

本章学习目标

- 掌握登录模块的库表设计。
- 了解登录模块的功能。
- 掌握登录模块服务端的功能实现。
- 掌握 Android App 登录页面的开发。

登录模块是整个系统的门户,所有操作都需要先登录系统才得以进行。登录模块的信息验证中不仅涉及用户名和密码的验证,还为登录成功的用户匹配唯一的 token(密钥),方便在后续章节中实现权限的验证功能。本章将带领读者从创建数据表开始正式学习登录模块的开发,以及接口的测试。

3.1 登录表设计

3.1.1 设计表结构

通常一个公司的用户表包含的字段很多,例如,性别、年龄和籍贯等,但员工表必需的字段有用户名和密码。此处为了项目演示方便,暂定四个字段,分别是 fuseraccount(员工工号)、fname(员工姓名)、mobilepassword(登录密码),以及 dapaderment_id(员工部门)。

3.1.2 创建数据表

创建用户表的 SQL 语句如下:

```
1  create table T_SEC_USER(
2      id int not null identity(1,1) primary key,
3      fuseraccount nvarchar(20),                      -- 员工工号
4      fname nvarchar(20),                            -- 员工姓名
5      mobilepassword varchar(50),                   -- 登录密码
6      dapaderment_id int                           -- 员工部门
7  )
```

3.1.3 用户登录、修改密码功能接口

在 SqlserverGenerator 类中指定需要生成的数据表名-> strategy.setInclude("T_SEC_USER"); 运行 SqlserverGenerator 类中的 main()方法,此时控制面板中将打印如图 3.1 所

示的提示信息。

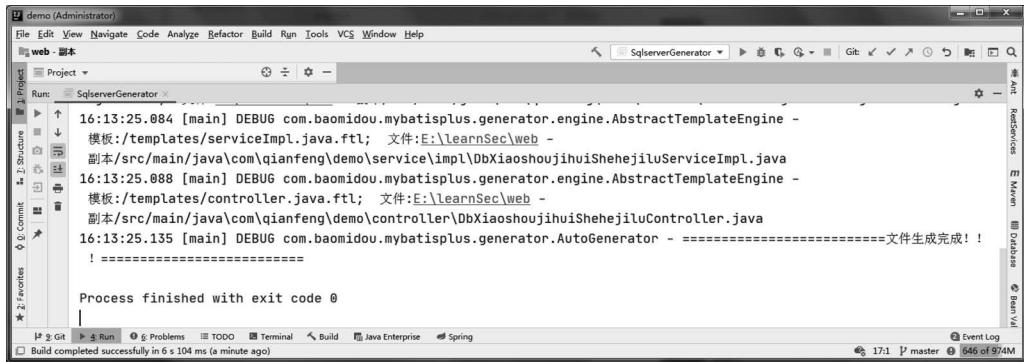


图 3.1 使用 MyBatis-Plus 生成代码

在项目的目录下可以发现,图 3.1 中的文件都已悉数生成。在项目目录下新建 utils 文件夹,创建 RedisOps 工具类,连接 Redis 数据库,并实现存取数据操作,代码如下所示。

```
1 public class RedisOps {
2     public static void set(String key, String value){
3         Jedis jedis = new Jedis ("localhost",6379);
4         jedis.set(key, value);
5         jedis.close();
6     }
7     public static String get(String key){
8         Jedis jedis = new Jedis ("localhost",6379);
9         String value = jedis.get(key);
10        jedis.close();
11        return value;
12    }
13    public static void setObject(String key, Object object){
14        Jedis jedis = new Jedis ("localhost",6379);
15        jedis.set(key.getBytes(), SerializeUtil.serialize(object));
16        jedis.close();
17    }
18    public static Object getObject(String key){
19        Jedis jedis = new Jedis ("localhost",6379);
20        byte[] bytes = jedis.get(key.getBytes());
21        jedis.close();
22        if(bytes == null) return null;
23        return SerializeUtil.deserialize(bytes);
24    }
25 }
```

