Image Search Application

Solution Details

**Requirement:**

1) Records can be fetched from https://jsonplaceholder.typicode.com/photos and are stored in Salesforce

2) Images are synchronized daily (please remember that photos removed from external API should also be removed from Salesforce)

3) Build lightning app to display records

4) Provide search engine to filter record based on the title

5) Allow to send list of filtered records to email address

6) Please provide a link to GitHub repo with your app

**Assumptions:**

1) Photo record update scenario has not been considered in solution design and development. There are no details in response from image source which indicates change of details. If all the existing records checked for possible updates in URL or title, then this will increase processing time reducing efficiency.

2) Photo id is unique and non-reusable. This assumption is used to create new record if matching id is not found in Salesforce existing records. Similarly, if existing record id found no match in retrieved records, then such record is being shortlisted for delete operation.

3) album id represents the clubbing of images and found no suitable use case linked with it in synchronization. Hence this id has not been used in the solution.

4) To display all the available photographs on page load, Solution is developed to fetch maximum 50k record at a time due to governor limit on SOQL result. View state issue are yet to observe. Currently total 5k records are in SFDC database and all 5k are being displayed without any issue using lightning component. Custom Pagination required if records are more than 50K.

5) Search logic to work for both exact match and partial match when search by Title.

6) To send email with search result details, email address to be received as user input. Multiple email address not supported in user input.

**Solution:**

1) Scheduler to fetch image values from third party system. Scheduler set to run daily.

2) Write logic in scheduler to create new image records in SFDC when new record created in third party system and delete the record from SFDC when same record found deleted in third party system.

3) Use HttpCallout to retrive images from third party system.

4) Create a lightning application, lightning tab and lightning component to display image records.

5) On the lightning page load display all the available image records.

6) On lightning page add a search button to search image with search string as image Title. Search string to be received as end user input.

7) Write search logic on the server side using apex class controller.

**Component details:**

**Apex:**

1) imageSearchEngine: Implemented Search logic.

2) ImageSyncScheduler: Scheduler class to fetch images daily from third party system.

3) IS\_ImageResponseDTO: Predefined Format/Template to receive and share callout response.

4) IS\_ImageSourceCallOut: Implemented HttpCallout logic

**Lightning:**

1) PhotoGallery: Lightning component to display images

2) Photo Gallery: Lightning tab to add lightning component for display.

2) PhotoGallery): Lightning Application to display "Photo Gallery" lightning tab and "Photographs" custom object tab

**Configurations:**

1. Remote site setting for endpoint <https://jsonplaceholder.typicode.com/photos>
2. Lightning Application “Photo Gallery”
3. Lightning Tab “Photo Gallery”

Script to schedule image sync for testing:

ImageSyncScheduler syncScheduler = new ImageSyncScheduler();

