

开发快速入手

1.环境要求

安装 jdk

Jdk8+（有些地方用了 jdk8 的新特性）

Maven3+

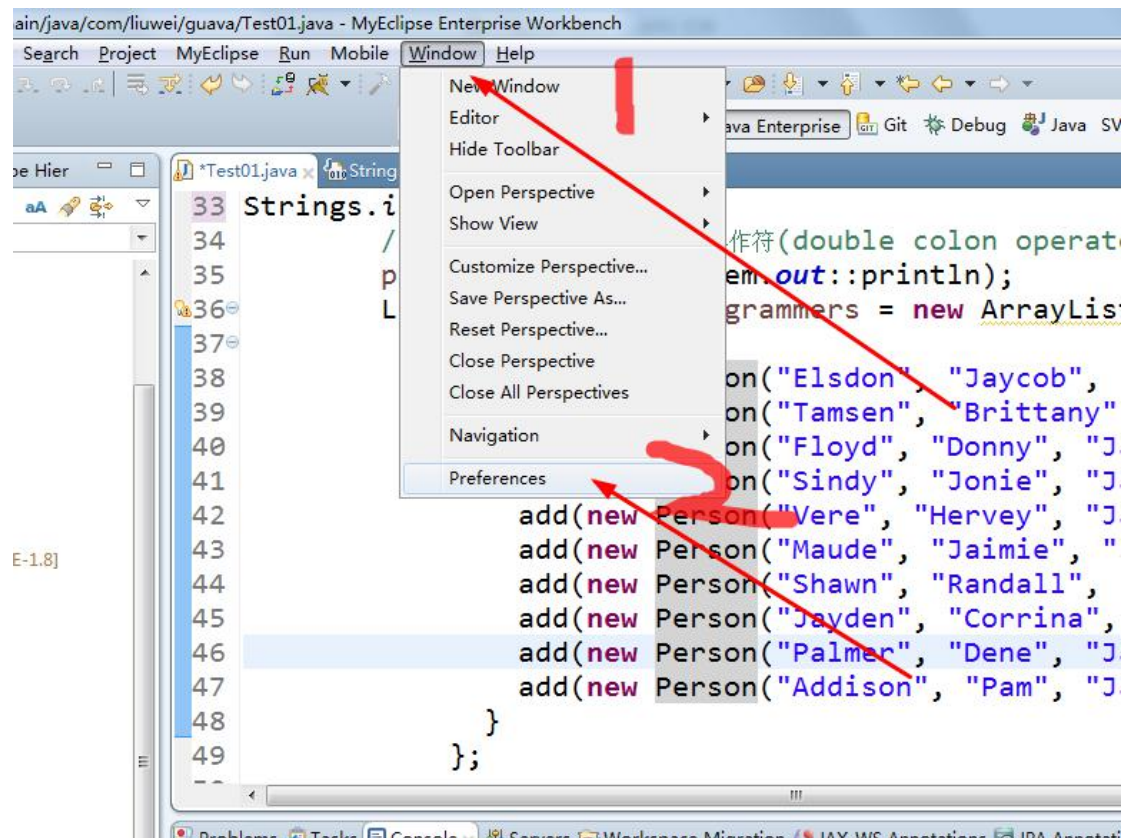
Eclipse，sts，myeclipse，DEA（推荐）

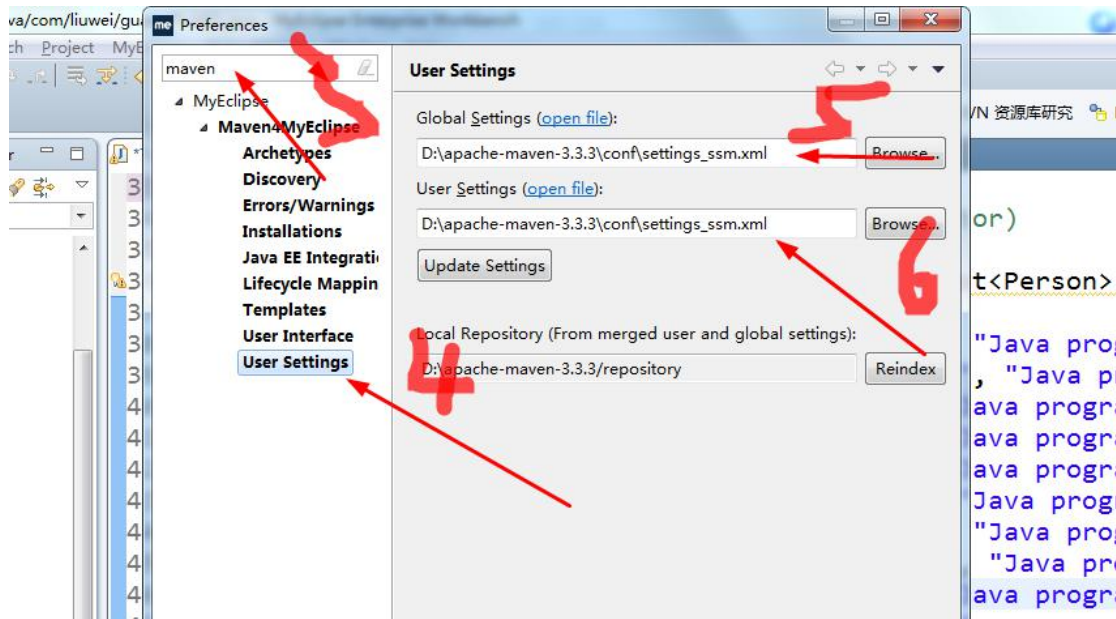
2.Maven 配置

1.如果只引用的话配置仓库地址就可以了

```
<repository>
  <id>nexus-releases</id>
  <name>Nexus Release Repository</name>
  <url>http://10.1.10.80:8081/nexus/content/groups/public/</url>
</repository>
```

