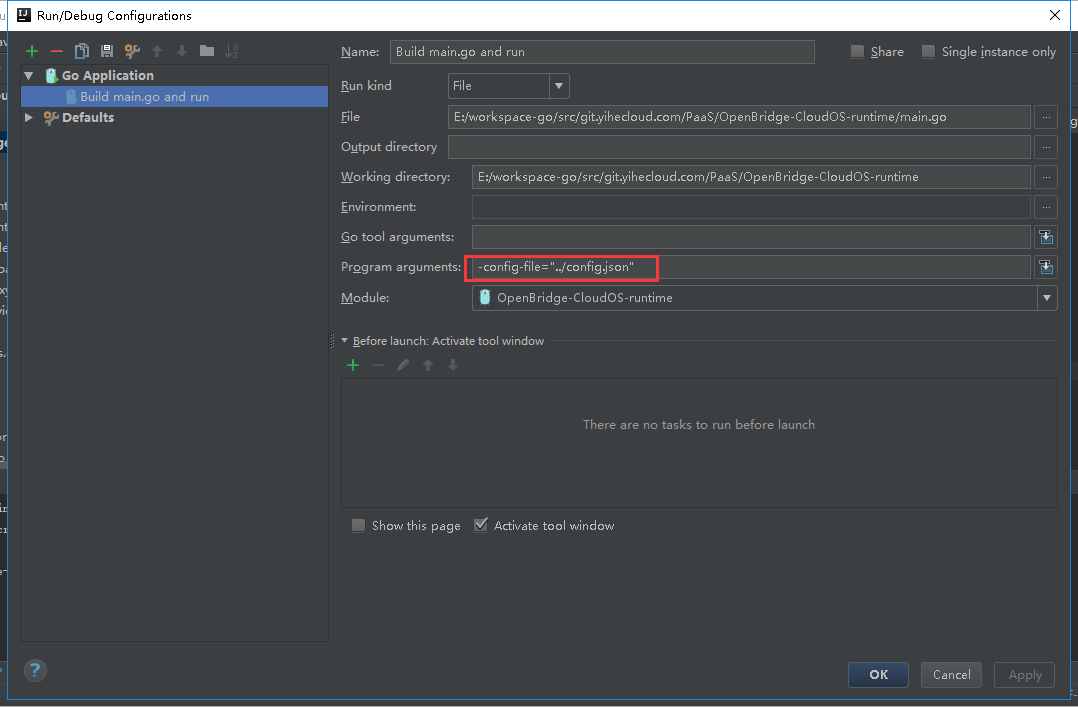
# IDEA配置

-config-file="../config.json"



# Prometheus编译

## 2017/5/26动态修改配置

E:\workspace\yh\OpenBridge-passos-proxy\open-falcon\src\prometheus\web\web.go

E:\workspace\yh\OpenBridge-passos-proxy\open-falcon\src\github.com\prometheus\prometheus\web\web.go

E:\workspace\yh\OpenBridge-passos-proxy\open-falcon\src\prometheus\vendor\github.com\prometheus\prometheus\web\web.go

E:\workspace\yh\OpenBridge-passos-proxy\open-falcon\src\github.com\prometheus\prometheus\vendor\github.com\prometheus\prometheus\web\web.go

router.Post("/config/:file", instrf("newConfig", h.newConfig))

**func** (h \*Handler) newConfig(w http.ResponseWriter, r \*http.Request) {  
 h.mtx.RLock()  
 **defer** h.mtx.RUnlock()  
  
 f := route.Param(route.Context(r), "file")  
 c, err := os.Create(f)  
 **if** err != nil {  
 log.Fatalf("writeLines: %s", err)  
 }  
 **defer** c.Close()  
  
 body, \_ := ioutil.ReadAll(r.Body)  
 log.Infoln(string(body))  
 c.Write(body)  
 c.Sync()  
 log.Infoln("rewrite config")  
}

## 2017/5/26动态修改配置

router.Post("/config/\*file", instrf("newConfig",h.newConfig))