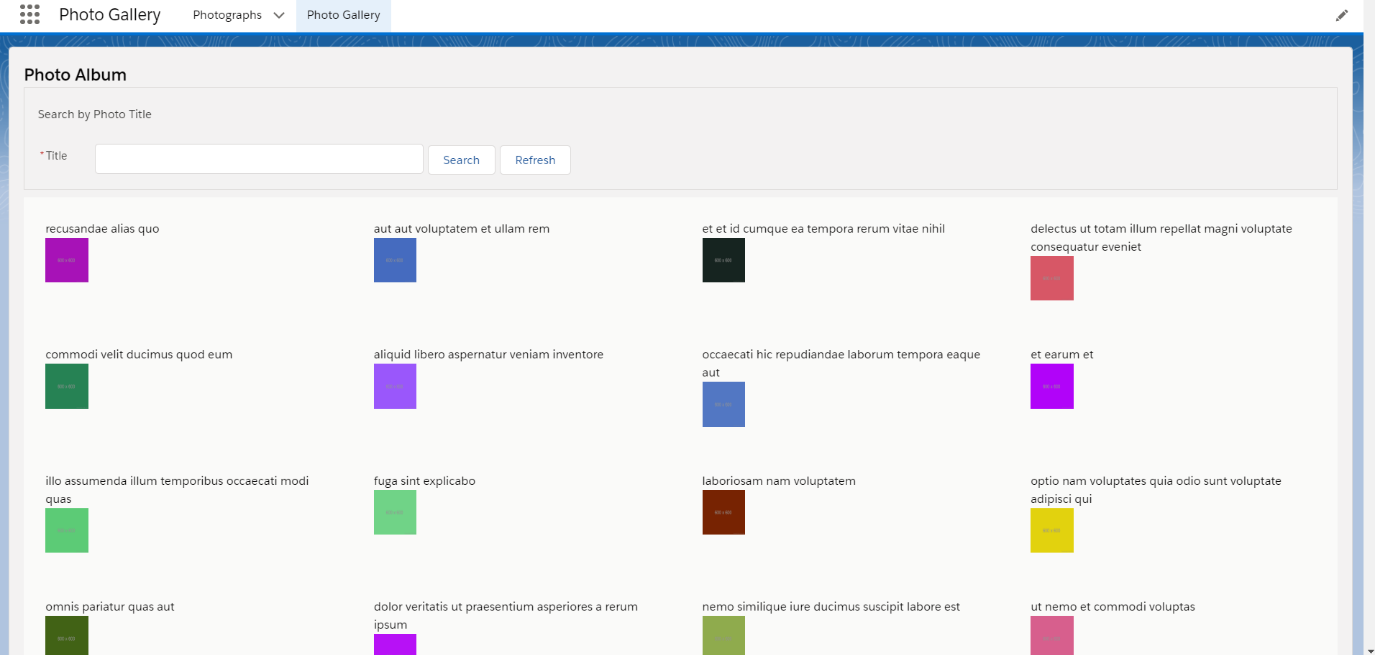
String sch = '0 <min> <hour> 1/1 \* ? \*'; //put hour and minute and replace the tags

String jobID = system.schedule('Scheduled Apex', sch, syncScheduler);

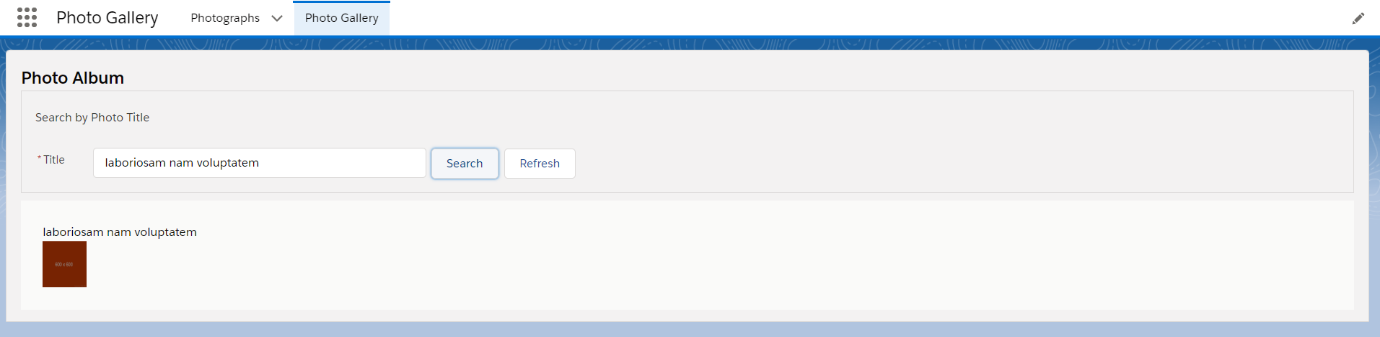
system.debug('job id: ' + jobID);