在以上代码中,导入 Jedis 包,分别实现了 String 类型和 Object 类型的数据在 Redis 中的存取功能。需要注意的是,在存取 Object 类型时,需要进行字节转换。

1. 编辑 TSecUserMapper.xml 文件

在 XML 文件中添加查询用户信息和修改密码的语句,代码如下。

```

1  <! -- 登录 -->
2      <select id = "getLogin" resultMap = "getLogin"
3      parameterType = "java.lang.String">
4          SELECT fuseraccount, mobilepassword, fname FROM T_SEC_USER where
5      fuseraccount = # {fuseraccount}
6      </select>
7  <! -- 修改密码 -->
8      <update id = "updatePassword">
9          UPDATE T_SEC_user
10         SET MOBILEPASSWORD = # {mobilepassword}
11         WHERE FUSERACCOUNT = # {fuseraccount}
12     </update>

```

2. 编辑 TSecUserMapper 接口文件

```

1  @Mapper
2  public interface TSecUserMapper extends BaseMapper < TSecUser > {
3      //登录
4      User getLogin(@Param("fuseraccount") String fuseraccount);
5      //修改密码
6      int updatePassword(@Param("fuseraccount") String fuseraccount,
7      @Param("mobilepassword") String mobilepassword);
8  }

```

3. 编写 TSecUserService 接口

```

1  public interface TSecUserService extends IService < TSecUser > {
2      //用户登录
3      User getLogin(@Param("fuseraccount") String fuseraccount);
4      //修改密码
5      int updatePassword(@Param("fuseraccount") String fuseraccount,
6      @Param("mobilepassword") String mobilepassword);
7  }

```

4. 编辑 TSecUserServiceImpl 实现类

```

1  @Service
2  public class TSecUserServiceImpl extends ServiceImpl < TSecUserMapper,
3  TSecUser > implements TSecUserService {
4      //用户登录
5      @Override
6      public User getLogin(String fuseraccount) {
7          return baseMapper.getLogin(fuseraccount);
8      }
9      //修改密码
10     @Override
11     public int updatePassword(String fuseraccount, String mobilepassword)
12     {
13         return baseMapper.updatePassword(fuseraccount, mobilepassword);
14     }

```

```
15 }
16 }
```

5. 编辑 TUserController

```
1  @RestController
2  @RequestMapping("/user")
3  public class TUserController {
4      @Resource
5          private TSecUserService tSecUserService;
6          //登录
7          @PostMapping("getLogin")
8          public Response getLogin(String username, String password) {
9              //对密码进行 md5 加密
10             Jedis jedis = new Jedis("localhost", 6379);
11             String pwd = DigestUtils.md5DigestAsHex(password.getBytes());
12             Response<User> response = new Response<>();
13             User user = tSecUserService.getLogin(username);
14             if (user == null) {
15                 response.setCode(-1);
16                 response.setMessage("用户不存在");
17                 return response;
18             }
19             if (!user.getMobilepassword().equals(pwd)) {
20                 response.setCode(-1);
21                 response.setMessage("密码错误");
22                 return response;
23             } else {
24                 List<RoleAndPower> list1 =
25                     getRolesAndPowersService.getRolesAndPowers(username);
26                 List<String> roles = tSecUserService.getRoles(username);
27                 user.setRoles(roles);
28                 String token = UUID.randomUUID().toString().replaceAll("-", "");
29
30                 jedis.set(("token_" + token).getBytes(),
31                     SerializeUtil.serizlize(list1));
32                 jedis.expire(("token_" + token), 60 * 30 * 2);
33                 System.out.println("token_" + token);
34                 System.out.println(RedisOps.getObject("token_" + token));
35                 response.setCode(200);
36                 response.setHeader("token_" + token);
37                 response.setResult(user);
38                 return response;
39             }
40         }
41     @PutMapping("updatePassword")
42     @ResponseBody
43     public Response updatePassword(String fuseraccount, String
44     mobilepassword) {
45         TSecUser tSecUser = tSecUserService.queryByCount(fuseraccount);
46         Response<List<String>> response = new Response<>();
47         if (tSecUser == null) {
```