**func** (h \*Handler) newConfig(w http.ResponseWriter, r \*http.Request) {  
 h.mtx.RLock()  
 **defer** h.mtx.RUnlock()  
  
 f := route.Param(route.Context(r), "file")  
 **if**(!strings.HasSuffix(f,"rules")&&!strings.HasSuffix(f,"yml")){  
 log.Warnf("is not config file!")  
 **return** }  
  
 **if**(strings.HasPrefix(f,"/")){  
 f = f[1:]  
 }  
  
 c, err := os.Create(f)  
 **if** err != nil {  
 log.Fatalf("writeLines: %s", err)  
 }  
 **defer** c.Close()  
  
 body, \_ := ioutil.ReadAll(r.Body)  
 log.Infoln(string(body))  
 c.Write(body)  
 c.Sync()  
 log.Infoln("rewrite config")  
}

### strings.HasPrefix

判断是否以某字符串打头/结尾  
strings.HasPrefix(s string, prefix string) bool => 对应python的str.startswith   
strings.HasSuffix(s string, suffix string) bool => 对应python的str.endswith

### Catch-All parameters

The second type are catch-all parameters and have the form \*name. Like the name suggests, they match everything. Therefore they must always be at the **end** of the pattern:

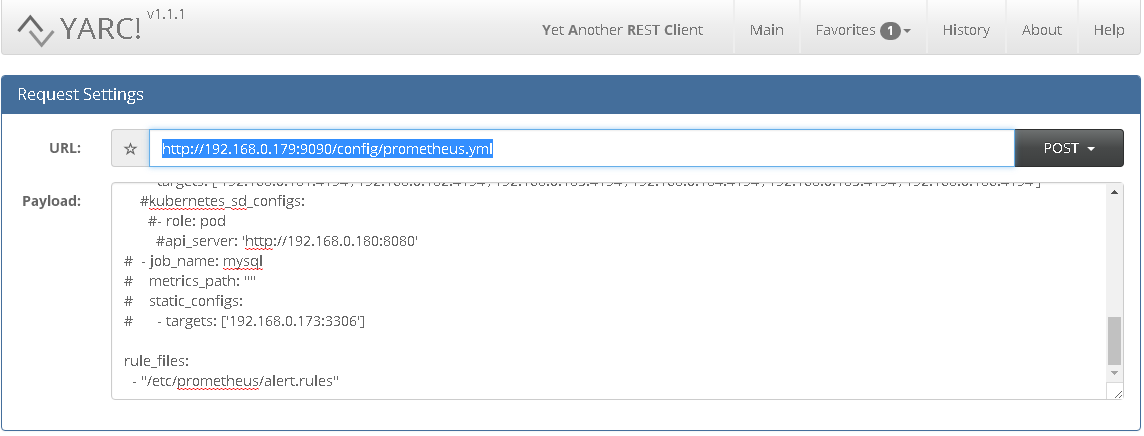
Pattern: /src/\*filepath

/src/ match

/src/somefile.go match

/src/subdir/somefile.go match

## Yet Another REST Client



http://192.168.0.179:9090/config/prometheus.yml

|  |
| --- |
| global:  scrape\_interval: 15s  evaluation\_interval: 15s  scrape\_configs:  # - job\_name: server  # static\_configs:  # - targets: ['192.168.0.179:9090']  - job\_name: federate  honor\_labels: true  metrics\_path: '/federate'  params:  match[]:  - '{\_\_name\_\_=~"^go.\*"}' # Request all job-level time series  - '{\_\_name\_\_=~"^c.\*"}'  static\_configs:  - targets:  - 192.168.0.180:9090  # - job\_name: linux  # static\_configs:  # - targets: ['192.168.0.179:9100']  # labels:  # instance: db1  - job\_name: container  static\_configs:  - targets: ['192.168.0.181:4194','192.168.0.182:4194','192.168.0.183:4194','192.168.0.184:4194','192.168.0.185:4194','192.168.0.186:4194']  #kubernetes\_sd\_configs:  #- role: pod  #api\_server: 'http://192.168.0.180:8080'  # - job\_name: mysql  # metrics\_path: ""  # static\_configs:  # - targets: ['192.168.0.173:3306']  rule\_files:  - "/etc/prometheus/alert.rules" |