ssm整合redis

准备工作：

jedis连接

添加jar支持

```
        <!-- redis依赖 -->
                <dependency>
                    <groupId>org.springframework.data</groupId>
                    <artifactId>spring-data-redis</artifactId>
                    <version>1.6.0.RELEASE</version>
                </dependency>
                <dependency>
                    <groupId>redis.clients</groupId>
                    <artifactId>jedis</artifactId>
                    <version>2.7.3</version>
                </dependency>
```

如果redis服务想被外部链接访问，需要修改redis.conf配置 （69行和88行）

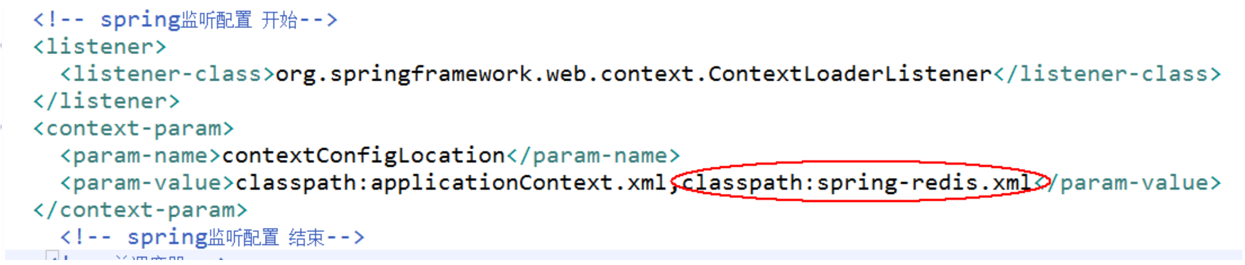
注释掉 #bind localhost或者127.0.0.1   69行

修改保护：protected-mode no 或者上面bind 相应IP 这里的保护可以默认

编码测试：

```
Jedis jedis=new Jedis("192.168.23.111",6379);
 //jedis.ping();
JedisPool jedisPool=new JedisPool("192.168.23.111",6379);
Jedis resource = jedisPool.getResource();
```

1，在web.xml里面添加



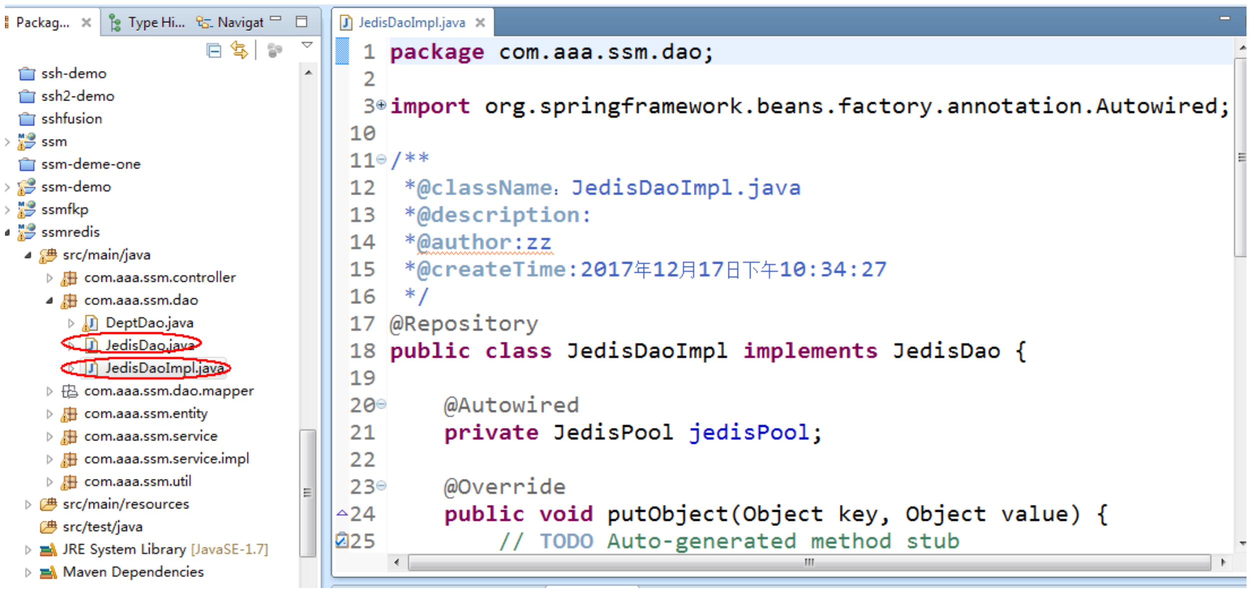
```
<!-- spring监听配置 开始-->
<listener>
  <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>classpath:applicationContext.xml,classpath:spring-redis.xml</param-value>
</context-param>
  <!-- spring监听配置 结束-->
```

