

W05-P1: Hooks Tutorial T14_66

The image shows a development environment with VS Code on the left and a web browser on the right. In VS Code, the file `T14_66-useState-object.js` is open, showing a function `T14_66_UseStateObject` that uses `useState` to manage a person's state. The state is initialized with `name: 'chenyukun', age: 21, message: 'random message'`. A changeMessage function is defined to update the message. The component returns a JSX element displaying the person's name, age, and message, along with a button to change the message. The web browser on the right shows the rendered output: "Tutorial 14 - 208410166", "Chenyukun", "21", and "Yukun Chen, 208410166". A "Change Message" button is also visible. The browser's developer tools show the component's props and the state managed by the hook.`

```
import React, { useState } from 'react';

const T14_66_UseStateObject = () => {
  const [person, setPerson] = useState({
    name: 'chenyukun',
    age: 21,
    message: 'random message'
  });

  const changeMessage = () => {
    setPerson({ ...person, message: 'yukun chen, 208410166' });
  }

  return <>
    <h3>{person.name}</h3>
    <h3>{person.age}</h3>
    <h4>{person.message}</h4>
    <button className="btn" onClick={changeMessage}>Change Message</button>
  </>;
};

export default T14_66_UseStateObject;
```

W05-P2: useEffect 3 cases demo

The image shows a development environment with VS Code on the left and a web browser on the right. In VS Code, the file `T21_66-useEffect-basics.js` is open, showing three different use cases for `useEffect`. The first case is a default effect that runs on every render. The second case is a one-time effect that runs only once. The third case is a conditional effect that runs when `value2` changes. The web browser on the right shows the rendered output: "Tutorial 21 - 208410166", "1", "Value+1", "3", and "Value2+1". The browser's developer tools show the component's props and the state managed by the hook.

```
// second parameter
const T21_66_UseEffectBasics = () => {
  const [value, setValue] = useState(0);
  const [value2, setValue2] = useState(0);

  // by default runs after every re-render
  useEffect(() => {
    console.log('value', value);
  });

  // run once
  useEffect(() => {
    console.log('loading...');
  }, []);

  // run when value2 changes
  useEffect(() => {
    console.log('value2', value2);
    if (value2 > 0) {
      document.title = `New Message(${value2})`;
    }
  }, [value2]);

  return <>
    <h1>{value}</h1>
  </>;
};
```

W05-P3: T23_66: fetch data from Github and display them

T23_66-useEffect-fetch-data.js

```
const [users, setUsers] = useState([]);

const getUsers = async () => {
  try {
    const response = await fetch(url);
    const data = await response.json();
    console.log('data', data);
    setUsers(data);
  } catch (err) {
    console.log('error', err);
  }
}

console.log('users', users);

// run once
useEffect(() => {
  getUsers();
}, []);

return (
  <ul className="users">
    { users.map( (user) => {
      const {id, login, avatar_url, html_url} = user;
      return (
        <li key={id}>
          <img src={avatar_url} alt={login} />
          <div>
            <h4>{login}</h4>
          </div>
        </li>
      );
    }) }
  </ul>
);
```

bash

Tutorial 23 - 208410166

MojomboProfile

DefunktProfile

PjhyettProfile

WycatsProfile

EzmobiusProfile

React 17. Learn more: <https://reactjs.org/link/switch-to-createroot>

T23_66-useEffect-fetch-data.js:20

users

data

users

react_devtools_backend.js:4082

W05-P4: T31_66: conditional rendering -- 3 cases

T31_66-multiple-returns.js

```
import React, { useState, useEffect } from 'react';
const url = 'https://api.github.com/users/chenken1104';

const T31_66_MultipleReturns = () => {
  const [isLoading, setIsLoading] = useState(true);
  const [isError, setIsError] = useState(false);
  const [user, setUser] = useState('default user');

  // run once
  useEffect(() => {
    fetch(url)
      .then( (response) => {
        if( response.status >= 200 && response.status <= 299 )
          return response.json();
      })
      .then( (user) => {
        console.log('user', user.login);
        setUser(user.login);
        setIsLoading(false);
      })
      .catch( (error) => {
        console.log(error);
        setIsError(true);
      });
  }, []);

  if(isLoading) { ... }
  if(isError) { ... }

  return <h2>{user}</h2>;
};

export default T31_66_MultipleReturns;
```

