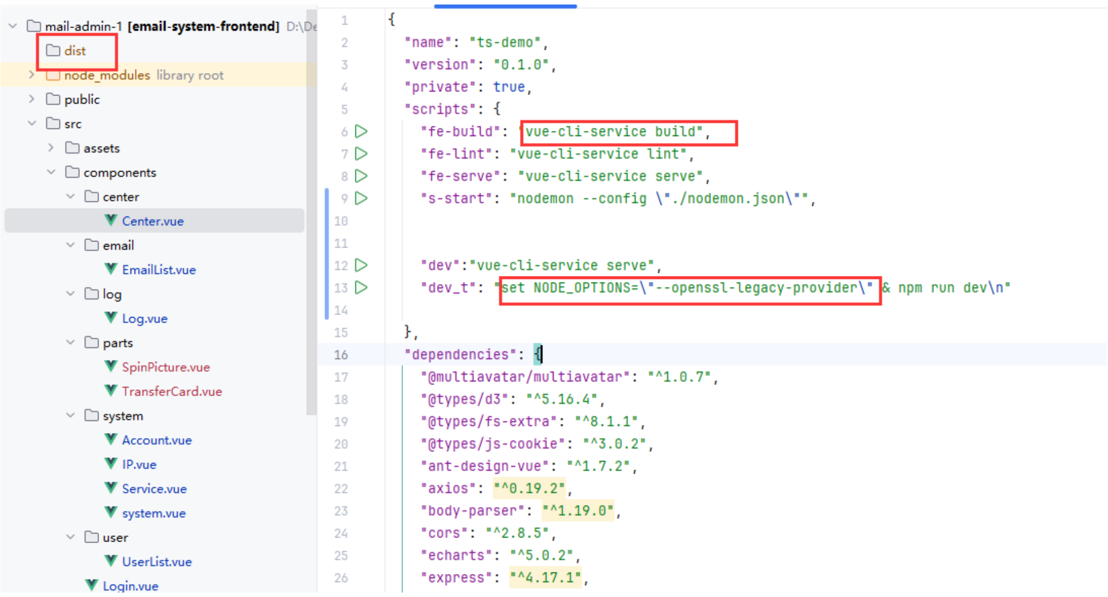
1.WEB 端

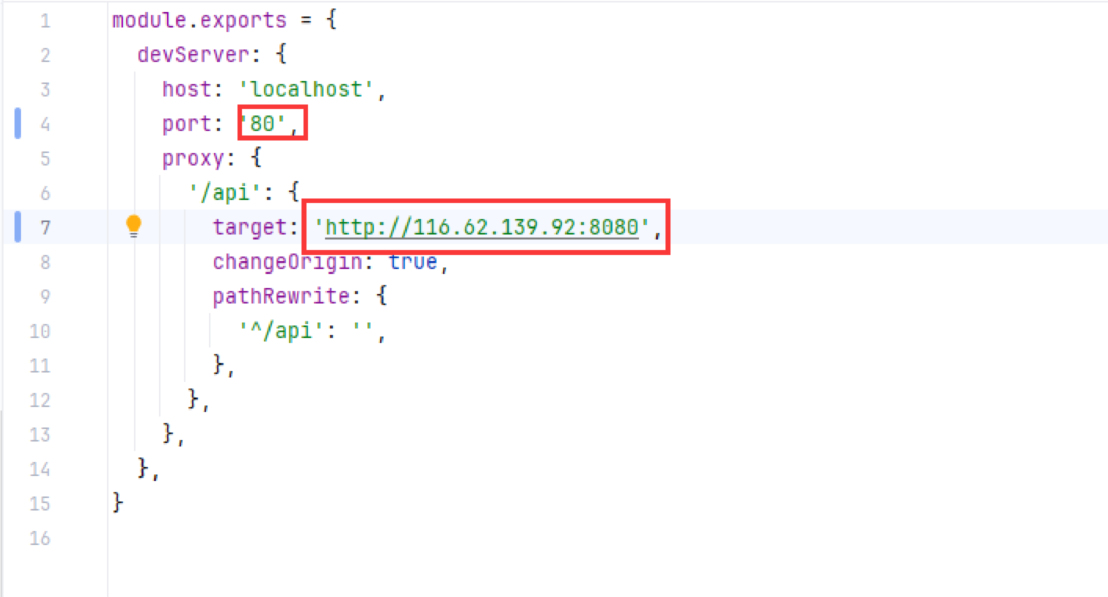
npm install 下载项目所用的依赖。
npm run fe-build 构建静态资源。



```
mail-admin-1 [email-system-frontend] D:\De
  dist
  node_modules library root
  public
  src
    assets
    components
      center
        Center.vue
      email
        EmailList.vue
      log
        Log.vue
      parts
        SpinPicture.vue
        TransferCard.vue
      system
        Account.vue
        IP.vue
        Service.vue
        system.vue
      user
        UserList.vue
      Login.vue
```

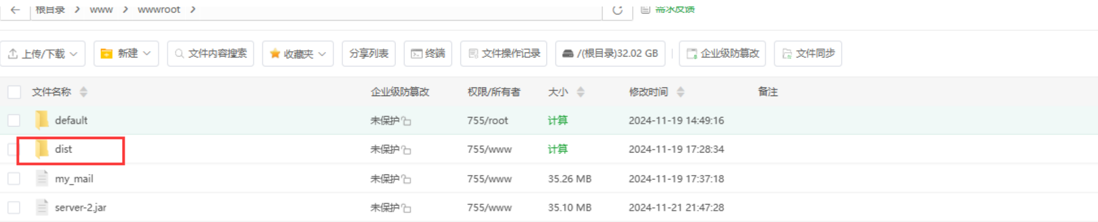
```
{
  "name": "ts-demo",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "fe-build": "vue-cli-service build",
    "fe-lint": "vue-cli-service lint",
    "fe-serve": "vue-cli-service serve",
    "s-start": "nodemon --config \"./nodemon.json\"",


    "dev":"vue-cli-service serve",
    "dev_t": "set NODE_OPTIONS=\"--openssl-legacy-provider\" & npm run dev\n"

  },
  "dependencies": {
    "@multiavatar/multiavatar": "^1.0.7",
    "@types/d3": "^5.16.4",
    "@types/fs-extra": "^8.1.1",
    "@types/js-cookie": "^3.0.2",
    "ant-design-vue": "^1.7.2",
    "axios": "^0.19.2",
    "body-parser": "^1.19.0",
    "cors": "^2.8.5",
    "echarts": "^5.0.2",
    "express": "^4.17.1",
```

