敏捷工程实践与规范

廖强

业务对技术的要求

所有重复的都应该被自动化

标准API定义

swagger:meta

```
//      Schemes: http, https
//      Host: localhost
//      BasePath: /v2
//      Version: 0.0.1
//      License: MIT http://opensource.org/licenses/MIT
//      Contact: liaoqiang<liaoqiang@g7> http://www.g7.com.cn
//
//      Consumes:
//      - application/json
//      - application/xml
//
//      Produces:
//      - application/json
//      - application/xml
//
// swagger:meta
```

swagger:route

```
// swagger:route GET /pets pets listPets
//
// Lists pets filtered by some parameters.
//
// This will show all available pets by default.
// You can get the pets that are out of stock
//
//     Consumes:
//     - application/json
//     - application/x-protobuf
//
//     Produces:
//     - application/json
//     - application/x-protobuf
//
//     Schemes: http, https, ws, wss
//
//     Security:
//       api_key:
//       oauth: read, write
//
//     Responses:
//       - default: genericError
//       - 200: someResponse
//       - 422: validationError
```

swagger:params

```go
// swagger:parameters get_goods_by_id
type GetGoodsByIDReq struct {
    // 商品ID
    // in : path
    // Required : true
    GoodsID uint64 `json:"goods_id"`
    // Sku筛选审核状态，默认0不作为筛选条件，1-审核通过 2-审核未通过
    // in : query
    // Required : false
    SkuCheckState int8 `json:"sku_check_state"`
    // Sku筛选上架状态，默认0不作为筛选条件，1-上架 2-下架
    // in : query
    // Required : false
    SkuOnlineState int8 `json:"sku_online_state"`
}
```

swagger:response

```go
// A ValidationError is an error that is used when the required input fails validation.
// swagger:response validationError
type ValidationError struct {
    // The error message
    // in: body
    Body struct {
        // The validation message
        //
        // Required: true
        Message string
        // An optional field name to which this validation applies
        FieldName string
    }
}
```

swagger:model

```go
// User represents the user for this application
//
// A user is the security principal for this application.
// It's also used as one of main axes for reporting.
//
// swagger:model
type User struct {
    // the id for this user
    //
    // required: true
    // min: 1
    ID int64 `json:"id"`

    // the name for this user
    // required: true
    // min length: 3
    Name string `json:"name"`

    // the email address for this user
    //
    // required: true
    Email Email `json:"login"`

    // the friends for this user
    Friends []User `json:"friends"`
}
```

其他

- swagger:allOf
- swagger:strfmt
- swagger: security

- 重复写注释
- 代码与注释无法保证同步

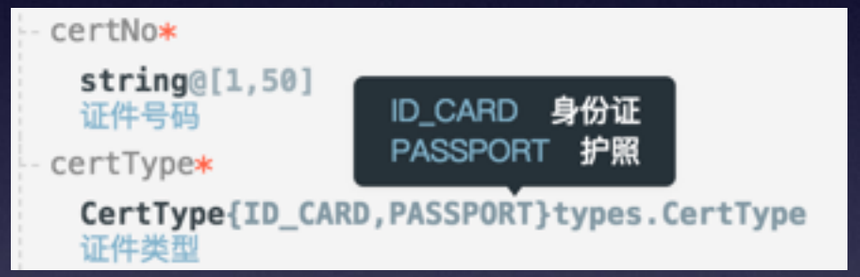
```
func main() {
    router := gin.Default()

    v1 := router.Group("/v1")
    {
        v1.POST("/login", loginEndpoint)
        v1.POST("/submit", submitEndpoint)
        v1.POST("/read", readEndpoint)
    }

    v2 := router.Group("/v2")
    {
        v2.POST("/login", loginEndpoint)
        v2.POST("/submit", submitEndpoint)
        v2.POST("/read", readEndpoint)
    }

    router.Run(":8080")
}
```

```
type GetGoodsByIDReq struct {
    // 商品ID
    GoodsID uint64 `json:"goods_id" in:"path" validate:"@uint64[1,]"`
    // Sku筛选审核状态，默认0不作为筛选条件，1-审核通过 2-审核未通过
    SkuCheckState int8 `json:"sku_check_state" in:"query" default:"0"`
    // Sku筛选上架状态，默认0不作为筛选条件，1-上架 2-下架
    SkuOnlineState int8 `json:"sku_online_state" in:"query" default:"0"`
}
```


```go
type ErrorField struct {
    // 出错字段路径
     // 这个信息为方便客户端定位错误原因
    // 例如输入中{"name":{ "alias" : "test"}} 中的alias出错 "name.alias"
    // 例如alias是数组，且第2个元素的a字段错误，则返回"name.alias[2].a"
    Field string `name:"field"`
    // 错误信息
    Msg string `name:"msg"`
    // 错误字段位置
    // body, query, header, path, formData
    In string `name:"in"`
}

type ErrorFields []*ErrorField

type StatusError struct {
    // 错误Key
    Key string `name:"key"`
    // 错误代码
    Code int64 `name:"code"`
    // 错误信息
    Msg string `name:"msg"`
    // 是否能作为错误话术
    CanDisplay bool `name:"canDisplay"`
    // 错误溯源
    Source []string `name:"source"`
    // 请求ID
    ID string `name:"id"`
    // 出错字段
    ErrorFields ErrorFields `name:"errorFields"`
}
```