bash

Tutorial 31 - 208410166

Chenken1104

Console

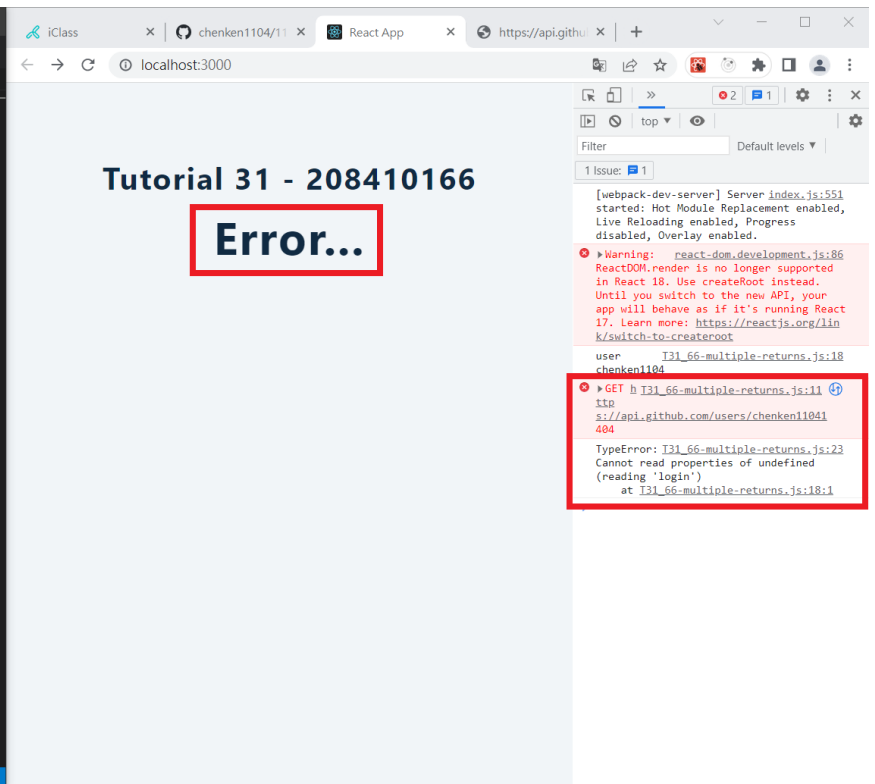
[webpack-dev-server] Server index.js:551 started: Hot Module Replacement enabled, Live Reloading enabled, Progress disabled, Overlay enabled.

Warning: react-dom.development.js:86 ReactDOM.render is no longer supported in React 18. Use createRoot instead. Until you switch to the new API, your app will behave as if it's running React 17. Learn more: <https://reactjs.org/link/switch-to-createroot>

