

# How To Fix The Caching Problem In Salesforce Lightning Component



Salesforce trying to optimize performance by caching components on the client-side. This is awesome for the end-users because page loads will be faster. This is a lot less awesome for developers when developing lightning components because they may think that the code is wrong. Often they have to hit hard refresh a few times while you wait for the cache to clear.

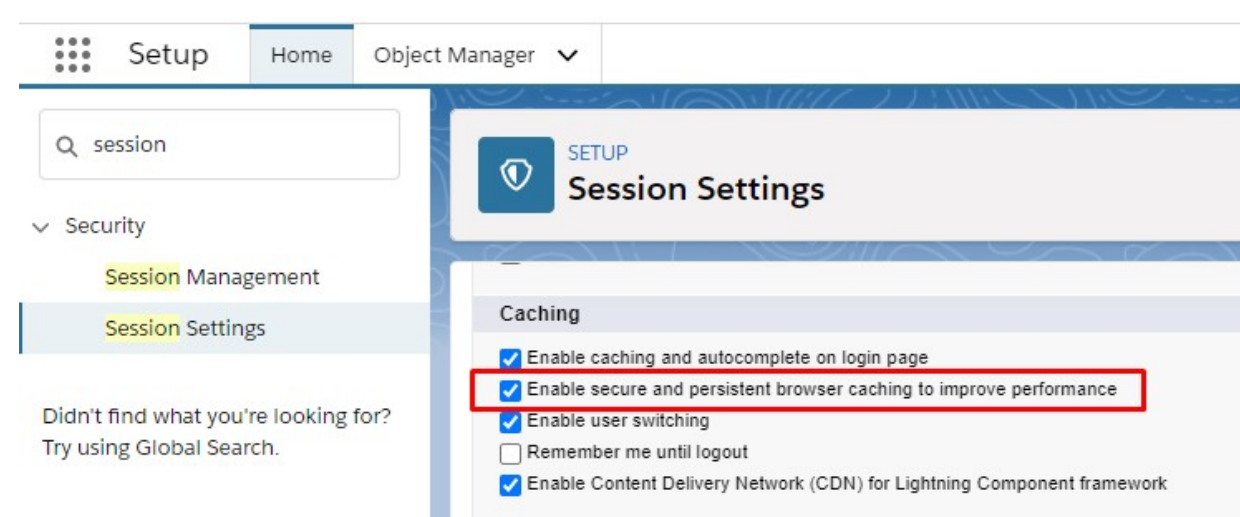
Usually, we have LWC which is doing some logic. The LWC is encapsulated in a Lightning Component and receives a Record Id from it because LWC can not be accessible from URL. When opening the parent Lightning Component for different records one after the other, the content is cached and it displays the content corresponding to the first record for all subsequent records.

Logic located in the `connectedCallback` method didn't execute for the second time when the user opens the Lightning component, `connectedCallback` is not executed at all.

There are several ways to fix caching problem in Lightning Components:

## 1. Session Caching

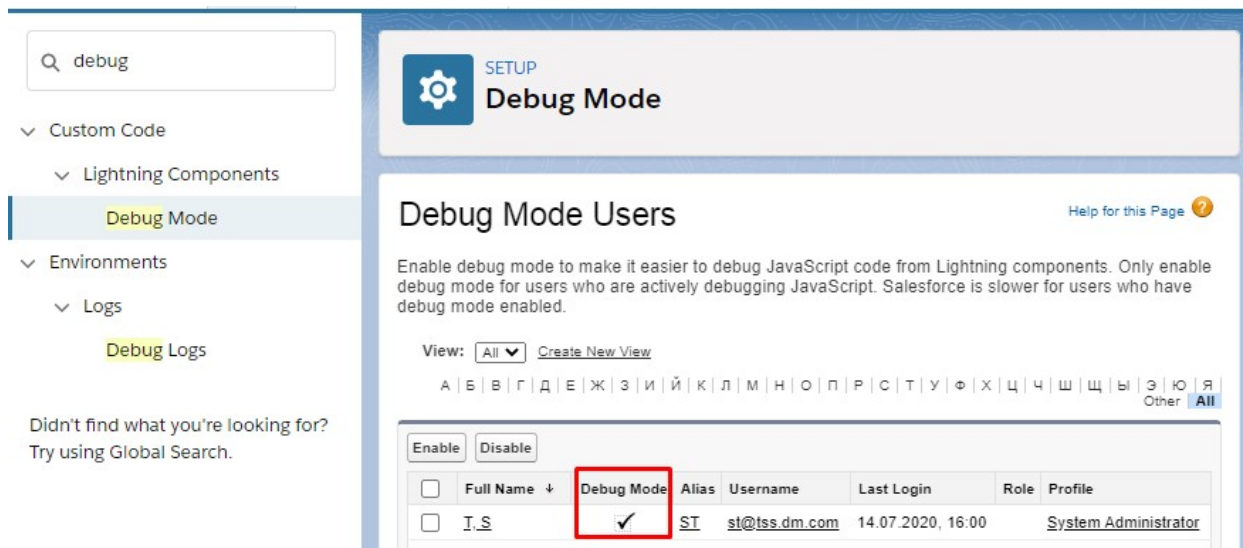
Click through `Setup > Session Settings > Caching` and deselect `Enable secure and persistent browser caching to improve performance`.



This is an Org-wide setting so it will affect all users.

## 2. Debug Mode

Click through `Setup > Custom Code > Lightning Component > Debug Mode`, then check the box next to your user and click `Enable`.



When you enable debug mode, JavaScript code isn't minified and is easier to read and debug. Debug mode also adds more detailed output for some warnings and errors. As with production mode, custom component code is not optimized or minified.

### 3. Refresh View Event

If none of the above methods isn't working try to refresh the component programmatically. You need to handle `pageReference` `change` event and fire `force:refreshView`. You can use the following code snippet.

Component

```
1 <aura:component implements="lightning:isUrlAddressable,
2   force:hasRecordId">
3   <aura:handler name="change" value="{!v.pageReference}"
4     action="{!c.onPageReferenceChanged}" />
5 </aura:component>
```

JavaScript Controller:

```
1 ({
2   onPageReferenceChanged: function(cmp, event, helper) {
3     $A.get('e.force:refreshView').fire();
4   }
5 })
```

### 4. Hard Reload

And the last one way is – Empty Cache and Hard Reload. You can press it several times to make sure that Lightning Component cache was invalidated.