枚举的可维护性

```go
// swagger:enum
type CertType uint8

// 个人证件类型
const (
    CERT_TYPE_UNKNOWN    CertType = iota
    CERT_TYPE__ID_CARD            // 身份证
    CERT_TYPE__PASSPORT           // 护照
)
```



输入 Operation ID 筛选

url

GetUrlOp
通过短链key查询原始url
GET

GetShortKeyOp
根据url查询短链地址
GET

CreateShortUrlOp
创建短链接
POST

GetUrlOp 通过短链key查询原始url
GET /r/{shortKey}

Parameters | URL Path | Header | Query | Cookie | FormData | Body

shortKey* 短链key

string@{1,15}

```
GET [redacted]/r/{shortKey} HTTP/1.0
Origin: [redacted]
Referer: [redacted]/STAGING/service-shorturl/url/GetUr
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_4) AppleWebKit/
```

Request

Responses

301

404 404032000 UrlNotFoundError
该url未创建短链
404032001 InvalidShortKeyError
非法的短链

StatusError{} application/json

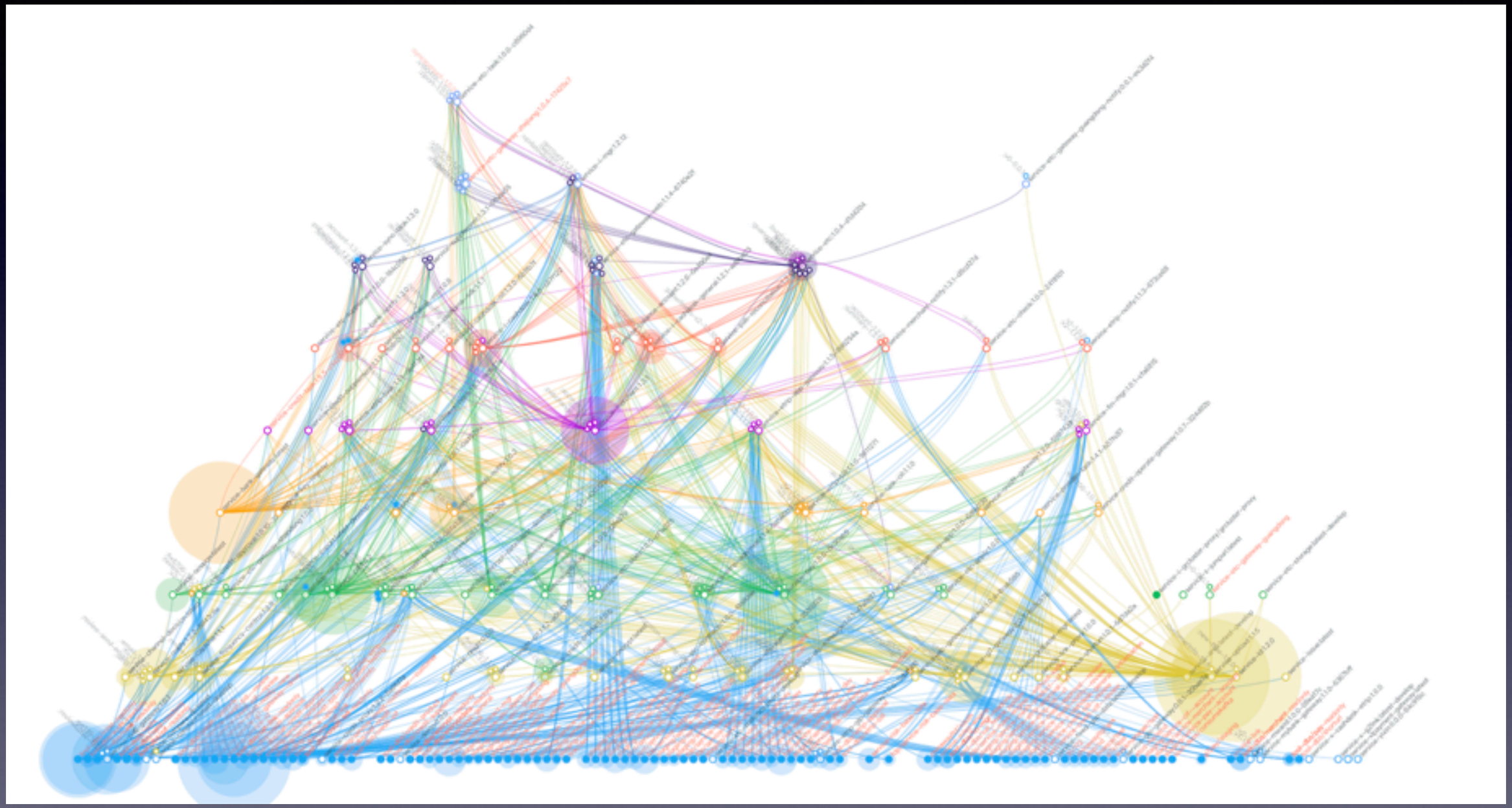
- canBeTalkError*
  - boolean
  - 是否能作为错误话术
- code*
  - integer(int64)
  - 错误代码
- desc*
  - string
  - 详细描述
- errorFields*
  - []ErrorField{}
    - field*
      - string
      - 出错字段路径
    - in*
      - string
      - 错误字段位置
    - msg*
      - string