注意这里要配置前端程序的端口号为 80，target 为端服务器 ip，已经对应端口。



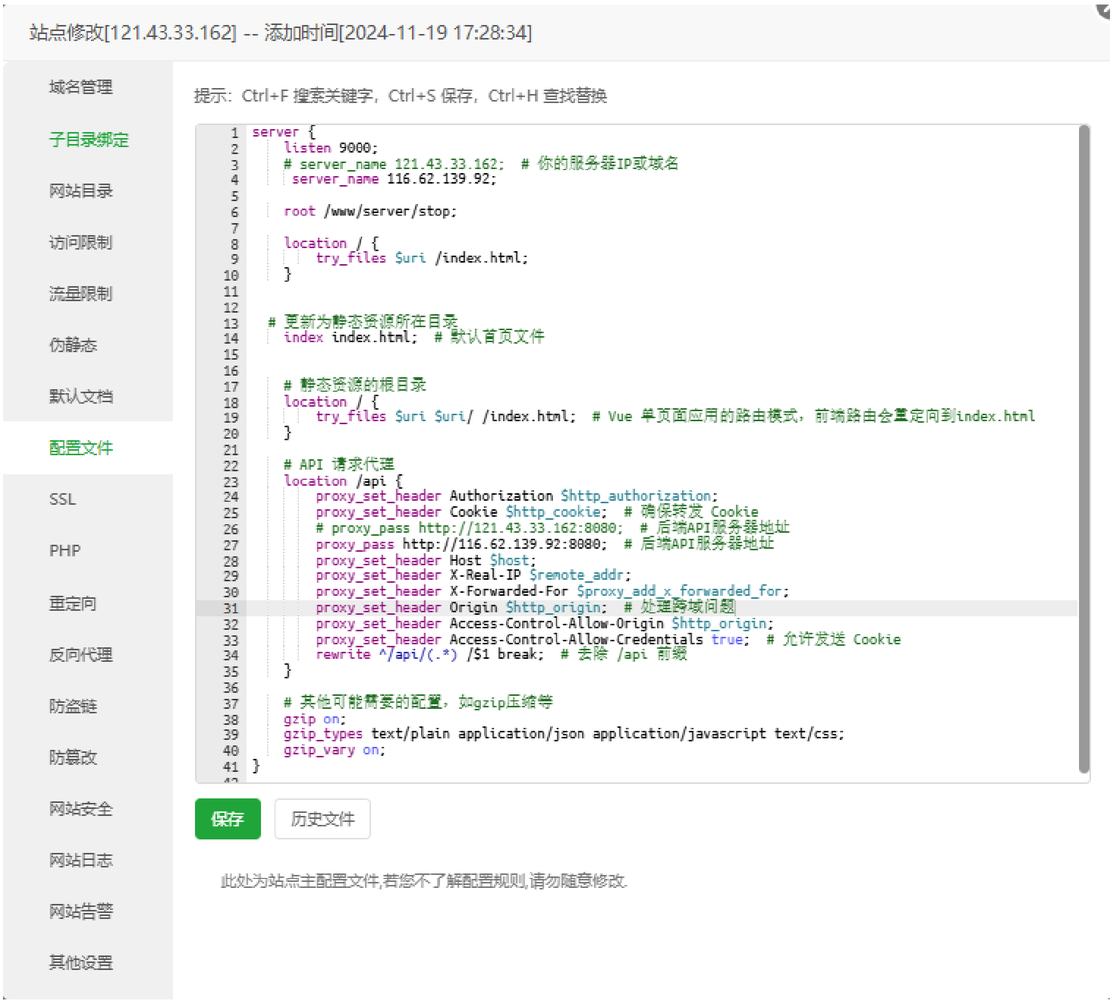
```
module.exports = {
  devServer: {
    host: 'localhost',
    port: '80',
    proxy: {
      '/api': {
        target: 'http://116.62.139.92:8080',
        changeOrigin: true,
        pathRewrite: {
          '^/api': '',
        },
      },
    },
  },
}

```

把 dist 文件夹上传到服务器上



文件名称	企业级防篡改	权限/所有者	大小	修改时间	备注
default	未保护	755/root	计算	2024-11-19 14:49:16	
dist	未保护	755/www	计算	2024-11-19 17:28:34	
my_mail	未保护	755/www	35.26 MB	2024-11-19 17:37:18	
server-2.jar	未保护	755/www	35.10 MB	2024-11-21 21:47:28	

添加 PHP 项目并运行，以下是配置文件。



站点修改[121.43.33.162] -- 添加时间[2024-11-19 17:28:34]

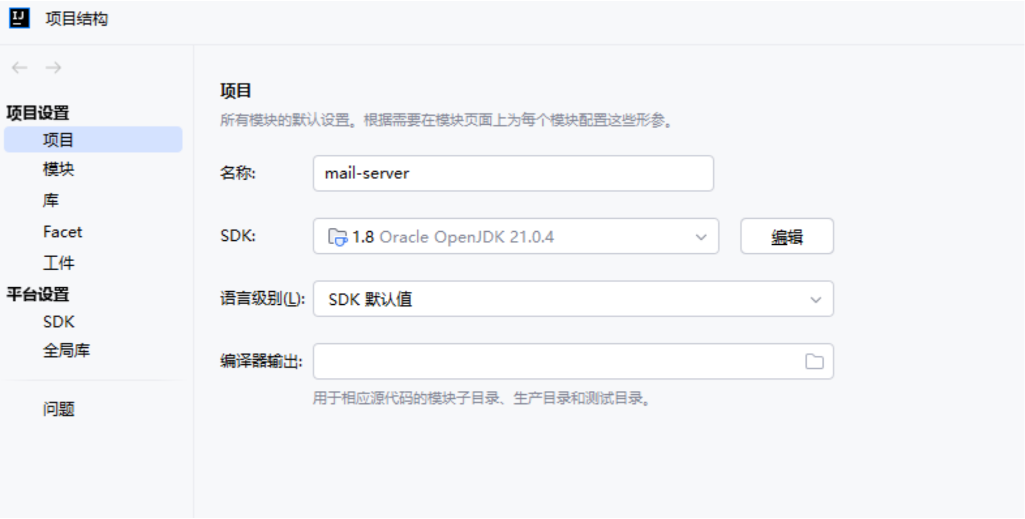
提示：Ctrl+F 搜索关键字，Ctrl+S 保存，Ctrl+H 查找替换

```
server {
    listen 9000;
    # server_name 121.43.33.162;  # 你的服务器IP或域名
     server_name 116.62.139.92;

    root /www/server/stop;

    location / {
        try_files $uri /index.html;
    }


  # 更新为静态资源所在目录
    index index.html;  # 默认首页文件


    # 静态资源的根目录
    location / {
        try_files $uri $uri/ /index.html;  # Vue 单页面应用的路由模式，前端路由会重定向到index.html
    }

    # API 请求代理
    location /api {
        proxy_set_header Authorization $http_authorization;
        proxy_set_header Cookie $http_cookie;  # 确保转发 Cookie
        # proxy_pass http://121.43.33.162:8080;  # 后端API服务器地址
        proxy_pass http://116.62.139.92:8080;  # 后端API服务器地址
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Origin $http_origin;  # 处理跨域问题
        proxy_set_header Access-Control-Allow-Origin $http_origin;
        proxy_set_header Access-Control-Allow-Credentials true;  # 允许发送 Cookie
        rewrite ^/api/(.*) /$1 break;  # 去除 /api 前缀
    }

    # 其他可能需要的配置，如gzip压缩等
    gzip on;
    gzip_types text/plain application/json application/javascript text/css;
    gzip_vary on;
}
```

此处为站点主配置文件,若您不了解配置规则,请勿随意修改.

2.服务器端

首先配置本地环境，在本地配置好之后，packages 成 jar 包，放到云端启动。
JDK 选择 1.8 版本。



项目

所有模块的默认设置。根据需要在模块页面上为每个模块配置这些形参。

名称: mail-server

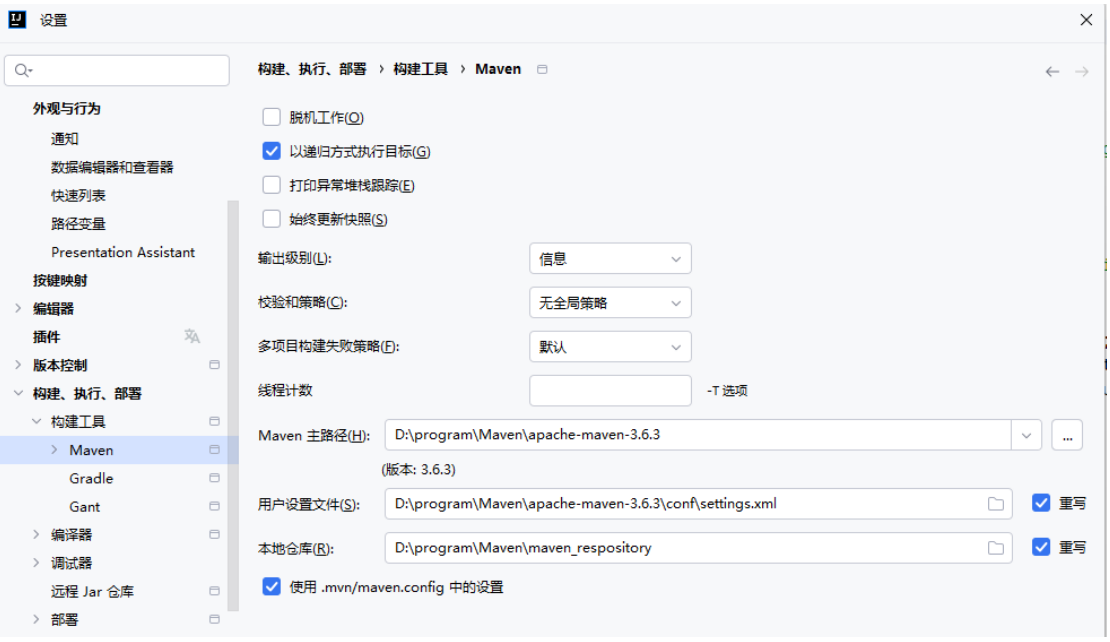
SDK: 1.8 Oracle OpenJDK 21.0.4

语言级别(L): SDK 默认值

编译器输出:

