

# **Error matchers in Ruby**

# Sunny



[github.com/sunny](https://github.com/sunny)

[twitter.com/sunfox](https://twitter.com/sunfox)

[sunfox.org](https://sunfox.org)



La place de marché  
dédiée à l'impression 3D.

[cults3d.com](https://cults3d.com)



**!FAIL!**

# Exceptions

```
def call(order)
  # do something...
rescue Net::OpenTimeout, Net::ReadTimeout
  try_again_later(order)
end
```

# Exceptions

```
def call(order)
  # do something...
rescue Net::OpenTimeout, Net::ReadTimeout
  try_again_later(order)
rescue Faraday::ConnectionFailed => error
  if error.message == "execution expired"
    try_again_later(order)
  end
end
```



```
module FaradayTimeoutErrorMatcher
  def self.===(error)
    error.is_a?(Faraday::ConnectionFailed) &&
      error.message == "execution expired"
  end
end
```





```
def call(order)
  # do something...
rescue Net::OpenTimeout,
       Net::ReadTimeout,
       FaradayTimeoutErrorMatcher
  try_again_later(order)
end
```

```
module NetworkTimeoutErrorMatcher
  def self.===(error)
    case error
    when Faraday::ConnectionFailed
      error.message == "execution expired"
    when Net::OpenTimeout,
         Net::ReadTimeout
      true
    end
  end
end
```



```
def call(order)
  # do something...
  rescue NetworkInterfaceError
  try_again_later(order)
end
```

# Retriable

---

```
def call(order)
  retrieable(on: NetworkTimeoutErrorMatcher) do
    # do something...
  end
end
```