代码生成工具v1.1开发

联系QQ：2816010068，加入会员群

二叉树出品

目录

- V1.0版本存在的问题
- Go template实战
- V1.1版本架构优化

V1.1版本存在的问题

- 代码写死，不好维护

```
    fmt.Fprintf(file, "var server = &controller.Server{}\n")
    fmt.Fprint(file, "\n\n")

    fmt.Fprintf(file, "var port= \":12345\"\n")
    fmt.Fprint(file, "\n\n")

    fmt.Fprintf(file, `

func main() {
    lis, err := net.Listen("tcp", port)
    if err != nil {
        log.Fatal("failed to listen: %v", err)
    }
    s := grpc.NewServer()
    hello.RegisterHelloServiceServer(s, server)
    s.Serve(lis)
}
```

V1.1版本存在的问题

```
29      fmt.Fprintf(file, "package main\n")
30      fmt.Fprintf(file, "import(\n")
31      fmt.Fprintf(file, `"net"`)
32      fmt.Fprintln(file)
33
34      fmt.Fprintf(file, `"log"`)
35      fmt.Fprintln(file)
36
37      fmt.Fprintf(file, `"google.golang.org/grpc"`)
38      fmt.Fprintln(file)
39
40      fmt.Fprintf(file, `"github.com/ibinarytree/koala/tools/koala/output/controller"`)
41      fmt.Fprintln(file)
42
43      fmt.Fprintf(file, `hello "github.com/ibinarytree/koala/tools/koala/output/generate"`)
44      fmt.Fprintln(file)
```

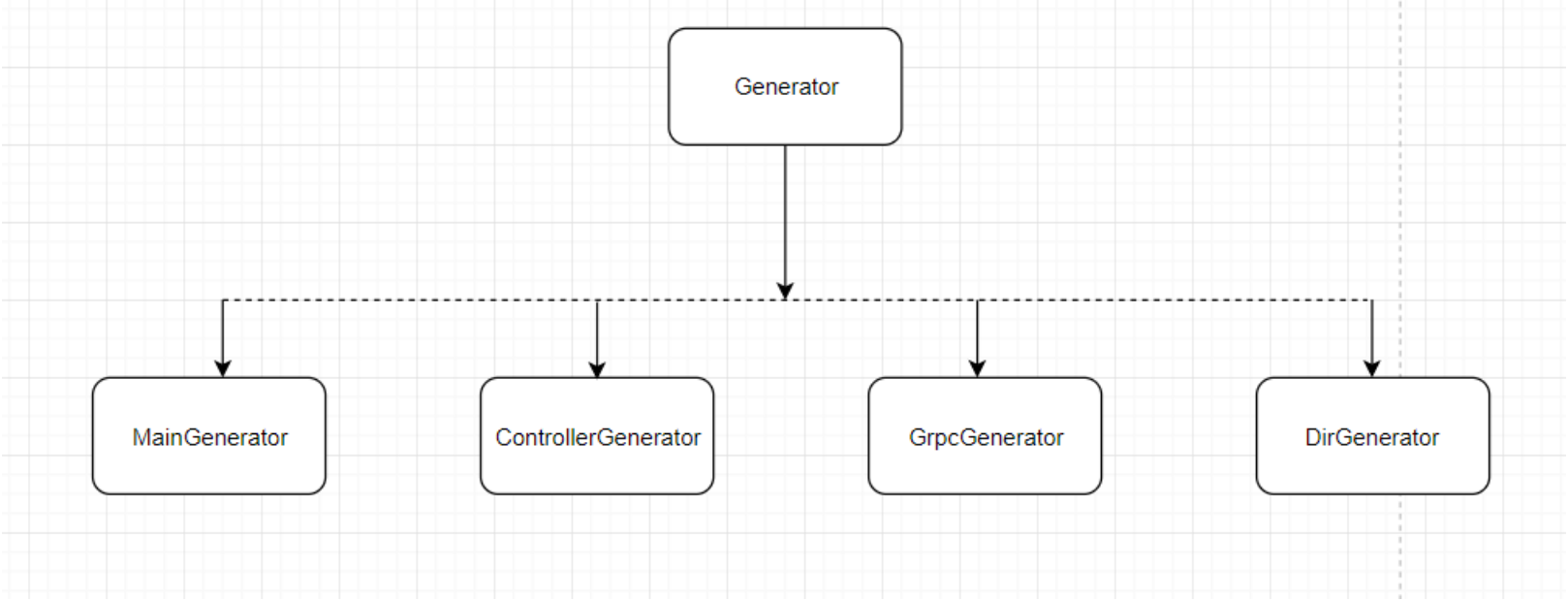
V1.1版本存在的问题

```
defer file.Close()
fmt.Fprintf(file, "package controller\n")
fmt.Fprintf(file, "import(\n")
fmt.Fprintf(file, `"context"`)
fmt.Fprintln(file)
fmt.Fprintf(file, `hello "github.com/ibinarytree/koala/tools/koala/output/generate"`)
fmt.Fprintln(file)
fmt.Fprintln(file, ")\n")
fmt.Fprintf(file, "type Server struct{}\n")
fmt.Fprint(file, "\n\n")

for _, rpc := range d.rpc {
    fmt.Fprintf(file,
        "func (s *Server) %s(ctx context.Context, r*hello.%s)(resp*hello.%s, err error){\nreturn\n
        rpc.Name, rpc.RequestType, rpc.ReturnsType)
}
```