**Demo screenshots of application:**

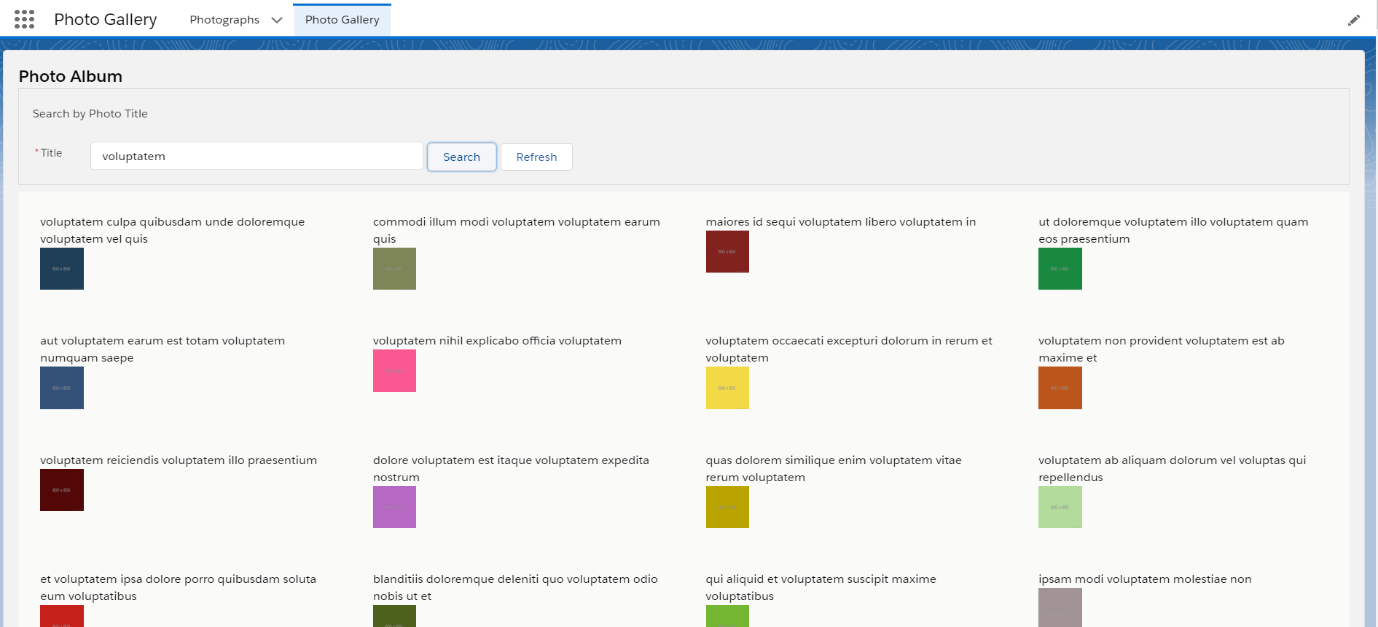
1) Page Load (All Records)



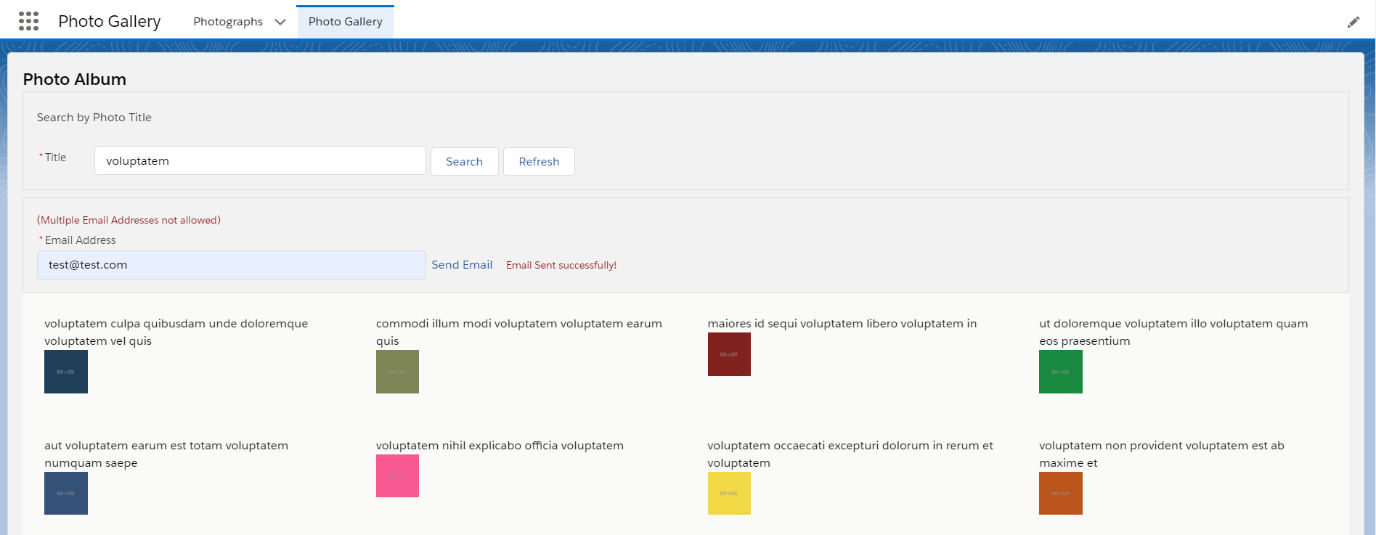
2) Search results (Exact match)



