

## -gen service

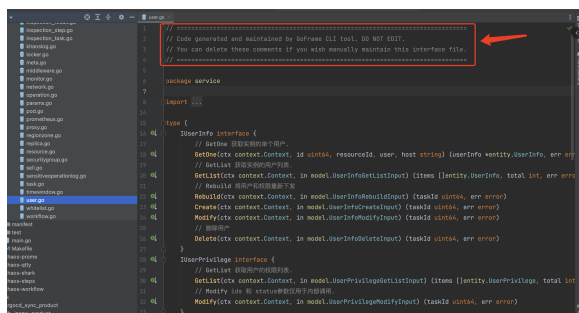
 v2.1

# GolangGolang

1. logiclogic
2. service
3. logicserviceservice

logicservice

servicelogicservice



logicservice



```
1. servicegologiclogic/xxx/*.gologic
2. service
```

<https://github.com/gogf/gf-demo-user>

```
gf gen service
```

## Content Menu

- ○
- ○
- ○
- ○
- - Goland /Idea
  - Visual Studio Code
- ○ Step1
- ○ Step2
- ○ Step3
- ○ Step4
- ○ Step5
- ○ Step6Start&Enjoy
- FAQ
  - logic

```
$ gf gen service -h
USAGE
  gf gen service [OPTION]

OPTION
  -s, --srcFolder      source folder path to be parsed. default:
internal/logic
  -d, --dstFolder      destination folder path storing automatically
generated go files. default: internal/service
  -f, --dstFileNameCase destination file name storing automatically
generated go files, cases are as follows:
                                | Case           | Example           |
                                |-----|-----|
                                | Lower          | anykindofstring  |
                                | Camel          | AnyKindOfString  |
                                | CamelLower     | anyKindOfString  |
                                | Snake          | any_kind_of_string|
default
                                | SnakeScreaming | ANY_KIND_OF_STRING|
                                | SnakeFirstUpper| rgb_code_md5      |
                                | Kebab          | any-kind-of-string|
                                | KebabScreaming | ANY-KIND-OF-STRING|
  -w, --watchFile      used in file watcher, it re-generates all
service go files only if given file is under
srcFolder
  -a, --stPattern       regular expression matching struct name for
generating service. default: ^s([A-Z]\\w+)$
  -p, --packages        produce go files only for given source packages
  -i, --importPrefix    custom import prefix to calculate import path
for generated importing go file of logic
  -l, --clear           delete all generated go files that are not
used any further
  -h, --help            more information about this command

EXAMPLE
  gf gen service
  gf gen service -f Snake
```

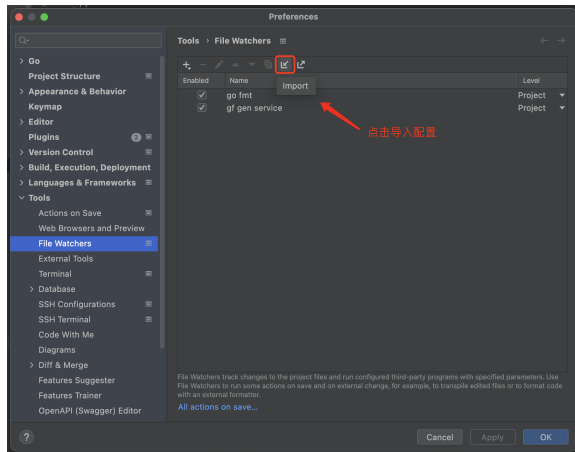


makemake service

srcFolder		internal/logic	logic						
dstFolder		internal/service							
dstFileNameCase		Snake							
stPattern		s([A-A]\w+)	s						
			<table><tr><td>logic</td><td>service</td></tr><tr><td>sUser</td><td>User</td></tr><tr><td>sMetaData</td><td>MetaData</td></tr></table>	logic	service	sUser	User	sMetaData	MetaData
			logic	service					
			sUser	User					
sMetaData	MetaData								
watchFile									
packages			JSON						
importPrefix									
overwrite		true							
clear		false	logic						

## Goland/Idea

GolandIDE[watchers.xml](#)



