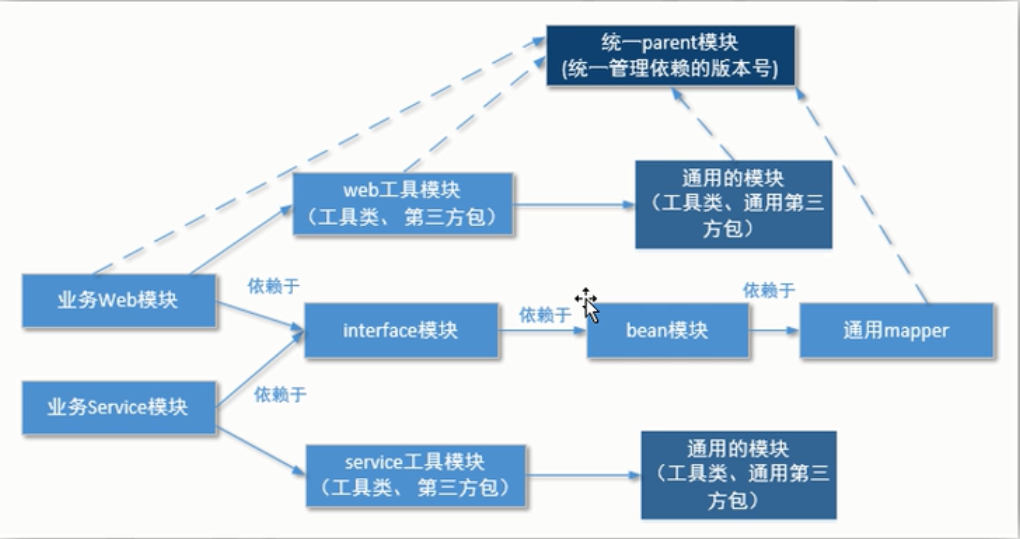
# 谷粒商城的架构



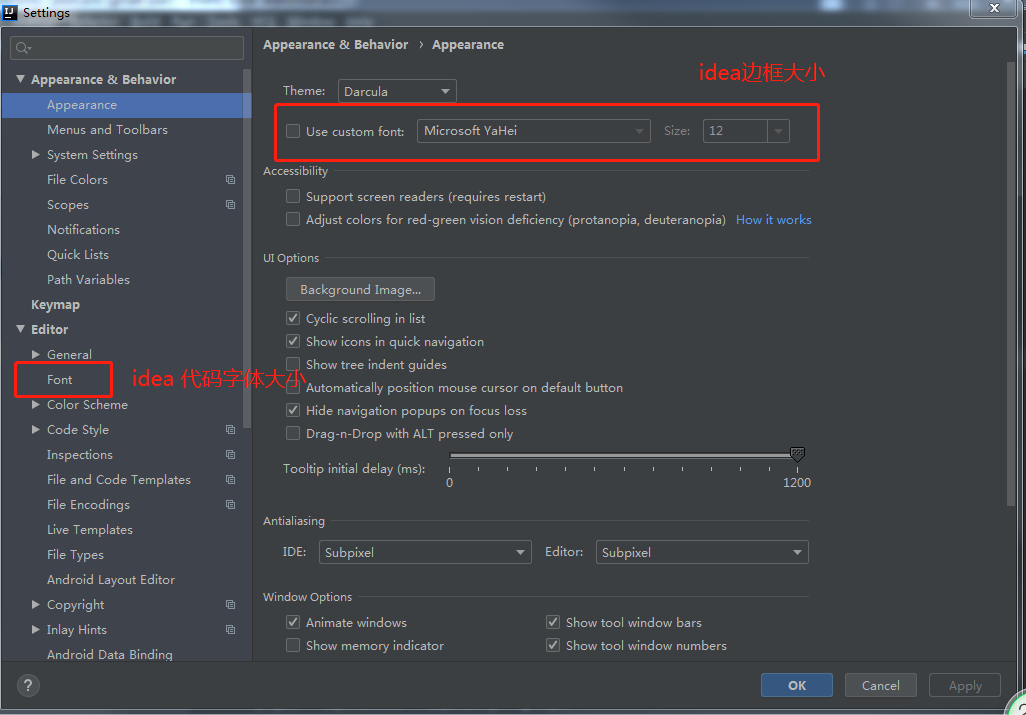
# 关于架构的介绍

|  |
| --- |
| 1 工程结构  以maven为基础，对项目的分层架构  2 项目架构  分布式（SOA）  Dubbo(ali) |

# gamll-parent 父依赖的创建

|  |  |
| --- | --- |
| 1 用maven 创建一个gmall-parent的工程  2 新建其他项目模块(子项目)的时候，继承gmall-parent  3 gmall-parent中的父依赖使用psringboot2.0.7   |  | | --- | | <parent>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-parent</artifactId>  <version>2.1.7.RELEASE</version>  <relativePath/> <!-- lookup parent from repository --> </parent>  <groupId>com.atguigu.gmall</groupId> <artifactId>gmall-parent</artifactId> <version>1.0-SNAPSHOT</version> <packaging>pom</packaging> |   4 在gmall-parent中定义好项目的技术框架各种版本  <?xml version="1.0" encoding="UTF-8"?> <project xmlns="http://maven.apache.org/POM/4.0.0"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  <modelVersion>4.0.0</modelVersion>   <parent>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-parent</artifactId>  <version>2.1.7.RELEASE</version>  <relativePath/> <!-- lookup parent from repository -->  </parent>   <groupId>com.atguigu.gmall</groupId>  <artifactId>gmall-parent</artifactId>  <version>1.0-SNAPSHOT</version>  <packaging>pom</packaging>   <properties>  <project.bulid.sourceEncoding>UTF-8</project.bulid.sourceEncoding>  <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>  <java.version>1.8</java.version>   <fastjson.version>1.2.57</fastjson.version>  <dubbo-starter.version>1.0.10</dubbo-starter.version>  <dubbo.version>2.7.3</dubbo.version>  <zkclient.version>0.10</zkclient.version>  <mybatis.version>2.1.0</mybatis.version>  <nekohtml.version>1.9.20</nekohtml.version>  <xml-apis.version>1.4.01</xml-apis.version>  <batik-ext.version>1.9.1</batik-ext.version>  <jsoup.version>1.11.2</jsoup.version>  <httpclient.version>4.5.5</httpclient.version>  <commons-lang3.version>3.8.1</commons-lang3.version>  <mapper-starter.version>2.0.2</mapper-starter.version>  <jedis.version>2.9.3</jedis.version>  <jest.version>5.3.3</jest.version>  <jna.version>4.5.1</jna.version>  <beanUtils.version>1.9.3</beanUtils.version>  </properties>    <dependencyManagement>  <dependencies>  <dependency>  <groupId>com.alibaba</groupId>  <artifactId>fastjson</artifactId>  <version>${fastjson.version}</version>  </dependency>   <dependency>  <groupId>com.101tec</groupId>  <artifactId>zkclient</artifactId>  <version>${zkclient.version}</version>  </dependency>   <dependency>  <groupId>com.gitee.reger</groupId>  <artifactId>spring-boot-starter-dubbo</artifactId>  <version>${dubbo-starter.version}</version>  </dependency>   <dependency>  <groupId>org.mybatis.spring.boot</groupId>  <artifactId>mybatis-spring-boot-starter</artifactId>  <version>${mybatis.version}</version>  </dependency>   <dependency>  <groupId>net.sourceforge.nekohtml</groupId>  <artifactId>nekohtml</artifactId>  <version>${nekohtml.version}</version>  </dependency>   <dependency>  <groupId>xml-apis</groupId>  <artifactId>xml-apis</artifactId>  <version>${xml-apis.version}</version>  </dependency>   <dependency>  <groupId>org.apache.xmlgraphics</groupId>  <artifactId>batik-ext</artifactId>  <version>${batik-ext.version}</version>  </dependency>   <!-- https://mvnrepository.com/artifact/org.jsoup/jsout -->  <dependency>  <groupId>org.jsoup</groupId>  <artifactId>jsoup</artifactId>  <version>${jsoup.version}</version>  </dependency>   <!-- https://mvnrepository.com/artifact/org.apache.httpcomponents/httpclient -->  <dependency>  <groupId>org.apache.httpcomponents</groupId>  <artifactId>httpclient</artifactId>  <version>${httpclient.version}</version>  </dependency>   <dependency>  <groupId>org.apache.commons</groupId>  <artifactId>commons-lang3</artifactId>  <version>${commons-lang3.version}</version>  </dependency>   <dependency>  <groupId>tk.mybatis</groupId>  <artifactId>mapper-spring-boot-starter</artifactId>  <version>${mapper-starter.version}</version>  </dependency>   <dependency>  <groupId>redis.clients</groupId>  <artifactId>jedis</artifactId>  <version>${jedis.version}</version>  </dependency>   <!-- https://mvnrepository.com/artifact/io.searchbox/jest -->  <dependency>  <groupId>io.searchbox</groupId>  <artifactId>jest</artifactId>  <version>${jest.version}</version>  </dependency>   <!-- https://mvnrepository.com/artifact/net.java.dev.jna/jna -->  <dependency>  <groupId>net.java.dev.jna</groupId>  <artifactId>jna</artifactId>  <version>${jna.version}</version>  </dependency>   <dependency>  <groupId>commons-beanutils</groupId>  <artifactId>commons-beanutils</artifactId>  <version>${beanUtils.version}</version>  </dependency>   </dependencies>  </dependencyManagement>  </project> |