48

```

48         response.setCode(-1);
49         response.setMessage("账户不存在");
50         return response;
51     } else {
52         //对密码进行 md5 加密
53         String pwd =
54             DigestUtils.md5DigestAsHex(mobilepassword.getBytes());
55         int i = tSecUserService.updatePassword(fuseraccount, pwd);
56         if (i == 0) {
57             response.setCode(-1);
58             response.setMessage("密码修改失败");
59             return response;
60         } else {
61             response.setCode(200);
62             response.setMessage("密码修改成功");
63             return response;
64         }
65     }
66 }
67 }
```

以上代码中通过用户名和密码实现登录功能,若用户输入的用户名和密码正确则随机生成 token 字符串,将其保存在 Redis 缓存中,通过 expire()方法设置 token 字符串在 Redis 中的存储时长,最后将用户信息和 token 返回给客户端,该 token 主要用作一些增、删、改操作时的权限验证,后续的章节中将逐步讲解。

3.1.4 用户登录、修改密码功能接口测试

1. 用户登录接口测试

在 postman 地址栏中输入登录测试接口地址,调用登录接口,如图 3.2 所示。

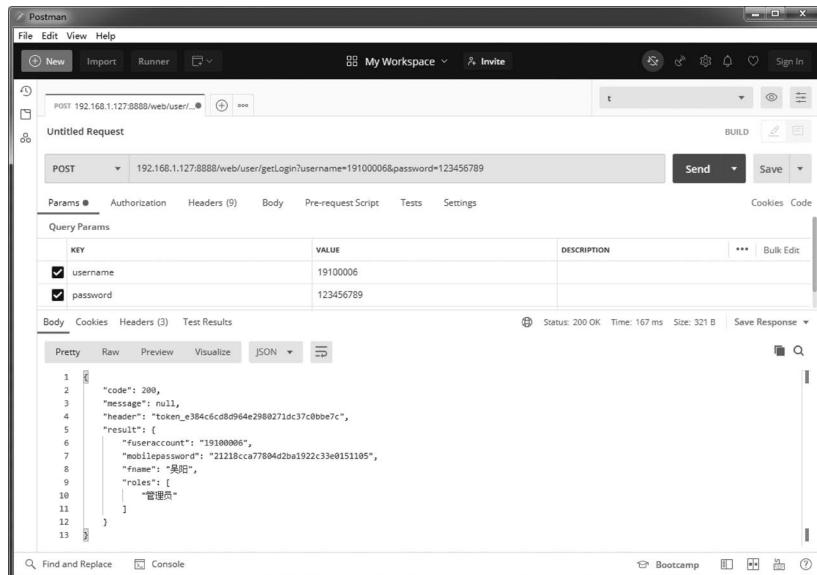


图 3.2 登录接口测试

2. 修改密码接口测试

在 postman 地址栏中输入修改密码测试接口地址, 调用登录接口, 如图 3.3 所示。

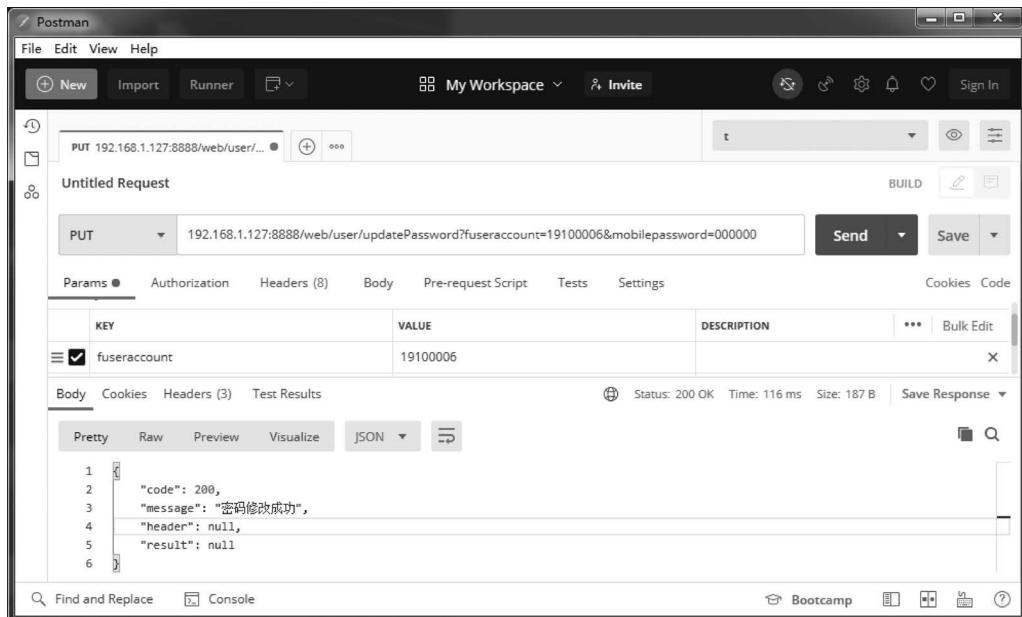


图 3.3 修改密码接口测试

3.2 实现登录功能

3.2.1 用户登录与记住密码

在实际项目开发过程中, 登录功能通常是必不可少的, 此功能主要包含用户登录、修改用户密码和记住用户密码。接下来通过讲解关键代码, 使读者理解与掌握登录功能的实现过程。

1. 编写登录 View

编写代码, 实现效果如图 1.4 所示。

```
<?xml version = "1.0" encoding = "utf - 8"?>
<LinearLayout xmlns: android = "http://schemas.android.com/apk/res/android"
    android: layout_width = "wrap_content"
    android: layout_height = "match_parent"
    android: background = "# ffffff"
    android: filterTouchesWhenObscured = "true"
    android: orientation = "vertical">
    <RelativeLayout
        android: id = "@ + id/login_layout"
        android: layout_width = "wrap_content"
        android: layout_height = "658dp"
        android: layout_marginLeft = "20dp"
```

```
        android: layout_marginRight = "20dp"
        android: gravity = "center"
        android: orientation = "vertical">
    < ImageView
        android: id = "@+id/logo"
        android: layout_width = "150dp"
        android: layout_height = "150dp"
        android: layout_centerHorizontal = "true"
        android: layout_marginTop = "30dp"
        android: layout_marginBottom = "20dp"
        android: src = "@drawable/linshi" />
    < RelativeLayout
        android: id = "@+id/one"
        android: layout_width = "match_parent"
        android: layout_height = "wrap_content"
        android: layout_below = "@+id/logo"
        android: layout_marginTop = "0dp"
        android: orientation = "vertical">
        < EditText
            android: id = "@+id/account"
            android: layout_width = "fill_parent"
            android: layout_height = "43dp"
            android: layout_marginTop = "5dp"
            android: hint = "请输入工号"
            android: maxLength = "20"
            android: paddingLeft = "45dp"
            android: paddingRight = "60dp"