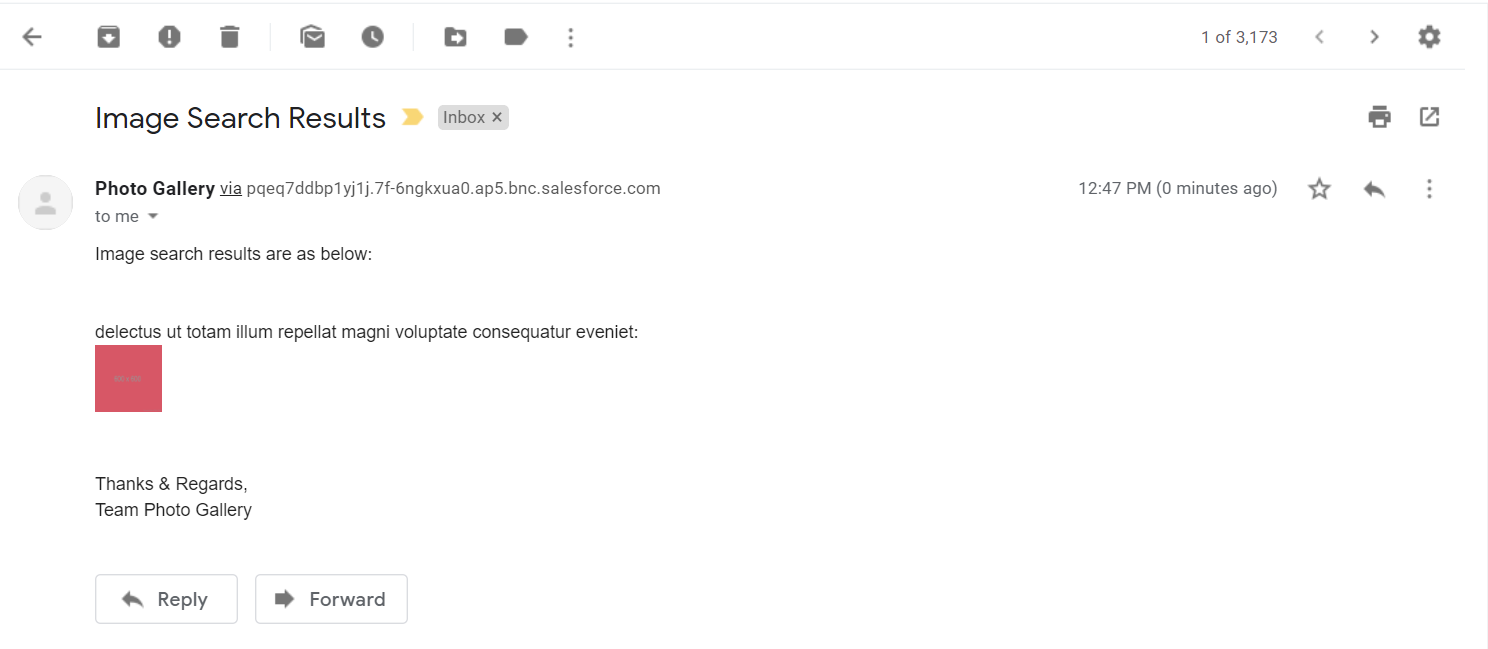
3) Search results (Partial match)



4) Email Address input screen



5) Email received with search result details



6) Scheduler setting for daily run

