



链滴

React 用 Await 组件来控制依赖 promise 的渲染

作者: [HerbertHe](#)

原文链接: <https://ld246.com/article/1621817853876>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<h2 id="背景">背景</h2>

<p>慢慢在前端挖坑，各类框架都有接触学习，在学 Svelte 的过程中看到了一个好玩的语法——<code>await</code> 块。然后我就寻思可不可以在 React 也实现这样的组件呢？因为总感觉类似于 ejs 的方式有点奇怪，然后就整出来了一个 @herberthe/react-await 这个库，挺好玩的。</p>

<h2 id="在Svelte中">在 Svelte 中</h2>

<p>下面的这个是官方的示例</p>

```
<code class="language-html highlight-chroma"><span class="highlight-line"><span class="highlight-cl">{#await promise}
</span></span><span class="highlight-line"><span class="highlight-cl">  <span class="highlight-p">&lt;</span><span class="highlight-nt">p</span><span class="highlight-p">&g
</span></span>...waiting<span class="highlight-p">&lt;</span></span><span class="highlight-nt">p</sp
n><span class="highlight-p">&gt;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">{:then number}
</span></span><span class="highlight-line"><span class="highlight-cl">  <span class="highlight-p">&lt;</span><span class="highlight-nt">p</span><span class="highlight-p">&g
</span></span>The number is {number}<span class="highlight-p">&lt;</span></span><span class="highl
ght-nt">p</span><span class="highlight-p">&gt;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">{:catch error}
</span></span><span class="highlight-line"><span class="highlight-cl">  <span class="highlight-p">&lt;</span><span class="highlight-nt">p</span> <span class="highlight-na">s
yle</span><span class="highlight-o">=</span><span class="highlight-s">"color: red"</sp
n><span class="highlight-p">&gt;</span>{error.message}<span class="highlight-p">&lt;</span></span>
<span class="highlight-nt">p</span><span class="highlight-p">&gt;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">{/await}
</span></span></code></pre>
```