用于相应源代码的模块子目录、生产目录和测试目录。

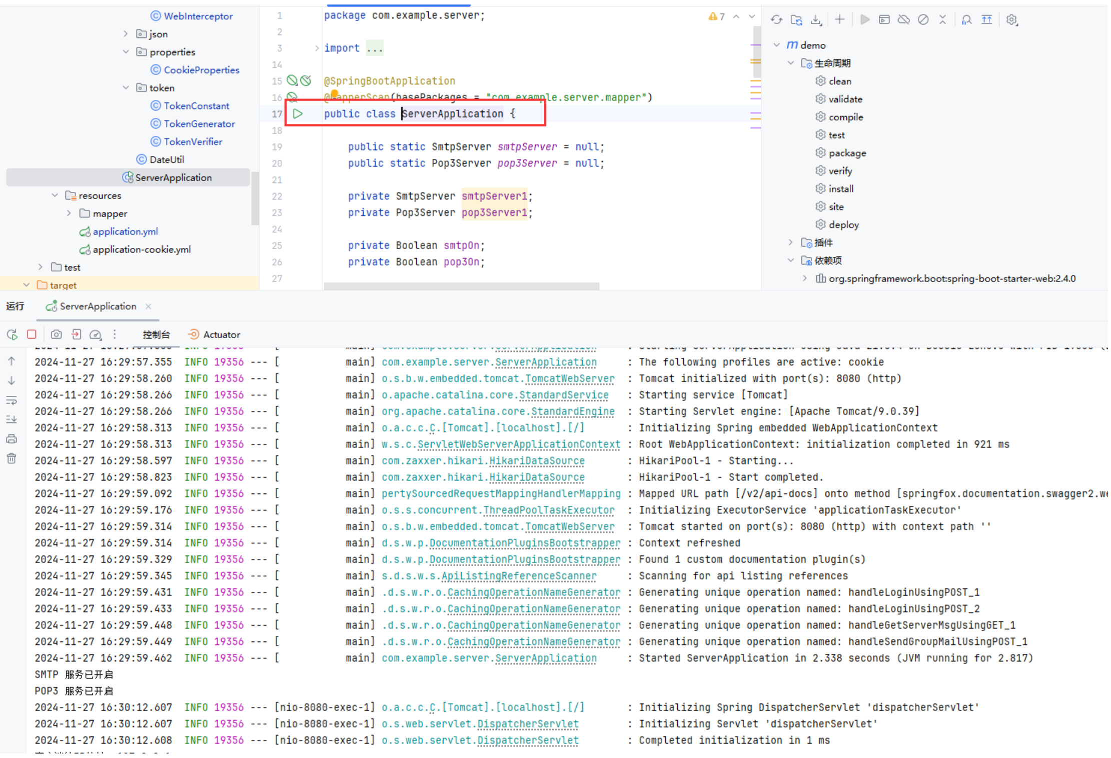
maven 选择自己本地的 maven 仓库，刷新后会自动下载。



□ 脱机工作(O)
☑ 以递归方式执行目标(G)
□ 打印异常堆栈跟踪(E)
□ 始终更新快照(S)

输出级别(L): 信息
校验和策略(C): 无全局策略
多项目构建失败策略(F): 默认
线程计数 -T 选项
Maven 主路径(H): D:\program\Maven\apache-maven-3.6.3
(版本: 3.6.3)
用户设置文件(S): D:\program\Maven\apache-maven-3.6.3\conf\settings.xml ☑ 重写
本地仓库(R): D:\program\Maven\maven_respository ☑ 重写
☑ 使用 .mvn/maven.config 中的设置

运行 ServerApplication 类，可以看见能够正常运行

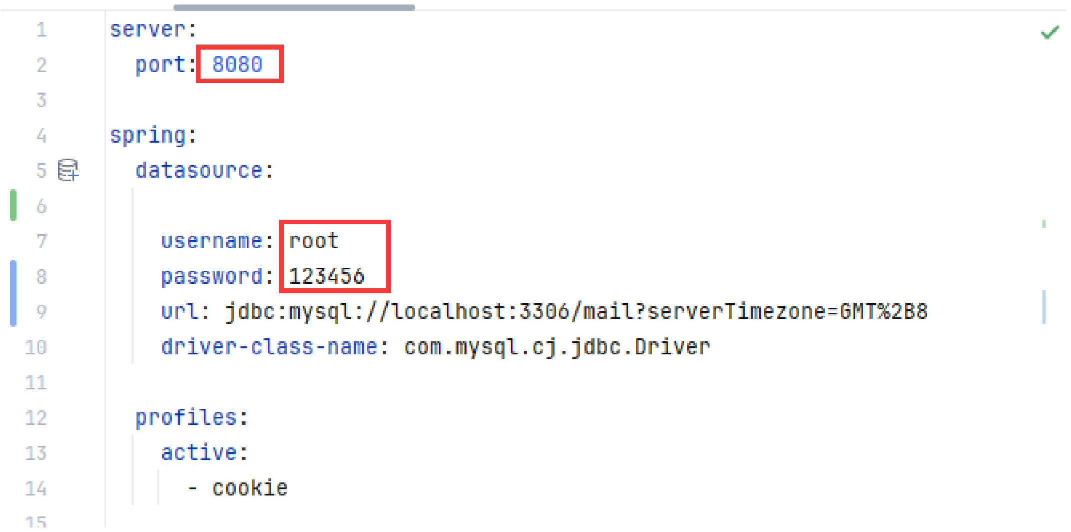


```
package com.example.server;

import ...

@SpringBootApplication
@MapperScan(basePackages = "com.example.server.mapper")
public class ServerApplication {

    public static SmtpServer smtpServer = null;
    public static Pop3Server pop3Server = null;

    private SmtpServer smtpServer1;
    private Pop3Server pop3Server1;

    private Boolean smtpOn;
    private Boolean pop3On;

