

# Equal

Equalerror

```
// Equal reports whether current error `err` equals to error `target`.
// Please note that, in default comparison for `Error`,
// the errors are considered the same if both the `code` and `text` of
// them are the same.
func Equal(err, target error) bool
```

```
Equal(target error) bool
```

GoFrame



```
func ExampleEqual() {
    err1 := errors.New("permission denied")
    err2 := gerror.New("permission denied")
    err3 := gerror.NewCode(gcode.CodeNotAuthorized, "permission
denied")
    fmt.Println(gerror.Equal(err1, err2))
    fmt.Println(gerror.Equal(err2, err3))

    // Output:
    // true
    // false
}
```

# Is

Iserrorerrorerror

```
// Is reports whether current error `err` has error `target` in its
// chaining errors.
// It is just for implements for stdlib errors.Unwrap from Go version 1.17.
func Is(err, target error) bool
```

```
Is(target error) bool
```

GoFrame

## Content Menu

- [Equal](#)
  - 
  -
- [Is](#)
  - 
  -



errorsIsIs

```
func ExampleIs() {
    err1 := errors.New("permission denied")
    err2 := gerror.Wrap(err1, "operation failed")
    fmt.Println(gerror.Is(err1, err1))
    fmt.Println(gerror.Is(err2, err2))
    fmt.Println(gerror.Is(err2, err1))
    fmt.Println(gerror.Is(err1, err2))

    // Output:
    // false
    // true
    // true
    // false
}
```