2.如果需要发布到私有仓库，需要更改相应的配置文件，指向私服仓库操作如下：

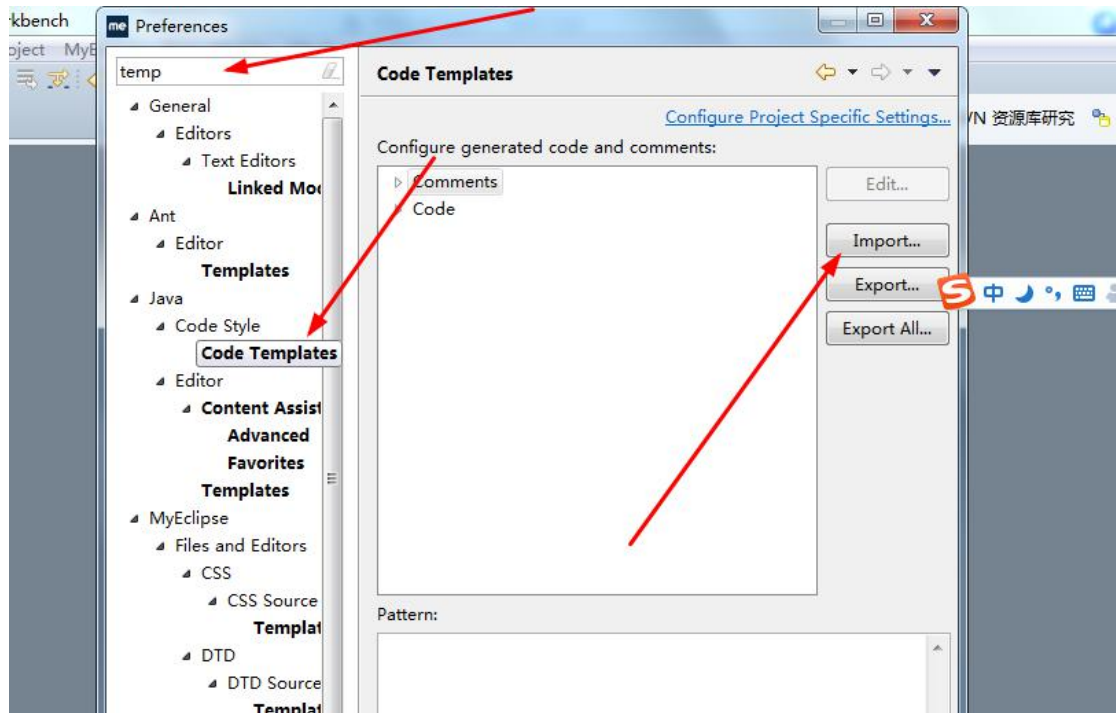


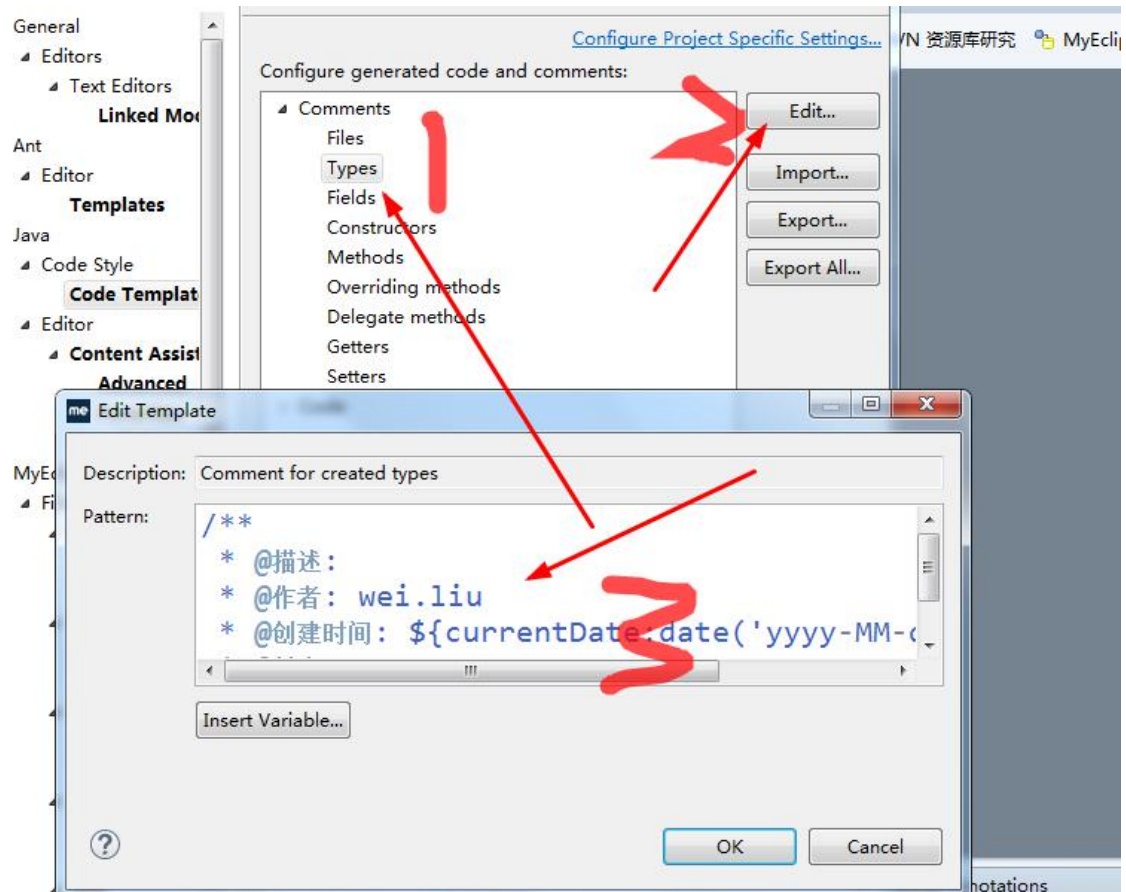


3.代码注释模板配置

1.把 **codetemplates.xml** 导入进去,

2.并改成自己的名字





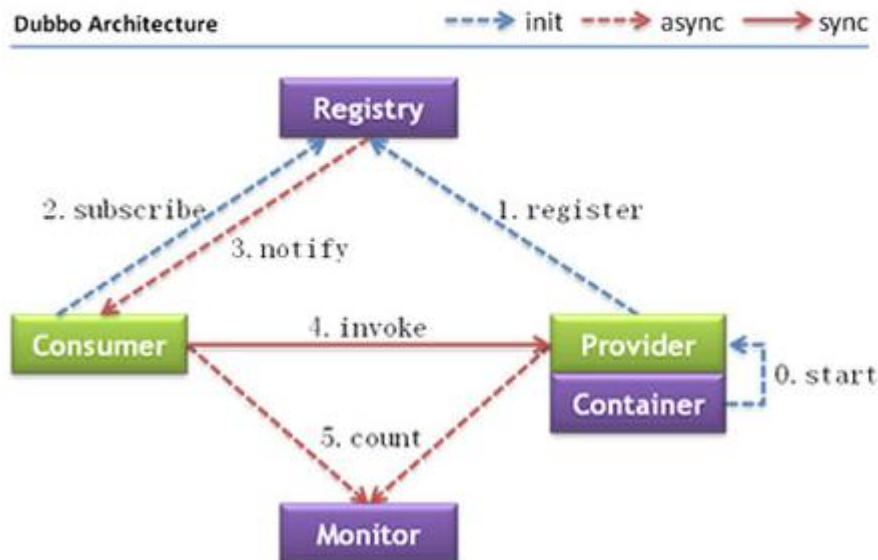
