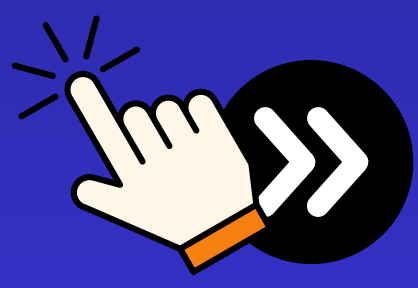
Sabarimani Ramamoorthi

Full Stack Developer

Mastering Axios in React

Unlock Effortless API Integration

Swipe for more



Sabarimani Ramamoorthi

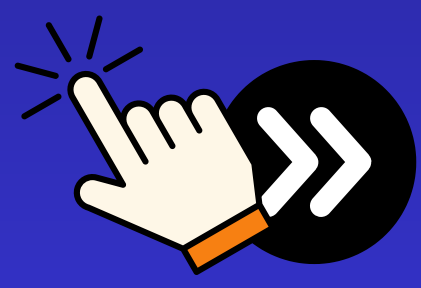
Full Stack Developer

Introduction to Axios for React Content

```
import axios from 'axios';

axios.get('https://api.example.com/items')
  .then(response => {
    console.log(response.data);
  })
  .catch(error => {
    console.error('Error fetching data:', error);
  });
```

Axios is a popular JavaScript library used to make HTTP requests. It provides a simple and easy-to-use API for interacting with APIs.

Swipe for more



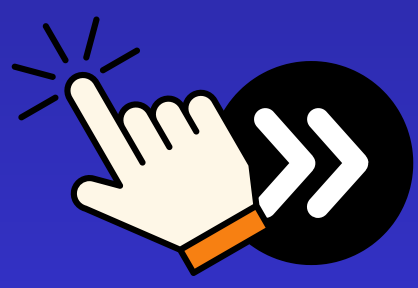
Sabarimani Ramamoorthi

Full Stack Developer

Making GET Requests

```
import axios from 'axios';

axios.get('https://api.example.com/items')
  .then(response => {
    console.log(response.data);
  })
  .catch(error => {
    console.error('Error fetching data:', error);
  });
```

Swipe for more



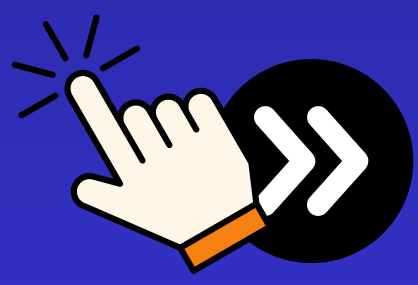
Sabarimani Ramamoorthi

Full Stack Developer

Making GET Requests with ID

```
import axios from 'axios';

const id = 1;
axios.get(`https://api.example.com/items/${id}`)
  .then(response => {
    console.log(response.data);
  })
  .catch(error => {
    console.error('Error fetching item:', error);
  });
```

Swipe for more



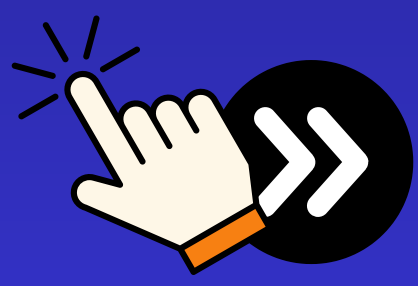
Sabarimani Ramamoorthi

Full Stack Developer

Making POST Requests

```
import axios from 'axios';

axios.post('https://api.example.com/items', {
  name: 'New Item',
  price: 100
})
  .then(response => {
    console.log('Item created:', response.data);
  })
  .catch(error => {
    console.error('Error creating item:', error);
  });
```

