

neonx-cli; neonx framework v2

Rg

# Current Problem

- Directory Structs
- Bff Interfaces
- Service Interfaces
- Freedom
- Package Manager
- Doc Generate
- Validate Generate
- Mock Server
- SQL Parser

# Solve ?



Figure 1:

Rg

neonx-cli; neonx framework v2

# Directory structs

```
xx_foo_system
  + services
  + bff
  + protocol
  + doc
  + vendor
  - go.mod
  - go.mod
  - README.md
```

# Services Directory

```
xx_foo_system
  + services
    + demo
      + config
        - config.go
        - config.yaml
        - config_test.yaml
      + impls
        + server.go
        + group
          - impl_change_permission_in_group.go
          - server.go
      + hook
        - hook.go
      - main.go
      - .k8s.yaml
      - k8s_test.yaml
```

# Bff Directory

```
+ bff
  + admin
    + codes
      - error_code.go
    + config
      - config.go
      - config.yml
      - config_test.yml
  + impls
    + group
      - impl_change_permission_in_group.go
      - types.go
    + department
      - impl_get_department_list.go
      - impl_login.go
  + router
```

```
+ protocol
  + demo
    - demo.proto
    - group.go
    - login.go
```

## Note

- +: Directory
- -: File

## Note

*structs uniq*

*not cross directory*



## Type

- s.i | service.interface
- s.i.r | service.interface.request
- s.i.r | service.interface.response
- b.i | bff.interface
- b.i.rt | bff.interface.request
- b.i.re | bff.interface.response

# Annotation

## Bff Annotation

### interface

```
// @type: b.i.rt
// @name: login
// @login: Y
// @page: xxxx / yyyy
// @uri: /api/admin/v1/login
// @des: xxxx
func LoginHandler(state *bff.State) {
}
```

*Option:* des

# Annotation

## Bff Annotation

### request

```
// @type: b.i.r
// @interface: LoginHandler
// @des: xxxx
type LoginRequest struct {
    GroupId      int64 `binding:"gte=1"` // User Group id
}
```

# Annotation

## Bff Annotation

### response

```
// @type: b.i.re  
// @interface: LoginHandler  
// @des: xxxx  
type LoginResponse struct {  
}
```

# Annotation

## Services Annotation

### interface

```
// @type: s.i  
// @path: demo  
// @des: Demo  
service Demo {  
    rpc Ping(PingRequest) returns (PingResponse);  
}
```

# Annotation

## Services Annotation

### PingRequest

```
// @type: s.i.rt  
// @des: Demo  
message PingRequest {}
```

# Annotation

## Services Annotation

### PingResponse

```
// @type: s.i.re  
message PingResponse {  
}
```

# Protocol

```
+ pb
  + demo
    - login.go
    - login.proto
  + foo
    - ping.proto
    - ping.go
```



- uri

# Datatabse Operate

- Create Table Mapping to Struct
- Add, Delete, Query, Update
- Split Page