4.开发基础框架介绍

Spring4.3+
Springmvc4.3+
Mybatis3.1.1+
Shiro1.2.3+
Easyui1.3.4+
Guava18.0+
Druid
Logback, log4j2

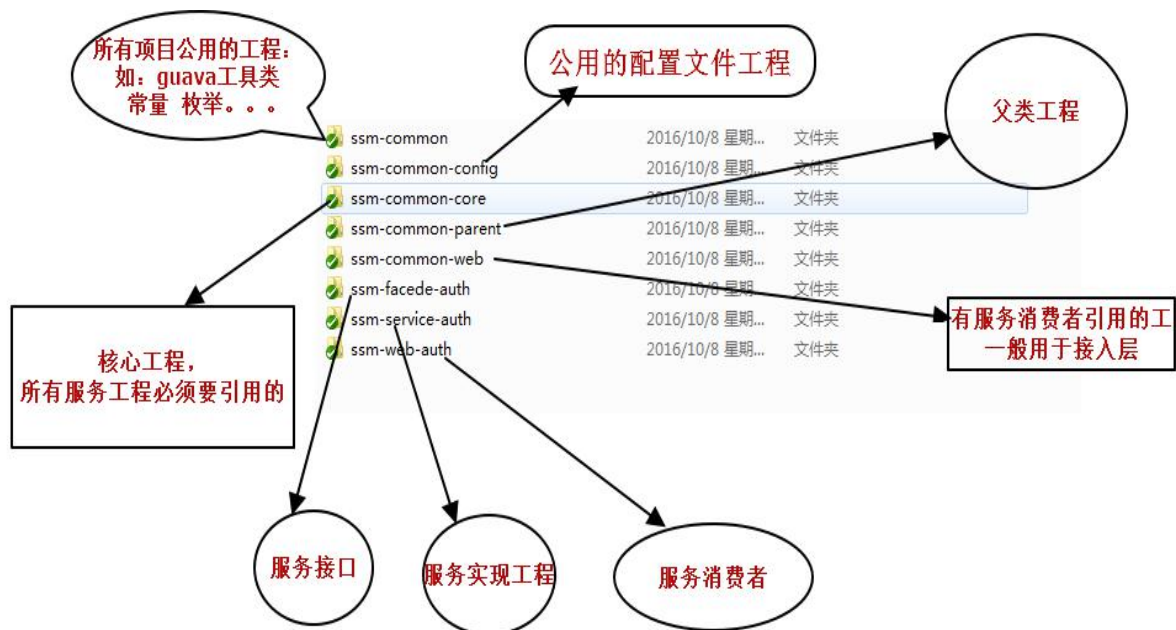
5.Dubbo 基本架构介绍

架构

(#)



6.开发工程结构介绍



命名规范:项目名+facade/service/web/timer..+服务名字

7.Swagger 使用

第一步引入 war 包

```
<dependency>  
    <groupId>org.lanqiao</groupId>  
    <artifactId>ssm-common-swagger</artifactId>  
    <type>war</type>  
</dependency>
```