V1.1版本存在的问题

- Generator 之间有依赖关系，执行的时候可能会出错



V1.1版本存在的问题

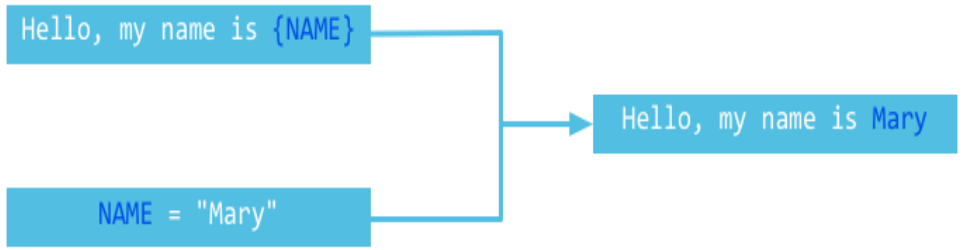
- Golang中的map是无序的，每次遍历都不一样

```
func (g *GeneratorMgr) Run(opt *Option) (err error) {

    for _, gen := range g.genMap {
        err = gen.Run(opt)
        if err != nil {
            return
        }
    }
    return
}
```

模板

- 模板替换
  - {{}}来包含需要在渲染时被替换的字段，{{.}}表示当前的对象
  - 通过{{.FieldName}}访问对象的属性



模板

- 模板替换

```
package main

import (
        "fmt"
        "os"
        "text/template"
)

type Person struct {
        Name string
        age  string
}

func main() {
        t, err := template.ParseFiles("./index.html")
        if err != nil {
                fmt.Println("parse file err:", err)
                return
        }
        p := Person{Name: "Mary", age: "31"}
        if err := t.Execute(os.Stdout, p); err != nil {
                fmt.Println("There was an error:", err.Error())
        }
}
```

模板

- If 判断

```
<html>
    <head>
    </head>
    <body>
        {{if gt .Age 18}}
        <p>hello, old man, {{.Name}}</p>
        {{else}}
        <p>hello,young man, {{.Name}}</p>
        {{end}}
    </body>
</html>
```

模板

- 条件

- not 非

```
{{if not .condition}}
{{end}}
```

- and 与

```
{{if and .condition1 .condition2}}
{{end}}
```

- or 或

```
{{if or .condition1 .condition2}}
{{end}}
```

- eq 等于

```
{{if eq .var1 .var2}}
{{end}}
```

- ne 不等于

```
{{if ne .var1 .var2}}
{{end}}
```

- lt 小于 (less than)

```
{{if lt .var1 .var2}}
{{end}}
```

- le 小于等于

```
{{if le .var1 .var2}}
{{end}}
```

- gt 大于

```
{{if gt .var1 .var2}}
{{end}}
```

- ge 大于等于

```
{{if ge .var1 .var2}}
{{end}}
```

模板

- With 语法

```
<html>
    <head>
    </head>
    <body>
        {{with .Name}}
        <p>hello, old man, {{.}}</p>
        {{end}}
    </body>
</html>
```

模板

- 循环

```
<html>
    <head>
    </head>
    <body>
        {{range .}}
            {{if gt .Age 18}}
            <p>hello, old man, {{.Name}}</p>
            {{else}}
            <p>hello,young man, {{.Name}}</p>
            {{end}}
        {{end}}
    </body>
</html>
```

