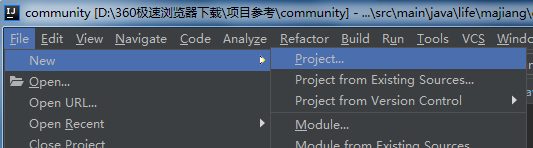
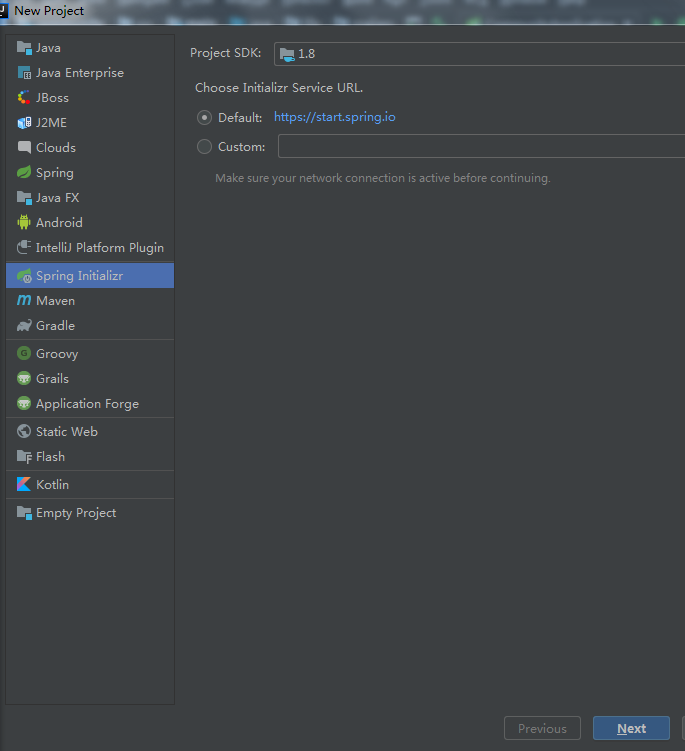
一、搭建SpringBoot项目，项目是使用IDEA2018.3.5 gradle管理依赖 MySQL数据库

