编写jce实现接口并调试修改

### 插入用户

测试

  it('saveUser', async () => {

        const req = **new** *Test*.*SaveUserInfoReq*();

        req.readFromObject({

            userinfo: {

                username: 'admin',

                password: '123456',

                gender: 'man',

                tel: '156xxx',

                age: '22'

            }

        });

        let ret = await proxy.saveUser(req);

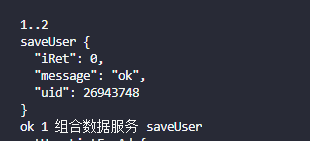
        ret = ret.response.arguments.stRsp.toObject();

*console*.log('saveUser', JSON.stringify(ret, null, 2));

        ret.iRet.should.eql(0);

    });

结果



### 修改用户

测试

    it('updateUser', async () => {

        const req = **new** *Test*.*SaveUserInfoReq*();

        req.readFromObject({

            userinfo: {

                id: 1,

                username: 'admin',

                password: '123456',

                gender: 'man',

                tel: '156xxx',

                age: '22'

            }

        });

        let ret = await proxy.saveUser(req);

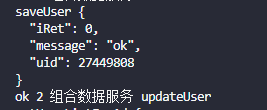
        ret = ret.response.arguments.stRsp.toObject();

*console*.log('saveUser', JSON.stringify(ret, null, 2));

        ret.iRet.should.eql(0);

    });

结果



### 获取用户列表

测试

  // 获取用户列表

    it('getUserListForAd', async () => {

        const req = **new** *Test*.*PageInfoReq*();

        req.readFromObject({

            page: 1,

            pageNum: 10

        });

        let ret = await proxy.getUserListForAd(req);

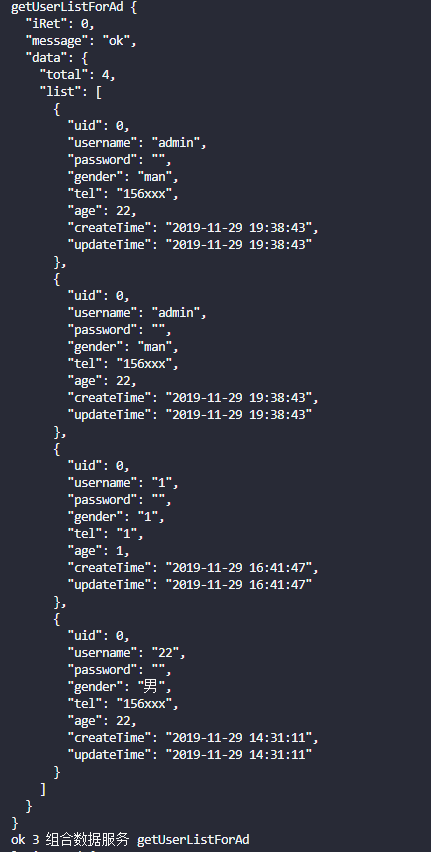
        ret = ret.response.arguments.stRsp.toObject();

*console*.log('getUserListForAd', JSON.stringify(ret, null, 2));

        ret.iRet.should.eql(0);

    });

结果



### 登录用户

测试

   it('loginForAd', async () => {

        const req = **new** *Test*.*LoginReq*();

        req.readFromObject({

            username: 'admin',

            password: 123456

        });

        let ret = await proxy.loginForAd(req);

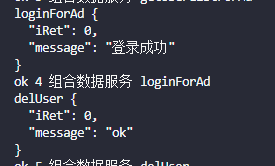
        ret = ret.response.arguments.stRsp.toObject();

*console*.log('loginForAd', JSON.stringify(ret, null, 2));

        ret.iRet.should.eql(0);

    });

结果



### 删除用户

测试

   it('delUser', async () => {

        const req = **new** *Test*.*QueryIdReq*();

        req.readFromObject({

            uid: 2,

        });

        let ret = await proxy.delUser(req);

        ret = ret.response.arguments.stRsp.toObject();

*console*.log('delUser', JSON.stringify(ret, null, 2));

        ret.iRet.should.eql(0);

    });

结果