Swipe for more



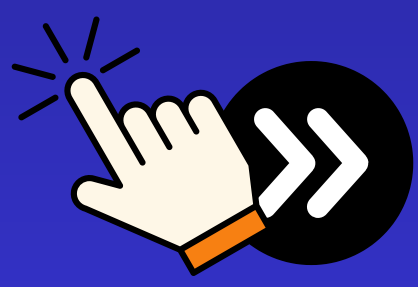
Sabarimani Ramamoorthi

Full Stack Developer

Making PUT Requests

```
import axios from 'axios';

const id = 1;
axios.put(`https://api.example.com/items/${id}`, {
  name: 'Updated Item',
  price: 120
})
  .then(response => {
    console.log('Item updated:', response.data);
  })
  .catch(error => {
    console.error('Error updating item:', error);
});
```

Swipe for more



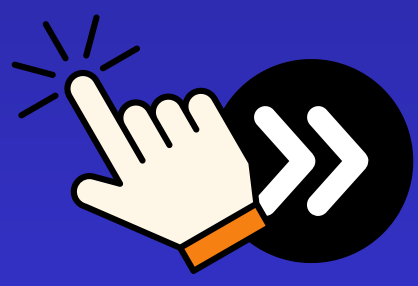
Sabarimani Ramamoorthi

Full Stack Developer

Making DELETE Requests

```
import axios from 'axios';

const id = 1;
axios.delete(`https://api.example.com/items/${id}`)
  .then(response => {
    console.log('Item deleted:', response.data);
  })
  .catch(error => {
    console.error('Error deleting item:', error);
  });
```

Swipe for more



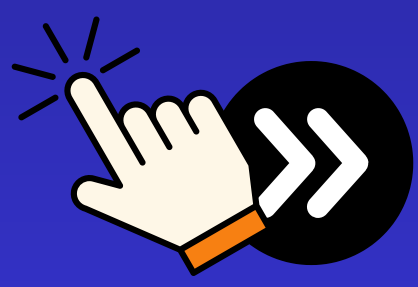
Sabarimani Ramamoorthi

Full Stack Developer

Making POST with JSON Body and URL Encoding

```
import axios from "axios";

axios
  .post(
    "https://api.example.com/auth",
    {
      username: "user",
      password: "pass",
    },
    {
      headers: {
        "Content-Type": "application/json",
      },
    }
  )
  .then((response) => {
    const token = response.data.token;
    return axios.get("https://api.example.com/data", {
      headers: {
        Authorization: `Bearer ${token}`,
      },
    });
  })
  .then((response) => {
    console.log("Data fetched with token:", response.data);
  })
  .catch((error) => {
    console.error("Error during request:", error);
  });
```

Swipe for more



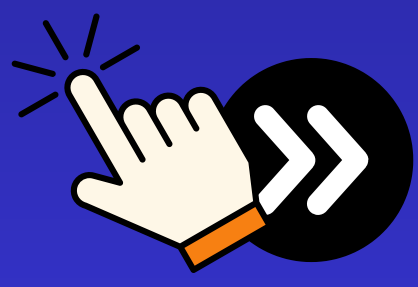
Sabarimani Ramamoorthi

Full Stack Developer

Advanced Error Handling in Axios

```
import axios from "axios";

axios
  .get("https://api.example.com/data")
  .then((response) => {
    console.log(response.data);
  })
  .catch((error) => {
    if (error.response) {
      // Server responded with a status other than 200 range
      console.error("Error response:", error.response.data);
    } else if (error.request) {
      // No response received
      console.error("No response received:", error.request);
    } else {
      // Error setting up the request
      console.error("Error setting up request:", error.message);
    }
  });
```

Swipe for more



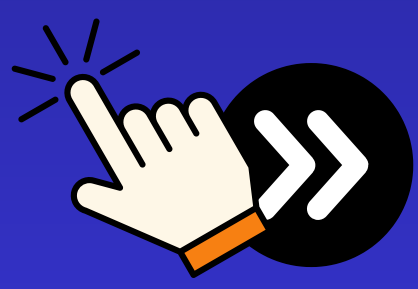
Sabarimani Ramamoorthi

Full Stack Developer

Implementing Retry Mechanism in Axios

```
import axios from "axios";

axios
  .get("https://api.example.com/data")
  .then((response) => {
    console.log(response.data);
  })
  .catch((error) => {
    if (error.response) {
      // Server responded with a status other than 200 range
      console.error("Error response:", error.response.data);
    } else if (error.request) {
      // No response received
      console.error("No response received:", error.request);
    } else {
      // Error setting up the request
      console.error("Error setting up request:", error.message);
    }
  });
```

