

# Why Did My Lightning Web Component Disappear?

A snowy Sunday morning in Colorado seemed like the perfect time to dive into [Lightning Web Components](#). All was going well with my simple Hello World component until I tried to fetch an Account's name using the [wire service](#). My component simply vanished from my Lightning Record Page.

Here's what my component's HTML file looked like:

```
<template>
  <lightning-card>
    Hello {name}!
  </lightning-card>
</template>
```

And here's the JavaScript file:

```
import { LightningElement, wire } from 'lwc';
import { getRecord } from 'lightning/uiRecordApi';
import ACCOUNT_NAME_FIELD from '@salesforce/schema/Account.Name';

export default class Example extends LightningElement {

  @wire(getRecord, { recordId: '0015600000BZoDoAAL', fields: [ACCOUNT_NAME_FIELD] })
  account

  get name() {
    return this.account.data.fields.Name.value;
  }
}
```

And... here's how it rendered in Salesforce:



Where'd my component go?

In an attempt to get to the bottom of this mystery, I created a function that would run when 'account' was populated with data:

```
@wire(getRecord, { recordId: '0015600000BZoDoAAL', fields: [ACCOUNT_NAME_FIELD] })
account({ error, data }) {
  if (data) {
    console.log('got data');
    console.log('data -> ' + JSON.stringify(data));
  } else if (error) {
    console.log('got error');
    console.log('error -> ' + JSON.stringify(error));
  }
}
```

In my browser console, I got exactly what I hoped for:

```
[Log] got data
[Log] data -> {"apiName":"Account","childRelationships":{},"fields":{"N
```

Next I added a similar debug statement to the name() function:

```
get name() {
  console.log('fetching name');
  console.log('data -> ' + JSON.stringify(this.account));
  return this.account.data.fields.Name.value;
}
```

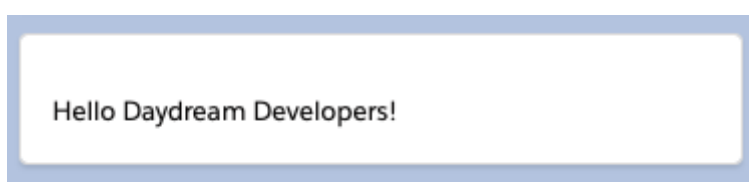
Here's what I found in the console log:

```
[Log] fetching name
[Log] data -> undefined
```

The component was attempting to fetch Name before it had a value. Apparently, doing that caused the component to fail silently in such a way that it refused to even render on the page.

**The solution was to add a conditional tag, `<template if:true={account.data}>` so that the name property is only referenced if account.data is not null.**

```
<template>
  <lightning-card>
    <template if:true={account.data}>
      <div class="slds-m-around_medium">
        Hello {name}!
      </div>
    </template>
  </lightning-card>
</template>
```



Two hours later, and I can now call off the search for my missing Lightning web component!

The key lesson learned here - if your Lightning web component disappears from the page silently - start your investigation by checking to see if you're referencing a null value. If you are, then use conditional logic to check for null before trying to reference the value.