            android: text = ""
            android: textSize = "20dp" />
        < ImageView
            android: layout_width = "28dp"
            android: layout_height = "30dp"
            android: layout_gravity = "left|center_horizontal"
            android: layout_marginStart = "8dp"
            android: layout_marginLeft = "8dp"
            android: layout_marginTop = "@dimen/px16"
            android: src = "@mipmap/account"
            android: visibility = "visible" />
    < RelativeLayout
        android: layout_width = "match_parent"
        android: layout_height = "wrap_content"
        android: layout_marginTop = "50dp"
        android: orientation = "horizontal">
        < EditText
            android: id = "@+id/password"
            android: layout_width = "fill_parent"
            android: layout_height = "43dp"
            android: hint = "请输入密码"
            android: inputType = "textPassword"
            android: maxLength = "20"
```

```
        android: paddingLeft = "45dp"
        android: paddingRight = "60dp"
        android: text = ""
        android: textSize = "20dp" />
    < ImageView
        android: layout_width = "28dp"
        android: layout_height = "30dp"
        android: layout_gravity = "left|center_vertical"
        android: layout_marginStart = "8dp"
        android: layout_marginLeft = "8dp"
        android: layout_marginTop = "@dimen/px8"
        android: src = "@mipmap/password" />
</RelativeLayout>
< Button
        android: id = "@+id/btn_click"
        android: layout_width = "fill_parent"
        android: layout_height = "40dp"
        android: layout_marginLeft = "5dp"
        android: layout_marginTop = "100dp"
        android: layout_marginRight = "5dp"
        android: background = "# ff336699"
        android: gravity = "center"
        android: text = "登 录"
        android: textColor = "@android: color/white" />
< CheckBox
        android: id = "@+id/checkBoxLogin"
        android: layout_width = "wrap_content"
        android: layout_height = "wrap_content"
        android: layout_marginLeft = "8dp"
        android: layout_marginTop = "145dp"
        android: layout_marginRight = "5dp"
        android: checked = "true"
        android: text = "记住用户名和密码"
        android: textColor = "# ff0000"
        android: textSize = "@dimen/px30" />
< TextView
        android: id = "@+id/btn_update"
        android: layout_width = "wrap_content"
        android: layout_height = "wrap_content"
        android: layout_marginLeft = "@dimen/px530"
        android: layout_marginTop = "150dp"
        android: background = "# ffffff"
        android: text = "更改密码"
        android: textColor = "# ff0000"
        android: textSize = "@dimen/px30" />
</RelativeLayout>
</RelativeLayout>
</LinearLayout>
```

