**CIRCUIT BREAKER**

The Circuit Breaker can prevent an application repeatedly trying to execute an operation that is likely to fail, allowing it to continue without waiting for the fault to be rectified.

Suppose an application is failing repeatedly due to some reason and is stuck then in this case, Circuit Breaker will prevent accessing that application and will give some alternate solution like an error or another operation which is to be performed so that other services, which are depending on this faulty operation, will not fail due to cascading or wait for too long for the error to be resolved.

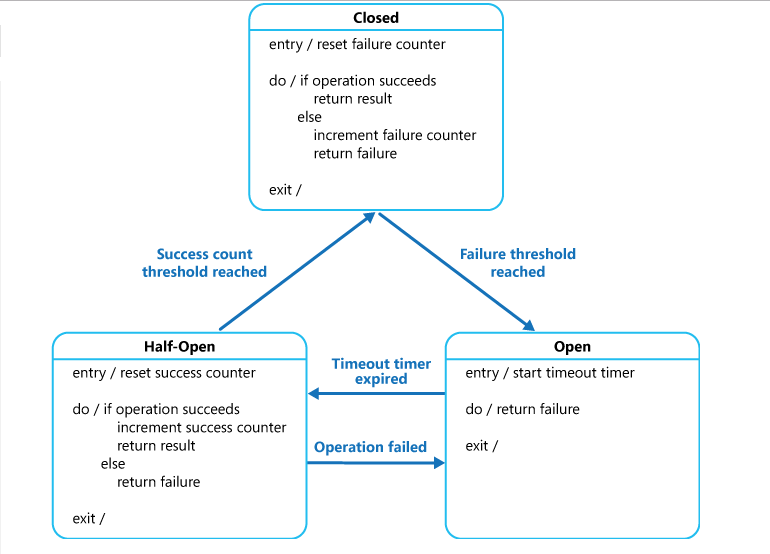
There may be situations where faults are due to unexpected events and may take much longer to rectify. In these situations it may be pointless for an application to continually retry performing an operation that is unlikely to succeed, and instead the application should quickly accept that the operation has failed and handle this failure accordingly.

Circuit Breaker is in one of the below three states at a time:

* Closed State- When the circuit breaker is in close state that means it is not active yet and will allow every request to pass through to that operation.
* Open State- Circuit breaker is in open state when the number of consecutive failures exceeds the threshold failure value within a given time period.

When Circuit breaker reaches Open state, a timer with ‘Timeout’ value will start after that Circuit breaker will go into Half-Open state. The purpose of this timer is to give the system time to rectify the problem that caused the failure before allowing the application to attempt to perform the operation again.

* Half-Open- There will be a single request or limited requests which will be allowed to go through to check whether the error has been rectified or not and that the operation is working properly. If it works, then Circuit breaker will go into Close state otherwise, it will go into Open state.



There are three breaker libraries which we can use in Golang-

1. Gobreaker
2. Hystrix
3. Circuit

Or, we can create our own implementation of breaker and then use it.

In my demo, I have created my own Circuit breaker implementation.

Source Code- <https://github.com/844nikhil/GO-Demos/blob/master/Circuit-breaker-pattern-go.zip>