

HOW @AURAENABLED (CACHEABLE=TRUE) WORKS?

Friday 8 May, 2020 Published in [Articles](#)

As we all know, Salesforce provided a simple tool to improve performance of your components.

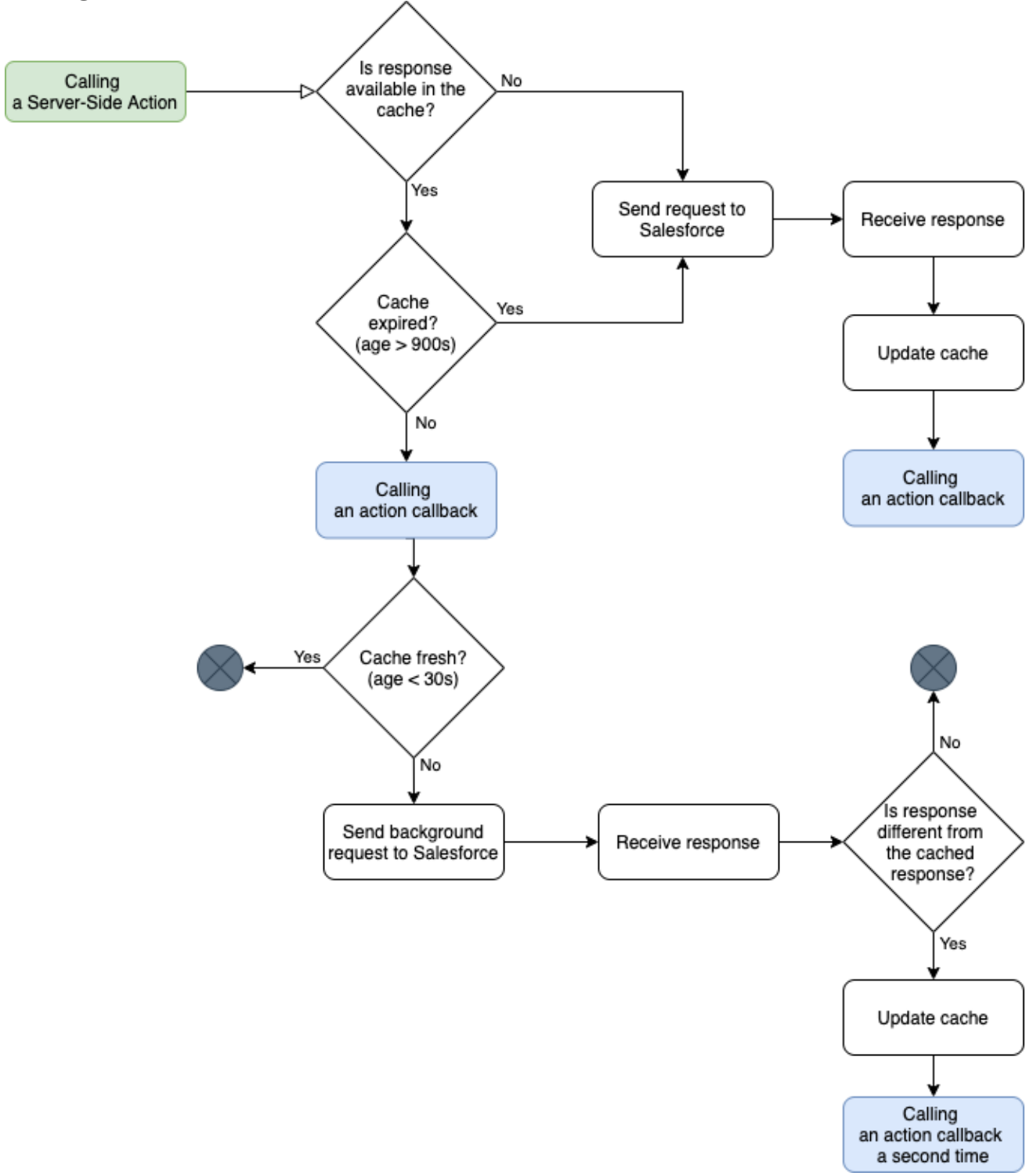
```
@AuraEnabled(cacheable=true)
public static List<Account> getAccounts() {
    // your code here
}
```

Let's get bogged down in how the sausage gets made. Making a method as cacheable improves your component performance by quickly showing cached data from the client-side storage without waiting for a server trip. But how long are cache kept alive? How does that work exactly? Let's find out it.

There are two parameters that set internally in the framework:

- **Expiration age** - in the Lightning Experience it sets to 900 seconds, controls how long the data is stored in the cache.
- **Refresh age** - in the Lightning Experience it sets to 30 seconds, controls how long cached data can be used without refreshing.

Now let's look at this diagram for better understanding of how does that work



There are possible three different scenarios:

1. Response is not available in the cache or it is expired. Salesforce calls the server method and caches the response. Then calls the action callback method.
2. Response is available in the cache and does not need to be refreshed. Salesforce calls the action callback method using cached response.
3. Response is available in the cache but needs to be refreshed. Salesforce calls the action callback method using cached response and additionally calls the server method in the background. If the response is different from the cached response, then Salesforce calls the action callback a second time.

Winter'21 Update: Increased Timeout for Client-Side Caching

When a component is generated on the server from metadata, the page loads from a client-side cache of the indexed database. The client-side cache timeout has increased from 15 minutes to 8 hours, with a refresh interval of 15 minutes. This increased timeout can result in faster page loads for users who bootstrap the application frequently or click links from outside the application to open a new browser window or tab to Lightning Experience.