Swipe for more



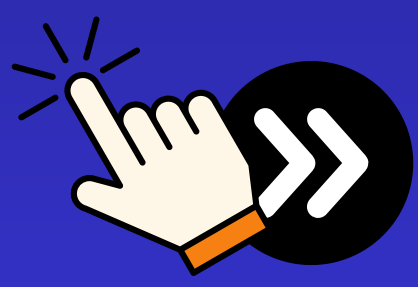
Sabarimani Ramamoorthi

Full Stack Developer

Custom Hook for Axios Requests

```jsx
import { useState, useEffect } from 'react';
import axios from 'axios';

const useAxios = (url) => {
  const [data, setData] = useState(null);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);

  useEffect(() => {
    axios.get(url)
      .then(response => {
        setData(response.data);
        setLoading(false);
      })
      .catch(error => {
        setError(error);
        setLoading(false);
      });
  }, [url]);

  return { data, loading, error };
};

export default
```

Swipe for more



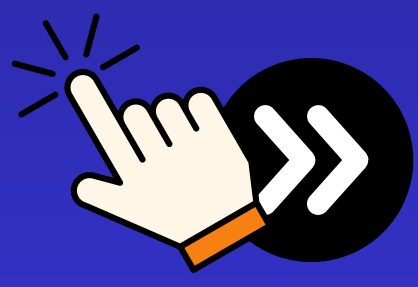
Sabarimani Ramamoorthi

Full Stack Developer

Handling Query Parameters with Axios

```
import axios from 'axios';

axios.get('https://api.example.com/items', {
  params: {
    category: 'electronics',
    sortBy: 'price'
  }
})
  .then(response => {
    console.log(response.data);
  })
  .catch(error => {
    console.error('Error fetching items:', error);
  });
```

Swipe for more



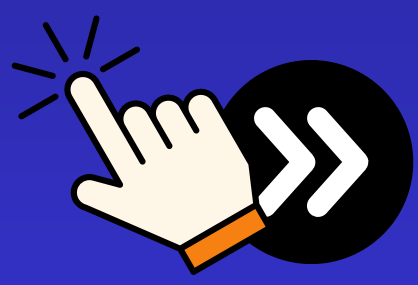
Sabarimani Ramamoorthi

Full Stack Developer

Axios Interceptors for Request Modification

```
import axios from "axios";

const axiosInstance = axios.create();

axiosInstance.interceptors.request.use(
  (config) => {
    config.headers["Authorization"] = "Bearer your_access_token";
    return config;
  },
  (error) => {
    return Promise.reject(error);
  }
);

axiosInstance
  .get("https://api.example.com/data")
  .then((response) => {
    console.log(response.data);
  })
  .catch((error) => {
    console.error("Error fetching data:", error);
  });
```

Swipe for more



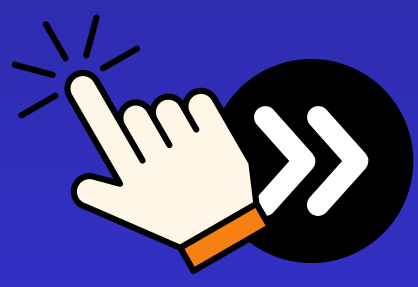
Sabarimani Ramamoorthi

Full Stack Developer

Handling Response Transformation

```
import axios from 'axios';

axios.get('https://api.example.com/data', {
  transformResponse: [function (data) {
    // Transform the response data here
    return JSON.parse(data).map(item => ({
      ...item,
      transformed: true
    }));
  }]
})
  .then(response => {
    console.log(response.data);
  })
  .catch(error => {
    console.error('Error transforming data:', error);
  });
```