在以上代码中，在 RelativeLayout 布局（即相对布局）中设置 android: orientation =

"vertical", 表示此时的排列方式为垂直方向。 android: gravity="center", 表示该布局中子控件要居中显示。 android: layout_marginLeft="20dp", 表示距离左边框的距离为 20dp。 android: layout_marginRight="20dp", 表示距离右边框的距离为 20dp。 android: layout_marginTop="30dp", 表示距离顶部边框的距离为 30dp。 android: layout_marginBottom="20dp", 表示距离底部边框的距离为 20dp。 在相对布局中, 设置 android: layout_centerHorizontal=" true ", 表示将控件置于水平方向的中心位置。设置 android: paddingLeft="45dp", padding 为内边框, 指该控件内部内容(如文本/图片距离该控件)的边距, 在本例中 id 为 account 的 EditText 控件中, 表示 hint 内容左内边距为 45dp。

2. 编写登录接口

```
@POST("user/getLogin")
Observable<Response<UserInfo>> login(@Query("username") String username,
@Query("password") String password);
```

该接口传入用户工号与密码即可实现登录操作。

3. 编写登录 Controller

在 com. qianfeng. mis. ui. login 包下新建 LoginActivity 类, 该类用于登录、自动登录与记住密码。具体代码如下所示。

```
public class LoginActivity extends BaseActivity implements
View.OnClickListener {
    private Button btnClick;
    private EditText userid;
    private EditText password;
    private TextView tvupdate;
    private String username;
    private String pwd;
    private CheckBox checkbox;
    @Override
    protected int setLayoutResId() {
        return R.layout.activity_login;
    }
    @Override
    public void initView() {
        btnClick = findViewById(R.id.btn_click);
        tvupdate = findViewById(R.id.btn_update);
        userid = findViewById(R.id.account);
        password = findViewById(R.id.password);
        checkbox = findViewById(R.id.checkBoxLogin);
        btnClick.setOnClickListener(LoginActivity.this);
        tvupdate.setOnClickListener(LoginActivity.this);
        checkbox.setOnClickListener(LoginActivity.this);
        String fuseraccount = SharedPreferencesHelperScan.getInstance(this).
        getStringValue("us");
        String pwd = SharedPreferencesHelperScan.getInstance(this).getStringValue("pa");
        userid.setText(fuseraccount);
        password.setText(pwd);
```

```
}

@Override
public void onClick(View view) {
    switch (view.getId()) {
        case R.id.btn_click:
            username = userid.getText().toString().trim();
            pwd = password.getText().toString().trim();
            if (TextUtils.isEmpty(username) || TextUtils.isEmpty(pwd)) {
                ToastUtil.showShort("请输入用户名或密码");
            }
            else {
                if (checkbox.isChecked()) {
                    //记住密码
                    SharedPreferencesHelperScan.getInstance(this).
                        putStringValue("us", username);
                    SharedPreferencesHelperScan.getInstance(this).putStringValue("pa", pwd);
                }
                else {
                    SharedPreferencesHelperScan.getInstance(this).putStringValue("us", "");