V1.1版本架构优化

- 模板化
  - 把原来在分散在代码中的模板代码，抽象到模板文件中

```
29      fmt.Fprintf(file, "package main\n")
30      fmt.Fprintf(file, "import(\n")
31      fmt.Fprintf(file, `"net"`)
32      fmt.Fprintln(file)
33
34      fmt.Fprintf(file, `"log"`)
35      fmt.Fprintln(file)
36
37      fmt.Fprintf(file, `"google.golang.org/grpc"`)
38      fmt.Fprintln(file)
39
40      fmt.Fprintf(file, `"github.com/ibinarytree/koala/tools/koala/output/controller"`)
41      fmt.Fprintln(file)
42
43      fmt.Fprintf(file, `hello "github.com/ibinarytree/koala/tools/koala/output/generate"`)
44      fmt.Fprintln(file)
```

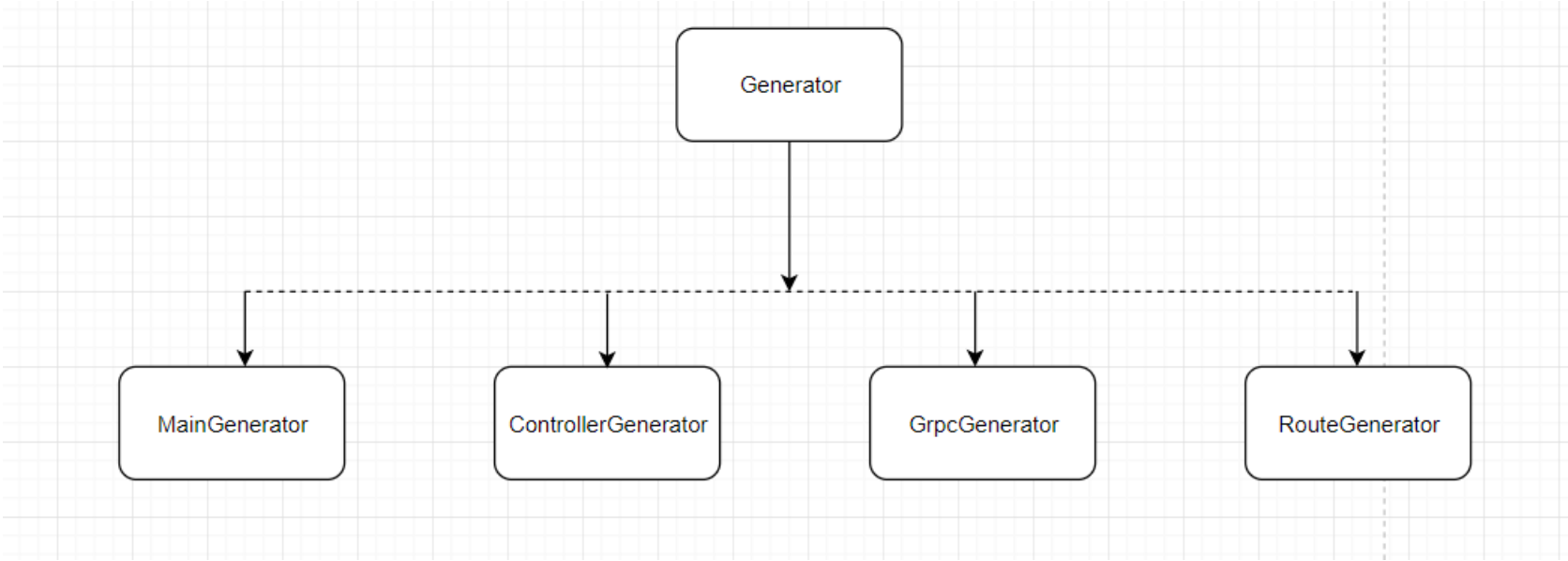
V1.1版本架构优化

- 接口重新定义

```
2
3   import (
4       "github.com/emicklei/proto"
5   )
6
7   type Generator interface {
8       Run(opt *Option, meta *ServiceMetaData) error
9   }
0
1   type ServiceMetaData struct {
2       service  *proto.Service
3       messages []*proto.Message
4       rpc      []*proto.RPC
5   }
6
```

V1.1版本架构优化

- DirGenerator去掉，在GeneratorMgr中进行预先创建好



V1.1版本架构优化

- 增加router模块
  - GRPC====》Router===》Controller
  - Router可以注入一些，打点统计，以及中间件

V1.1版本架构优化

- Controller改造
  - Controller每个方法，单独抽象一个类进行处理