Swipe for more



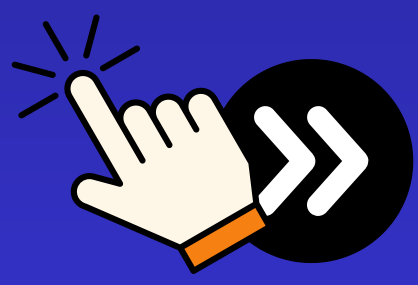
Sabarimani Ramamoorthi

Full Stack Developer

Caching Responses with Axios

```
import axios from 'axios';

const cache = {};

const fetchData = async (url) => {
  if (cache[url]) {
    return cache[url];
  }

  try {
    const response = await axios.get(url);
    cache[url] = response.data;
    return response.data;
  } catch (error) {
    console.error('Error fetching data:', error);
    throw error;
  }
};

fetchData('https://api.example.com/data')
  .then(data => {
    console.log(data);
  });
```

Swipe for more



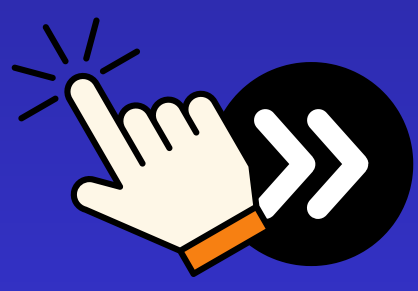
Sabarimani Ramamoorthi

Full Stack Developer

File Upload with Axios

```
import axios from 'axios';

const formData = new FormData();
formData.append('file', fileInput.files[0]);

axios.post('https://api.example.com/upload', formData, {
  headers: {
    'Content-Type': 'multipart/form-data'
  }
})
  .then(response => {
    console.log('File uploaded successfully:', response.data);
  })
  .catch(error => {
    console.error('Error uploading file:', error);
  });
```

Swipe for more



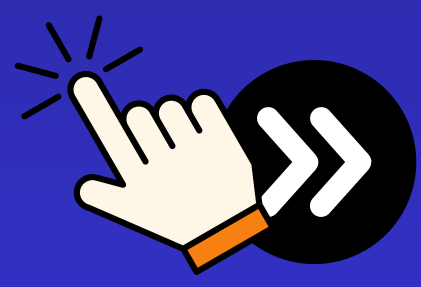
Sabarimani Ramamoorthi

Full Stack Developer

Handling Timeouts in Axios

```
import axios from 'axios';

axios.get('https://api.example.com/data', {
  timeout: 5000 // 5 seconds timeout
})
  .then(response => {
    console.log(response.data);
  })
  .catch(error => {
    if (error.code === 'ECONNABORTED') {
      console.error('Request timed out:', error.message);
    } else {
      console.error('Error fetching data:', error);
    }
  });
```

Swipe for more



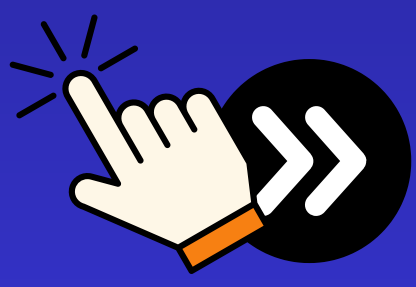
Sabarimani Ramamoorthi

Full Stack Developer

Cancelling Requests with Axios

```
import axios from 'axios';

const CancelToken = axios.CancelToken;
const source = CancelToken.source();

axios.get('https://api.example.com/data', {
  cancelToken: source.token
})
  .then(response => {
    console.log(response.data);
  })
  .catch(thrown => {
    if (axios.isCancel(thrown)) {
      console.log('Request canceled:', thrown.message);
    } else {
      console.error('Error fetching data:', thrown);
    }
  });

// Cancel the request
source.cancel('Request cancelled by the user');
```