```

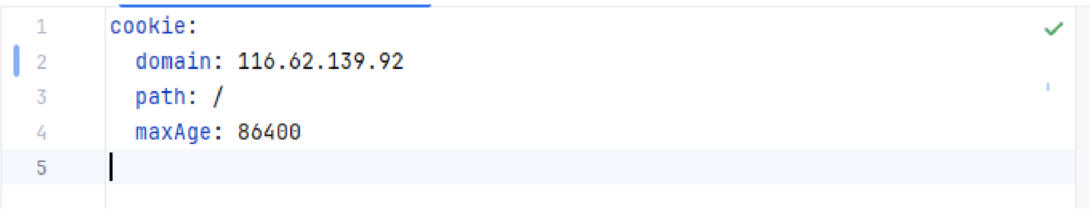
```
2024-11-27 16:29:57.355  INFO 19356 --- [           main] com.example.server.ServerApplication     : The following profiles are active: cookie
2024-11-27 16:29:58.260  INFO 19356 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8080 (http)
2024-11-27 16:29:58.266  INFO 19356 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2024-11-27 16:29:58.266  INFO 19356 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.39]
2024-11-27 16:29:58.313  INFO 19356 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2024-11-27 16:29:58.313  INFO 19356 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 921 ms
2024-11-27 16:29:58.597  INFO 19356 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2024-11-27 16:29:58.823  INFO 19356 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2024-11-27 16:29:59.092  INFO 19356 --- [           main] pertySourcedRequestMappingHandlerMapping : Mapped URL path [/v2/api-docs] onto method [springfox.documentation.swagger2.we
2024-11-27 16:29:59.176  INFO 19356 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing ExecutorService 'applicationTaskExecutor'
2024-11-27 16:29:59.314  INFO 19356 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080 (http) with context path ''
2024-11-27 16:29:59.314  INFO 19356 --- [           main] d.s.w.p.DocumentationPluginsBootstrapper : Context refreshed
2024-11-27 16:29:59.329  INFO 19356 --- [           main] d.s.w.p.DocumentationPluginsBootstrapper : Found 1 custom documentation plugin(s)
2024-11-27 16:29:59.345  INFO 19356 --- [           main] s.d.s.w.s.ApiListingReferenceScanner     : Scanning for api listing references
2024-11-27 16:29:59.431  INFO 19356 --- [           main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: handleLoginUsingPOST_1
2024-11-27 16:29:59.433  INFO 19356 --- [           main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: handleLoginUsingPOST_2
2024-11-27 16:29:59.448  INFO 19356 --- [           main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: handleGetServerMsgUsingGET_1
2024-11-27 16:29:59.449  INFO 19356 --- [           main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: handleSendGroupMailUsingPOST_1
2024-11-27 16:29:59.462  INFO 19356 --- [           main] com.example.server.ServerApplication     : Started ServerApplication in 2.338 seconds (JVM running for 2.817)
SMTP 服务已开启
POP3 服务已开启
2024-11-27 16:30:12.607  INFO 19356 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherServlet 'dispatcherServlet'
2024-11-27 16:30:12.607  INFO 19356 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2024-11-27 16:30:12.608  INFO 19356 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 1 ms
```

这里要配置 application.yml 文件，port 是本程序在服务器上要运行的端口，username 和 password 是云服务器数据库的用户名和密码。

这里使用宝塔配置环境，首先开放端口，这里需要开放后端程序的 8080 端口、25 端口和 110 端口。

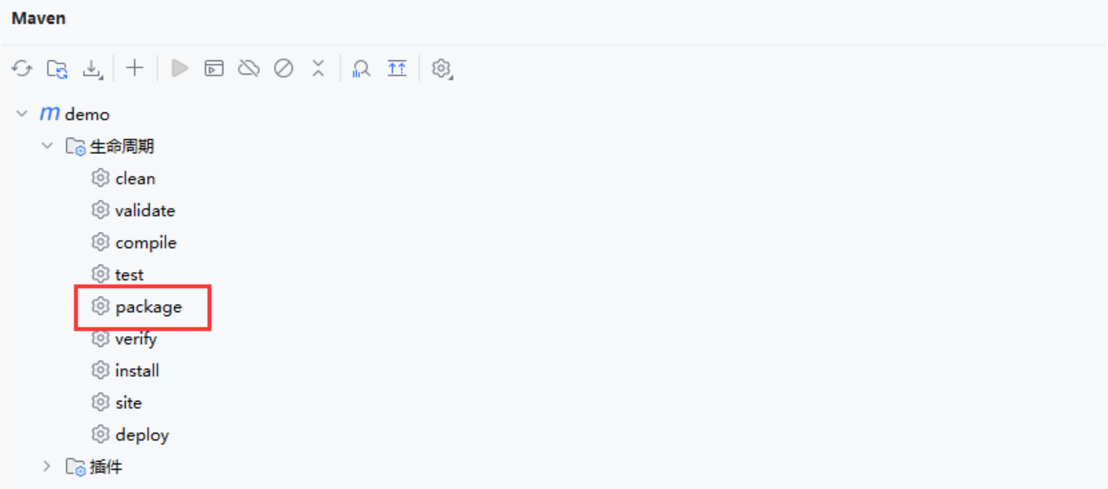


```yaml
server:
  port: 8080

spring:
  datasource:

    username: root
    password: 123456
    url: jdbc:mysql://localhost:3306/mail?serverTimezone=GMT%2B8
    driver-class-name: com.mysql.cj.jdbc.Driver

  profiles:
    active:
      - cookie
```

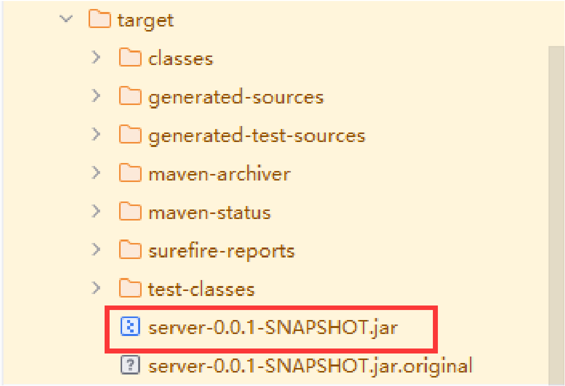


```yaml
cookie:
  domain: 116.62.139.92
  path: /
  maxAge: 86400
```

使用 maven 打包，生成 jar 包

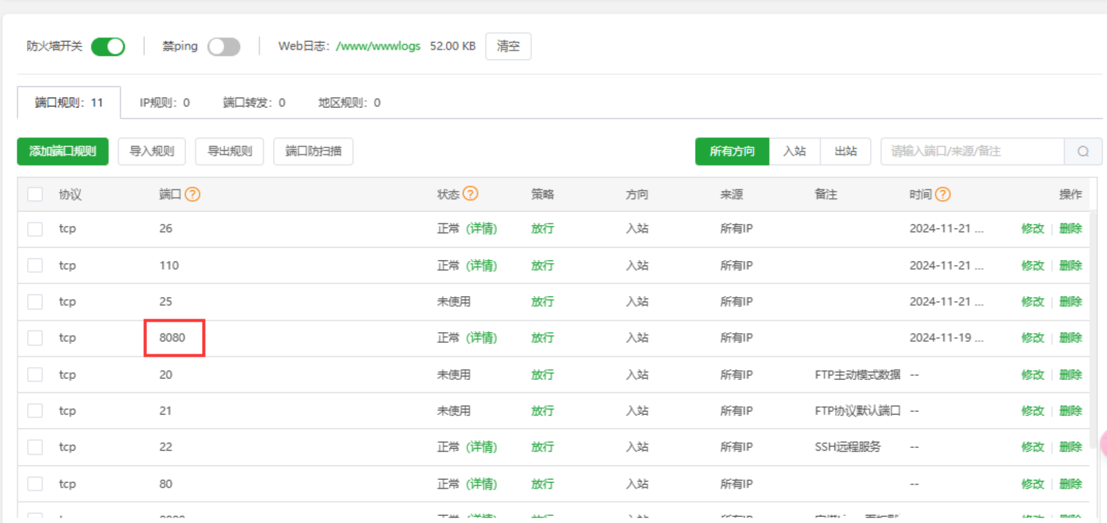


```
Maven
m demo
  生命周期
    clean
    validate
    compile
    test
    package
    verify
    install
    site
    deploy
  插件
```



