Difference Between Salesforce Classic And Salesforce Lightning – Part 2

NOVEMBER 18, 2019 PRATYUSH KUMAR 1. SALESFORCE
27,572

In the first blog of the **Salesforce Lightning Migration** series, we discussed the Lightning Platform and the Lightning App Builder in detail. Now, it's time to differentiate between Salesforce Classic and **Salesforce Lightning** to give a clear picture of the two and how the latter one can prove to be more beneficial.

When compared to other CRM contenders, **Salesforce Classic** feels quite outdated to some degree. But with the Lightning Experience in the picture, the mindsets of users have changed in favor of the world's no. 1 CRM. the platform has a refreshed UI that focuses on displaying the information visually, as opposed to relying on numbers.

Differences Between Salesforce Classic vs Lightning

To provide more clarity on the topic, here are a few points that depict how Salesforce Classic and **Salesforce Lightning** differ from each other:

1. Enhanced User Experience

One of the major differences between Salesforce Classic and **Salesforce Lightning** is the user interface. The latter offers a much better **user interface** experience to its users that includes features like the drag-and-drop functionality that can be achieved without any code. Instead of hiring a **Salesforce developer** to create a Salesforce page or modify it, the page components can easily be rearranged by an admin according to their liking. Furthermore, Lightning helps in toning down the need for **Visualforce** for every task. The codes that are created during any kind of development need to be tested and then deployed. In case of missed bugs, the code is sent back to the developer to fix and the process starts over again. But with the help of **Salesforce Lightning**, these types of tedious processes can be avoided.

Lightning does not mean that companies don't need **Visualforce developers** anymore. Lightning provides a helping hand to the developers by moving minor customization tasks to the admins, which allows developers to focus on larger **Salesforce app development** projects.

2. Higher Security

With **Salesforce Lightning** comes enhanced security. For instance, **LockerService** is a feature that separates Lightning components for them to interact with each other. This helps in safeguarding the platform from malicious data. No such feature can be found in the Classic mode.

Permissions, too, work quite distinctly in **Salesforce Lightning**. The platform does not allow users to raise their assurance levels, say from standard to high, in-session. For that, they will have to log out of the Lightning platform and sign in again with authentication with a higher assurance level.

3. Einstein (Wave) Analytics

While an enhanced and upgraded user interface and security are the points strong enough for the comparison, **Salesforce Lightning** also provides users with access to **Einstein (Wave) Analytics** reporting, which the Classic does not. For creating graphs, charts, and lists, the Salesforce Classic reports depend on standard reporting types. At the time the data is refreshed, these dashboards prove to be a great option for capturing a view of important metrics.

When we talk about **Einstein Analytics**, it is a whole different deal. The platform carries its own database that is fetched from Salesforce and updates each hour automatically. Also, the dashboard displays the most recent, refreshed data.

That's not all. Einstein Analytics comes with various other features, like:

- Dashboard exports as pictures
- Widget editing right on the dashboard
- Advanced formula calculation

4. Progressive platform

In its initial days, Lightning was looked down at because of its transition and compatibility issues with objects, custom code, and apps. But gone are those days, and the platform has evolved with the ability to support all custom metadata objects, making it a lot simpler for companies to transition their existing apps and workflows with no requirements of building from scratch.

*Looking for a professional **Salesforce Lightning consultant**? Algoworks is right here!*

5. Hassle-free Lead Generation

Although Salesforce Classic enables you to create leads, the **Lightning Experience** offers more components in order to manage sales processes better. For instance, the Activity timeline in the Lightning mode allows users to identify what's been achieved for a specific lead and presents the details of every meeting, task, or call. The Path component enables the tracking of various stages involved in the business process, whereas, the News component offers updates about the leads on time.

What is Difference between classic and Lighting?

While the User Interface(UI) upgrade alone is superior enough compared to Salesforce Classic reporting, Lightning gives users access to Einstein Wave Analytics reporting. The platform has its own database pulled from Salesforce and is automatically updated every hour.

Triggers and Order of Execution

When you save a record with an insert, update, or upsert statement, Salesforce performs the following events in order.
On the server, Salesforce:

1. Loads the original record from the database or initializes the record for an upsert statement.
2. Loads the new record field values from the request and overwrites the old values.

