Contents

Errors are inevitable when performing a save: maybe a database constraint is violated, a login lacks permissions, or the database is unreachable. Whatever the cause, you'll need to **handle save errors**.

Handling the exception

If an *EntityManagerSaveException* is raised, there are several ways you can trap it.

One, of course, is via try/catch logic in your code. Save exceptions are also passed to your *EntityServerError* handler; and to your *Saved* event handler; and available in the *SaveResult* from a "try save changes".

You should prepare your code to trap and analyze the save exception. The EntityManagerSaveException has the information you need to help diagnose and handle the problem.

Your application should determine how to process save errors, and recover from then when possible.

EntityManagerSaveException

The *EntityManagerSaveException* is raised for all save-related exceptions.

If you've added a handler to the *EntityManager's* *EntityServerError* event, that handler will be called first when a save-related exception occurs. If there is no handler or it doesn't handle the exception, the *EntityManager* throws it again, now in the context of the save changes call.

We recommend that you do not handle save exceptions in an *EntityServerError* handler; leave that to the code near your save changes call that traps and interprets save failures.

The *EntityManagerSaveException* inherits from EntityServerException, supplementing it with information about which entity or entities in the local cache caused the problem.

There are several properties on the *EntityManagerSaveException* that can be very useful in diagnosing the problem that occurred.

Property	Property type	Description
EntitiesWithErrors	IList<Object>	A list of the entities with errors. In practice, this will usually be a list of one -- the first entity to fail --, since database saves are transactional. If the error was triggered by a validation failure, then the list will contain all the entities with validation errors.
InnerException	Exception	The precipitating exception, whether from an attempt to connect to the data source or an exception from the data source itself such as a concurrency conflict or referential integrity violation.
FailureType	FailureType	A classification of the error that cause the problem. Several FailureTypes are possible.

FailureType
Connection
Data
Concurrency
Validation
Other

What happens to any local entities after a save fails

These entities remain in the cache and retain exactly the values and setting they had before the save attempt. This means that you may be able to correct the entities and attempt to re-save them.