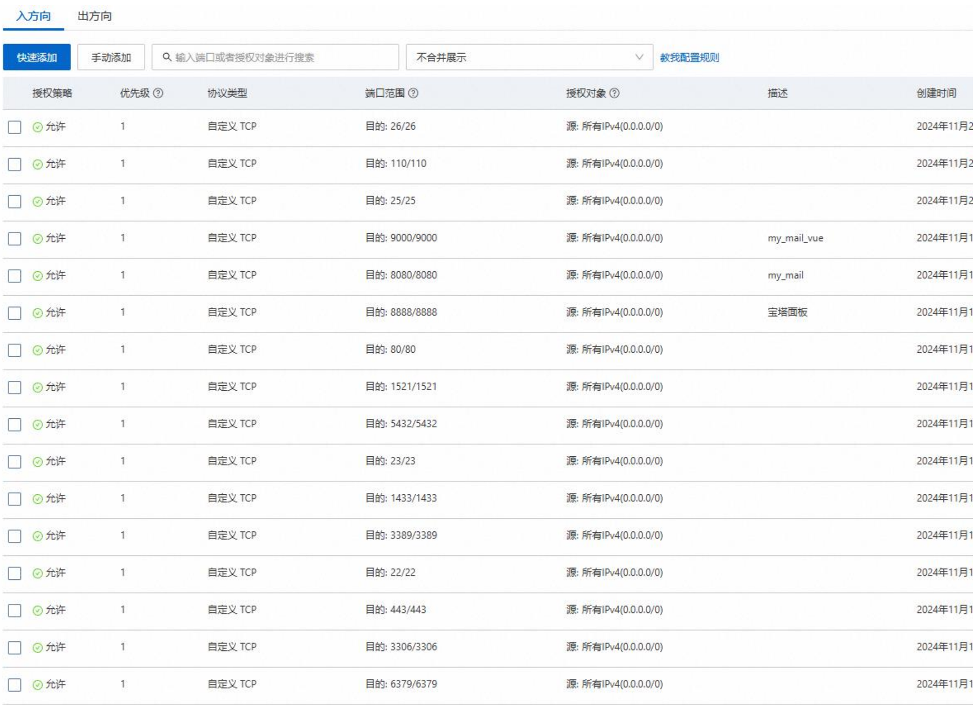
```
target
  classes
  generated-sources
  generated-test-sources
  maven-archiver
  maven-status
  surefire-reports
  test-classes
  server-0.0.1-SNAPSHOT.jar
  server-0.0.1-SNAPSHOT.jar.original
```

这里使用宝塔配置环境，首先开放端口，这里需要开放后端程序的 8080 端口、25 端口和 110 端口。



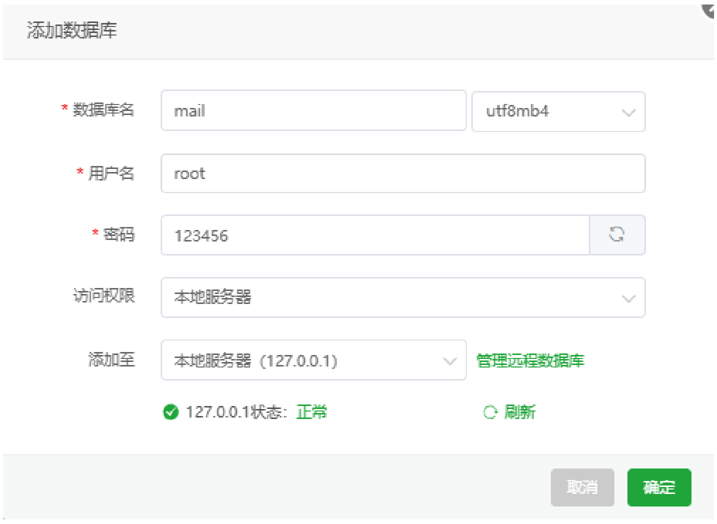
□	协议	端口	状态	策略	方向	来源	备注	时间	操作
□	tcp	26	正常 (详情)	放行	入站	所有IP		2024-11-21 ...	修改 | 删除
□	tcp	110	正常 (详情)	放行	入站	所有IP		2024-11-21 ...	修改 | 删除
□	tcp	25	未使用	放行	入站	所有IP		2024-11-21 ...	修改 | 删除
□	tcp	8080	正常 (详情)	放行	入站	所有IP		2024-11-19 ...	修改 | 删除
□	tcp	20	未使用	放行	入站	所有IP	FTP主动模式数据	--	修改 | 删除
□	tcp	21	未使用	放行	入站	所有IP	FTP协议默认端口	--	修改 | 删除
□	tcp	22	正常 (详情)	放行	入站	所有IP	SSH远程服务	--	修改 | 删除
□	tcp	80	正常 (详情)	放行	入站	所有IP		--	修改 | 删除

服务器也要设置对应的安全组。



□	授权策略	优先级	协议类型	端口范围	授权对象	描述	创建时间
□	✓ 允许	1	自定义 TCP	目的: 26/26	源: 所有IPv4(0.0.0.0/0)		2024年11月2
□	✓ 允许	1	自定义 TCP	目的: 110/110	源: 所有IPv4(0.0.0.0/0)		2024年11月2
□	✓ 允许	1	自定义 TCP	目的: 25/25	源: 所有IPv4(0.0.0.0/0)		2024年11月2
□	✓ 允许	1	自定义 TCP	目的: 9000/9000	源: 所有IPv4(0.0.0.0/0)	my_mail_vue	2024年11月1
□	✓ 允许	1	自定义 TCP	目的: 8080/8080	源: 所有IPv4(0.0.0.0/0)	my_mail	2024年11月1
□	✓ 允许	1	自定义 TCP	目的: 8888/8888	源: 所有IPv4(0.0.0.0/0)	宝塔面板	2024年11月1
□	✓ 允许	1	自定义 TCP	目的: 80/80	源: 所有IPv4(0.0.0.0/0)		2024年11月1
□	✓ 允许	1	自定义 TCP	目的: 1521/1521	源: 所有IPv4(0.0.0.0/0)		2024年11月1
□	✓ 允许	1	自定义 TCP	目的: 5432/5432	源: 所有IPv4(0.0.0.0/0)		2024年11月1
□	✓ 允许	1	自定义 TCP	目的: 23/23	源: 所有IPv4(0.0.0.0/0)		2024年11月1
□	✓ 允许	1	自定义 TCP	目的: 1433/1433	源: 所有IPv4(0.0.0.0/0)		2024年11月1
□	✓ 允许	1	自定义 TCP	目的: 3389/3389	源: 所有IPv4(0.0.0.0/0)		2024年11月1
□	✓ 允许	1	自定义 TCP	目的: 22/22	源: 所有IPv4(0.0.0.0/0)		2024年11月1
□	✓ 允许	1	自定义 TCP	目的: 443/443	源: 所有IPv4(0.0.0.0/0)		2024年11月1
□	✓ 允许	1	自定义 TCP	目的: 3306/3306	源: 所有IPv4(0.0.0.0/0)		2024年11月1
□	✓ 允许	1	自定义 TCP	目的: 6379/6379	源: 所有IPv4(0.0.0.0/0)		2024年11月1

下载 MYSQL，并导入 sql 脚本，配置好用户名和密码



添加数据库

* 数据库名：mail　utf8mb4

* 用户名：root

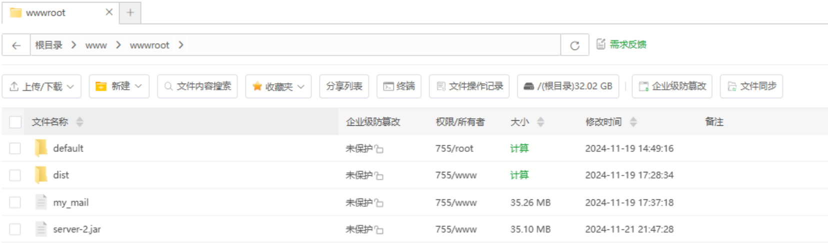
* 密码：123456

访问权限：本地服务器

添加至：本地服务器（127.0.0.1）　管理远程数据库

✓ 127.0.0.1状态：正常　刷新

导入 jar 包



wwwroot

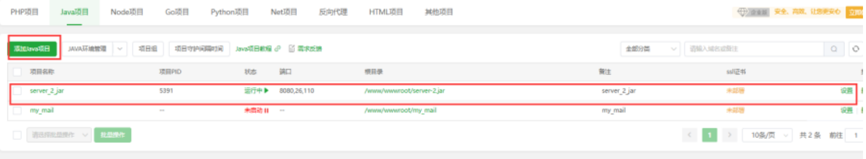
根目录 > www > wwwroot >

需求反馈

上传/下载 | 新建 | 文件内容搜索 | 收藏夹 | 分享列表 | 终端 | 文件操作记录 | /(根目录)32.02 GB | 企业级防篡改 | 文件同步

文件名称	企业级防篡改	权限/所有者	大小	修改时间	备注
default	未保护	755/root	计算	2024-11-19 14:49:16	
dist	未保护	755/www	计算	2024-11-19 17:28:34	
my_mail	未保护	755/www	35.26 MB	2024-11-19 17:37:18	
server-2.jar	未保护	755/www	35.10 MB	2024-11-21 21:47:28	

