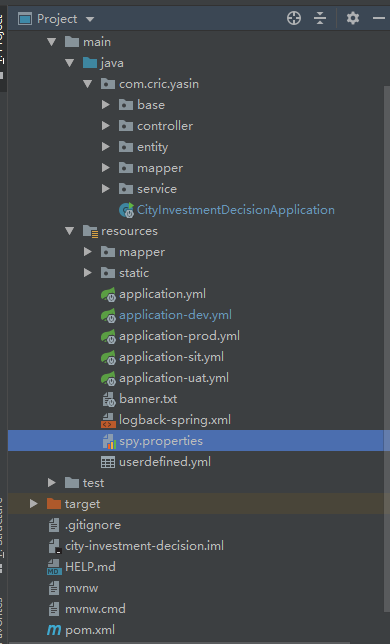
项目的project结构



配置文件：

Application.yml，springboot的配置文件，application-\*.yml，不同环境下的配置文件；

Bnner.txt，标志

Logback-spring.xml，日志配置文件

Spy.properties,打印sql的配置文件

Userdefined.yml,参数配置

1. 前端

Vue.js

1. 后端
   1. Springboot
      1. aop切面编程

首先需要引入aop-starter;

注意几个概念：

目标类target:需要被代理的类，即将被加强的类；

连接点joinpoint:就是即将被连接的方法，target类的所有方法；

切入点pointcut:被增强的连接点；

增强点advice:增强的方法；

Weaving织入：将advice应用到target来创建proxy的过程；

Proxy代理；

Aspect切面：pointcut和advice的方法构成切面

* + 1. Threadlocal
    2. RequestContextHolder

HttpServletRequest request = ((ServletRequestAttributes)RequestContxtHolder.getRequestAttributes()).getRequest();

* + 1. 多线程
    2. 日志

采用logback-spring.xml，springboot默认读取；

* + 1. driud连接池

Spring.datasource.dynamic.p6spy:采用整合p6spy打印真正的执行sql;

Spting.datasource.dynmic.primary:多数据源配置，采用读写库分离，pom中增加

<groupId>com.baomidou</groupId>  
<artifactId>dynamic-datasource-spring-boot-starter</artifactId>  
<version>2.5.4</version>

通过@DS注解在service上切换数据库，建议在service方法上添加（可以在类上，不过方法优先于类）

* + 1. 如何读写库分离？Mysql,sqlserver,oracl是怎么做的？

1. 为什么要读写分离？

减缓数据库压力，提高性能;

1. 怎么用？
2. 有没有可以替代的方法？

读写分离主要是提高性能缓解数据库压力，缓存技术也可以实现这样的效果，目前主要是使用redis缓存，redis是单线程数据结构可以存储上万数据，采用key-value形式，缓存快，读取速度快，采用主从结构实现redis的高性能，一主多从，主写读从，也可以采用集群模式实现高可用，缺点也有，系统复杂度，程序复杂度变高，对于程序员的要求变高；也可以分组的形式将单个数据库分割开，可以增加性能；

* + 1. Mybaties-plus插件
    2. 事务机制
    3. 锁机制

问题：

1. 什么时候使用到web-starter

在需要使用到tomcat，restful,springmvc的时候需要引入

1. 如何在service中获取request,response请求？

使用RequestContextHolder