## Visual Studio Code

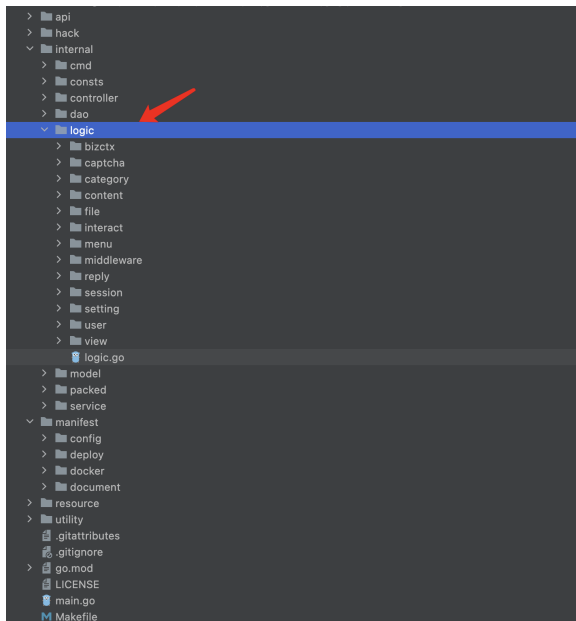
Visual Studio Code [RunOnSave](#)

```
"emeraldwalk.runonsave": {
  "commands": [
    {
      "match": ".*logic.*go",
      "isAsync": true,
      "cmd": "gf gen service"
    }
  ]
}
```

## Step1

Goland IDE[watchers.xml](#)

## Step2



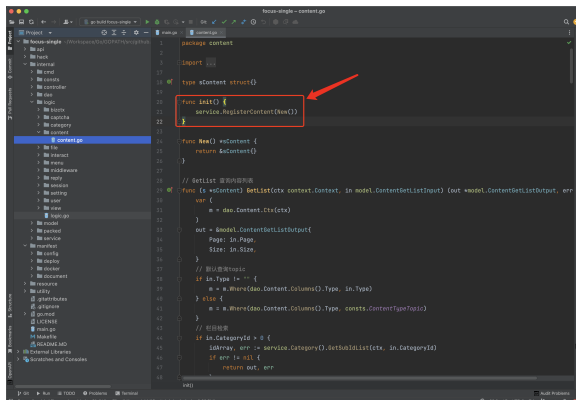
### Step3

Step1service

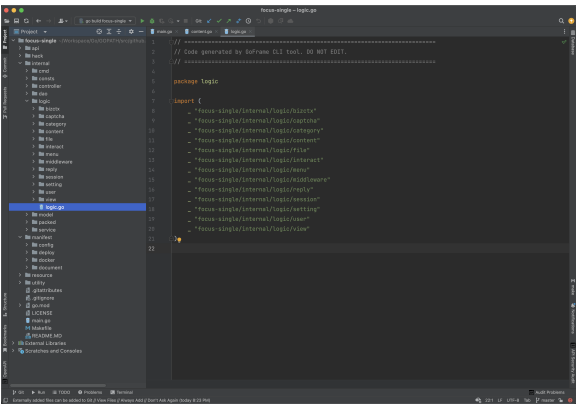
```
/logic gf gen service
```



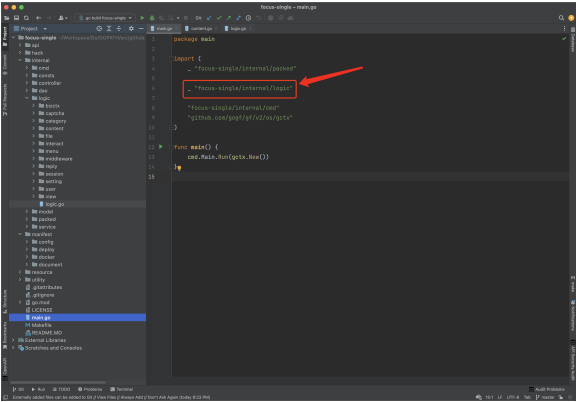
### Step4



Step5



mainimportpackedpacked



Step6Start&Enjoy

main.go

FAQ

logic

service

> Goland IDEGolandGoland