添加 java 项目并运行



PHP项目 | Java项目 | Node项目 | Go项目 | Python项目 | Net项目 | 反向代理 | HTML项目 | 其他项目

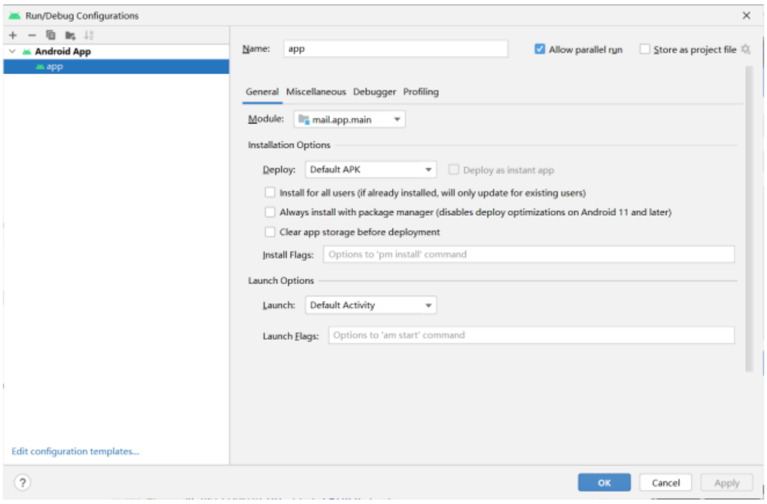
添加Java项目 | JAVA环境管理 | 项目组 | 项目守护间隔时间 | Java项目教程 | 需求反馈

项目名称	项目PID	状态	端口	根目录	备注	ssl证书	
server_2_jar	5391	运行中 ▶	8080,26,110	/www/wwwroot/server-2.jar	server_2_jar	未部署	设置
my_mail	--	未启动 ⏸	--	/www/wwwroot/my_mail	my_mail	未部署	设置

共 2 条

3.Android 端

编辑配置信息



Run/Debug Configurations

Android App
- app

Name: app

☑ Allow parallel run □ Store as project file

General Miscellaneous Debugger Profiling

Module: mail.app.main

Installation Options

Deploy: Default APK □ Deploy as instant app

□ Install for all users (if already installed, will only update for existing users)

□ Always install with package manager (disables deploy optimizations on Android 11 and later)

□ Clear app storage before deployment

Install Flags: Options to 'pm install' command

Launch Options

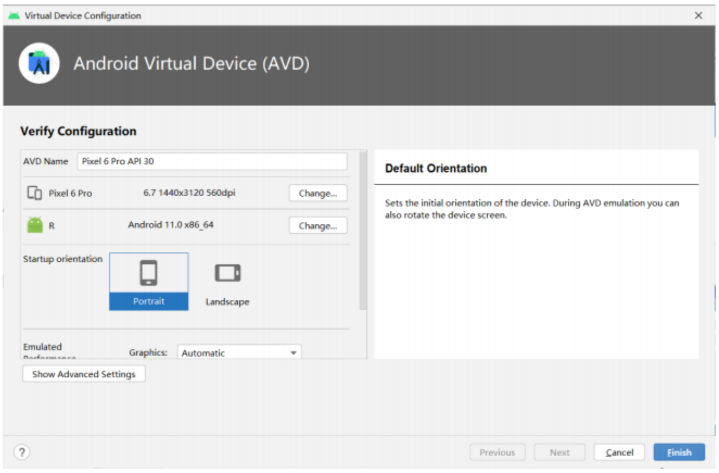
Launch: Default Activity

Launch Flags: Options to 'am start' command

Edit configuration templates...

OK | Cancel | Apply

配置虚拟机相关信息：



Virtual Device Configuration

Android Virtual Device (AVD)

Verify Configuration

AVD Name: Pixel 6 Pro API 30

Pixel 6 Pro 6.7 1440x3120 560dpi Change...

R Android 11.0 x86_64 Change...

Startup orientation: Portrait | Landscape

Emulated Performance: Graphics: Automatic

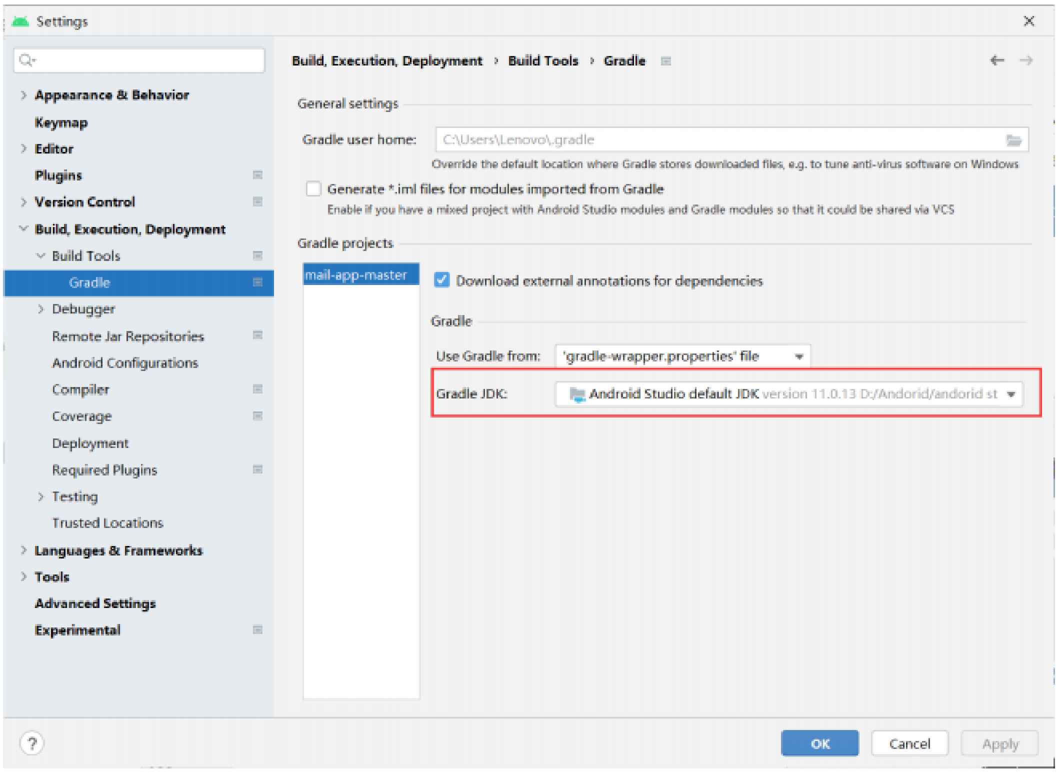
Show Advanced Settings

Default Orientation

Sets the initial orientation of the device. During AVD emulation you can also rotate the device screen.