                    SharedPreferencesHelperScan.getInstance(this).putStringValue("pa", "");
                }
                login(username, pwd);    //登录
            }
            break;
        case R.id.btn_update:
            Intent intent = new Intent(LoginActivity.this,
                UpdatePasswordActivity.class);
            startActivity(intent);
        default:
            break;
    }
}

private void login(String username, String pwd) {
    RestClient.getInstance()
        .getStatisticsService()
        .login(username, pwd)
        .subscribeOn(Schedulers.io())
        .compose(bindToLifecycle())
        .observeOn(AndroidSchedulers.mainThread())
        .subscribe(response -> {
            if (response.getCode() == 200) {
                //登录成功,保存用户所有信息
                SharedPreferencesHelperScan.getInstance(this).setUserBean(response.getResult());
                //保存 token
                if (response.getHeader() != null) {
                    SharedPreferencesHelperScan.getInstance(this).putStringValue("token",
                        response.getHeader());
                }
                //将用户名保存下来
                SharedPreferencesHelperScan.getInstance(this).putStringValue("username", username);
            }
        });
}
```

```
SharedPreferencesHelperScan.getInstance(this).putStringValue("fname", response.getResult().getFname());
        ToastUtil.showShort("登录成功");
        //跳转到主界面
        Intent intent = new Intent(LoginActivity.this, WorkActivity.class);
        intent.putExtra("fname", response.getResult().getFname());
        intent.putExtra("fuseraccount", response.getResult().getFuseraccount());
        startActivity(intent);
    } else {
        ToastUtil.showShort("登录失败 " + response.getMessage());
    }
}, throwable -> {
    ToastUtil.showShort(throwable.getMessage());
});
}
}
```

在 LoginActivity 类中,首先对各个控件进行初始化与设置监听事件,通过 SharedPreferencesHelperScan 帮助类取出用户保存的工号与密码并设置到对应的 EditText 中。在 onClick(View view)方法中,对用户输入的工号与密码进行判断,如果选择记住密码,则调用 SharedPreferencesHelperScan 帮助类的 putStringValue()方法将工号与密码保存,然后调用 login(String username, String pwd)方法完成登录操作,并且把 UserBean 与 token 保存下来用于后面的操作。

3.2.2 修改用户密码

1. 编写修改用户密码 View

编写代码，实现如图 3.4 所示的效果。

```
<?xml version = "1.0" encoding = "utf - 8"?>
< LinearLayout xmlns: android = " http://schemas.
android. com/apk/res/android"
    android: layout_width = "wrap_content"
    android: layout_height = "match_parent"
    android: orientation = "vertical"
    android: background = "# ffffff"
    android: filterTouchesWhenObscured = "true">
< TextView
    android: id = "@ + id/tv_scanResult"
    android: layout_width = "match_parent"
    android: layout_height = "40dp"
    android: layout_marginTop = "0dp"
    android: gravity = "center"
    android: text = "密码修改页面"
    android: textSize = "20dp"
```