If the request came from a standard UI edit page, Salesforce runs system validation to check the record for:

- Compliance with layout-specific rules
- Required values at the layout level and field-definition level
- Valid field formats
- Maximum field length

When the request comes from other sources, such as an Apex application or a SOAP API call, Salesforce validates only the foreign keys. Before executing a trigger, Salesforce verifies that any custom foreign keys do not refer to the object itself.

Salesforce runs custom validation rules if multiline items were created, such as quote line items and opportunity line items.

3. Executes record-triggered flows that are configured to run before the record is saved.
4. Executes all before triggers.
5. Runs most system validation steps again, such as verifying that all required fields have a non-null value, and runs any custom validation rules. The only system validation that Salesforce doesn't run a second time (when the request comes from a standard UI edit page) is the enforcement of layout-specific rules.
6. Executes duplicate rules. If the duplicate rule identifies the record as a duplicate and uses the block action, the record is not saved and no further steps, such as after triggers and workflow rules, are taken.
7. Saves the record to the database, but doesn't commit yet.
8. Executes all after triggers.
9. Executes assignment rules.
10. Executes auto-response rules.
11. Executes workflow rules. If there are workflow field updates:
    - Updates the record again.
    - Runs system validations again. Custom validation rules, flows, duplicate rules, processes, and escalation rules are not run again.
    - Executes before update triggers and after update triggers, regardless of the record operation (insert or update), one more time (and only one more time)
12. Executes escalation rules.
13. Executes the following Salesforce Flow automations, but not in a guaranteed order.
    - Processes
    - Flows launched by processes
    - Flows launched by workflow rules (flow trigger workflow actions pilot)

When a process or flow executes a DML operation, the affected record goes through the save procedure.

14. Executes entitlement rules.
15. Executes record-triggered flows that are configured to run after the record is saved.
16. If the record contains a roll-up summary field or is part of a cross-object workflow, performs calculations and updates the roll-up summary field in the parent record. Parent record goes through save procedure.

17. If the parent record is updated, and a grandparent record contains a roll-up summary field or is part of a cross-object workflow, performs calculations and updates the roll-up summary field in the grandparent record. Grandparent record goes through save procedure.
18. Executes Criteria Based Sharing evaluation.
19. Commits all DML operations to the database.
20. After the changes are committed to the database, executes post-commit logic such as sending email and executing enqueued asynchronous Apex jobs, including queueable jobs and future methods.

Additional Considerations

Note the following when working with triggers.

- If a workflow rule field update is triggered by a record update, Trigger.old doesn't hold the newly updated field by the workflow after the update. Instead Trigger.old holds the object before the initial record update was made. For example, an existing record has a number field with an initial value of 1. A user updates this field to 10, and a workflow rule field update fires and increments it to 11. In the update trigger that fires after the workflow field update, the field value of the object obtained from Trigger.old is the original value of 1, and not 10. See Trigger.old values before and after update triggers.
- If a DML call is made with partial success allowed, triggers are fired during the first attempt and are fired again during subsequent attempts. Because these trigger invocations are part of the same transaction, static class variables that are accessed by the trigger aren't reset. See Bulk DML Exception Handling.
- If more than one trigger is defined on an object for the same event, the order of trigger execution isn't guaranteed. For example, if you have two before insert triggers for Case and a new Case record is inserted. The order in which these two triggers are fired isn't guaranteed.
- If an object has multiple active record-triggered flows that are configured to run before or after the record is saved, the order in which those flows are executed isn't guaranteed.
- To learn about the order of execution when you insert a non-private contact in your org that associates a contact to multiple accounts, see AccountContactRelation.
- To learn about the order of execution when you're using before triggers to set `Stage` and `Forecast Category`, see Opportunity.

API	What you can do with it
SOAP API	Integrate your org's data with other applications using standard SOAP protocols.
REST API	Access objects in your org using standard REST protocols.
Metadata API	Manage customizations in your org and build tools that manage your metadata mo
Tooling API	Build custom development tools for platform applications.

API	What you can do with it
Marketing Cloud API	Expose Marketing Cloud capabilities with the REST API and get comprehensive acce functionality with the SOAP API.
Bulk API	Load, delete, and perform asynchronous queries on large data sets.
Streaming API	Send and receive notifications securely and efficiently. Notifications can reflect data custom events.
Connect REST API	Build UI for Commerce, CMS-Managed Content, Experience Cloud Sites, Files, Notifi
Mobile SDK	While it's technically a software development kit, it's worth including here. Integrate apps directly with Salesforce.