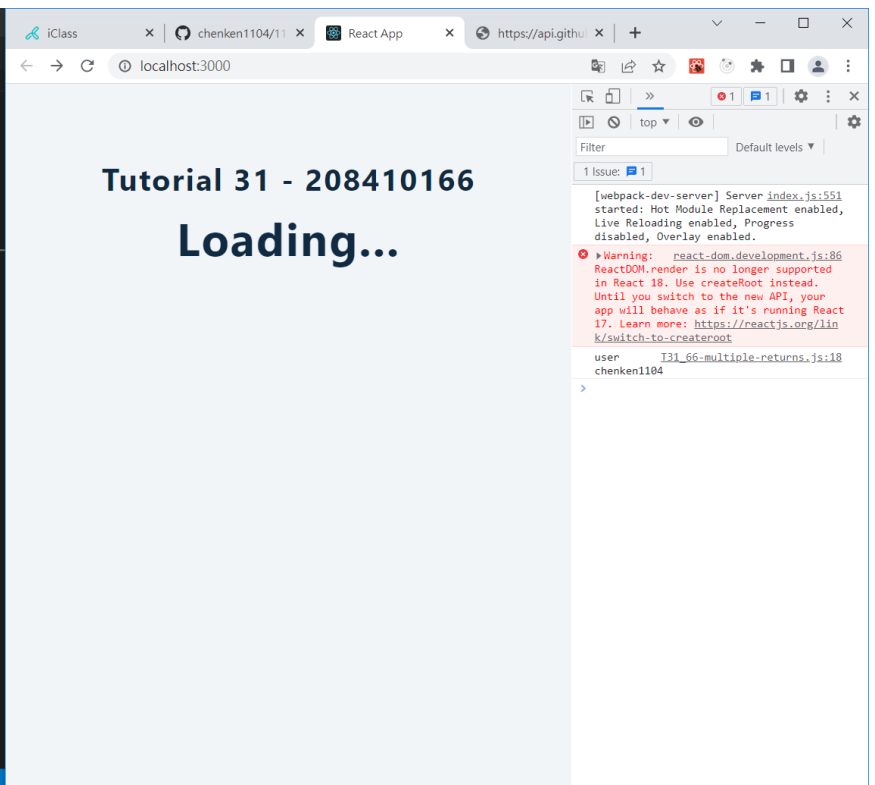
user

chenken1104

```
src > tutorial > 3-conditional-rendering > T31_66-multiple-returns.js > url
1 import React, { useState, useEffect } from 'react';
2 const url = 'https://api.github.com/users/chenken11041';
3
4 const T31_66_MultipleReturns = () => {
5   const [isLoading, setIsLoading] = useState(true);
6   const [isError, setIsError] = useState(false);
7   const [user, setUser] = useState('default user');
8
9   // run once
10  useEffect(() => {
11    fetch(url)
12      .then((response) => {
13        if (response.status >= 200 && response.status <= 299)
14          return response.json();
15      })
16      .then((user) => {
17        console.log('user', user.login);
18        setUser(user.login);
19        setIsLoading(false);
20      })
21      .catch((error) => {
22        console.log(error);
23        setIsError(true);
24      })
25  }, []);
26
27 > if(isLoading) { ...
28
29 > if(isError) {
30   return (
31     <div>
32       <h1>Error...</h1>
33     </div>
34   )
35 }
36
37 > if(isLoading) { ...
38
39 > if(isError) {
40   return (
41     <div>
42       <h1>Error...</h1>
43     </div>
44   )
45 }
```



```
1,66-multiple-returns.js > T31_66-MultipleReturns > useEffect() callback > then() callback
1 import React, { useState, useEffect } from 'react';
2 const url = 'https://api.github.com/users/chenken1104';
3
4 const T31_66_MultipleReturns = () => {
5   const [isLoading, setIsLoading] = useState(true);
6   const [isError, setIsError] = useState(false);
7   const [user, setUser] = useState('default user');
8
9   // run once
10  useEffect(() => {
11    fetch(url)
12      .then((response) => {
13        if (response.status >= 200 && response.status <= 299)
14          return response.json();
15      })
16      .then((user) => {
17        console.log('user', user.login);
18        setUser(user.login);
19        // setIsLoading(false);
20      })
21      .catch((error) => {
22        console.log(error);
23        setIsError(true);
24      })
25  }, []);
26
27 > if(isLoading) {
28   return (
29     <div>
30       <h1>Loading...</h1>
31     </div>
32   )
33 }
34
35 > if(isError) { ...
36
37 > if(isError) {
38   return (
39     <div>
40       <h1>Error...</h1>
41     </div>
42   )
43 }
```



W05-logs: show all four logs done today

```
User@E201-89 MINGW64 /d/1111-wp1-hooks-208410166 (main)
$ git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2022-10-04"
d5f84421 chenken1104 Wed Oct 5 21:32:59 2022 +0800 T31_66: conditional rendering -- 3 cases
2a2235c0 chenken1104 Wed Oct 5 20:19:22 2022 +0800 T23_66: fetch data from Github and display them
2eef7ac9 chenken1104 Wed Oct 5 19:46:06 2022 +0800 useEffect 3 cases demo
c8ee0910 chenken1104 Wed Oct 5 18:55:51 2022 +0800 Hooks Tutorial T14_66
```