第二步引入合并 war 包

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-war-plugin</artifactId>
  <version>2.4</version>
  <configuration>
    <overlays>
      <overlay>
        <groupId>org.lanqiao</groupId>
        <artifactId>ssm-common-swagger</artifactId>
      </overlay>
    </overlays>
  </configuration>
</plugin>
```

第三步 在 spring 配置文件中初始化配置

加上下面这行代码

```
<bean class="org.lanqiao.ssm.common.web.config.SwaggerConfig"></bean>
```

官方资料

<http://swagger.io/>

8.代码生成使用

ssm-common-generator 这个是工程

1.配置文件详解

generatorConfig.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE generatorConfiguration PUBLIC "-//mybatis.org//DTD MyBatis Generator Configuration 1.0//EN" "http://mybatis.org/dtd/mybatis-generator-config_1_0.dtd">
```

深圳市蓝泰源信息技术有限公司

```
<generatorConfiguration>
  <!-- 数据库驱动包位置 -->
  <classPathEntry location="D:\generator\mysql-jdbc.jar" />
  <!--<classPathEntry location="C:\oracle\product\10.2.0\db_1\jdbc\lib\ojdbc14.jar"
    /> -->
  <context id="DB2Tables" targetRuntime="MyBatis3">
    <commentGenerator>
      <property name="suppressAllComments" value="true" />
    </commentGenerator>
    <!-- 数据库链接 URL、用户名、密码 -->
    <jdbcConnection driverClass="com.mysql.jdbc.Driver"
      connectionURL="jdbc:mysql://localhost:3306/ssm" userId="root"
      password="1234567">
      <!--<jdbcConnection
        driverClass="oracle.jdbc.driver.OracleDriver"
        connectionURL="jdbc:oracle:thin:@localhost:1521:orcl"
        userId="msa" password="msa"> -->
    </jdbcConnection>
    <javaTypeResolver>
      <property name="forceBigDecimals" value="false" />
    </javaTypeResolver>
    <!-- 生成模型的包名和位置 -->
    <javaModelGenerator targetPackage="org.lanqiao.ssm.model"
      targetProject="D:\git\ssm-common-generator\src\main\java">
      <property name="enableSubPackages" value="true" />
      <property name="trimStrings" value="true" />
    </javaModelGenerator>
    <!-- 生成的映射文件包名和位置 -->
    <sqlMapGenerator targetPackage="org.lanqiao.ssm.mapping"
      targetProject="D:\git\ssm-common-generator\src\main\java">
      <property name="enableSubPackages" value="true" />
    </sqlMapGenerator>
    <!-- 生成 DAO 的包名和位置 -->
    <javaClientGenerator type="XMLMAPPER"
      targetPackage="org.lanqiao.ssm.dao"
      targetProject="D:\git\ssm-common-generator\src\main\java">
      <property name="enableSubPackages" value="true" />
    </javaClientGenerator>

    <!-- 要生成那些表(更改 tableName 和 domainObjectName 就可以) -->
    <table tableName="t_auth_user_role" domainObjectName="UserRole"
      enableCountByExample="false" enableUpdateByExample="false"
      enableDeleteByExample="false" enableSelectByExample="false"
      selectByExampleQueryId="false" />
    <table tableName="t_auth_role_resource" domainObjectName="RoleResource"
```


深圳市蓝泰源信息技术有限公司

```
enableCountByExample="false" enableUpdateByExample="false"
enableDeleteByExample="false" enableSelectByExample="false"
selectByExampleQueryId="false" />
<table tableName="t_auth_role_menu" domainObjectName="RoleMenu"
enableCountByExample="false" enableUpdateByExample="false"
enableDeleteByExample="false" enableSelectByExample="false"
selectByExampleQueryId="false" />

<!-- <table tableName="t_auth_%"> </table> -->
</context>
</generatorConfiguration>
```

