-gutil

Content Menu

gutil

```
import "github.com/gogf/gf/v2/util/gutil"
```

https://pkg.go.dev/github.com/gogf/gf/v2/util/gutil

Dump

- Dumpvalues
-

  ```
  Dump(values ...interface{})
  ```

-

```
type User struct {
        Name string
        Age int
}

type Location struct {
        Province string
        City string
}

type UserInfo struct {
        U User
        L Location
}

func main() {
        userList := make([]UserInfo, 0)
        userList = append(userList, UserInfo{
                U: User{
                        Name: "",
                        Age:  18,
                },
                L: Location{
                        Province: "",
                        City:     "",
                },
        })
        userList = append(userList, UserInfo{
                U: User{
                        Name: "",
                        Age:  18,
                },
                L: Location{
                        Province: "",
                        City:     "",
                },
        })

        gutil.Dump(userList)
}

// Output:
[
    {
        U: {
            Name: "",
            Age:  18,
        },
        L: {
            Province: "",
            City:     "",
        },
    },
    {
        U: {
            Name: "",
            Age:  18,
        },
        L: {
            Province: "",
            City:     "",
        },
    },
]
```

DumpWithType

- `DumpWithType Dump`
-

```
DumpWithType(values ...interface{})
```

-

```
type User struct {
        Name string
        Age int
}

type Location struct {
        Province string
        City string
}

type UserInfo struct {
        U User
        L Location
}

func main() {
        userList := make([]UserInfo, 0)
        userList = append(userList, UserInfo{
                U: User{
                        Name: "",
                        Age:  18,
                },
                L: Location{
                        Province: "",
                        City:     "",
                },
        })
        userList = append(userList, UserInfo{
                U: User{
                        Name: "",
                        Age:  18,
                },
                L: Location{
                        Province: "",
                        City:     "",
                },
        })

        gutil.DumpWithType(userList)
}

// Output:
[]main.UserInfo(2) [
    main.UserInfo(2) {
        U: main.User(2) {
            Name: string(6) "",
            Age:  int(18),
        },
        L: main.Location(2) {
            Province: string(6) "",
            City:     string(6) "",
        },
    },
    main.UserInfo(2) {
        U: main.User(2) {
            Name: string(6) "",
            Age:  int(18),
        },
        L: main.Location(2) {
            Province: string(6) "",
            City:     string(6) "",
        },
    },
]
```

DumpTo

- DumpTovaluewrite
-

```
DumpTo(writer io.Writer, value interface{}, option DumpOption)
```

-
-

```
package main

import (
        "bytes"
        "fmt"
        "github.com/gogf/gf/v2/util/gutil"
        "io"
)

type UserInfo struct {
        Name     string
        Age      int
        Province string
        City     string
}

type DumpWriter struct {
        Content string
}

func (d *DumpWriter) Write(p []byte) (n int, err error) {
        buffer := bytes.NewBuffer(nil)
        buffer.WriteString("I'm Start!\n")
        buffer.WriteString(string(p))
        buffer.WriteString("\nI'm End!\n")

        d.Content = buffer.String()

        return buffer.Len(), nil
}

func main() {
        u := UserInfo{
                "a", 18, "b", "c",
        }

        var dw io.Writer = &DumpWriter{}

        gutil.DumpTo(dw, u, gutil.DumpOption{})

        fmt.Println(dw.(*DumpWriter).Content)
}

// Output:
I'm Start!
{
    Name:     "a",
    Age:      18,
    Province: "b",
    City:     "c",
}
I'm End!
```