<p>其中的 then 块和 catch 块都可以省去</p>

<h2 id="在React中的实现">在 React 中的实现</h2>

<p>下面的例子来源于库的文档</p>

```
<code class="language-ts highlight-chroma"><span class="highlight-line"><span class="highlight-cl"><span class="highlight-p">&lt;</span><span class="highlight-nt">Await</span>
</span></span><span class="highlight-na">promise</span><span class="highlight-o">=</span><span class="highlight-p">{</span><span class="highlight-nx">promise</span><span class="highlight-p">}&gt;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">  <span class="highlight-p">{</span><span class="highlight-cm">/* Here are code for pending */</span><span class="highlight-p">}</span>
</span></span><span class="highlight-line"><span class="highlight-cl">  <span class="highlight-p">&lt;</span><span class="highlight-nt">Then</span><span class="highlight-p">
&gt;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">    <span class="highlight-p">{</span><span class="highlight-cm">/* Here are code for resolve */</span><span class="highlight-p">}</span>
</span></span><span class="highlight-line"><span class="highlight-cl">    <span class="highlight-p">&lt;</span><span class="highlight-nt">Then</span><span class="highlight-p">
">&gt;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">    <span class="highlight-p">&lt;</span><span class="highlight-nt">Catch</span><span class="highlight-p">
">&gt;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">    <span class="highlight-p">{</span><span class="highlight-cm">/* Here are code for reject */</span><span class="highlight-p">}</span>
</span></span><span class="highlight-line"><span class="highlight-cl">    <span class="highlight-p">{</span><span class="highlight-cm">/* Here are code for reject */</span><span class="highlight-p">}</span>
</span></span></code></pre>
```

```

highlight-p">&lt;/span> <span class="highlight-nt">Catch</span> <span class="highlight
p">&gt;</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
ight-p">&lt;/span> <span class="highlight-nt">Await</span> <span class="highlight-p">
gt;</span>
</span> </span> </code> </pre>
<p>完整的举例如下</p>
<pre> <code class="language-ts highlight-chroma"> <span class="highlight-line"> <span clas
="highlight-cl"> <span class="highlight-kr">interface</span> <span class="highlight-nx">IC
ustomResolveProps</span> <span class="highlight-p">{</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="h
ghlight-nx">awaitvalue</span>:<span class="highlight-kt"> any</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
ight-p">}</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
ight-kr">const</span> <span class="highlight-nx">CustomResolve</span>:<span class="h
ghlight-kt"> FC</span> <span class="highlight-p">&lt;/span> <span class="highlight-nt">I
ustomResolveProps</span> <span class="highlight-p">&gt;</span> <span class="highlight
o">=</span> <span class="highlight-p">({</span> <span class="highlight-nx">awaitvalue
</span> <span class="highlight-p">})</span> <span class="highlight-o">=&gt;</span> <sp
n class="highlight-p">&lt;/span> <span class="highlight-nt">h1</span> <span class="highl
ght-p">&gt;{</span> <span class="highlight-nx">awaitvalue</span> <span class="highlight
p">}&lt;/span> <span class="highlight-nt">h1</span> <span class="highlight-p">&gt;</s
an>
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
ight-kr">interface</span> <span class="highlight-nx">ICustomRejectProps</span> <span c
lass="highlight-p">{</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="h
ghlight-nx">awaiterror</span>:<span class="highlight-kt"> any</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
ight-p">}</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
ight-kr">const</span> <span class="highlight-nx">CustomReject</span>:<span class="hi
hlight-kt"> FC</span> <span class="highlight-p">&lt;/span> <span class="highlight-nt">IC
ustomRejectProps</span> <span class="highlight-p">&gt;</span> <span class="highlight-o"
=</span> <span class="highlight-p">({</span> <span class="highlight-nx">awaiterror</sp
n> <span class="highlight-p">})</span> <span class="highlight-o">=&gt;</span> <span cl
ss="highlight-p">&lt;/span> <span class="highlight-nt">h1</span> <span class="highlight
p">&gt;{</span> <span class="highlight-nx">awaiterror</span> <span class="highlight-p">
&lt;/span> <span class="highlight-nt">h1</span> <span class="highlight-p">&gt;</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="high
ight-kr">const</span> <span class="highlight-nx">App</span>:<span class="highlight-kt
"> FC</span> <span class="highlight-o">=</span> <span class="highlight-p">()</span> <s
an class="highlight-o">=&gt;</span> <span class="highlight-p">{</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="h
ghlight-kr">const</span> <span class="highlight-p">[</span> <span class="highlight-nx">
romise</span> <span class="highlight-p">,</span> <span class="highlight-nx">setPromise
</span> <span class="highlight-p">]</span> <span class="highlight-o">=</span> <span cla
s="highlight-nx">useState</span> <span class="highlight-p">&lt;/span> <span class="high
ight-nt">Promise</span> <span class="highlight-err">&lt;/span> <span class="highlight-n