数据库操作的重复性

```go
//go:generate tools gen model --with-comments --database DBContractMgr Contract
// @def primary ID
// @def unique_index I_from From FromID
type Contract struct {
    // 合同id
    ID uint64 `db:"F_id" json:"contractId,string" sql:"bigint unsigned NOT NULL"`
    // 合同来源
    From types.BusinessFrom `db:"F_from" json:"from" sql:"tinyint unsigned NOT NULL"`
    // 第三方的id
    FromID string `db:"F_from_id" json:"fromId" sql:"varchar(64) NOT NULL"`
    // 逻辑标志
    Enabled enumeration.Bool `db:"F_enabled" json:"-" sql:"tinyint unsigned NOT NULL"`
    // 操作时间
    presets.OperateTime
}
```

client代码的重复性和准确性

```
// go:generate tools gen client --spec-url http://xxxxx.com/user
// go:generate tools gen client --spec-url http://xxxxx.com/data
// go:generate tools gen client --spec-url http://xxxxx.com/pic
```

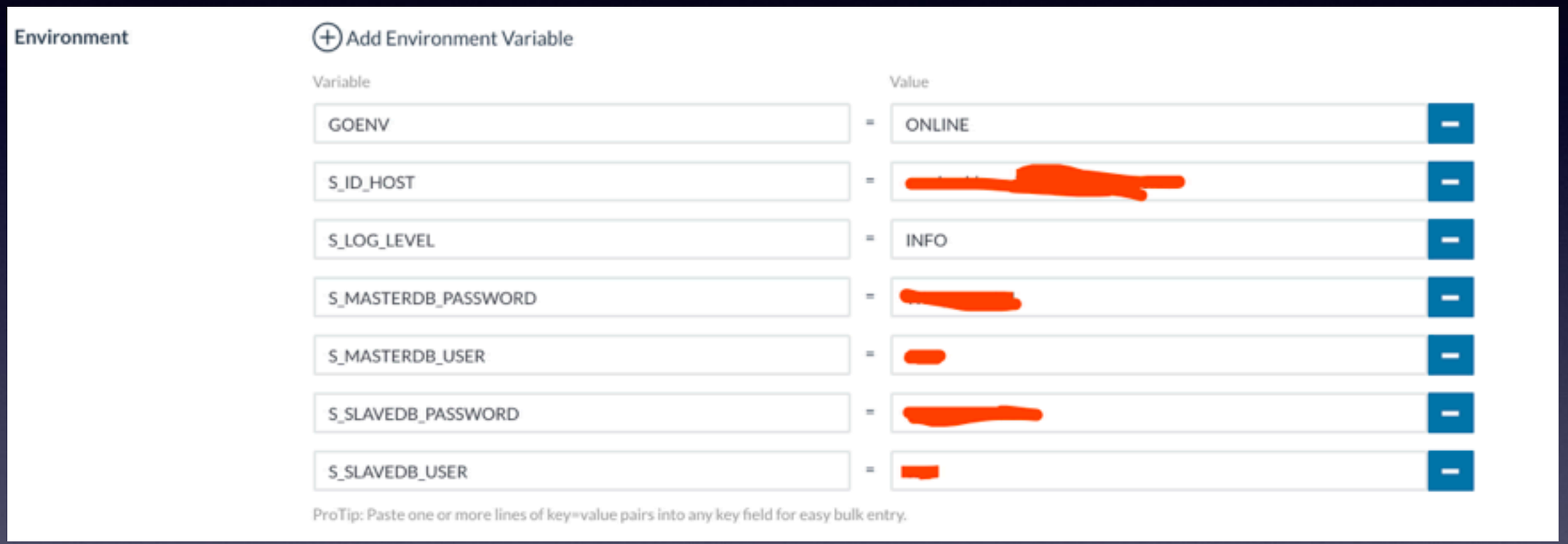


协议与序列化

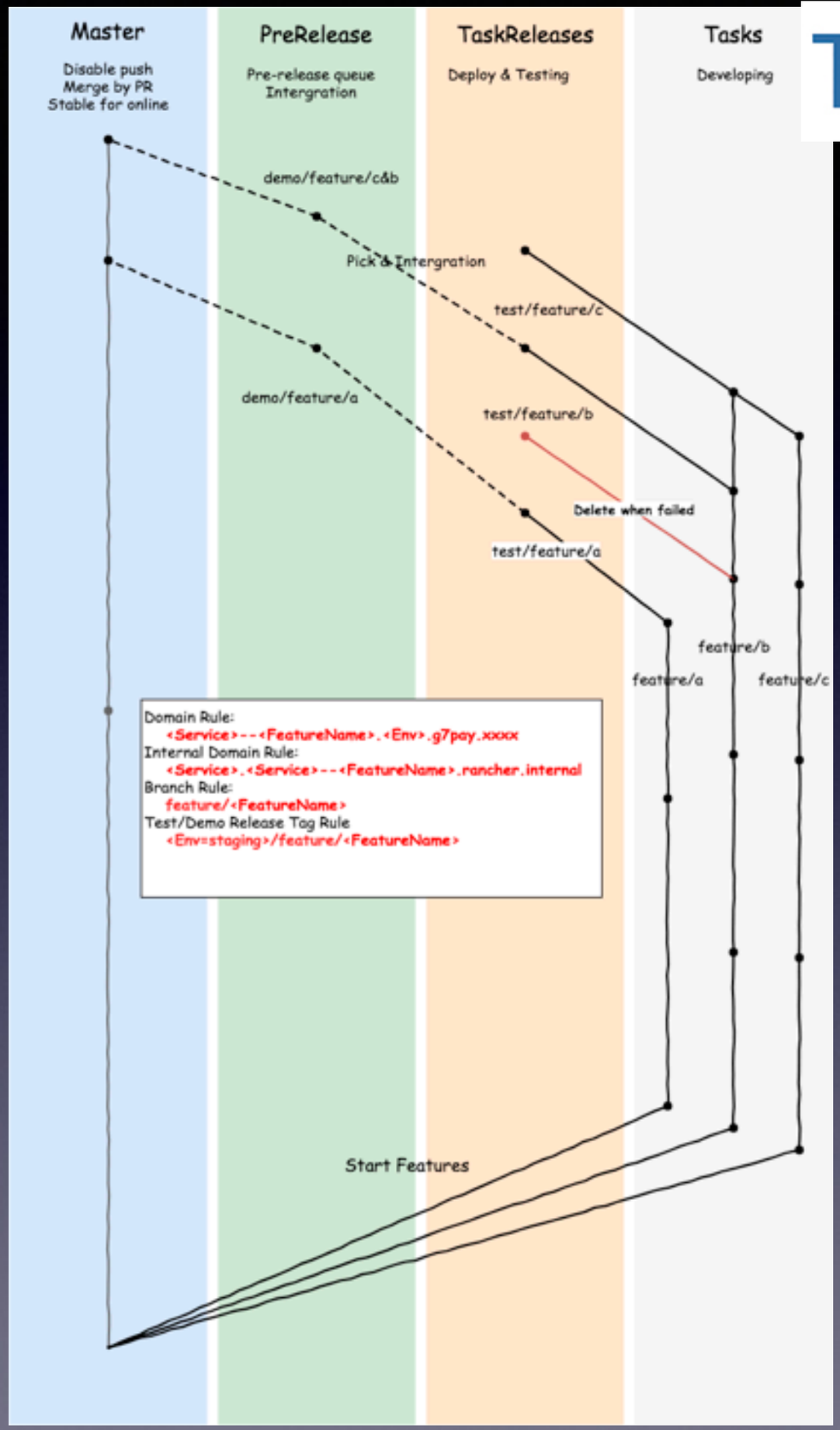
Mock

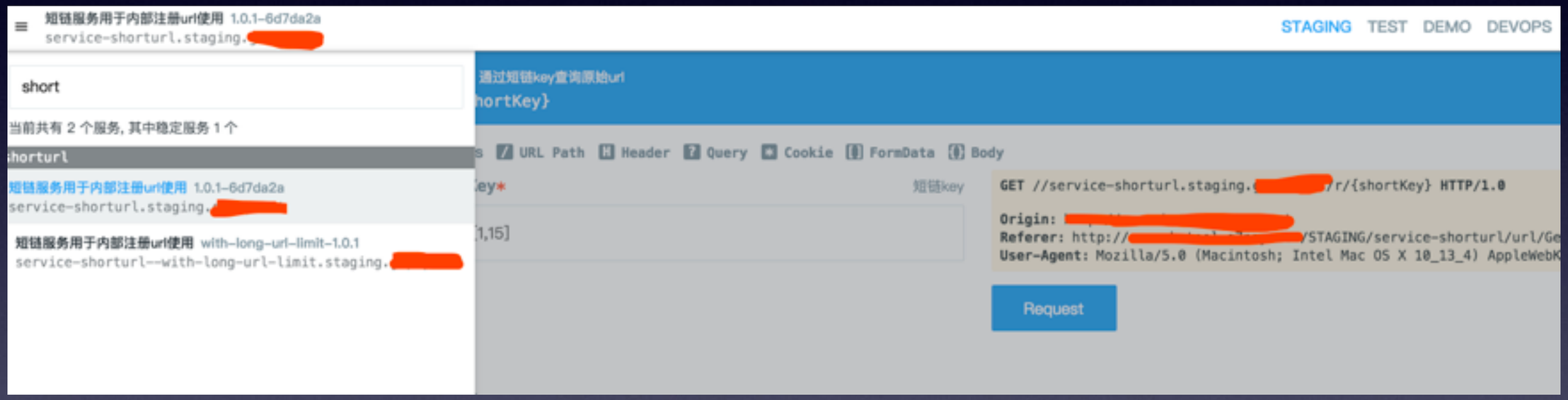
配置管理

```
var Config = struct {
    Log         *log.Log
    Server      transport_http.ServeHTT
    Upload      client_upload.ClientUpload `conf:"env"`
    MasterDB    mysql.MySQL
}{
    Log: &log.Log{
        Level: "DEBUG",
    },
    Server: transport_http.ServeHTTP{
        WithCORS: true,
        Port:     8000,
    },
    Upload: client_upload.ClientUpload{
        Client: client.Client{
            Host: “service-upload—customerinfo.xxx.com",
        },
    },
    MasterDB: mysql.MySQL{
        Name:     "service-contract-mgr",
        Host:     "staging.xxx.com",
        User:     "root",
        Password: "root",
    },
}
```