因为前面课程没有多个spring配置文件，命名规范不统一 ，所以这样配置

2，添加spring-redis.xml

```
<!-- 开启扫描 -->
    <context:component-scan base-package="com.aaa.ssm.dao">
</context:component-scan>
    <!-- 初始化Jedis连接池-->
    <bean id="poolConfig" class="redis.clients.jedis.JedisPoolConfig">
             <!--最大连接数, 默认8个-->
           <property name="maxTotal" value="50" />
        <!--最大空闲连接数, 默认8-->
            <property name="maxIdle" value="10" />
             <!--连接时的最大等待毫秒数-->
            <property name="maxWaitMillis" value="1000" />
            <!--获得一个jedis实例的时候是否检查连接可用性-->
            <property name="testOnBorrow" value="true" />
    </bean>
    <!-- 把jedisPool交给spring管理 -->
    <bean   class="redis.clients.jedis.JedisPool" >
        <constructor-arg name="poolConfig" ref="poolConfig">
</constructor-arg>
        <constructor-arg name="host" value="192.168.152.180">
</constructor-arg>
        <constructor-arg name="port" value="6379"></constructor-arg>
    </bean>
```

3，编写JedisDao及实现类



```
package com.aaa.ssm.dao;

import org.springframework.beans.factory.annotation.Autowired;

/**
 *@className: JedisDaoImpl.java
 *@description:
 *@author:zz
 *@createTime:2017年12月17日下午10:34:27
 */
@Repository
public class JedisDaoImpl implements JedisDao {

    @Autowired
    private JedisPool jedisPool;

    @Override
    public void putObject(Object key, Object value) {
        // TODO Auto-generated method stub
```

JedisDao

```
    package com.aaa.ssm.dao;
/**
 *@className: JedisDao.java
 *@description:
 *@author:zz
 *@createTime:2017年12月17日下午10:33:35
 */
 interface JedisDao {
        /**
         * 放入缓存
         * @param key
         * @param value
         */
        void putObject(Object key, Object value);
        /**
         * 清除缓存
         * @param arg0
         * @return
         */
        Object removeObject(Object arg0);
        /**
         * 从缓存中获取
         * @param arg0
```

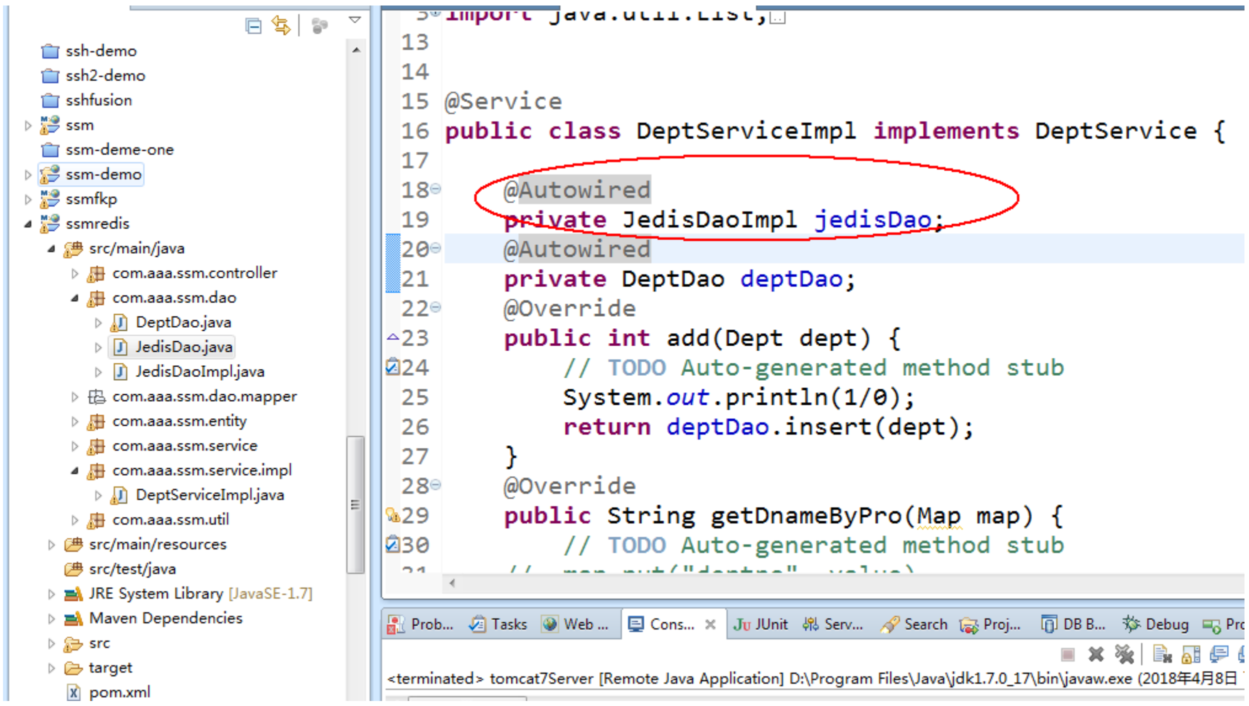
```
          * @return
          */
         Object getObject(Object arg0);
}
```

