

[Главная](#) /

Походы викингов

Эпоха викингов была периодом с 8 по 11 век, в течение которого они приобрели непревзойденные навыки в области кораблестроения, навигации и мореплавания, что позволило викингам путешествовать по всей Северной и Западной Европе, Северной Атлантике, в Средиземное море, вплоть до рек России и Украины.

[Города](#) /[Поиск](#)

Ставангер (Норвегия)

Ставангер был важным торговым и кораблестроительным центром на юго-западном побережье Норвегии. Гор.....

[подробнее](#)

Альта (Норвегия)

Альта, расположенная в северной части Норвегии, известна своими наскальными рисунками и служила баз.....

[подробнее](#)

Бирка (Швеция)

Бирка был важным торговым центром викингов, расположенным на острове в озере Меларен. Город играл к.....

[подробнее](#)

Гардарики (Россия)

Гардарики, расположенный на территории современной России, был важным торговым центром на пути вики.....

[подробнее](#)[Города](#) / [Альта \(Норвегия\)](#) /

Альта (Норвегия)

Альта, расположенная в северной части Норвегии, известна своими наскальными рисунками и служила базой для экспедиций в Арктические регионы.

Площадь: 500km².

Без бэкенд (бэкенд недоступен) (django container off)

Города /

Поиск...

Поиск



Ставангер (Норвегия)

Ставангер был важным торговым и кораблестроительным центром на юго-западном побережье Норвегии. Гор.....

[подробнее](#)



Альта (Норвегия)

Альта, расположенная в северной части Норвегии, известна своими наскальными рисунками и служила баз.....

[подробнее](#)



Бирка (Швеция)

Бирка был важным торговым центром викингов, расположенным на острове в озере Меларен. Город играл к.....

[подробнее](#)



Гардарики (Россия)

Гардарики, расположенный на территории современной России, был важным торговым центром на пути вики.....

[подробнее](#)



Лунд (Швеция)

Лунд был одним из первых викингских городов и важным центром вероисповедания. Город известен своими.....

[подробнее](#)



Рейкьявик (Исландия)

Рейкьявик был основан викингами и со временем стал столицей Исландии. Город стал важным центром для.....

[подробнее](#)

```

12 function App() {
13
14
15     const [selectedPlace, setSelectedPlace] = useState<T_Place | null>(null)
16
17     const [isMock, setIsMock] = useState(false);
18
19     const [placeName, setPlaceName] = useState<string>("")
20
21     return (
22
23         <div>
24             <Header/>
25             <Container className="pt-4">
26                 <Row className="mb-3">
27                     <Breadcrumbs selectedPlace={selectedPlace} />
28                 </Row>
29                 <Row>
30                     <Routes>
31                         <Route path="/" element={<HomePage />} />
32                         <Route path="/places/" element={<PlacesListPage places={places} setPlaces={setPlaces} isMock={isMock} setIsMock={setIsMock} placeName={placeName} setPlaceName={setPlaceName} />} />
33                         <Route path="/places/:id" element={<PlacePage selectedPlace={selectedPlace} setSelectedPlace={setSelectedPlace} isMock={isMock} setIsMock={setIsMock} />} />
34                     </Routes>
35                 </Row>
36             </Container>
37         </div>
38     )
39 }
40

```

In app.tsx component, using (useState hook) getting list of services which is passed through props to the (index.tsx >place_list component) of services list page())()

//////////

Placeslistpage/index.tsx component

- Stores the filtering parameter in its state
- Gets a list of services from above via props and renders them
- Renders a search field for filtering services

```

1 import {Button, Col, Container, Form, Input, Row} from "reactstrap";
2 import {T_Place} from "src/modules/types.ts";
3 import PlaceCard from "components/PlaceCard";
4 import {PlaceMocks} from "src/modules/mocks.ts";
5 import {FormEvent, useEffect} from "react";
6 import * as React from "react";
7
8 type PlacesListPageProps = {
9   places: T_Place[];
10   setPlaces: React.Dispatch<React.SetStateAction<T_Place[]>>;
11   isMock: boolean;
12   setIsMock: React.Dispatch<React.SetStateAction<boolean>>;
13   placeName: string;
14   setPlaceName: React.Dispatch<React.SetStateAction<string>>;
15 };
16
17 const PlacesListPage = ({places, setPlaces, isMock, setIsMock, placeName, setPlaceName}: PlacesListPageProps) => {
18
19   const fetchData = async () => {
20     try {
21       const response = await fetch(`/api/places/?place_name=${placeName.toLowerCase}`);
22       const data = await response.json();
23       setPlaces(data.places);
24       setIsMock(false);
25     } catch {
26       createMocks();
27     }
28   }
29
30   const createMocks = () => {
31     setIsMock(true);
32     setPlaces(PlaceMocks.filter(place => place.name.toLowerCase().includes(placeName)));
33   }
34
35   const handleSubmit = async (e:FormEvent) => {
36     e.preventDefault();
37     if (isMock) {
38       createMocks();
39     } else {
40       await fetchData();
41     }
42   }
43
44   useEffect(() => {
45     fetchData();
46   }, []);
47
48   return (
49     <Container>
50       <Row className="mb-5">
51         <Col md="6">
52           <Form onSubmit={handleSubmit}>
53             <Row>
54               <Col md="8">
55                 <Input value={placeName} onChange={(e) => setPlaceName(e.target.value)} />
56               </Col>
57               <Col>
58                 <Button color="primary" className="w-100">Поиск</Button>
59               </Col>
60             </Row>
61           </Form>
62         </Col>
63         <Col>
64           <Row>
65             <Col>
66               {places?.map(place => (
67                 <PlaceCard place={place} isMock={isMock} />
68               ))}
69             </Col>
70           </Row>
71         </Col>
72       </Row>
73     </Container>
74   );
75   export default PlacesListPage

