

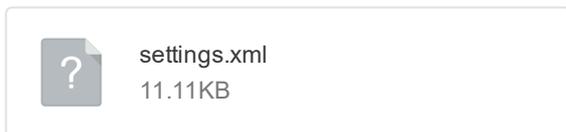
开发注意事项：

数据库使用的是mysql，地址是M1->57:3308，S1->57:3309

数据库中间件mycat->202:8066 root/root

maven:3.6.0 nexus: 57

maven 配置文件：



代码地址：

前端：<http://gitlab.mrbird.cc/FEBS-dev-org/FEBS-Cloud-Web>

后台：<http://gitlab.mrbird.cc/FEBS-dev-org/FEBS-Cloud>

代码管理使用的是gitea,需要先注册，注册之后需要加入到组织里才能看到代码，

gitlab.mrbird.cc 地址其实是：192.168.11.196，需要在

hosts (C:\Windows\System32\drivers\etc) 里添加记录：

192.168.11.196 gitlab.mrbird.cc

下载好代码后先切换到dev分支：

git checkout -b dev origin/dev //当你从远程库 clone 时，默认情况下，只能看到本地的 master 分支。要在 dev 分支上开发，就必须创建远程 origin 的 dev 分支到本地，于是用这个命令创建本地 dev 分支。

然后使用dev 分支，创建自己的分支，进行代码开发：

(1) git checkout dev //切换到 dev 分支

(2) git checkout -b dev_001 //然后创建并切换到开发者自身分支,这样新创建的分支是 dev 上的代码。

(3) 在 dev_001 上开发代码。

注意：

1,切换分支前需要add 并commit 代码，切换分支代码才自动变。

2,拉取代码：先切换到dev 分支 拉取远程dev 的代码，拉取下来后，再合并到自己的分支，不能在自己的分支拉取 dev的代码(不夸分支拉取代码)

本地启动需要配置一些项目环境变量：

febs-gateway=192.168.11.193;

mysql.url=192.168.11.202;

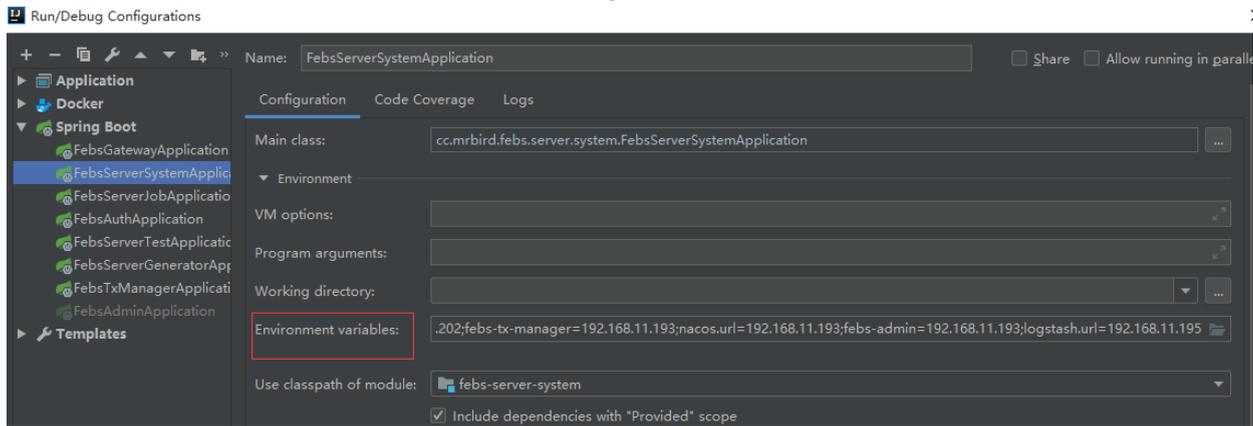
febs-tx-manager=192.168.11.193;

nacos.url=192.168.11.193;

febs-admin=192.168.11.193;
logstash.url=192.168.11.195;
mongo.url=192.168.11.195;
redis.url=192.168.11.57;

