集成 Redis 二级缓存

一、阐述

○ 原始项目中没有引入缓存的概念，每一次执行操作都是与数据库交互（如下图），

```
Creating a new SqlSession
SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@3a629162] was not registered for synchronization because synchronization is not active
Cache Hit Ratio [cn.shiyun.dao.UserDao]: 0.0
JDBC Connection [ProxyConnection[PooledConnection[com.mysql.jdbc.JDBC4Connection@79672e56]]] will not be managed by Spring
==>  Preparing: SELECT UserName,Password,Description from Users WHERE enabled=? and UserName=? and Password=?
==> Parameters: true(Boolean), admin(String), 1(String)
<==    Columns: UserName, Password, Description
<==        Row: admin, 1, null
<==      Total: 1
Closing non transactional SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@3a629162]
```
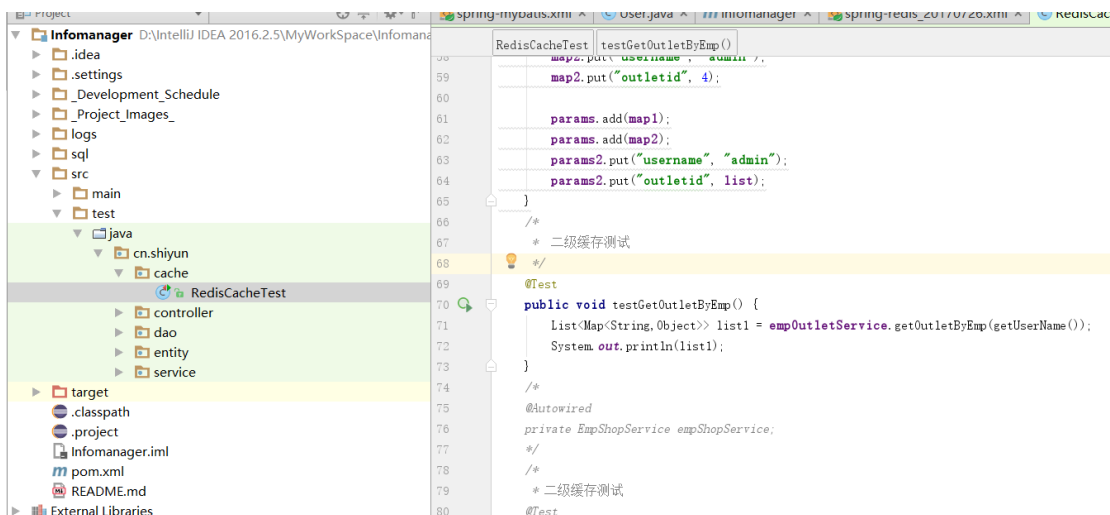
引入 Redis 缓存机制后，对于相同的操作没必要每次都与数据库交互（如下图）；

```
Creating a new SqlSession
SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@436646f7] was not registered for synchronization because synchronization is not active
Cache Hit Ratio [cn.shiyun.dao.UserDao]: 0.5
Closing non transactional SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@436646f7]
Creating a new SqlSession
SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@6de26eae] was not registered for synchronization because synchronization is not active
JDBC Connection [ProxyConnection[PooledConnection[com.mysql.jdbc.JDBC4Connection@79672e56]]] will not be managed by Spring
==>  Preparing: select authority from authorities where authority !='Admin'
==> Parameters:
<==    Columns: authority
<==        Row: Anonymous
<==        Row: Authenticated
           Row: bb
```

Redis 数据库中数据：

缓存的数据

○ 添加单元测试

```
        map2.put("username", "admin");
        map2.put("outletid", 4);

        params.add(map1);
        params.add(map2);
        params2.put("username", "admin");
        params2.put("outletid", list);
    }

    /*
     * 二级缓存测试
     */
    @Test
    public void testGetOutletByEmp() {
        List<Map<String,Object>> list1 = empOutletService.getOutletByEmp(getUserName());
        System.out.println(list1);
    }

    /*
    @Autowired
    private EmpShopService empShopService;
    */

    /*
     * 二级缓存测试
    @Test
```

二、计划项（工作重心放到前端的优化上，并接上之前的（2017.07.21）的工作安排.）

◆计划开发时间：2017.07.28 至 2017.07.30

√ 消除原始项目中的 java 的代码片段，并将 xxx.jsp 文件优化成 xxx.html

◆计划开发时间：2017.07.31 至 ~

√ 研究基于 Anngular.js 的单页面(SPA)应用并用其优化前端代码

三、项目地址（暂时将优化的项目放到个人的 github 上，后期 master 下.）

https://github.com/TaoPengFei/Infomanager