```

by using (useEffect) when opens page that call function search which that sends query GET TO BACKEND FOR GETTING MARRIED) I THINK()

```

42   }
43
44   useEffect(() => {
45     |   fetchData()
46   }, []);
47
48   return (
49     <Container>
50       <Row className="mb-5">
51         <Col md="6">
52           <Form onSubmit={handleSubmit}>
53             <Row>
54               <Col md="8">
55                 <Input value={placeName} onChange={(e) => setPlaceName(e.target.value)} placeholder="Введите название места" />
56               </Col>
57               <Col>
58                 <Button color="primary" className="w-100">Поиск</Button>
59               </Col>
60             </Row>
61           </Form>
62         </Col>
63       </Row>
64       <Row>
65         {places?.map(place => (
66           <Col key={place.id} xs="4">
67             <PlaceCard place={place} isMock={isMock} />
68           </Col>
69         ))}
70       </Row>
71     </Container>
72   );
73 };
74
75 export default PlacesListPage

```

С помощью хука `useEffect` после маунтинга компонента `placelistpage` вызывается функция `fetchData`, которая отправляет GET запрос на бэкенд для получения списка услуг, передавая в него, в качестве `query` параметра, значение параметра фильтрации. Вызывается калбэк `setplace` который обновляет значения стейта списка услуг в компоненте `App.tsx` => меняется значение пропсов в компоненте `placelistpage` => компонент `placelistpage` перерендеривается => обновляется список усл. Также функцию `fetchData` можно вызвать, нажав на кнопку поиска.

```
11 function App() {
12
13
14   const [places, setPlaces] = useState<T_Place[]>([])
15
16   const [selectedPlace, setSelectedPlace] = useState<T_Place | null>(null)
17
18   const [isMock, setIsMock] = useState(false);
19
20   const [placeName, setPlaceName] = useState<string>("")
21
22   return (
23     <div>
24       <Header/>
25       <Container className="pt-4">
26         <Row className="mb-3">
27           <Breadcrumb selectedPlace={selectedPlace} />
28         </Row>
29         <Row>
30           <Routes>
31             <Route path="/" element={<HomePage />} />
32             <Route path="/places/" element={<PlacesListPage places={places} setPlaces={setPlaces} isMock={isMock} setIsMock={setIsMock} placeName={placeName} setPlaceName={setPlaceName} />} />
33             <Route path="/places/:id" element={<PlacePage selectedPlace={selectedPlace} setSelectedPlace={setSelectedPlace} isMock={isMock} setIsMock={setIsMock} />} />
34           </Routes>
35         </Row>
36       </Container>
37     </div>
38   )
39 }
```

PROBLEMS (16) OUTPUT DEBUG CONSOLE TERMINAL PORTS

(env) ahmedkashima@ahmedkashima-Latitude-E5470:~/Desktop/WEBF_IU5_BNSTU/VikingsServers

моки;

Если нет доступа к бэкэнду, моки выводятся (src/modules/mocks.ts)

В состоянии компонента App.tsx хранится логический флаг — выводить моки или нет, который передается через props компонентам страницы списка служб и одной страницы служб

After sending a GET request to get a list of services, the frontend waits for one hour and if no response is received from the backend, it calls the createMocks function, which calls the callbacks setIsMock(true) and place(mock)

After that, the placelistpage component renders the mocks

```
34 const handleSubmit = async (e:FormEvent) => {
35   e.preventDefault()
36   if (isMock) {
37     createMocks()
38   } else {
39     await fetchData()
40   }
41 }
42
43
44 useEffect(() => {
45   fetchData()
46 }, []);
47
48 return ()
49
50 <Container>
51   <Row className="mb-5">
52     <Col md="6">
53       <Form onSubmit=handleSubmit>
54         <Row>
55           <Col md="8">
56             <input value={placeName} onChange={e => setPlaceName(e)} />
57           </Col>
58           <Col>
59             <Button color="primary" className="w-100">Поиск</Button>
60           </Col>
61         </Row>
62       </Form>
63     </Row>
64     <Row>
65       {places?.map(place => (
66         <Col key={place.id} xs="4">
67           <PlaceCard place={place} isMock={isMock} />
68         </Col>
69       ))}
70     </Row>
71   </Container>
72
73 );
74
75 export default PlacesListPage
```

```
2 export const PlaceMocks:T_Place[] = [
3
4   {
5     id: 1,
6     name: "Ставангер (Норвегия)",
7     status: 1,
8     image: "",
9     description: "Ставангер был важным торговым и кораблестроительным центром",
10    square: 71
11  },
12  {
13    id: 2,
14    name: "Альта (Норвегия)",
15    status: 1,
16    image: "",
17    description: "Альта, расположенная в северной части Норвегии, известна св",
18    square: 500
19  },
20  {
21    id: 3,
22    name: "Бирка (Швеция)",
23    status: 1,
24    image: "",
25    description: "Бирка был важным торговым центром викингов, расположенным н",
26    square: 300
27  },
28  {
29    id: 4,
30    name: "Гардарики (Россия)",
31    status: 1,
32    image: "",
33    description: "Гардарики, расположенный на территории современной России,",
34    square: 1200
35  }
36 ]
```