# 调整idea的大小

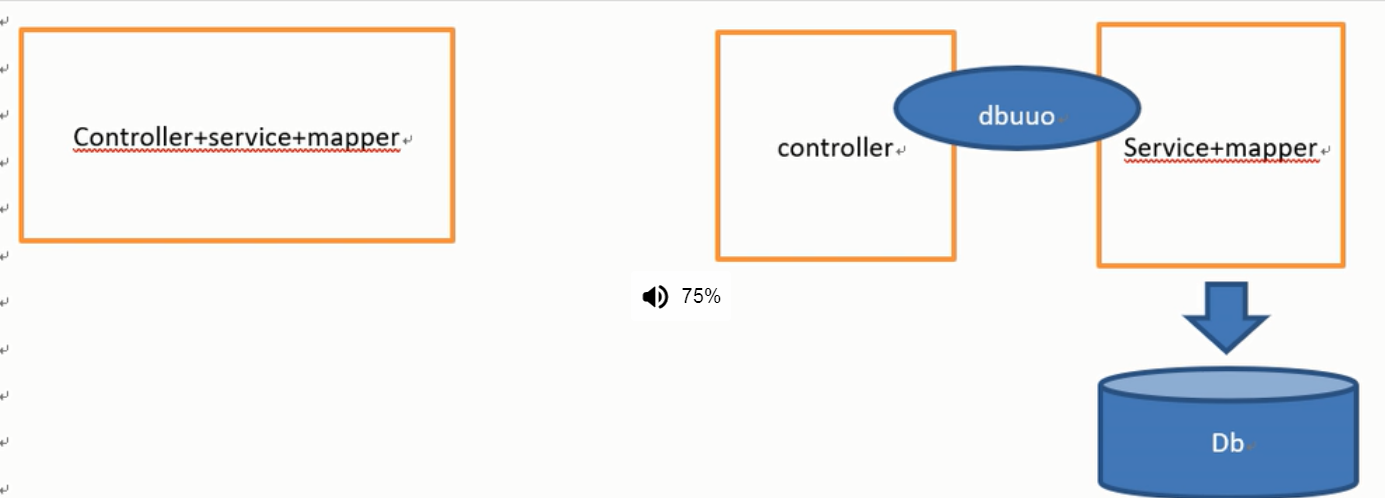


# 抽取api工程(负责管理项目中所有的接口和bean)

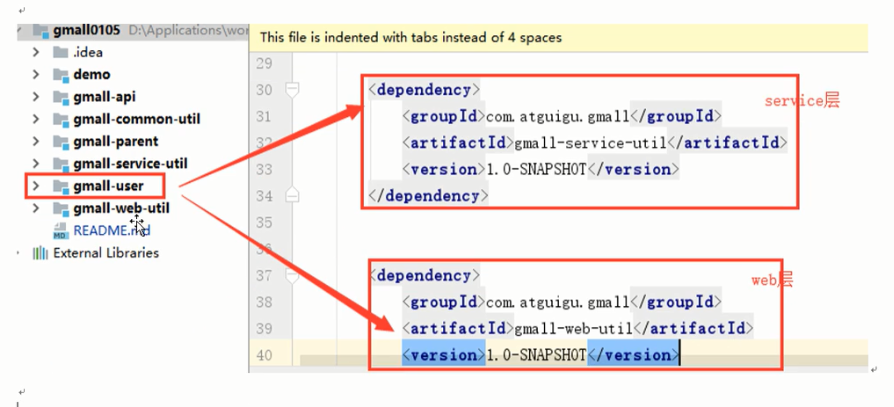
|  |
| --- |
| 1 首先用maven创建一个gmall-api的工程  2 应用tk通用的mapper(映射类)  3 将XXXService接口和所有的bean类都放到api中  4 service\serviceImp\controller\mapper\mapper.xml中所有的bean的引入全部修改引入路径 |