2.使用不同配置文件生成不同项目的 MyBatis 文件

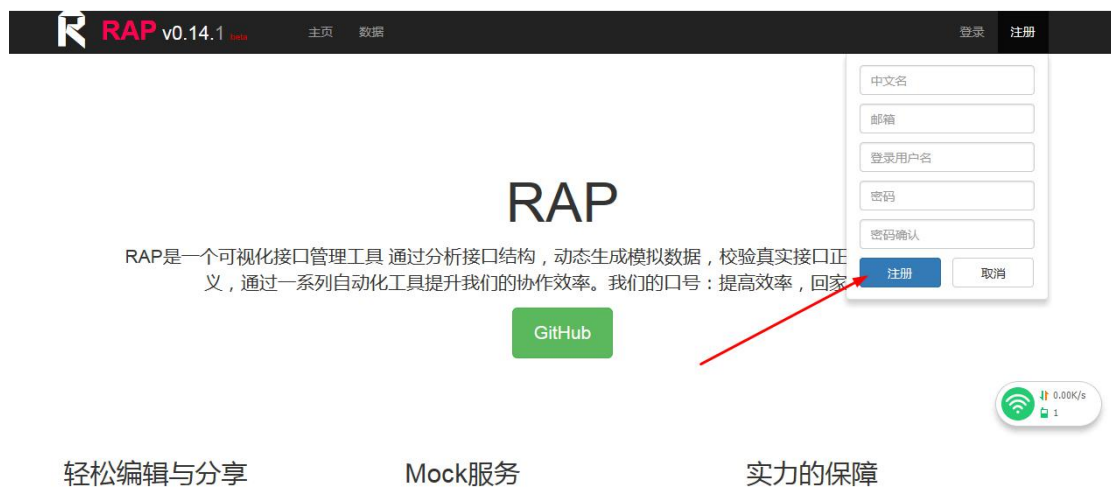
```
install mybatis-generator:generate -DconfigurationFile=generatorConfig-scheduler.xml
install mybatis-generator:generate -DconfigurationFile=generatorConfig-sys.xml
install mybatis-generator:generate -DconfigurationFile=generatorConfig-biz.xml
install mybatis-generator:generate -DconfigurationFile=generatorConfig.xml
```

具体配置查看详细工程

9.前后端分离

1.可视化接口测试平台

<http://10.1.10.79:2016> 先注册:



深圳市蓝泰源信息技术有限公司

响应参数列表

变量名	含义	类型	备注
data [1-5]	构造一个数组，含有 1-5 个元素	array<object>	
cn [1-5]	重复1-5次 '汉字'	string	@mock=汉字
date	随即生成一段日期字符串，默认格式为 yyyy-MM-dd	string	@mock=@DATE
datetime	随机生成一段时间字符串，默认格式为 yyyy-MM-dd HH:mm:ss	string	@mock=@DATETIME
dummyimage	复合对象属性	object	
background	随机生成一个颜色作为背景色	string	@mock=@COLOR
foreground	随机生成一个颜色作为前景色	string	@mock=@COLOR
format [1]	从数组中随机挑选一个元素做为属性值	array<string>	@mock=png,gif,jpg
size	图片尺寸	string	@mock=@AD_SIZE
text	随机获得一段连续的字符	string	@mock=@WORD
url	引用mockjs已生成的数值	string	@mock=http://ex.com/@size/@background
email	随即生成一个 Email	string	@mock=@EMAIL
float [1],1-10	随机生成一个浮点数，整数部分为1，保留1到10位小数	number	

