# 软件工程项目笔记

## 关于项目的部署

本软件项目使用Spring Boot作为底层启动框架，软件项目管理工具使用的是build.gradle。

其他有关的框架有：

1 使用Spring Data JPA作为持久层访问的框架，其特点是使用一系列@Annotaion作为实体的标记工具，用于控制数据库的同步生成。以及对数据持久层的访问。

2 使用Spring MVC作为web层的核心架构。页面模板引擎为Thymeleaf，其主要特点是可以快速显示Model中存储的信息，比较快速和方便。

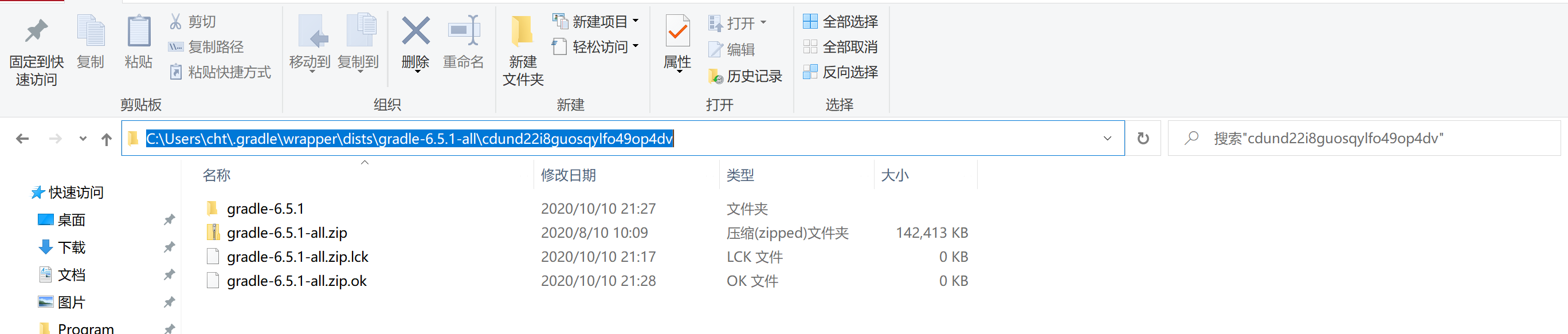
3 前端主要使用纯html和少量的js代码展示页面。

软件的部署分以下几部完成：

### 1 构建项目

用IntelliJ IDEA打开server文件夹，点击IDE右上角的build按钮进行项目构建。在构建过程中可能会出现下列问题。

a) 网络异常，无法下载gradle-6.5.1-all.zip。原因：gradle构建包有时候会由于网络问题无法下载，这个时候需要手动下载包到缓存目录。首先找到用户目录（常常为：c:/Users/${userName}），找到./gradle/wrapper/dists文件夹（如果没有则创建，然后），然后找到gradle-6.5.1-all/${base64}（为base64的一串码）/文件夹。然后将gradle-6.5.1-all.zip放到这个目录。



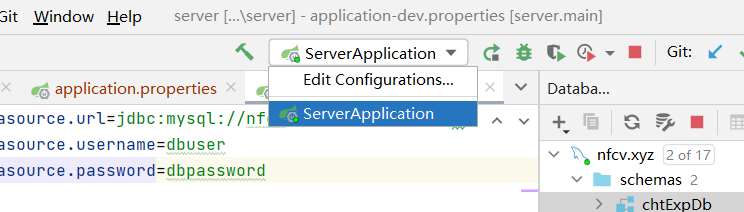
重新启动IDE，并开始测试。

### 2 配置数据库

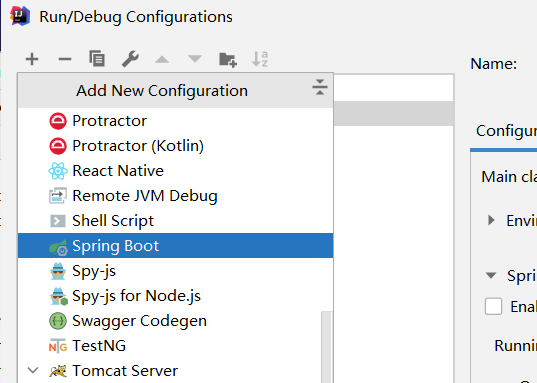
a) 在application.properties的同级目录中创建application-dev.properties，然后加入与数据库有关的配置项。例如：

|  |
| --- |
| spring.datasource.url=jdbc:mysql://nfcv.xyz:3306/chtLearning spring.datasource.username=dbuser spring.datasource.password=dbpassword |

