

React Native - HTTP

In this chapter, we will show you how to use **fetch** for handling network requests.

App.js

```
import React from 'react';
import HttpExample from './http_example.js'

const App = () => {
  return (
    <HttpExample />
  )
}
export default App
```

Using Fetch

We will use the **componentDidMount** lifecycle method to load the data from server as soon as the component is mounted. This function will send GET request to the server, return JSON data, log output to console and update our state.

http_example.js

```
import React, { Component } from 'react'
import { View, Text } from 'react-native'

class HttpExample extends Component {
  state = {
    data: ''
  }
  componentDidMount = () => {
    fetch('https://jsonplaceholder.typicode.com/posts/1', {
      method: 'GET'
    })
    .then((response) => response.json())
    .then((responseJson) => {
      console.log(responseJson);
      this.setState({
        data: responseJson
      })
    })
    .catch((error) => {
      console.error(error);
    });
  }
}
```

```
}  
render() {  
  return (  
    <View>  
      <Text>  
        {this.state.data.body}  
      </Text>  
    </View>  
  )  
}  
}  
export default HttpExample
```

Output