JedisDaoImpl

```
package com.aaa.ssm.dao;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Repository;

import redis.clients.jedis.Jedis;
import redis.clients.jedis.JedisPool;

import com.aaa.ssm.util.SerializeUtil;

/**
 *@className：JedisDaoImpl.java
 *@description:
 *@author:zz
 *@createTime:2017年12月17日下午10:34:27
 */
@Repository
public class JedisDaoImpl implements JedisDao {

        @Autowired
        private JedisPool jedisPool;

        @Override
        public void putObject(Object key, Object value) {
                // TODO Auto-generated method stub
                Jedis resource = jedisPool.getResource();
                resource.set(SerializeUtil.serialize(key.toString()),
                SerializeUtil.serialize(value));
                resource.close();
        }
        @Override
        public Object removeObject(Object arg0) {
                // TODO Auto-generated method stub
                 Jedis resource = jedisPool.getResource();
                 Object expire = resource.expire(
                        SerializeUtil.serialize(arg0.toString()), 0);
                 resource.close();
                return expire;
        }
        @Override
        public Object getObject(Object arg0) {
                // TODO Auto-generated method stub
                  Jedis resource = jedisPool.getResource();
                Object value = SerializeUtil.unserialize(resource.get(
                        SerializeUtil.serialize(arg0.toString())));
                resource.close();
                return value;
        }

}
/**
 *@className：SerializeUtil.java
 *@description:
 *@author:zz
 *@createTime:2017年12月14日下午5:25:01
 */
public class SerializeUtil {
          public static byte[] serialize(Object object) {
                ObjectOutputStream oos = null;
                ByteArrayOutputStream baos = null;
                try {
                    // 序列化
                    baos = new ByteArrayOutputStream();
                    oos = new ObjectOutputStream(baos);
                    oos.writeObject(object);
```

```
            byte[] bytes = baos.toByteArray();
            return bytes;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static Object unserialize(byte[] bytes) {
        if (bytes == null)
            return null;
        ByteArrayInputStream bais = null;
        try {
            // 反序列化
            bais = new ByteArrayInputStream(bytes);
            ObjectInputStream ois = new ObjectInputStream(bais);
            return ois.readObject();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }
}
```

4，在服务层注入接口测试缓存功能



```
ssh-demo
ssh2-demo
sshfusion
ssm
ssm-deme-one
ssm-demo
ssmfkp
ssmredis
  src/main/java
    com.aaa.ssm.controller
    com.aaa.ssm.dao
      DeptDao.java
      JedisDao.java
      JedisDaoImpl.java
    com.aaa.ssm.dao.mapper
    com.aaa.ssm.entity
    com.aaa.ssm.service
    com.aaa.ssm.service.impl
      DeptServiceImpl.java
    com.aaa.ssm.util
  src/main/resources
  src/test/java
  JRE System Library [JavaSE-1.7]
  Maven Dependencies
  src
  target
  pom.xml
```

```java

@Service
public class DeptServiceImpl implements DeptService {

    @Autowired
    private JedisDaoImpl jedisDao;
    @Autowired
    private DeptDao deptDao;
    @Override
    public int add(Dept dept) {
        // TODO Auto-generated method stub
        System.out.println(1/0);
        return deptDao.insert(dept);
    }
    @Override
    public String getDnameByPro(Map map) {
        // TODO Auto-generated method stub
```

```
<terminated> tomcat7Server [Remote Java Application] D:\Program Files\Java\jdk1.7.0_17\bin\javaw.exe (2018年4月8日
```

具体代码

```java
@Override
	public Dept getById(int deptno) {
		// TODO Auto-generated method stub
		//获取缓存对象
		Object object = jedisDao.getObject("dept");
		//存在返回
		if(object!=null)
			return (Dept)object;
		//获取对象
		Dept byId = deptDao.getById(deptno);
		//不存在放入
		if(object==null)
			jedisDao.putObject("dept", byId);
		return byId;
	}
```