Swipe for more



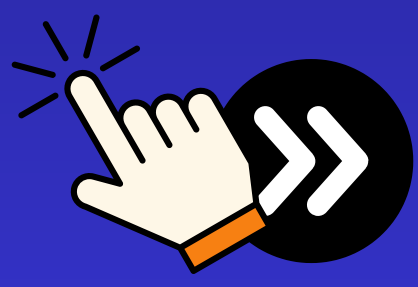
Sabarimani Ramamoorthi

Full Stack Developer

Upload Progress with Axios

```
import axios from 'axios';

const CancelToken = axios.CancelToken;
const source = CancelToken.source();

axios.get('https://api.example.com/data', {
  cancelToken: source.token
})
  .then(response => {
    console.log(response.data);
  })
  .catch(thrown => {
    if (axios.isCancel(thrown)) {
      console.log('Request canceled:', thrown.message);
    } else {
      console.error('Error fetching data:', thrown);
    }
  });

// Cancel the request
source.cancel('Request cancelled by the user');
```

Swipe for more



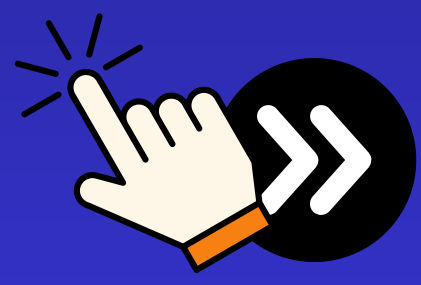
Sabarimani Ramamoorthi

Full Stack Developer

Working with Pagination in Axios

```
import axios from 'axios';

const formData = new FormData();
formData.append('file', fileInput.files[0]);

axios.post('https://api.example.com/upload', formData, {
  headers: {
    'Content-Type': 'multipart/form-data'
  },
  onUploadProgress: progressEvent => {
    const percentCompleted = Math.round((progressEvent.loaded * 100) / progressEvent.total);
    console.log(`Upload progress: ${percentCompleted}%`);
  }
})
  .then(response => {
    console.log('File uploaded successfully:', response.data);
  })
  .catch(error => {
    console.error('Error uploading file:', error);
  });
```

Swipe for more



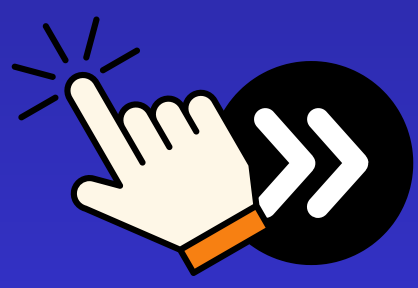
Sabarimani Ramamoorthi

Full Stack Developer

Upload Progress with Axios

```
import axios from 'axios';

const formData = new FormData();
formData.append('file', fileInput.files[0]);

axios.post('https://api.example.com/upload', formData, {
  headers: {
    'Content-Type': 'multipart/form-data'
  },
  onUploadProgress: progressEvent => {
    const percentCompleted = Math.round((progressEvent.loaded * 100) / progressEvent.total);
    console.log(`Upload progress: ${percentCompleted}%`);
  }
})
  .then(response => {
    console.log('File uploaded successfully:', response.data);
  })
  .catch(error => {
    console.error('Error uploading file:', error);
  });
```

Swipe for more



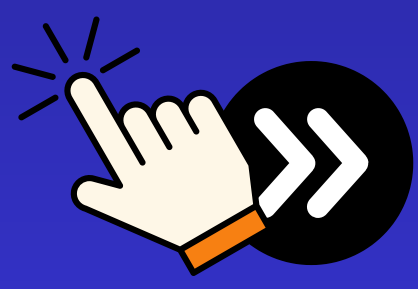
Sabarimani Ramamoorthi

Full Stack Developer

Working with Pagination in Axios

```jsx
import axios from 'axios';
import { useState, useEffect } from 'react';

const usePaginatedData = (url, page) => {
  const [data, setData] = useState([]);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);

  useEffect(() => {
    setLoading(true);
    axios.get(url, {
      params: { page }
    })
      .then(response => {
        setData(prevData => [...prevData, ...response.data.items]);
        setLoading(false);
      })
      .catch(error => {
        setError(error);
        setLoading(false);
      });
  }, [url, page]);

  return { data, loading, error };
};

export default usePaginatedData;
```

Swipe for more



Sabarimani Ramamoorthi

Full Stack Developer

If you find this helpful, please like and share it with your friends

@sabarimani.r