

# CoralExceptionClassification

Primary Owner [glacier-core \(Idap\)](#)

Last modified 2 years ago by [kesting](#).

## Overview

When a Coral call is made it can fail for a variety of reason. For classification purposes we sort these exceptions into three separate classes: Error, Fault, and Failure.

<https://w.amazon.com/index.php/Coral/Metrics/Common#Exceptions>

## Error

### Definition

Errors are purely client side errors, generally due to bad input. These do not reflect any fault of the service itself.

### Example

In GMS, if a client calls `getGrid()` with an invalid `gridId`, an `InvalidInputException` will be thrown. This reflects a problem with the supplied input and not with GMS itself.

### Suggested Rollback Strategy

100% tolerance to errors. If Errors are categorized correctly the appearance of one should not be a reflection of the state of the service.

### Coral Model Definition

Errors are declared within the `CoralModel` without the `fault` tag.

Example:

```
<documentation target="InvalidInputException">
  Could not execute the request because the passed in input is invalid.
</documentation>
<structure name="InvalidInputException" isa="GMSException" />
<exception target="InvalidInputException" />
```

## Fault

### Definition

Faults represent acknowledged server side errors. These oftentimes come in the form of dependency failures. These are errors that we know can exist but are not necessarily the fault of the service itself.

### Example

In GMS if we receive a `ProvisionedThroughputExceededException` while processing a `getGrid()` call we will throw a `DependencyException`, which is marked as a fault.

### Suggested Rollback Strategy

1% tolerance to faults over 2 consecutive data points. The reason for this is that we want to be able to tolerate a small number of faults for a short period of time without rolling back. A common cause of this could be a sudden spike in traffic that could in turn briefly cause `ProvisionedThroughputExceptions` to occur while reading from DDB.

### Coral Model Definition

Errors are declared within the `CoralModel` with the `fault` tag.

Example:

```
<documentation target="DependencyException">
  Thrown when the service receives an exception from one of its dependencies when trying to
  process a request.
</documentation>
<structure name="DependencyException" isa="GMSEException" />
<exception target="DependencyException" />
<fault target="DependencyException" />
```

# Failure

## Definition

Failures represent unacknowledged server side errors. The presence of one represents a problem with the service itself.

## Example

Any bugs, such as NullPointerExceptions, should manifest as failures

## Suggested Rollback Strategy

0% tolerance to failures. We should rollback upon seeing even one error

## Coral Model Definition

Failures are not represented within the Coral Model. Thus any unexpected exceptions will be classified as failures