



# BETTER SECURITY THROUGH CODE HYGIENE

---

DR. PHILIPPE DE RYCK

## *A template to render HTML in the application*

---

```
1 <h3>${title}</h3>
2 <p>${review}</p>
```

---

## *A review submitted by a malicious user*

---

```
1 This restaurant is highly recommended. The food
2 is exquisite and the service is impeccable.
3 <script>alert('Go there, now!');</script>
```

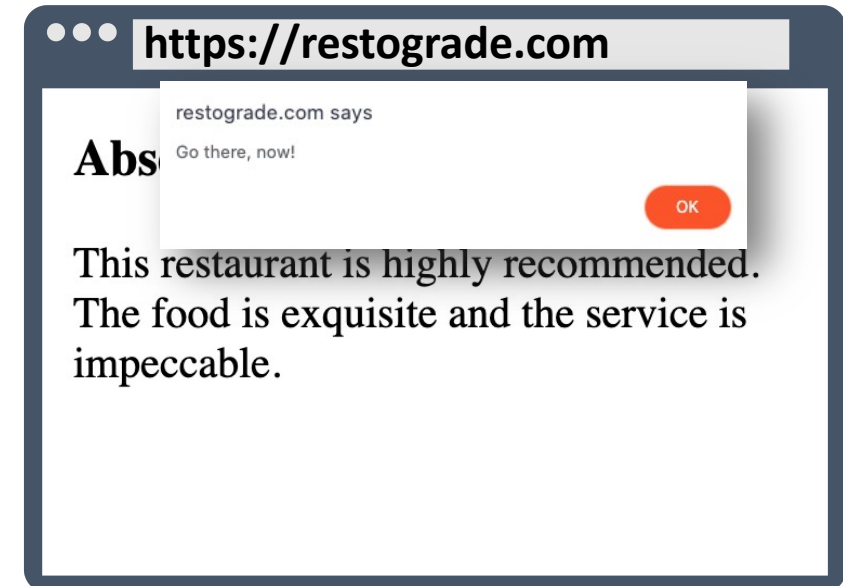
---

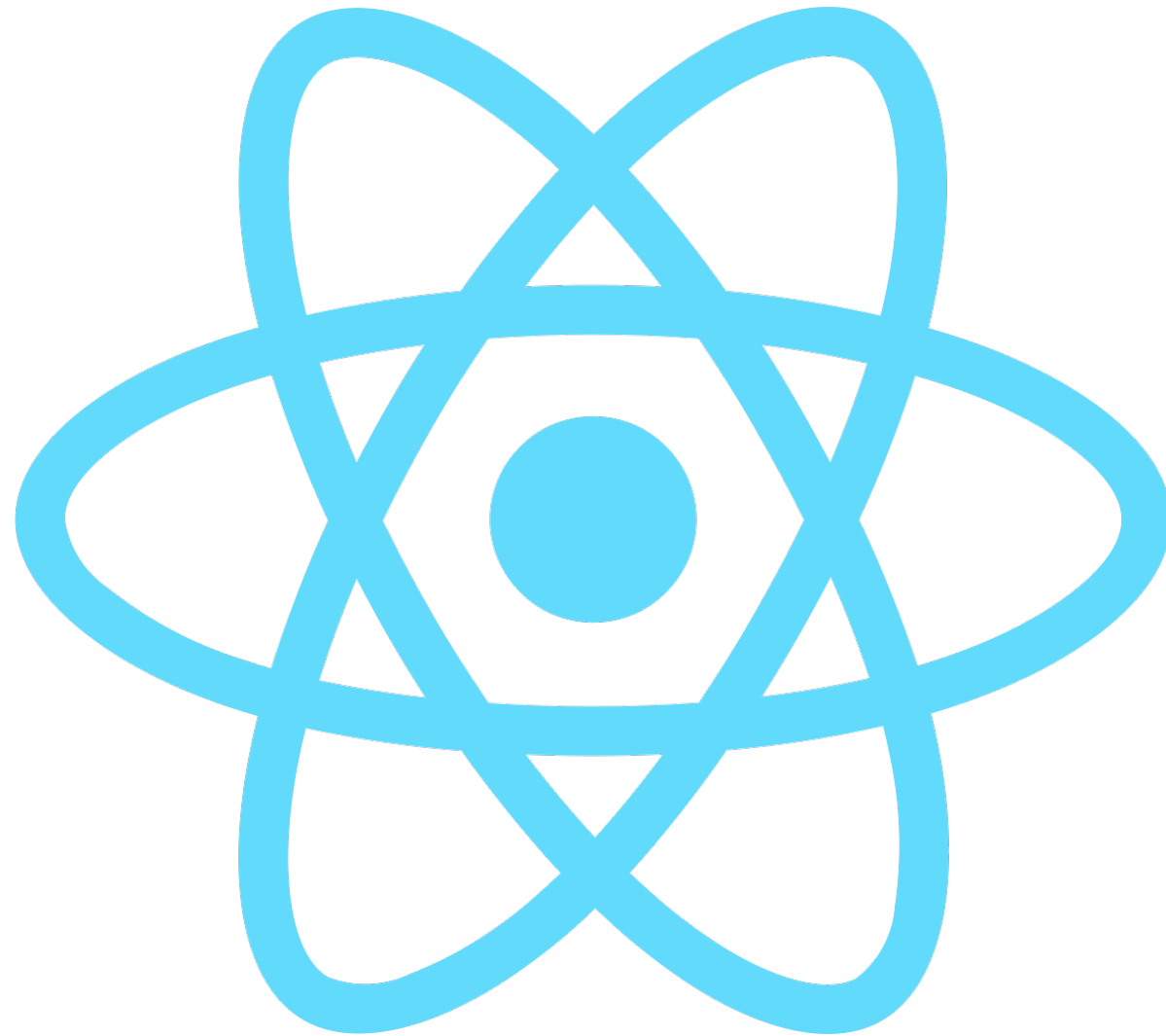
## *The HTML page sent to the browser*

---

```
1 <h3>Absolutely awesome</h3>
2 <p>This restaurant is highly recommended. The
3 food is exquisite and the service is impeccable.
4 <script>alert('Go there, now!');</script></p>
```

---





I am *Dr. Philippe De Ryck*



**Founder of Pragmatic Web Security**



**Google Developer Expert**



**Auth0 Ambassador**



**SecAppDev organizer**

I help developers with security



**Hands-on in-depth security training**



**Advanced online security courses**



**Security advisory services**

<https://t.me/learningnets>



<https://pragmaticwebsecurity.com>

## A JSX template to combine data with HTML

---

```
1 return ( <div>
2   <h3>{ title }</h3>
3   <p>{ review }</p>
4 </div>);
```

By default, React escapes values embedded in JSX before rendering them

## A review submitted by a malicious user

---

```
1 This restaurant is <b>highly recommended</b>. The
2 food is exquisite and the service is impeccable. <a
3 href="https://pics.example.com">Check out my story
4 here!</a>
```



# Some of the greatest things you learn from traveling

One of the great things on earth traveling teaches us by example. Here are some of the most precious lessons I've learned over the years of traveling.



Leaving your comfort zone might lead you to such beautiful sceneries like this one.

## Appreciation of diversity

Getting used to an entirely different culture can be challenging. While it's also nice to learn about cultures online or from books, nothing comes close to experiencing cultural diversity in person. It helps you learn to appreciate each and every single one of the differences while you become more open-minded and fluid.

## A JSX template to render user-provided HTML

```
1 return ( <div>
2 <h3>{ title }</h3>
3 <p dangerouslySetInnerHTML={{__html: review}}></p>
4 </div>);
```

*dangerouslySetInnerHTML*  
exposes the *innerHTML*  
property

## A review submitted by a malicious user

```
1 This restaurant is <b>highly recommended</b>. The
2 food is exquisite and the service is impeccable. <a
3 href="https://pics.example.com">Check out my story
4 here!</a>
```