图 3.4 修改密码

```
    android: background = "# 336699"
    android: textColor = "# ffffff"/>
<RelativeLayout
    android: id = "@ + id/login_layout"
    android: layout_width = "wrap_content"
    android: layout_height = "658dp"
    android: layout_marginLeft = "20dp"
    android: layout_marginRight = "20dp"
    android: gravity = "center"
    android: orientation = "vertical">
<RelativeLayout
    android: id = "@ + id/one"
    android: layout_width = "match_parent"
    android: layout_height = "match_parent"
    android: layout_marginTop = "150dp"
    android: orientation = "vertical">
<EditText
    android: id = "@ + id/account"
    android: layout_width = "fill_parent"
    android: layout_height = "43dp"
    android: layout_marginTop = "5dp"
    android: hint = "请输入工号"
    android: maxLength = "20"
    android: paddingLeft = "55dp"
    android: paddingRight = "60dp"
    android: text = ""
    android: textSize = "20dp" />
<ImageView
    android: layout_width = "31dp"
    android: layout_height = "35dp"
    android: layout_gravity = "left|center_horizontal"
    android: layout_marginStart = "8dp"
    android: layout_marginLeft = "8dp"
    android: src = "@mipmap/account"
    android: visibility = "visible" />
<RelativeLayout
    android: layout_width = "match_parent"
    android: layout_height = "match_parent"
    android: layout_marginTop = "50dp"
    android: orientation = "horizontal">
<EditText
    android: id = "@ + id/password"
    android: layout_width = "fill_parent"
    android: layout_height = "43dp"
    android: hint = "请输入新密码"
    android: inputType = "textPassword"
    android: maxLength = "20"
    android: paddingLeft = "55dp"
    android: paddingRight = "60dp"
```

```
        android: text = ""
        android: textSize = "20dp" />
< ImageView
        android: layout_width = "31dp"
        android: layout_height = "35dp"
        android: layout_gravity = "left|center_vertical"
        android: layout_marginStart = "8dp"
        android: layout_marginLeft = "8dp"
        android: src = "@mipmap/password" />
< RelativeLayout
        android: layout_width = "match_parent"
        android: layout_height = "match_parent"
        android: layout_marginTop = "50dp"
        android: orientation = "vertical">
< EditText
        android: id = "@+id/repassword"
        android: layout_width = "fill_parent"
        android: layout_height = "43dp"
        android: hint = "请再次输入新密码"
        android: inputType = "textPassword"
        android: maxLength = "20"
        android: paddingLeft = "55dp"
        android: paddingRight = "60dp"
        android: text = ""
        android: textSize = "20dp" />
< ImageView
        android: layout_width = "31dp"
        android: layout_height = "35dp"
        android: layout_gravity = "left|center_vertical"
        android: layout_marginStart = "8dp"
        android: layout_marginLeft = "8dp"
        android: src = "@mipmap/password" />
</RelativeLayout>
</RelativeLayout>
< Button
        android: id = "@+id/btn_commit"
        android: layout_width = "fill_parent"
        android: layout_height = "40dp"
        android: layout_marginLeft = "5dp"
        android: layout_marginRight = "5dp"
        android: layout_marginTop = "200dp"
        android: background = "# ff336699"
        android: gravity = "center"
        android: text = "提 交"
        android: textColor = "@android: color/white" />
</RelativeLayout>
</RelativeLayout>
</LinearLayout>
```

在该页面,所用的布局与控件较为简单,主要通过三个 EditText 来收集数据,通过

Button 提交相关数据。

2. 编写修改用户密码接口

```
@PUT("user/updatePassword")
Observable < Response < UserInfo >> updatePassword (@ Query ("fuseraccount") String fuseraccount, @Query("mobilepassword") String mobilepassword);
```

