

Set Approval Process Lock And Unlock Records Using Apex Code

Biswajeet July 27, 2018 1 Comment

Sometimes we have faced business requirement to Lock or Unlock records in Salesforce. We can use apex `lock()` and `unlock()` methods in the `System.Approval` namespace to lock and unlock records by passing in record IDs or sObjects.

To enable this feature, go to Setup | Search Automation Settings in the Quick Find box | click on Automation Settings. Then, select Enable record locking and unlocking in Apex.

Specify a default workflow user. Salesforce recommends choosing a user with system administrator privileges.

Default Workflow User

By default, the email approval sender is the approval submitter. To override the default sender, select an organization-wide email address.

Email Approval Sender **Approval Submitter** ▼

Enabling email approval response lets users reply to email approval requests by typing APPROVE or REJECT in the first line and adding comments in the second line.

Enable email approval response ☐

⚠ By enabling the email approval response feature, you agree to allow Salesforce to process email approval responses, update approval requests for all active users in your organization, and update the approval object on behalf of your organization's users.

Let users pause flows when they need to wait for more information. Once you enable this setting, the Pause button appears on every screen that has "Show Pause button" selected.

Let users pause flows ☐

Let users resume interviews that are shared with them, either directly or via the role hierarchy. When disabled, interviews can be resumed only by the user who paused the interview or a flow admin who has view access.

Let users resume shared flow interviews ☒

Flows launched from a URL or from Setup use the Lightning runtime experience instead of the classic runtime experience. Only Lightning runtime supports two-column flow screens.

Enable Lightning runtime for flows ☐

When creating or updating records, the flow can use sObject variables to define the field values. If the running user doesn't have edit access to those fields, by default the flow fails to create or update the records. This setting filters the inaccessible fields from the request for Fast Create and Fast Update elements.

Filter inaccessible fields from flow requests ☒

Allows Apex code to set and remove approval process locks.

Enable record locking and unlocking in Apex ☒

When a process or flow interview fails, Salesforce sends an email with details about everything that was executed and what failed.

Send Process or Flow Error Email to **User Who Last Modified the Process or Flow** ▼

Lock Record Example:

```
1 //Get records to lock
2 List<case> caseList = [SELECT Id From Case LIMIT 10];
3 //Lock records
4 List<Approval.LockResult> lrList = Approval.lock(caseList, false);
5
6 // Iterate through each returned result
7 for(Approval.LockResult lr : lrList) {
8     if (lr.isSuccess()) {
9         //Operation was successful, so get the ID of the record that was processed
```

```

10     System.debug('Successfully locked account with ID: ' + lr.getId());
11 }
12 else {
13     //Operation failed, so get all errors
14     for(Database.Error err : lr.getErrors()) {
15         System.debug('The following error has occurred.');
```

Unlock Record Example:

```

1 //Get records to unlock
2 List<case> caseList = [SELECT Id From Case LIMIT 10];
3 //Check locked records
4 List<case> caseLockList = new List<Case>();
5 for(Case c :caseList){
6     if(Approval.isLocked(c.id)){
7         caseLockList.add(c);
8     }
9 }
10 //Unlock record
11 if(!caseLockList.isEmpty()){
12     //Unlock records
13     List<Approval.UnlockResult> ulrList = Approval.unlock(caseLockList, false);
14
15     // Iterate through each returned result
16     for(Approval.UnlockResult ulr : ulrList) {
17         if (ulr.isSuccess()) {
18             //Operation was successful, so get the ID of the record that was processed
19             System.debug('Successfully locked account with ID: ' + ulr.getId());
20         }
21         else {
22             //Operation failed, so get all errors
23             for(Database.Error err : ulr.getErrors()) {
24                 System.debug('The following error has occurred.');
```