2.最终生成的 Mock 数据

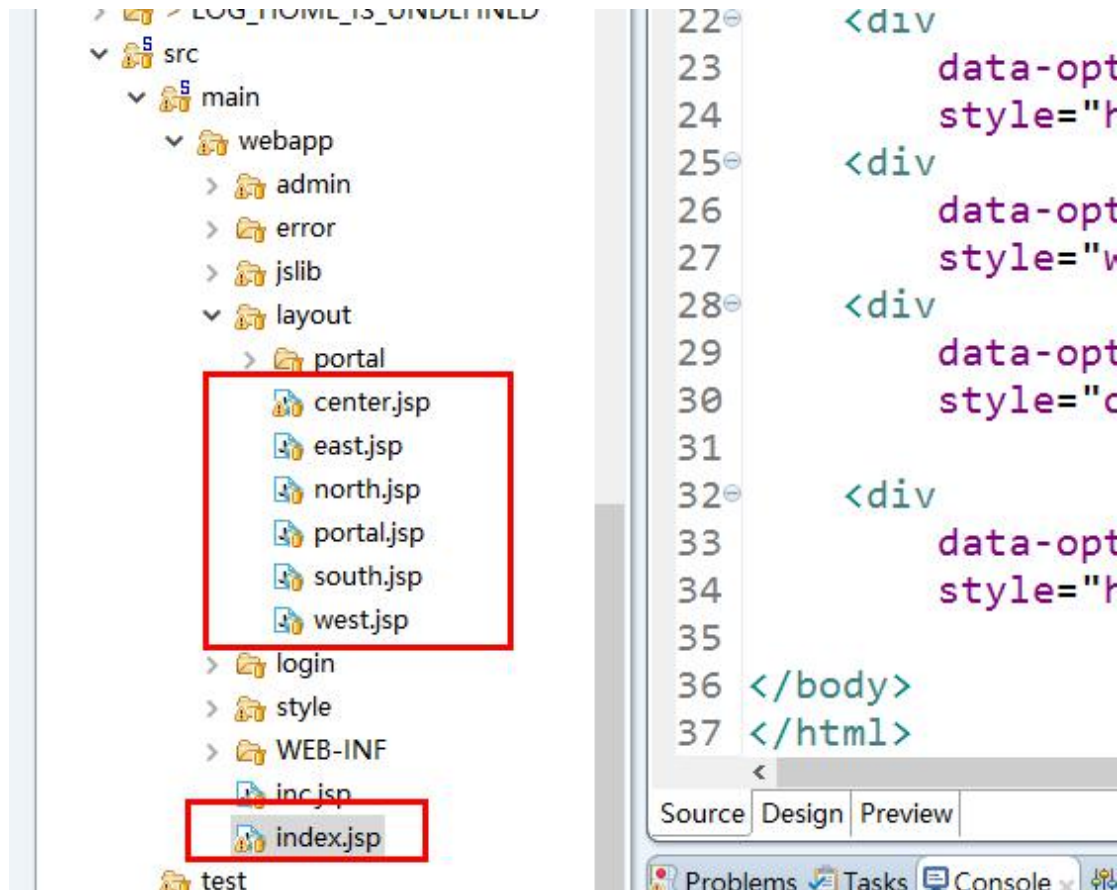
```
{
  "data": [
    {
      "cn": "汉字汉字", "date": "1987-09-22", "datetime": "2002-06-04 14:36:58", "dummyimage": {
        "background": "#b776ae", "foreground": "#4ad59d", "format": ["png,gif,jpg"], "size": "120x90", "text": "btrd", "url": "http://ex.com/120x90/#b776ae"
      }, "email": "e.thompson@anderson.com", "float1": 54164.5113442, "float2": 42.8886, "float3": 999.321, "float4": 40530.72383, "grade1": 91, "grade2": 94, "id": 100, "img1": "http://dummyimage.com/200x200", "img2": "http://dummyimage.com/468x60", "img3": "http://dummyimage.com/@size", "img4": "http://dummyimage.com/728x90", "method": "POST", "param": "abc=123", "published": true, "repeat": "AAAAAAAAA", "size": "468x60", "star": "★★", "time": "22:46:43", "title": "Exsjfjq Vbdpkf Loebdyex Yxqvfyxno Rqpub", "url1": "http://dummyimage.com/@size?abc=123", "url2": "http://dummyimage.com/728x90?810000197607039271&id=100"
    },
    {
      "cn": "汉字汉字", "date": "2002-11-17", "datetime": "1998-06-16 20:12:18", "dummyimage": {
        "background": "#b8ab36", "foreground": "#4779ea", "format": ["png,gif,jpg"], "size": "300x600", "text": "ytijn", "url": "http://ex.com/300x600/#b8ab36"
      }, "email": "f.taylor@perez.net", "float1": 54164.76383534, "float2": 11.85486, "float3": 999.371, "float4": 40530.8258741694, "grade1": 16, "grade2": 95, "id": 101, "img1": "http://dummyimage.com/200x200", "img2": "http://dummyimage.com/234x60", "img3": "http://dummyimage.com/@size", "img4": "http://dummyimage.com/234x60", "method": "DELETE", "param": "abc=123", "published": true, "repeat": "AAAAAAAAA", "size": "720x300", "star": "★★★★", "time": "04:43:57", "title": "Gxlv Mcfudzfs Dpam Zaydyji Yutloqqzrv Syniwpvpv Paxqnkl chn", "url1": "http://dummyimage.com/@size?abc=123", "url2": "http://dummyimage.com/234x60?410000197606133333&id=101"
    }
  ]
}
```