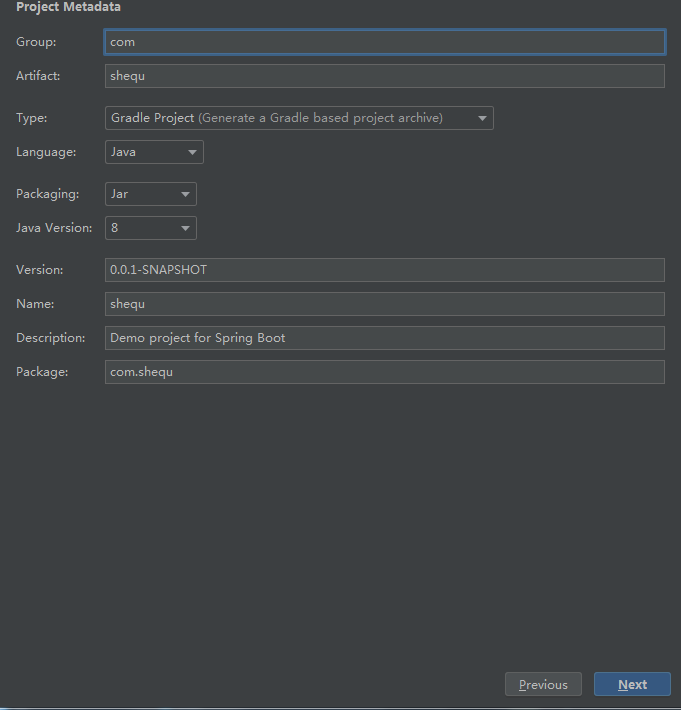
1.新建project



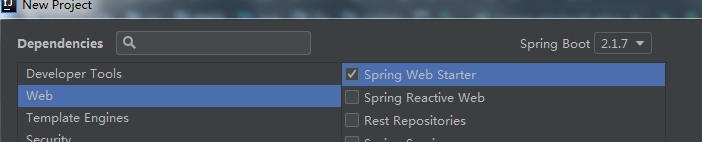
1. 选择spring Initializr，jdk默认，直接Next

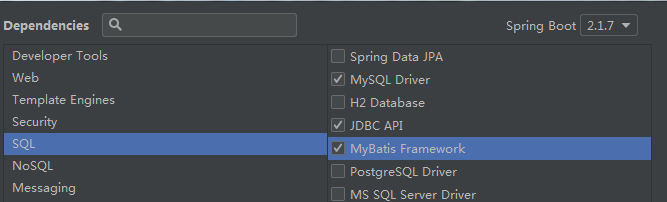


1. Group 团队 一般写国籍.公司，Artifact 工程名，Type 这里选择Gradle Project

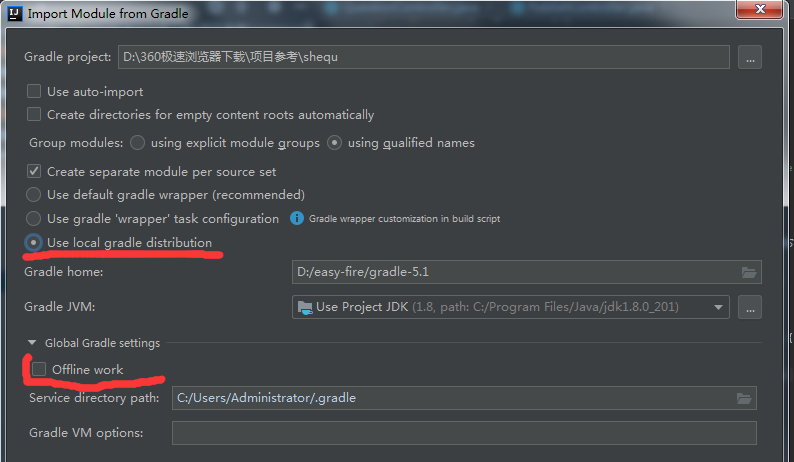


1. 选择依赖，暂时选择数据库相关和web依赖，下一步是选择路径，这里不做解释





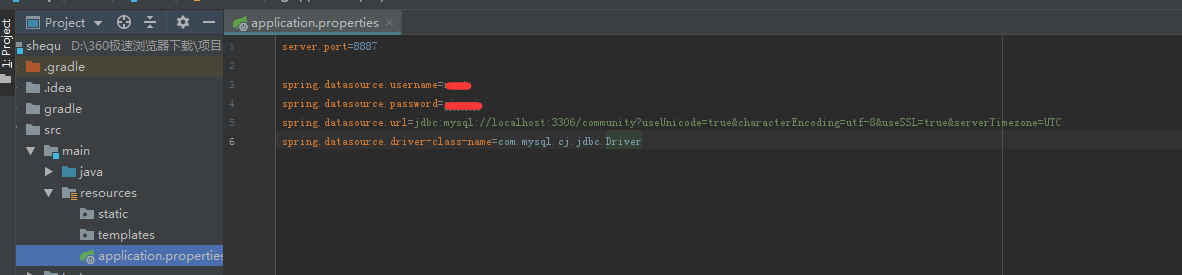
1. Gradle仓库地址选择，选择本地地址，注意 Offline work 不要勾选，否则无法从网上仓库下载依赖



OK，这就是一个web项目的基本内容。

1. 配置数据库

在application.properties文件中配置，或者在application.yml文件，配置端口号和数据库



1. swagger2

由于项目是前后端分离，我们写后端代码的时候也需要在页面上测试接口，需要有数据交互，所以推荐使用swagger2接口管理，或者使用postman来测试接口。这里采用swagger2。

1. 引入swagger2的依赖

//swagger依赖  
implementation 'io.springfox:springfox-swagger-ui:2.7.0'  
implementation 'io.springfox:springfox-swagger2:2.7.0'

1. 在主目录下新建config包，编写SwaggerConfig.java

@Configuration  
@EnableSwagger2  
public class SwaggerConfig {  
  
  
 @Value("${swagger.enabled}")  
 private boolean enableSwagger;  
 @Bean  
 public Docket RecruitApi(){  
 return new Docket(DocumentationType.*SWAGGER\_2*)  
 .groupName("Recruit")  
 .genericModelSubstitutes(DeferredResult.class)  
 .useDefaultResponseMessages(false)  
 .forCodeGeneration(true)  
 .pathMapping("")  
 .select()  
 .apis(RequestHandlerSelectors.*basePackage*("com.shequ.controller"))  
 .build()  
 .apiInfo(RecruitApiInfo());  
 }  
  
 private ApiInfo RecruitApiInfo() {  
 return new ApiInfoBuilder()  
 .title("Here is the name for system")  
 .description("name-system Server's REST API")  
 .version("V1.0")  
 .termsOfServiceUrl("NO terms of service")  
 .contact(new Contact("easy.fire","","easy.fire@aorise.org"))  
 .license("GPL")  
 .licenseUrl("")  
 .build();  
 }  
}

1. 配置文件中设置swagger.enabled值为true ，即在application.properties写入

Swagger.enabled=true

1. 浏览器中输入 localhost://8888/swagger-ui.html
2. GitHub登录

有了接口管理，我们就可以开始写接口了。

首先写登录接口，这里我们直接借用GitHub的登录