Previous | Next | Cancel | Finish

配置 Gradle JDK



Build, Execution, Deployment › Build Tools › Gradle

General settings

Gradle user home: C:\Users\Lenovo\.gradle

Override the default location where Gradle stores downloaded files, e.g. to tune anti-virus software on Windows

□ Generate *.iml files for modules imported from Gradle

Enable if you have a mixed project with Android Studio modules and Gradle modules so that it could be shared via VCS

Gradle projects

mail-app-master

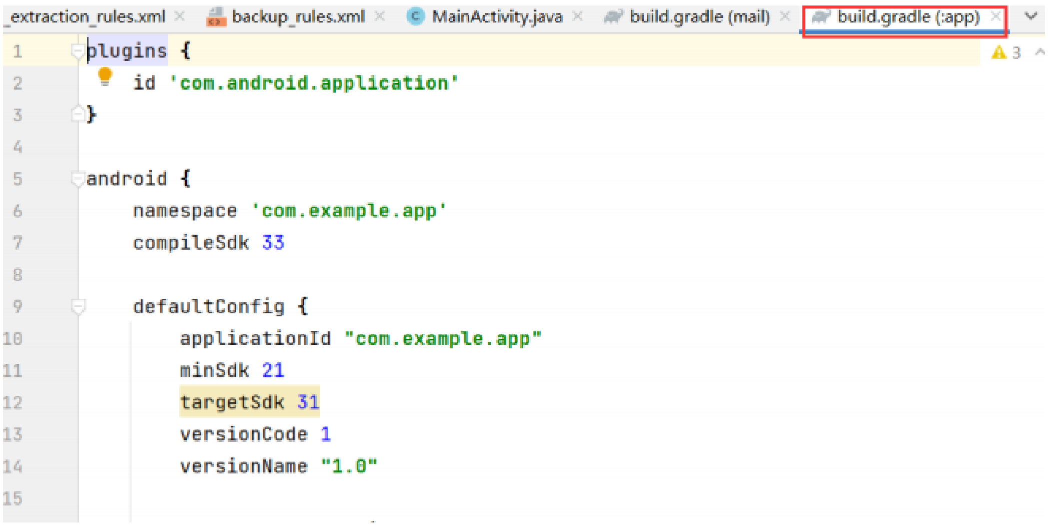
☑ Download external annotations for dependencies

Gradle

Use Gradle from: 'gradle-wrapper.properties' file

Gradle JDK: Android Studio default JDK version 11.0.13 D:/Andorid/andorid st

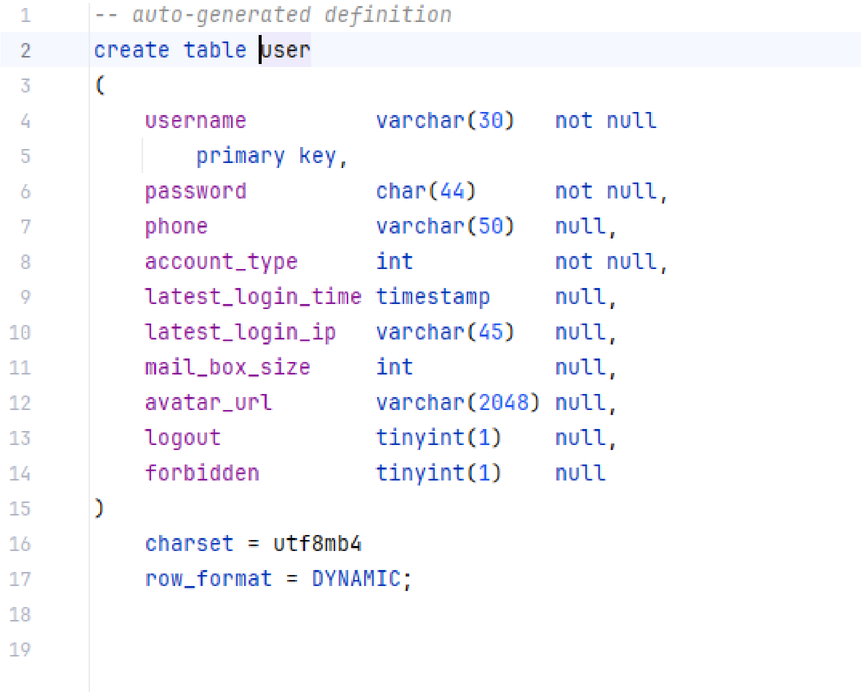
配置 SDK



```
plugins {
    id 'com.android.application'
}

android {
    namespace 'com.example.app'
    compileSdk 33

    defaultConfig {
        applicationId "com.example.app"
        minSdk 21
        targetSdk 31
        versionCode 1
        versionName "1.0"