3.官方参考资料文档

<http://mockjs.com/>

10.Easyui 快速入门

1.整个布局和基本示例



2.官方参考资料文档

<http://www.jeasyui.com/>

11.dubbo 工程入手

1.服务接口

(#)

完整安装步骤，请参见：[示例提供者安装 \(+\)](#)

定义服务接口: (该接口需单独打包，在服务提供方和消费方共享)

```
package com.alibaba.dubbo.demo;

public interface DemoService {

    String sayHello(String name);

}
```

DemoService

2.服务实现

在服务提供方实现接口: (对服务消费方隐藏实现)

```
package com.alibaba.dubbo.demo.provider;

import com.alibaba.dubbo.demo.DemoService;

public class DemoServiceImpl implements DemoService {

    public String sayHello(String name) {
        return "Hello " + name;
    }

}
```

DemoService

深圳市蓝泰源信息技术有限公司

用Spring配置声明暴露服务：

```
provider.xml

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans
http://code.alibabatech.com/schema/dubbo http://code.alibabatech.com/schema/dubbo/dubbo.xsd"

    <!-- 提供方应用信息，用于计算依赖关系 -->
    <dubbo:application name="hello-world-app" />

    <!-- 使用multicast广播注册中心暴露服务地址 -->
    <dubbo:registry address="multicast://224.5.6.7:1234" />

    <!-- 用dubbo协议在20880端口暴露服务 -->
    <dubbo:protocol name="dubbo" port="20880" />

    <!-- 声明需要暴露的服务接口 -->
    <dubbo:service interface="com.alibaba.dubbo.demo.DemoService" ref="demoService" />

    <!-- 和本地bean一样实现服务 -->
    <bean id="demoService" class="com.alibaba.dubbo.demo.provider.DemoServiceImpl" />

</beans>
```

加载Spring配置：

```
Provider.java

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Provider {

    public static void main(String[] args) throws Exception {
        ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext(new String[] {
            "provider.xml"
        });
        context.start();

        System.in.read(); // 按任意键退出
    }
}
```


3.服务消费

服务消费者

(#)

✔ 完整安装步骤，请参见：示例消费者安装 (+)

通过Spring配置引用远程服务：

```
consumer.x

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans
http://code.alibabatech.com/schema/dubbo http://code.alibabatech.com/schema/dubbo/dubbo.xs

    <!-- 消费方应用名，用于计算依赖关系，不是匹配条件，不要与提供方一样 -->
    <dubbo:application name="consumer-of-helloworld-app" />

    <!-- 使用multicast广播注册中心暴露发现服务地址 -->
    <dubbo:registry address="multicast://224.5.6.7:1234" />

    <!-- 生成远程服务代理，可以和本地bean一样使用demoService -->
    <dubbo:reference id="demoService" interface="com.alibaba.dubbo.demo.DemoService" />

</beans>
```

加载Spring配置，并调用远程服务：(也可以使用IoC注入)

```
Consumer.java

import org.springframework.context.support.ClassPathXmlApplicationContext;
import com.alibaba.dubbo.demo.DemoService;

public class Consumer {

    public static void main(String[] args) throws Exception {
        ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext(new String[] {"http://1
context.start();

        DemoService demoService = (DemoService)context.getBean("demoService"); // 获取远程服务代理
        String hello = demoService.sayHello("world"); // 执行远程方法

        System.out.println( hello ); // 显示调用结果
    }
}
```

注 意 点 ： <import
resource="classpath:spring/dubbo-consumer.xml" />

4.官方参考资料

<http://dubbo.io/>