Trailblazer COMMUNITY GROUPS

salesforce

Asynchronous Apex

Niki Vankerk

Vankerk Solutions Inc

Let's succeed together!

Trailblazer COMMUNITY GROUPS

salesforce

Niki Vankerk

Salesforce Architect, Vankerk Solutions Inc
13x Certified, Partner since 2007

SALESFORCE CERTIFIED
Application Architect

SALESFORCE CERTIFIED
System Architect

SALESFORCE CERTIFIED
Platform Developer II

SALESFORCE CERTIFIED
Sales Cloud Consultant

SALESFORCE CERTIFIED
Administrator

SALESFORCE CERTIFIED
Community Cloud Consultant

Agenda

- Async Apex: What and Why?
- Future
- Queueable
- Scheduled Apex
- Batch Apex
- Exception Handling

Asynchronous Apex

- More efficient for end user if automations can run in background while they continue working
- More scalable to queue up jobs when resources become available
- Higher limits means more data can be processed (ie 100 SOQL Queries goes up to 200 for async apex, double heap size, 6X CPU time)
- Can schedule work to happen nightly without a person starting the job
- Can execute call outs to other systems which are not allowed in triggers

Trailblazer COMMUNITY GROUPS

Types of Async Apex

Async Apex Feature	What/When
Future Method	Easy to implement (@future annotation and only take in primitives) Allows callouts, avoid mixed DML
Queueable Apex	Allows passing complex data types (ie. list of Leads) Can chain multiple jobs
Scheduled Apex	Allow jobs to run independently on schedule Can schedule batch jobs
Batch Apex	Chunks up data for processing that requires many queries or other actions that would take too long Allows processing large volumes of data

Trailblazer COMMUNITY GROUPS

Future Apex

- Add @future annotation to method
- Can only take in primitive data types (list/set of Ids, String) so to work with data in Salesforce you need to query for the values in the method
- No return value, must be void
- Limit to 50 future calls per transaction and could be delayed due to flow control if your org is queueing up too many
- Can’t trigger future call in a future call (ie future method updates a case and case has a trigger that executes a future call)
- Use System.isFuture() in code to avoid call a future method

Queueable Apex

- Requires a separate class with constructor (optional) and execute
- Invoke one by using System.enqueueJob() passing in the class object
- Can enqueue up to 50 jobs in a synchronous transaction (ie trigger) but only 1 while in asynchronous transaction (ie running a queueable)
- Can chain queueables together by enqueueing the next job at the end of execute – but must check Test.isRunningTest() as you get an error in tests if a queueable calls another.
- No limit on how many can be chained together

Scheduled Apex

- Requires a separate class with constructor (optional) and execute
- Invoke one by using System.Schedule() passing in the class object, Cron expression and job Name
- Limit of 100 scheduled jobs at a time
- No callouts from scheduled apex, call future or queueable methods from scheduled apex instead

Batch Apex

- Requires a separate class with constructor (optional), start, execute and finish
- Invoke one by using Database.executeBatch() passing in the class object, and optionally batch size – 200 default
- Can use stateful if you want to track values over the span of all batches but slows things down
- Test class can only run 1 batch so don’t test with more than 200 records
- Batches can also chain together and execute batch from its finish method

Limits and Considerations

- Triggering async processing while in an async process:
  - From the initial synchronous call, you may trigger up to 50 future calls and 50 queueable calls
  - Future calls may not trigger another future call
  - Queueable apex may trigger only 1 future and 1 queueable call

- Chaining Queueables: from the finish method of a queueable you can call another queueable

Trailblazer COMMUNITY GROUPS

Resources

- Trailhead Module on Async: https://trailhead.salesforce.com/content/learn/modules/asynchronous_apex
- Dev Guide on Async: https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_async_overview.htm
- Scheduled Apex Cron details (bottom of page: https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_scheduler.htm

Trailblazer COMMUNITY GROUPS

Trailblazer
COMMUNITY GROUPS

salesforce

Thank You!

Let's succeed together!