

# Тестовое задание на позицию React Developer в компанию Core Line.

## Введение

Необходимо реализовать мобильную версию веб-приложения новостного сайта. Макет разрабатываемой страницы нарисован в Figma. Backend часть реализована на стороне REST API New York Times.

## Материалы

<https://www.figma.com/file/sc2xxWzzgeeFgW7MgMjpYT/Besider---React-тестовое?type=design&node-id=0-1&mode=design&t=U2Y6w3zdzS5bahWf-0>

[developer.nytimes.com](https://developer.nytimes.com)

<https://developer.nytimes.com/docs/archive-product/1/routes/{year}/{month}.json/get>

## Условия выполнения

Используйте React + Redux

Используйте Typescript

Верстка страницы должна быть рассчитана на мобильные устройства

Пользователь должен иметь возможность листать список новостей до любой доступной даты

Нет ограничений на использование сторонних библиотек

## Функциональные требования

В шапке страницы левая кнопка должна открывать меню

Новости расположены по убыванию даты начиная с текущей

Блоки с новостями за различные дни должны быть разделены датами

Синие круги внизу макета - иконка загрузка

Раз в 30 секунд должна происходить подгрузка только что вышедших новостей (которые еще не показаны в списке)

При клике на новость, пользователя должно переводить на сайт New York Times

## Backend

Апи-ключ: `rJ7XaUF0IQZG7UYu0jp85Mdqpeu5MnbP` (Или можете создать свой в личном кабинете)

Необходимый метод:

`https://api.nytimes.com/svc/archive/v1/{year}/{month}.json`

Заголовок статьи: `abstract`

Ссылка на статью: `web_url`

Фото: `multimedia`

Дата публикации: `pub_date`

Источник: `source`

## CORS (Для справки)

✓ Если ты используешь Create React App (CRA)

В корне проекта добавь файл:

```
package.json
{
  // ...
  "proxy": "https://example-backend.com"
}
```

Теперь в React ты можешь делать:

```
fetch('/api/some-endpoint')
```

А CRA будет проксировать это на  
`https://example-backend.com/api/some-endpoint`.

✓ Если ты используешь Vite

В `vite.config.js`:

```
export default {
  server: {
    proxy: {
      '/api': {
        target: 'https://example-backend.com',
        changeOrigin: true,
        rewrite: path => path.replace(/^\/api/, '')
      }
    }
  }
};
```

Теперь запросы на `/api/xyz` идут через прокси.

 Альтернатива: временное использование CORS-прокси

Просто:

```
fetch('https://cors-anywhere.herokuapp.com/https://example-backend.com/api/data')
```