This property is dangerous,  
since React does not apply  
any protection at all



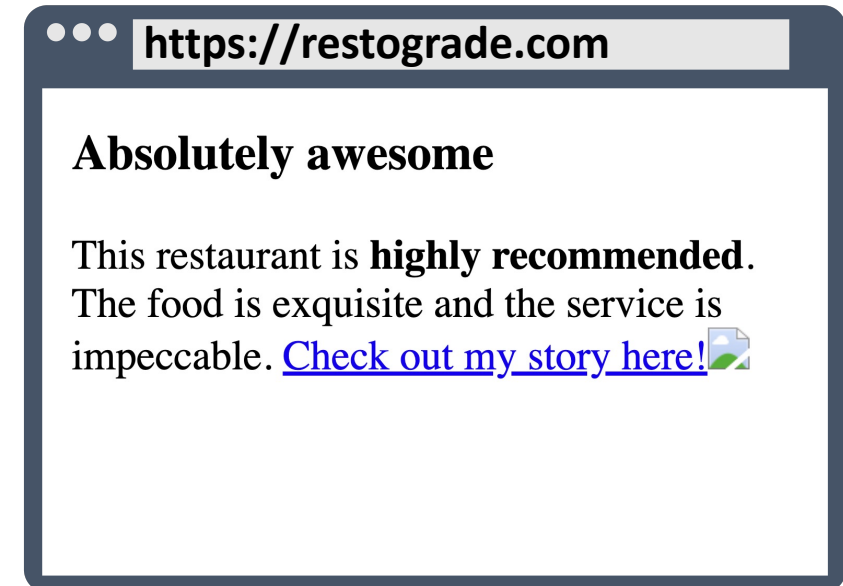
## A JSX template to render user-provided HTML

```
1 import DOMPurify from 'dompurify';
2
3 return ( <div>
4   <h3>{ title }</h3>
5   <p dangerouslySetInnerHTML={{__html: DOMPurify.sanitize(review)}}></p>
6 </div>);
```

DOMPurify turns untrusted HTML in safe HTML, making it safe to include in the page

## A review submitted by a malicious user

```
1 This restaurant is <b>highly recommended</b>. The
2 food is exquisite and the service is impeccable. <a
3 href="https://pics.example.com">Check out my story
4 here!</a>
```





**THAT WASN'T TOO BAD,  
NOW WAS IT?**



A terminal window with a dark background and light text. The window title bar shows three colored window control buttons (red, yellow, green) on the left, the text "-zsh" in the center, and a keyboard shortcut icon (⌘⌘1) on the right. The main content area contains a single line of text: "\$ semgrep --config 'p/react'".







OH, CRAP!

## *A JSX template directly using dangerouslySetInnerHTML (not recommended)*

---

```
1 import DOMPurify from 'dompurify';
2
3 return ( <div>
4   <h3>{ title }</h3>
5   <p dangerouslySetInnerHTML={{__html: DOMPurify.sanitize(review)}}></p>
6 </div>);
```

---

## *A JSX template using a SafeHtml component (recommended)*

---

```
1 import SafeHtml from './SafeHtml';
2
3 return ( <div>
4   <h3>{ title }</h3>
5   <SafeHtml element="p" html={{review}}></SafeHtml>
6 </div>);
```

---

## A JSX template using a SafeHtml component (recommended)

---

```
1 import SafeHtml from './SafeHtml';
2
3 return ( <div>
4   <h3>{ title }</h3>
5   <SafeHtml element="p" html={{review}}></SafeHtml>
6 </div>);
```

---

## The SafeHtml component

---

```
1 import React from 'react';
2 import DOMPurify from 'dompurify';
3
4 // This function will render HTML safely using DOMPurify
5 function SafeHtml({ element, html }){
6   return React.createElement(element, {
7     dangerouslySetInnerHTML: { __html: DOMPurify.sanitize(html) }
8   });
9 }
10 export default SafeHtml;
```

---



SafeHtml.js — xss-react

JS SafeHtml.js M ×

src > JS SafeHtml.js > SafeHtml

```
1 | import React from 'react';
2 | import DOMPurify from 'dompurify';
3 |
4 | // This function will render HTML safely using DOMPurify
5 | function SafeHtml({ element, html }) {
6 |     return React.createElement(element, { dangerouslySetInnerHTML: { __html: DOMPurify.sanitize(html) }
7 |     }); // nosemgrep typescript.react.security.audit.react-dangerouslysetinnerhtml.
8 |     react-dangerouslysetinnerhtml
9 | }
10|
11| export default SafeHtml;
```

master\* 0 0 Ln 6, Col 213 Spaces: 4 UTF-8 LF JavaScript



# USING CODE HYGIENE FOR SECURITY SUCCESS



*Developers should focus on development, not on fine-grained security rules*

*Encapsulate dangerous features in secure components*

*Use code analysis techniques to flag direct use of dangerous features*

# Cutting-edge



## Security

Live training starting  
October 15th

HOSTED BY  
**JIM MANICO**



PRESENTED BY  
**DR. PHILIPPE DE RYCK**

# API Security best practices

Live training starting  
October 25th

HOSTED BY  
**JIM MANICO**



PRESENTED BY  
**DR. PHILIPPE DE RYCK**



# THANK YOU!

*Follow me on Twitter to stay up to date  
on security resources and courses*



**@PhilippeDeRyck**