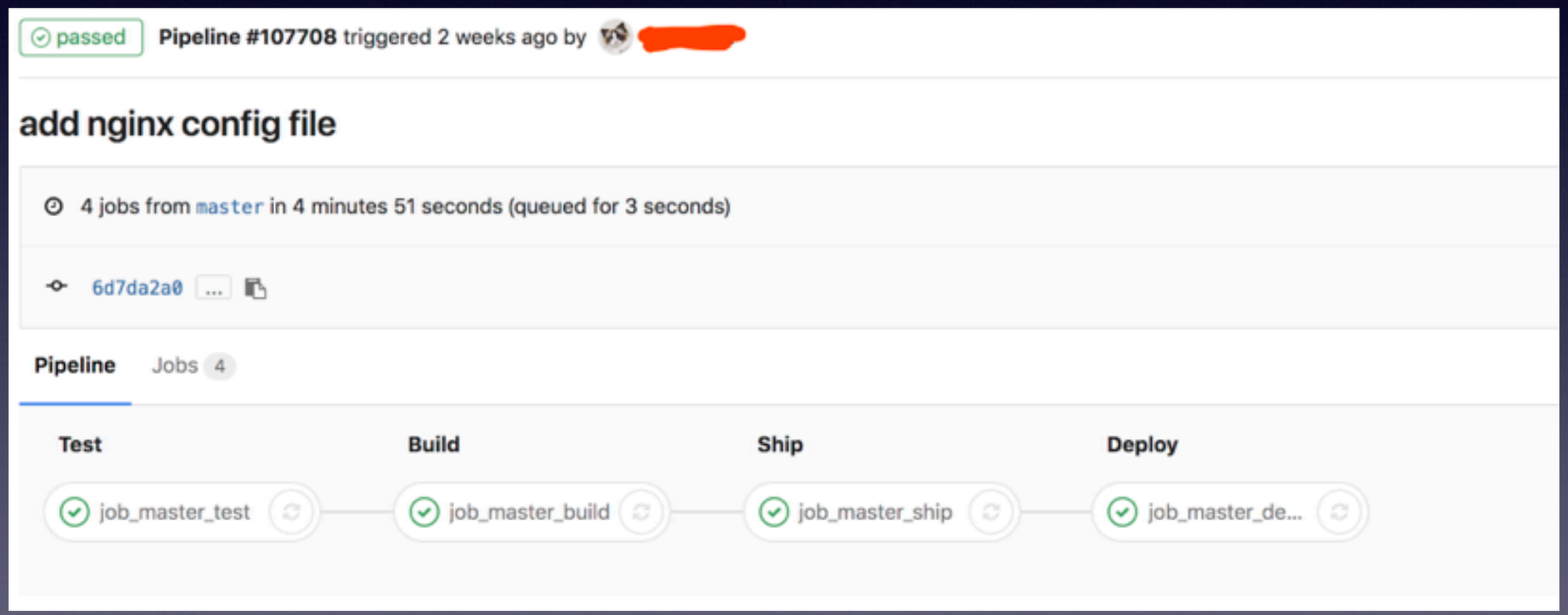


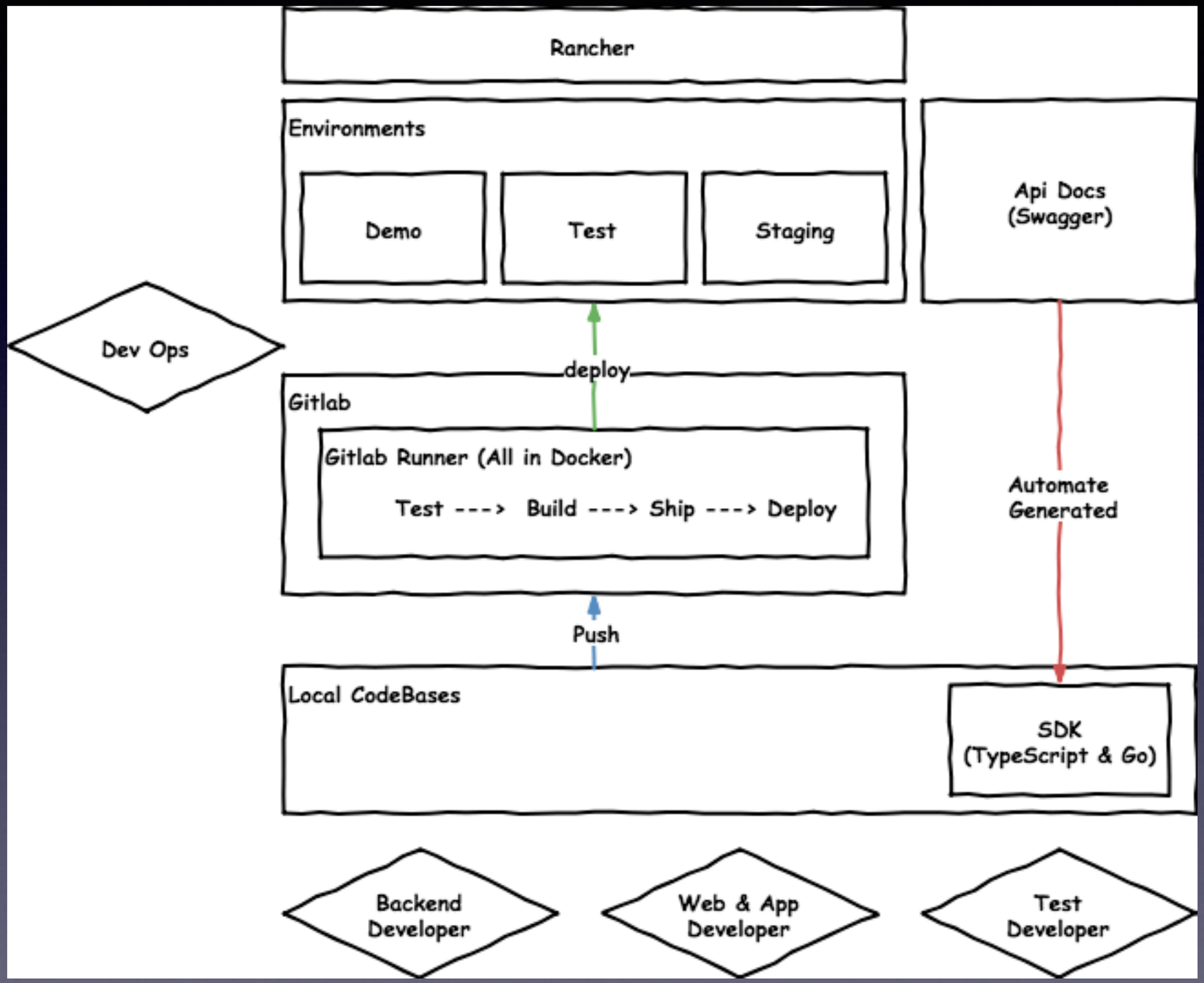
CI Flow


Master
Disable push
Merge by PR
Stable for online
PreRelease
Pre-release queue
Intergration
TaskReleases
Deploy & Testing
Tasks
Developing
demo/feature/c&b
Pick & Intergration
test/feature/c
demo/feature/a
test/feature/b
Delete when failed
test/feature/a
feature/b
feature/a
feature/c
Domain Rule:
<Service>--<FeatureName>.<Env>.g7pay.xxxx
Internal Domain Rule:
<Service>.<Service>--<FeatureName>.rancher.internal
Branch Rule:
feature/<FeatureName>
Test/Demo Release Tag Rule
<Env=staging>/feature/<FeatureName>
Start Features


短链服务用于内部注册url使用 1.0.1-6d7da2a
service-shorturl.staging.
STAGING TEST DEMO DEVOPS
short
当前共有 2 个服务, 其中稳定服务 1 个
horturl
短链服务用于内部注册url使用 1.0.1-6d7da2a
service-shorturl.staging.
短链服务用于内部注册url使用 with-long-url-limit-1.0.1
service-shorturl--with-long-url-limit.staging.
通过短链key查询原始url
hortKey}
URL Path
Header
Query
Cookie
FormData
Body
短链key
[1,15]
GET //service-shorturl.staging.
r/{shortKey} HTTP/1.0
Origin:
Referer: http://
/STAGING/service-shorturl/url/Ge
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_4) AppleWebK
Request

gitlab CI


passed
Pipeline #107708 triggered 2 weeks ago by
add nginx config file
4 jobs from master in 4 minutes 51 seconds (queued for 3 seconds)
6d7da2a0
...
Pipeline
Jobs 4
Test
Build
Ship
Deploy
job_master_test
job_master_build
job_master_ship
job_master_de...


Rancher
Environments
Demo
Test
Staging
Api Docs
(Swagger)
Dev Ops
deploy
Gitlab
Gitlab Runner (All in Docker)
Test ---> Build ---> Ship ---> Deploy
Automate
Generated
Push
Local CodeBases
SDK
(TypeScript & Go)
Backend
Developer
Web & App
Developer
Test
Developer

未来待做的

- 开发测试并行化

Think | IT大咖说 知识共享平台

谢谢