

项目组织及文件说明

注：粗体表示文件夹，斜体表示重要文件（夹）

backend :

—— .idea	Intellij 自动配置的文件
—— .mvn	maven 文件自动配置的依赖
—— src	本次项目的源代码
—— target	项目运行中产生的.class 等文件
—— .gitignore	项目加入 GitHub 时需要忽略的文件
—— backend.iml	用 intellij 运行项目会得到的临时文件
—— mvnw (mvnw.cmd)	Maven Wrapper/Maven 保持构建工具版本一致的工具
—— <i>pom.xml</i>	maven 配置环境最重要的文件 描述了 maven 坐标，依赖关系，开发者需要遵循的规则，缺陷管理系统，组织和 licenses

src :

—— main	程序代码部分
—— java	程序开发代码及配置文件
—— <i>com.advancedweb.backend</i>	程序开发代码
—— resources	配置文件
—— static	直接访问的静态(HTML)文件
—— templates	直接访问的动态(HTML)文件
—— <i>application.properties</i>	统一的配置（部署）文件
—— test	程序测试代码
—— java.com.advancedweb.backend	
—— <i>BackendApplicationTests.java</i>	测试代码放在其中

src/main/java/com/advancedweb/backend :

（实现了 SpringMVC 中的 Model 与 Controller）

—— controller	控制层实现，实现了控制器的所有代码
—— json_model	前后端 json 交互中的数据类型模型
—— AssignmentController.java	作业模块所需的控制器
—— CourseController.java	课程模块
—— CoursewaresController.java	课件模块
—— MainController.java	主页模块
—— MaterialsController.java	资源模块
—— MindmapController.java	思维导图模块

—— model —— UserTemp.java 等	与数据库交互中的数据类型模型，具体类型详见数据库 ER 图与类图
—— repository —— UserTempRepository.java 等	持久层实现，与数据库实际交互的 dao 层，详见《接口规范文档.md》
—— service —— mailService.java 等	业务逻辑层实现
—— BackendApplication.java	启动类，可执行文件，main 函数

frontend :

—— e2e	端到端测试目录，做自动化测试
—— node_modules	第三方依赖包存放目录
—— src	应用的源代码目录
—— .editorconfig	webstorm 的配置文件
—— .gitignore	git 的配置文件
—— angular.json	angular 的配置文件
—— package.json	标准的 npm 工具的配置文件，文件里列出了该应用程序所用的第三方依赖包
—— tsconfig.json	项目的基础文档、说明文档
—— tslint.json	tslint 的配置文件，用来定义 TypeScript 代码质量检查的规则

src :

—— app	
—— component	组件（在之后具体介绍）
—— component.component.css	组件的 css 样式
—— component.component.html	组件的 html 页面
—— component.component.spec.ts	用于自动化测试
—— component.component.ts	组件的业务数据和逻辑
—— service	服务（在之后具体介绍）
—— service.service.ts	服务的业务逻辑和操作
—— service.service.spec.ts	用于自动化测试
—— class	类
—— class.ts	类的具体声明
—— assets	存放静态资源，比如图片
—— bootstrap	导入的 bootstrap

—— jsmind	导入的 jsmind
—— environments	环境配置
—— index.html	整个应用的根 HTML
—— karma.conf.js	给 karma 的单元测试配置
—— main.ts	整个应用脚本执行的入口点
—— polyfills.ts	导入一些库, 来适应一些版本比较老的浏览器
—— styles.css	应用的全局样式
—— test.ts	做自动化测试
—— tsconfig.app.json	Typescript 编译器的配置文件
—— tsconfig.spec.json	Typescript 编译器的配置文件

src/app :

—— add-course	添加课程组件
—— courses	课程组件 (包括显示课程和控制弹出添加课程组件)
—— courseware	课件组件
—— details	详情组件 (包括思维导图页、作业页、课件页、资源页)
—— homework	作业组件
—— index	主页组件 (包括控制弹出登录组件和注册组件)
—— login	登录组件
—— main	主组件
—— mindmap	思维导图组件
—— modify-password	修改密码组件
—— register	注册组件
—— resources	资源组件
—— stu-add-course	学生的添加课程组件
—— stu-details	学生的详情组件
—— stu-homework	学生的作业组件
—— stu-main	学生的主组件
—— stu-mindmap	学生的思维导图组件

src/service :

—— course.service.ts	课程对象的服务 (包括添加课程、显示课程)
—— mindmap.service.ts	思维导图对象的服务 (包括创建思维导

	图、切换思维导图)
—— node.service.ts	思维导图节点的服务 (包括增删改查节点 ; 对节点发布、完成作业 ; 对节点上传、下载课件和资源)
—— user.service.ts	用户的服务 (包括登录、注册、发送验证码、修改密码)