB



Name

INDONESIA FRINSA MANIS

Price

800

Description

Сочный фруктовый микс из личи, абрикосового джема, персиков и вишни, молочного улуна, сбалансированная кислотность, сладкое, продолжительное послевкусие, шелковистое тело.

Score

10

Country

Indonesia

Cafe

bloom-n-brew

bloom-n-brew

Brrrew!

```
{
  "cafe": 2,
  "name": "INDONESIA FRINSA MANIS",
  "price": "800",
  "description": "Сочный фруктовый микс из личи, абрикосового джема, персиков и вишни, молочного улуна, сбалансированная кислотность, сладкое, продолжительное послевкусие, шелковистое тело.",
  "score": "10",
  "country": "Indonesia"
}
```