```

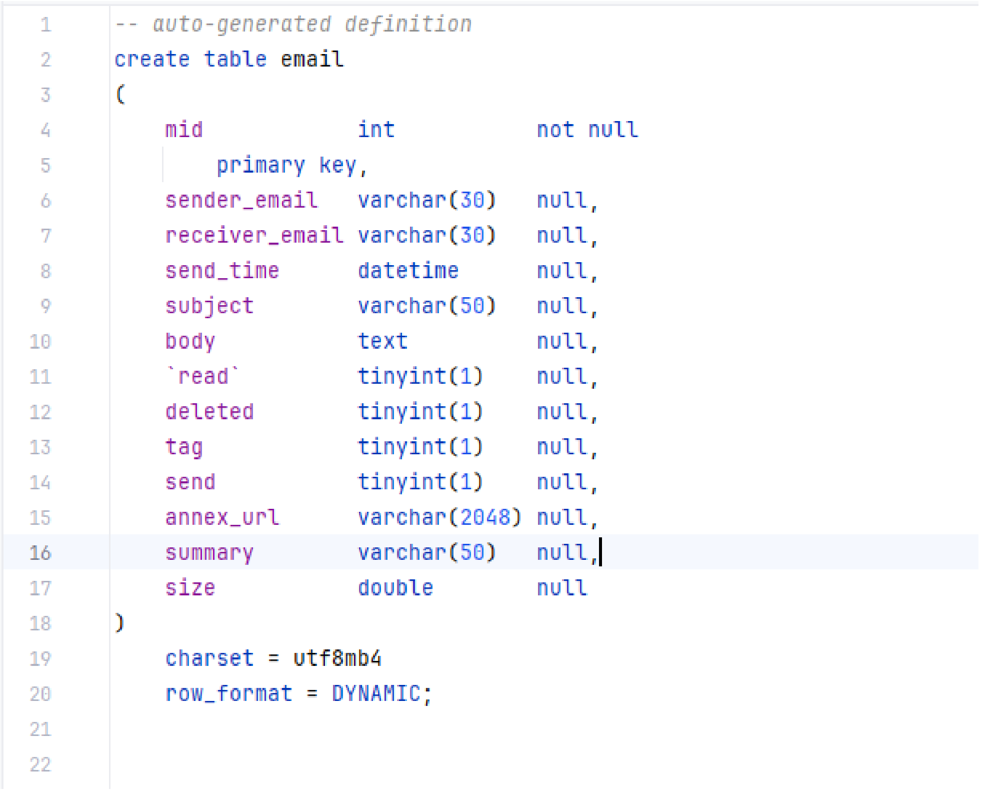
4.数据库

user



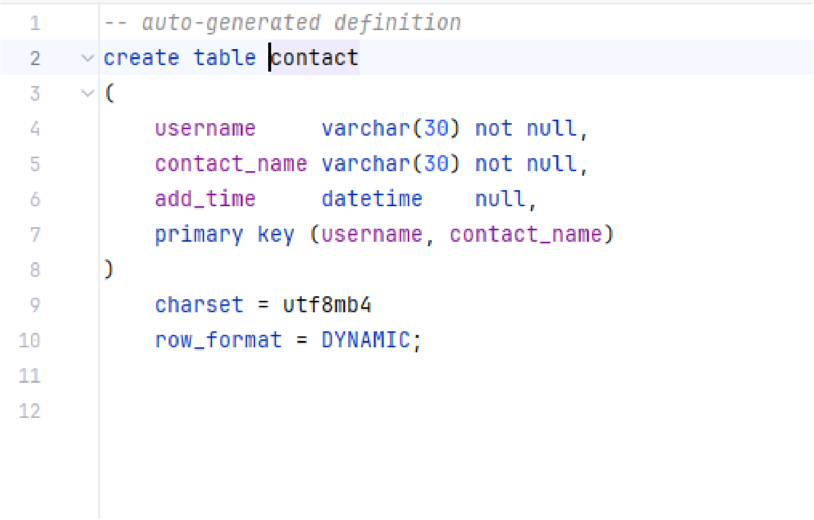
```sql
-- auto-generated definition
create table user
(
    username          varchar(30)   not null
        primary key,
    password          char(44)      not null,
    phone             varchar(50)   null,
    account_type      int           not null,
    latest_login_time timestamp     null,
    latest_login_ip   varchar(45)   null,
    mail_box_size     int           null,
    avatar_url        varchar(2048) null,
    logout            tinyint(1)    null,
    forbidden         tinyint(1)    null
)
    charset = utf8mb4
    row_format = DYNAMIC;
```

email



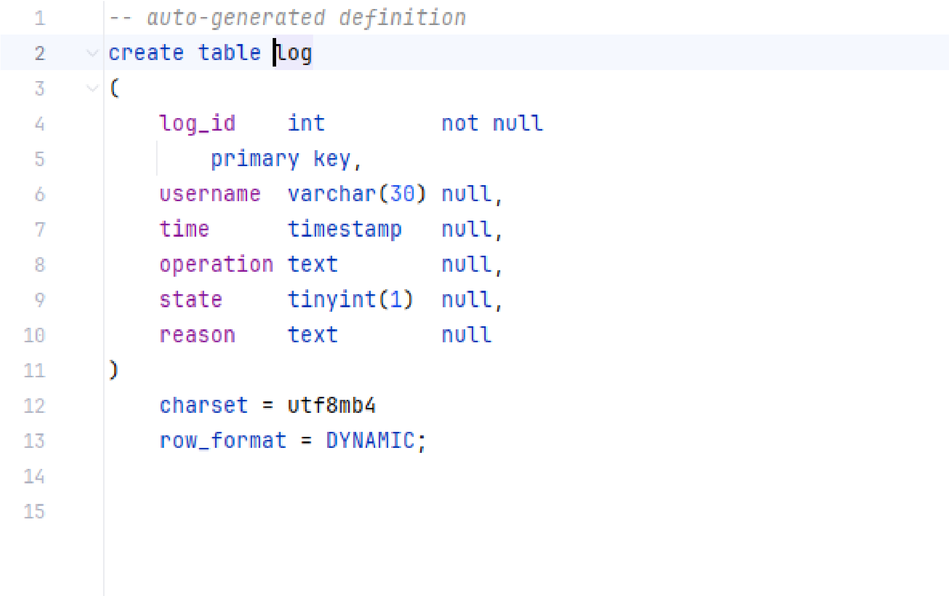
```sql
-- auto-generated definition
create table email
(
    mid            int           not null
        primary key,
    sender_email   varchar(30)   null,
    receiver_email varchar(30)   null,
    send_time      datetime      null,
    subject        varchar(50)   null,
    body           text          null,
    `read`         tinyint(1)    null,
    deleted        tinyint(1)    null,
    tag            tinyint(1)    null,
    send           tinyint(1)    null,
    annex_url      varchar(2048) null,
    summary        varchar(50)   null,
    size           double        null
)
    charset = utf8mb4
    row_format = DYNAMIC;
```

contact



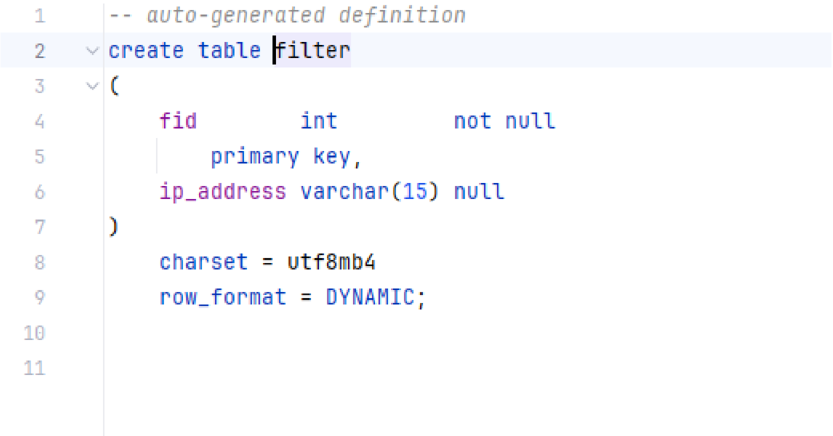
```sql
-- auto-generated definition
create table contact
(
    username     varchar(30) not null,
    contact_name varchar(30) not null,
    add_time     datetime    null,
    primary key (username, contact_name)
)
    charset = utf8mb4
    row_format = DYNAMIC;
```

log



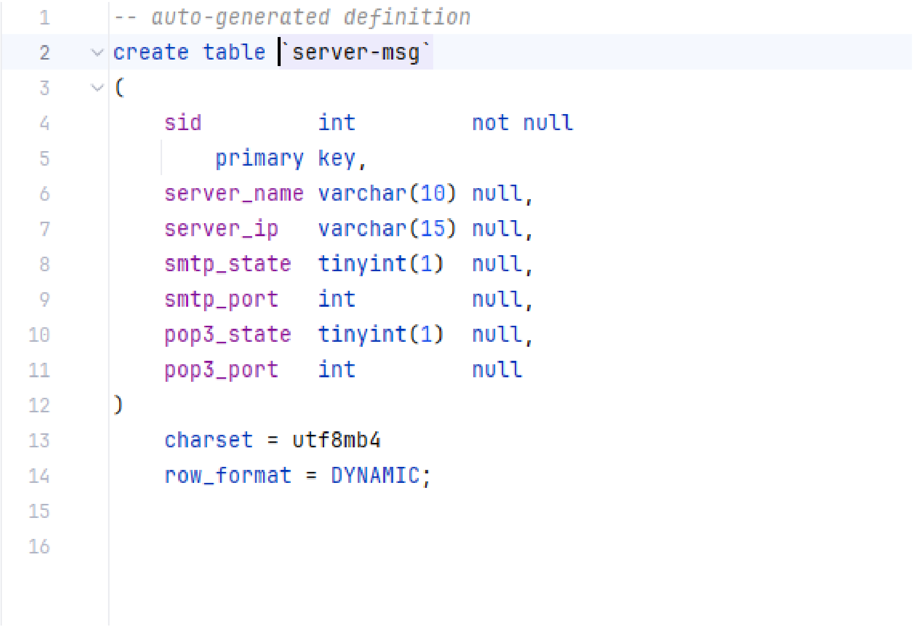
```sql
-- auto-generated definition
create table log
(
    log_id    int          not null
        primary key,
    username  varchar(30) null,
    time      timestamp   null,
    operation text        null,
    state     tinyint(1)  null,
    reason    text        null
)
    charset = utf8mb4
    row_format = DYNAMIC;
```

filter



```sql
-- auto-generated definition
create table filter
(
    fid        int         not null
        primary key,
    ip_address varchar(15) null
)
    charset = utf8mb4
    row_format = DYNAMIC;
```

server-msg



```sql
-- auto-generated definition
create table `server-msg`
(
    sid         int         not null
        primary key,
    server_name varchar(10) null,
    server_ip   varchar(15) null,
    smtp_state  tinyint(1)  null,
    smtp_port   int         null,
    pop3_state  tinyint(1)  null,
    pop3_port   int         null
)
    charset = utf8mb4
    row_format = DYNAMIC;


```