位置：（ 按需添加，可以看nacos的配置文件，nacos地址：

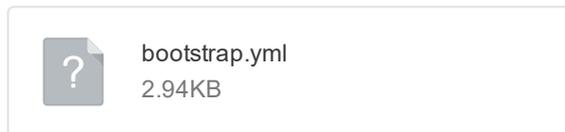
<http://192.168.11.193:8001/nacos/#/login> febs/123456)



多人开发，前后分离时候如何通过gateway指定调用某一个应用：

以febs-server-system 为例，首先使用自己的gateway：

其中的配置文件不使用nacos-config,而使用自己本地的：



```
- id: FEBS-Server-System
  uri: http://192.168.11.183:8201
  predicates:
    - Path=/system/**
  filters:
    - name: Hystrix
      args:
        name: systemfallback
        fallbackUri: forward:/fallback/FEBS-Server-System
```

这个ip是要调用的地址。你要调用谁的应用，就填谁的ip:port

然后CMD管理员执行：

```
route -p add 172.23.0.0 MASK 255.255.0.0 192.168.11.193 METRIC 3
```

即匹配到172.23的网络段就转发到192.168.11.193网络上。暂时解决，通过本地网关不能访问docker内应用的问题。

前端调用谁，就填谁的网关地址：

```
33 errors: true
34 },
35 proxy: {
36 // change xxx-api/login => mock/login
37 // detail: https://cli.vuejs.org/config/#devserver-proxy
38 [process.env.VUE_APP_BASE_API]: {
39 // target: 'http://api.mrbird.cc/',
40 target: 'http://192.168.11.193:8301/',
41 changeOrigin: true,
42 pathRewrite: {
43 ['^' + process.env.VUE_APP_BASE_API]: ''
44 }
45 }
46 }
47 },
48 configureWebpack: {
49 // provide the app's title in webpack's name field, so that
50 // it can be accessed in index.html to inject the correct title.
51 name: name,
52 resolve: {
```

顺便说下代码生成问题（目前只能生成单表，需要启动febs-server-generator）

基础配置：

基础配置 × 生成代码 ×

* 作者名称 MrBird * 基础包名 cc.mrbird.febs.server.generator.gen

* entity包名 entity * mapper包名 mapper

* mapperXml包名 mapper * service包名 service

* serviceImpl包名 service.impl * controller包名 controller

是否去除表前缀 是 否 表前缀 t_

修改 重置

其实基础配置一般只需要修改 作者名称，表前缀，和是否去除前缀。根据需要而定。

然后是代码生成页面：

| 表名 | 备注 | 数据量 (行) | 创建时间 | 更新时间 | 操作 |
|------------------------|-------------|---------|---------------------|---------------------|-------------------|
| t_dict | 字典表 | 2 | 2020-11-03 14:55:29 | 2020-11-05 11:10:04 | ↓ |
| t_dept_position | | 0 | 2020-11-03 14:55:29 | | ↓ |
| t_dept | 部门表 | 7 | 2020-11-03 14:55:29 | 2020-11-03 14:55:30 | ↓ |
| t_data_permission_test | 用户权限测试 | 8 | 2020-11-03 14:55:29 | 2020-11-03 14:55:30 | ↓ |
| oauth_client_details | 客户端配置表 | 3 | 2020-11-03 14:55:29 | 2020-11-03 14:55:30 | ↓ |
| t_export | Excel导入导出测试 | 32 | 2020-11-03 14:55:30 | 2020-11-03 14:55:30 | ↓ |
| t_dict_item | 字典项 | 4 | 2020-11-03 14:55:30 | 2020-11-03 14:55:31 | ↓ |
| t_job | 定时任务表 | 4 | 2020-11-03 14:55:30 | 2020-11-03 14:55:31 | ↓ |
| t_generator_config | 代码生成配置表 | 1 | 2020-11-03 14:55:30 | 2020-11-05 14:48:49 | ↓ |
| t_login_log | 登录日志表 | 186 | 2020-11-03 14:55:31 | 2020-11-03 14:55:31 | ↓ |

选择对应的数据库，对应的表就出来了，点击下载即可。然后将代码复制到项目里即可。