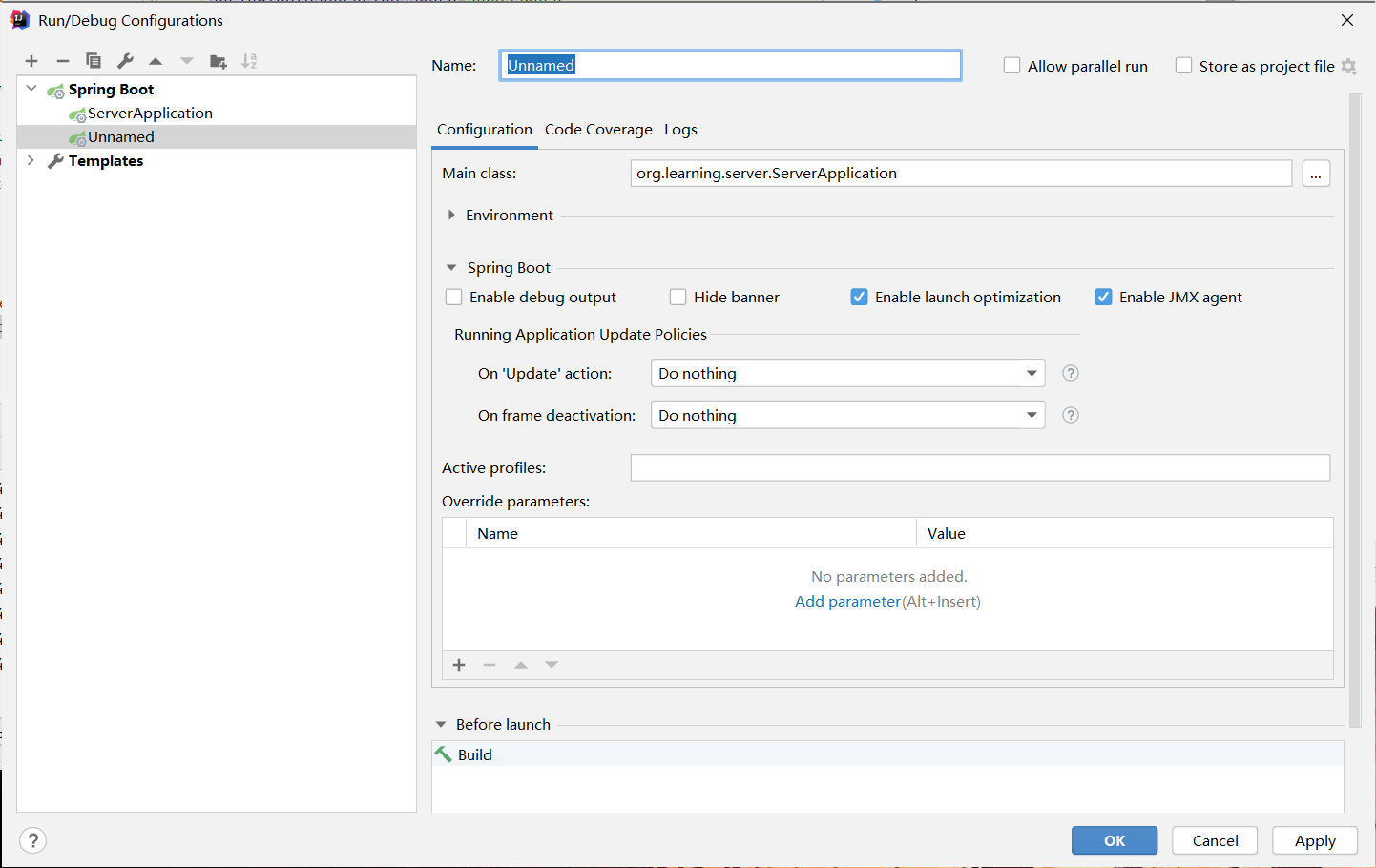
### 3 配置启动项



点击Edit Configuration…



找到Spring Boot



修改Name和Main class（org.learning.server.ServerApplication）。

## 关于build.gradle文件

build.gradle是部署与配置这个项目的核心。其中repositories指定软件依赖库的仓储。而dependencies指定软件的依赖。

在repositories中

|  |
| --- |
| maven **{** url 'https://maven.aliyun.com/repository/google' **}** maven **{** url 'https://maven.aliyun.com/repository/public' **}** |

两行制定了国内的maven源，主要是由于国际源常常由于网络的原因，连接不到。

dependencies中的各项即软件的依赖项，使用implementation/compile + 软件描述符的方式进行描述。其中软件描述符有3部分组成：groupName:archiveName:version。当软件需要依赖时，在其中添加依赖并build即可。

## 包的作用和说明

entity：存放与数据库实体相关的类。

repository：存放dao类型，其为一个接口，继承CrudRepository，从而实现数据库的Crud操作。

service：存放服务，业务的核心部分，一般能通过单元测试。其中实现类放在service.impl包中。

controller：控制器类，用于控制url->页面（数据）的跳转关系。

form：存放表单类型，一般用于存放可验证的表单类。

form.pattern：存放要用到的pattern信息

## 1.2 用户基础设置

使用到的教程：

1 如何进行表单验证（配合thymeleaf）<https://www.jianshu.com/p/48a4feab7541>

2 thymeleaf入门：<https://zhuanlan.zhihu.com/p/24988845?refer=dreawer>

数据库设计：请见实体entity.User

注册、登录：

使用到的Form：UserRegisterForm

使用到的其他类型：UserController，User，ActionResult，UserRepository，UserService

## 15周验收内容

1 组织使用树型结构进行管理。（使用回溯表节点进行管理）。分成多种树型的结构。

2 课程内容的设置参考Mooc。

3 体现组织之间管理的关系，例如上级的管理员可以对下级的节点进行管理。

4 课程资料库。课程。和课程开设的关系。

5 课程进度的记录，以及课程的管理等。