Fluxzero cheatsheet

Annotations

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.

Web Message Handling

@Handle[Get] M

Handles web request message with given HTTP method like GET or PUT.

@[Query]Param P

Injects a web request parameter like a URI path or query segment.

@Path P C M

Specifies the path of a web request handler.

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.

Scheduling

@Periodic C M

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

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.

F = Field M = Method C = Constructor P = Parameter C = Class P = Package

Fluxzero cheatsheet

Functions

Message Publication

send(Command)

Sends specific message types such as commands or queries and returns a CompletableFuture with the result.

(query)AndWait

Sends specific message types such as commands or queries and blocks until receipt of the result.

sendAndForgetCommand

Sends a command without waiting for its result.

publishEvent

Publishes an event.

Scheduling

schedule

Appends a message to the Schedules log to be released in the future.

scheduleCommand

Schedules a command for some time in the future.

schedulePeriodic

Schedules a periodic message, see @Periodic.

cancelSchedule

Cancels a scheduled message or command.

Utilities

currentTime

Get the current time Instant from the internal clock used by Fluxzero.

generateId

Generates a new identifier using the identity provider used by Fluxzero.

Search

index

Indexes an object for search, replacing any document with the same id.

search

Prepares a search in one or more collections.

getDocument

Gets a document by id.

deleteDocument

Deletes a document by id.

deleteCollection

Deletes a collection.

Search.match

Match documents that contain one or more words.

Search.lookAhead

Match documents with words that start with a given string.

Search.matchFacet

Match documents with matching facets, see @Facet.

Search.inPeriod

Match documents that fall within a certain time period.

Search.between

Match documents with properties that fall within a certain range (min/max).

Search.fetch

Fetches a maximum number of matching documents.

Search.fetchAll

Fetches all matching documents.

Search.fetchFirst

Fetches the first matching document.

Search.facetStats

Count occurrences of facets in all matching documents, see @Facet.

Search.delete

Deletes all matching documents.

Modeling

loadAggregate

Loads the root entity with given id.

loadAggregateFor

Loads the root entity for a (child) entity with given id.

loadEntity

Loads the root or child entity with given id.

Entity.assertLegal

Check if a given command is legal given the state of the entity using @AssertLegal annotations.

Entity.apply

Apply an update to the entity using @Apply annotations.

Entity.assertAndApply

Combination of assertLegal and apply.

Testing

TestFixture.*create*

Creates a synchronous test fixture for given-when-then tests. Handlers are registered as local handlers.

TestFixture.*createAsync*

Creates an asynchronous test fixture for given-when-then tests. Handlers are registered as in a real application.

TestFixture.given(…)

Specify preconditions that have occurred prior to the behaviour you want to test.

TestFixture.when(…)

Specify the action you want to test the behaviour of.

ResultValidator.then(…)

Specify the expected behaviour of the ‘when’ phase.