

URL-gurl

URL

```
import "github.com/gogf/gf/v2/encoding/gurl"
```

<https://pkg.go.dev/github.com/gogf/gf/v2/encoding/gurl>

Content Menu

- [URL](#)
- [URL](#)
- [URL](#)

URL

```
package main

import (
    "fmt"
    "github.com/gogf/gf/v2/encoding/gurl"
    "net/url"
)

func main() {
    // url
    values := url.Values{}
    values.Add("name", "gopher")
    values.Add("limit", "20")
    values.Add("page", "7")

    // URL limit=20&name=gopher&page=7
    urlStr := gurl.BuildQuery(values)
    fmt.Println(urlStr)
}
```

```
limit=20&name=gopher&page=7
```

URL

```
package main

import (
    "fmt"
    "github.com/gogf/gf/v2/encoding/gurl"
    "log"
)

func main() {
    // URL
    encodeStr := gurl.Encode("limit=20&name=gopher&page=7")
    fmt.Println(encodeStr)

    // URL
    decodeStr, err := gurl.Decode("limit%3D20%26name%3Dgopher%26page%3D7")
    if err != nil {
        log.Fatal(err)
    }
    fmt.Println(decodeStr)
}
```

```
limit%3D20%26name%3Dgopher%26page%3D7
limit=20&name=gopher&page=7
```

URL

component :

-1	all
1	scheme
2	host
4	port
8	user
16	pass
32	path
64	query
128	fragment

```
package main

import (
    "fmt"
    "github.com/gogf/gf/v2/encoding/gurl"
    "log"
)

func main() {
    // URL
    data, err := gurl.ParseURL("http://127.0.0.1:8199/goods?
limit=20&name=gopher&page=7", -1)
    if err != nil {
        log.Fatal(err)
    }
    fmt.Println(data)
    fmt.Println(data["host"])
    fmt.Println(data["query"])
    fmt.Println(data["path"])
    fmt.Println(data["scheme"])
    fmt.Println(data["fragment"])
}
```

```
map[fragment: host:127.0.0.1 pass: path:/goods port:8199 query:
limit=20&name=gopher&page=7 scheme:http user:]
127.0.0.1
limit=20&name=gopher&page=7
/goods
http
```