

REUSABLE LIGHTNING NOTIFICATION ON RECORD UPDATE (BACKEND/API/UI) USING LIGHTNING DATA SERVICE

A very common scenario faced in various organizations is that multiple users are working on same set of records and end up editing the same record which has been edited by a different user. The user who last clicks on the save button gets error message saying “error...being edited by...”. All last typed text is lost. The possibility of our edits colliding in the same record is a real risk. So, it is only a matter of time before one of us has to retype an entry because one of us entered, edited, and saved before the other was finished with an edit. This is especially necessary for those of us that use Salesforce as a full-fledged service desk.

Salesforce does not prevent multiple users from editing a record / custom object at the same time so when they try to save their edits, they receive an error and have to start all over. One of the workarounds to this scenario is to use Lightning Data Service(LDS).

WHAT IS LIGHTNING DATA SERVICES?

We use Lightning Data Service to load, create, edit, or delete a record in your component without using Apex code. In addition to not needing Apex, Lightning Data Service improves performance and user interface consistency. LDS is the Lightning Components counterpart to the Visualforce standard controller, providing access to the data displayed on a page. Without LDS, each component within an app makes independent calls to the server to perform CRUD(create, read, update, delete) operations on a record, even if all components in the app pull from the same record data. Each server call reduces performance, leaving users twiddling their thumbs instead of working with their data. These independent server calls can also lead to inconsistencies, creating situations where a server call refreshes one component, leaving other components out of date.

HYPOTHETICAL SITUATION TIME!

So, you are editing records of any object which has many fields to be filled. While you are editing, fields of that object get updated by integration from the back-end. In this case you might end up filling all the fields and clicking on save button which would result in loss of your valuable data and time. This could be frustrating if the number of fields to be updated is more. We can escape from this situation by using LDS. When saving a record through LDS, the local cache isn't updated until the save is complete. When the save is successfully completed on the server, the cache is updated to the latest version of the record from the server, and all components with a reference to that record are notified. You don't have to worry about manually reloading the record into the cache after saving and making repeated server calls to check the update status. LDS handles all the work for you. When the record gets updated by some user, the other users working on the same record gets a “Toast Message Notification” notifying about the record updating without refreshing the record page thus no critical data loss occurs.

Type

Prospect

Phone

Website

[absyz.com](#)

Account Owner

[Abhishek Raj](#)

Account Site

Industry

Technology

RELATED

DETAILS

NEWS

Account Owner

Abhishek Raj

Rating

Hot

* Account Name

ARaj

Phone

Parent Account

Search Accounts...

Fax

Account Number

Website

[absyz.com](#)

Account Site

Ticker Symbol

Cancel

Save

Manually editing the record. At the same instant the record gets updated from the backend.

Execute Anonymous - Google Chrome

Secure

https://cunning-raccoon-133991-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexExecAnon

```

1 Account acc=[select name from account where id='0017F00000Q60ADQAZ'];
2 acc.name='Abhishek Raj';
3 acc.phone='432-533-8552';
4 update acc;

```

Open Log

Execute

Execute Highlighted

When the record is updated from the backend, the user who is editing same record gets toast message notification that the record is updated.

We can see in the above image that the Account name and Phone number has been updated while the user is still updating the record. So, now the user can stop making any further changes and save any critical data, thus saving his time and effort.

Other Solution

CHANGE DATA CAPTURE EVENTS – KEEP YOUR UI UPDATED WITH LATEST DATA

Have you ever faced the issue when you are viewing the record and at the same time other user has modified the same record and the data you are seeing is not up to date !! How to tackle the issue ? Here is the solution for that ... Let's start!!

CDC Events tracks all changes and all fields in an object and if any of the fields are modified the event will be triggered. This makes it very easy to listen for all type of changes in a record. **You can use Streaming API for a similar purpose, but there you specifically need to specify which fields and change type you want to track while creating PushTopics.**

IMPLEMENTATION EXAMPLE

Let's discuss an implementation scenario of CDC events where you want to keep your UI updated with latest changes

- User “Ron Weasley” is viewing an Opportunity record which is being viewed by the user “Harry Potter” as well at the same time.
- Harry Potter has changed the record data and has modified Opportunity amount.
- Ron Weasley who is unaware of this change and still viewing the old record, has to make a very important decision based on Opportunity amount and send the same to higher management.
- Now in this situation, Ron Weasley may take an incorrect decision and send incorrect data to higher management.

How to avoid the above situation? Well, lets leverage Change Data Capture events and lightning:empApi (Winter'19 release) component to track changes in Opportunity Record, and update the data on Ron Weasley's screen as soon as the data is modified by Harry Potter.