# 抽取util工程

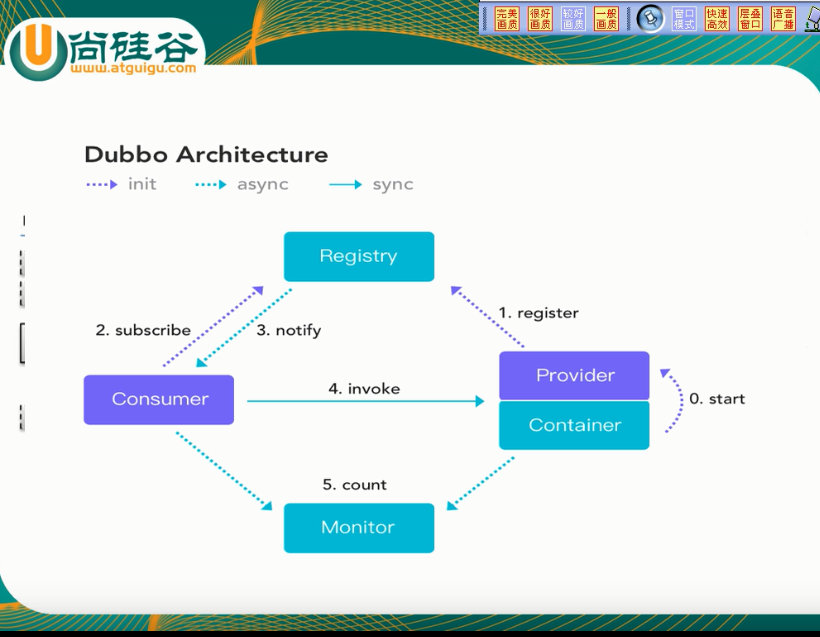
|  |  |  |  |
| --- | --- | --- | --- |
| 1 项目中的通用框架，是所有应用工程需要应用的包  例如：springboot\common-langs\common-beanutils   |  | | --- | | /groupId>  <artifactId>httpclient</artifactId>  </dependency>   <dependency>  <groupId>org.apache.commons</groupId>  <artifactId>commons-lang3</artifactId>  </dependency>   <dependency>  <groupId>commons-beanutils</groupId>  <artifactId>commons-beanutils</artifactId>  </dependency>   <dependency>  <groupId>commons-codec</groupId>  <artifactId>commons-codec</artifactId>  </dependency>   </dependencies> |   2 基于soa的架构理念，项目分为web前端controller(webUtil)  Jsp\thymeleaf\cookie工具类  加入commonUtil   |  | | --- | | <parent>  <groupId>com.atguigu.gmall</groupId>  <artifactId>gmall-parent</artifactId>  <version>1.0-SNAPSHOT</version> </parent>  <groupId>com.atguigu.gmall</groupId> <artifactId>gmall-web-util</artifactId> <version>1.0-SNAPSHOT</version>  <dependencies>  <dependency>  <groupId>com.atguigu.gmall</groupId>  <artifactId>gmall-common-util</artifactId>  <version>1.0-SNAPSHOT</version>  </dependency>   <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-thymeleaf</artifactId>  </dependency> </dependencies> |   2 基于soa的架构理念，项目分为web后端service(serviceUtil)  Mybatis\mysql\redis  加入commonUtil   |  | | --- | | <parent>  <groupId>com.atguigu.gmall</groupId>  <artifactId>gmall-parent</artifactId>  <version>1.0-SNAPSHOT</version> </parent>   <dependencies>   <dependency>  <groupId>com.atguigu.gmall</groupId>  <artifactId>gmall-common-util</artifactId>  <version>1.0-SNAPSHOT</version>  </dependency>    <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-jdbc</artifactId>  </dependency>    <dependency>  <groupId>org.mybatis.spring.boot</groupId>  <artifactId>mybatis-spring-boot-starter</artifactId>  <version>2.1.0</version>  </dependency>   <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  <scope>runtime</scope>  </dependency>   <dependency>  <groupId>redis.clients</groupId>  <artifactId>jedis</artifactId>  </dependency> </dependencies> | |



|  |
| --- |
| 1 新建一个web的前端controller模块的项目  Controller=parent+ api +WebUtil  2 新建一个web的后端端service模块的项目  service=parent+ api +WebUtil |



# Soa面向服务(以dubbo为基础)



|  |
| --- |
| 1 dubbo的soa的工作原理，和springcloud类似  2 dubbo和springcloud的区别在于dubbo由自己的dubbo协议通讯，spingcloud是由htt协议通讯(rest风格)  3 dubbo有一个注册中心的客户端在实时同步注册中心的服务信息  4 dubbo有一个javaweb的监控中心，负责监控服务的注册信息，甚至可以配置负载均衡 |

7 将dbubbo框架引入到项目中

|  |
| --- |
| 1 将dubbo监控中心和tomcat上传至linux服务器  2 用unzip 命令解压dubbo-admin.war监控中心  Upzip file文件名 -d 配置名称 （ unzip dubbo-admin-2.6.0.war -d dubbo）  3 配置tomcat的server.xml配置文件  <Context path="/dubbo" docBase="/opt/dubbo" debug="0" privileged="true"/>  4 启动tomcat ，打开监控中心 |