```

```

">unknown</span><span class="highlight-p">&gt;</span><span class="highlight-o">&gt;
</span><span class="highlight-p">()</span>
</span></span><span class="highlight-line"><span class="highlight-cl">  <span class="h
ghlight-nx">useEffect</span><span class="highlight-p">()</span> <span class="highlight
o">=&gt;</span> <span class="highlight-p">{</span>
</span></span><span class="highlight-line"><span class="highlight-cl">    <span class=
highlight-kr">const</span> <span class="highlight-nx">prom</span> <span class="highlig
t-o">=</span> <span class="highlight-k">new</span> <span class="highlight-nx">Promis
</span><span class="highlight-p">((</span><span class="highlight-nx">resolve</span><span class="h
ghlight-p">,</span><span class="highlight-nx">reject</span><span class="h
ghlight-p">)</span> <span class="highlight-o">=&gt;</span> <span class="highlight-p">
</span>
</span></span><span class="highlight-line"><span class="highlight-cl">    <span clas
="highlight-nx">setTimeout</span><span class="highlight-p">()</span> <span class="highl
ight-o">=&gt;</span> <span class="highlight-p">{</span>
</span></span><span class="highlight-line"><span class="highlight-cl">    <span c
lass="highlight-nx">resolve</span><span class="highlight-p">(</span><span class="highli
ht-s2">"Data Here!"</span><span class="highlight-p">)</span>
</span></span><span class="highlight-line"><span class="highlight-cl">    <span clas
="highlight-p">},</span> <span class="highlight-mi">2000</span><span class="highlight-
">)</span>
</span></span><span class="highlight-line"><span class="highlight-cl">    <span class=
highlight-p">})</span>
</span></span><span class="highlight-line"><span class="highlight-cl">    <span class=
highlight-nx">setPromise</span><span class="highlight-p">(</span><span class="highligh
-nx">prom</span><span class="highlight-p">)</span>
</span></span><span class="highlight-line"><span class="highlight-cl">  <span class="h
ghlight-p">},</span> <span class="highlight-p">[]</span>
</span></span><span class="highlight-line"><span class="highlight-cl">  <span class="h
ghlight-k">return</span> <span class="highlight-p">(</span>
</span></span><span class="highlight-line"><span class="highlight-cl">    <span class=
highlight-p">&lt;</span> <span class="highlight-nt">Await</span> <span class="highlight-
a">promise</span><span class="highlight-o">=</span><span class="highlight-p">{</spa
><span class="highlight-nx">promise</span><span class="highlight-p">}&gt;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">    <span clas
="highlight-p">&lt;</span> <span class="highlight-nt">h1</span> <span class="highlight-p
">&gt;</span><span class="highlight-nx">pending</span><span class="highlight-o">~</s
an> <span class="highlight-p">&lt;</span> <span class="highlight-nt">h1</span> <span cl
ss="highlight-p">&gt;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">    <span clas
="highlight-p">&lt;</span> <span class="highlight-nt">p</span> <span class="highlight-p"
">&gt;</span> <span class="highlight-nx">waiting</span> <span class="highlight-k">for</s
an> <span class="highlight-nx">values</span> <span class="highlight-o">~</span> <span c
lass="highlight-p">&lt;</span> <span class="highlight-nt">p</span> <span class="highligh
-p">&gt;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">    <span clas
="highlight-p">&lt;</span> <span class="highlight-nt">Then</span> <span class="highlight
p">&gt;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">    <span c
lass="highlight-p">&lt;</span> <span class="highlight-nt">CustomResolve</span> <span cl
ss="highlight-p">/&gt;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">    <span clas
="highlight-p">&lt;</span> <span class="highlight-nt">Then</span> <span class="highligh
-p">&gt;</span>

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">      <span clas
="highlight-p">&lt;</span><span class="highlight-nt">Catch</span><span class="highligh
-p">&gt;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">      <span c
ass="highlight-p">&lt;</span><span class="highlight-nt">CustomReject</span><span clas
="highlight-p">/&gt;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">      <span clas
="highlight-p">&lt;</span><span class="highlight-nt">Catch</span><span class="highligh
t-p">&gt;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">      <span class=
highlight-p">&lt;</span><span class="highlight-nt">Await</span><span class="highligh
t-p">&gt;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">      <span class="h
ghlight-p">)</span></span>
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-p">}</span></span>
</span></span></code></pre>

```

实现思路

实现思路比较简单, 其实使用自定义 hook 包裹 promise 也可以很好实现同样的效果。这里的思是通过分别向 `Await` 的子组件 `Then` 和 `Catch` 注入 `props.awaitvalue` 和 `props.awaiterror` 属性。然后再由 `Then` 和 `Catch` 向它们的子组件 (除了字符串类型) 分别注入了上的属性 (深度为 1), 这样组件就可以拿到数据和错误信息了。

因为 props 和 states 的改变都会触发组件的渲染, 因此通过 `shouldComponentUpdate` 手动进行了性能优化以避免了标签页崩溃的问题。

现在咱们也可以在 React 里面, 使用类似于 Svelte Await Block 的语法了!