

Message Handling

@Handle(Command) (M)

Handles specific message types such as commands or events.

@LocalHandler (P) (C) (M)

Handles messages locally in the thread that publishes a message.

@Trigger (P)

Injects the message that triggered the current one.

@Stateful (C)

Makes handlers persist across invocations, maintaining historical context for handling subsequent messages.

@Association (M) (F)

Associates stateful handlers with specific message properties for targeted processing.

@Consumer (C) (P)

Configures an independent message consumer that tracks a message log.

@TrackSelf (C)

Creates a non-local handler for a payload class with handler methods. Without this, the message would be handled in the publishing thread.

@Timeout (C)

Overrides the default maximum timeout for requests that are sent using sendAndWait-like methods.

Scheduling

@Periodic (C) (M)

Schedules periodic messages that will automatically be (re)scheduled at set intervals.

Validation

@AssertLegal (M) (F)

Validates actions against business rules, based on entity state.

@RequiresUser (P) (C) (C) (M)

Ensures the message is sent by an authenticated user.

@RequiresRole (P) (C) (C) (M)

Checks if the user holds the required role.

@ForbidsRole (P) (C) (C) (M)

Prevents users with certain roles from performing specific actions.

@ValidateWith (C)

Allows custom validation logic for message payloads.

Event sourcing

@Apply (C) (M)

Applies state changes to entities and stores them for event sourcing.

@InterceptApply (M)

Intercepts and modifies state changes before application.

Modeling

@Aggregate (C)

Defines a root entity that groups related objects.

@EntityId (M) (F)

Designates the primary identifier of an entity.

@Alias (M) (F)

Marks an alternative identifier for the entity.

@Member (F)

Marks properties containing sub-entities.

Search

@Facet (C) (M) (F)

Adds faceted search functionality for filtering across multiple criteria.

@SearchExclude (C) (M) (F)

Excludes properties or classes from search indexing.

@SearchInclude (C) (M) (F)

Includes specific properties for search, overriding exclusions.

Versioning

@Upcast (M)

Converts older data versions to the latest format.

@Downcast (M)

Converts newer data to older formats for compatibility.

@Revision (C)

Specifies the revision or version of a data object.

Filtering

@FilterContent (M)

Filters out sensitive or unnecessary data based on user permissions.

@ProtectData (C) (M)

Removes sensitive data from a message before transmission and returns it before handling.

@DropProtectedData (C) (M)

Removes sensitive data from a message before transmission and returns it before handling.

Message Publication
<i>send(Command)</i>
Sends specific message types such as commands or queries and returns a <code>CompletableFuture</code> with the result.
<i>(query)AndWait</i>
Sends specific message types such as commands or queries and blocks until receipt of the result.
<i>sendAndForgetCommand</i>
Sends a command without waiting for its result.
<i>publishEvent</i>
Publishes an event.

Scheduling
<i>schedule</i>
Appends a message to the Schedules log to be released in the future.
<i>scheduleCommand</i>
Schedules a command for some time in the future.
<i>schedulePeriodic</i>
Schedules a periodic message, see <code>@Periodic</code> .
<i>cancelSchedule</i>
Cancels a scheduled message or command.

Utilities
<i>currentTime</i>
Get the current time Instant from the internal clock used by Flux.
<i>generateId</i>
Generates a new identifier using the identity provider used by Flux

Search
<i>index</i>
Indexes an object for search, replacing any document with the same id.
<i>search</i>
Prepares a search in one or more collections.
<i>getDocument</i>
Gets a document by id.
<i>deleteDocument</i>
Deletes a document by id.
<i>deleteCollection</i>
Deletes a collection.
<i>Search.match</i>
Match documents that contain one or more words.
<i>Search.lookAhead</i>
Match documents with words that start with a given string.
<i>Search.matchFacet</i>
Match documents with matching facets, see <code>@Facet</code> .
<i>Search.inPeriod</i>
Match documents that fall within a certain time period.
<i>Search.between</i>
Match documents with properties that fall within a certain range (min/max).
<i>Search.fetch</i>
Fetches a maximum number of matching documents.
<i>Search.fetchAll</i>
Fetches all matching documents.
<i>Search.fetchFirst</i>
Fetches the first matching document.
<i>Search.facetStats</i>
Count occurrences of facets in all matching documents, see <code>@Facet</code> .
<i>Search.delete</i>
Deletes all matching documents.

Modeling
<i>loadAggregate</i>
Loads the root entity with given id.
<i>loadAggregateFor</i>
Loads the root entity for a (child) entity with given id.
<i>loadEntity</i>
Loads the root or child entity with given id.
<i>Entity.assertLegal</i>
Check if a given command is legal given the state of the entity using <code>@AssertLegal</code> annotations.
<i>Entity.apply</i>
Apply an update to the entity using <code>@Apply</code> annotations.
<i>Entity.assertAndApply</i>
Combination of <code>assertLegal</code> and <code>apply</code> .

Testing
<i>TestFixture.create</i>
Creates a synchronous test fixture for given-when-then tests. Handlers are registered as local handlers.
<i>TestFixture.createAsync</i>
Creates an asynchronous test fixture for given-when-then tests. Handlers are registered as in a real application.
<i>TestFixture.given(...)</i>
Specify preconditions that have occurred prior to the behaviour you want to test.
<i>TestFixture.when(...)</i>
Specify the action you want to test the behaviour of.
<i>ResultValidator.then(...)</i>
Specify the expected behaviour of the 'when' phase.