该接口有两个参数，分别是工号 fuseraccount 与新密码 mobilepassword，通过调用该接口实现修改用户密码。

3. 编写修改用户密码 Controller

在 com.qianfeng.mis.ui.login 包下新建 UpdatePasswordActivity，该 Activity 用于修改密码。关键代码如下所示。

```
public class UpdatePasswordActivity extends BaseActivity implements
View.OnClickListener {
    private Button btncommit;
    private EditText userid;
    private EditText password;
    private EditText repassword;
    private String username;
    private String pwd;
    private String repwd;
    @Override
    protected int setLayoutResId() {
        return R.layout.activity_update_password;
    }
    @Override
    public void initView() {
        btncommit = findViewById(R.id.btn_commit);
        userid = findViewById(R.id.account);
        password = findViewById(R.id.password);
        repassword = findViewById(R.id.repassword);
        btncommit.setOnClickListener(UpdatePasswordActivity.this);
    }
    @Override
    public void onClick(View view) {
        username = userid.getText().toString().trim();
        pwd = password.getText().toString().trim();
        repwd = repassword.getText().toString().trim();
        if(TextUtils.isEmpty(username)&&TextUtils.isEmpty(pwd)){
            ToastUtil.showShort("用户名或密码不能为空");
        }else if (pwd.equals("")&&repwd.equals("")) {
            //判断两次密码是否为空
            Toast.makeText(getApplicationContext(), "密码不能为空",
                    Toast.LENGTH_SHORT).show();
        }else if (!pwd.equals(repwd)){
            Toast.makeText(getApplicationContext(),"密码不一致,请重新输入",
                    Toast.LENGTH_SHORT).show();
        }
    }
}
```

58

```
        Toast.LENGTH_SHORT).show();
    }else {
        updatePassword(username, pwd);
    }
}

public void updatePassword(String username, String pwd){
    RestClient.getInstance()
        .getStatisticsService()
        .updatePassword(username, pwd)
        .subscribeOn(Schedulers.io())
        .compose(bindToLifecycle())
        .observeOn(AndroidSchedulers.mainThread())
        .subscribe(response -> {
            if (response.getCode() == 200) {
                ToastUtil.showShort("密码修改成功,请重新登录");
                Intent intent = new Intent(UpdatePasswordActivity.this,
                    LoginActivity.class);
                startActivity(intent);
                finish();
            } else {
                ToastUtil.showShort(response.getMessage());
            }
        }, throwable -> {
    });
}
}
```

在 UpdatePasswordActivity 中，首先对各个控件进行初始化并设置监听事件，在 onClick(View view) 方法中，获取用户输入的内容并做判断，然后调用 updatePassword (String username, String pwd) 方法进行密码的修改。

3.3 实现首页页面

3.3.1 首页页面

1. 编写首页面面 View

编写代码，实现效果如图 1-5 所示

```
<?xml version = "1.0" encoding = "utf - 8"?>
< LinearLayout
    xmlns: android = "http://schemas.android.com/apk/res/android"
    android: layout_width = "match_parent"
    android: layout_height = "match_parent"
    android: clipChildren = "false"
    android: orientation = "vertical">
< FrameLayout
    android: id = "@ + id/f1_mains"
```

```
    android: layout_width = "match_parent"
    android: layout_height = "0dp"
    android: layout_weight = "1">
</FrameLayout >
< View
    android: layout_width = "match_parent"
    android: layout_height = "0.3dp"
    android: background = "# 33666666" />
< RadioGroup
    android: id = "@ + id/radioGroup"
    android: layout_width = "match_parent"
    android: layout_height = "56dp"
    android: layout_gravity = "bottom|center"
    android: background = "# eee"
    android: clipChildren = "false"
    android: gravity = "center"
    android: orientation = "horizontal">
    < RadioButton
        android: id = "@ + id/rb_home"
        android: layout_width = "0dp"
        android: layout_height = "match_parent"
        android: layout_weight = "1"
        android: background = "@null"
        android: button = "@null"
        android: drawablePadding = "6dp"
        android: gravity = "center"
        android: padding = "5dp"
        android: text = "工作台"
        android: textColor = "@color/navigator_color" />
    < RadioButton
        android: id = "@ + id/rb_pond"
        android: layout_width = "0dp"
        android: layout_height = "match_parent"
        android: layout_weight = "1"
        android: background = "@null"
        android: button = "@null"
        android: drawablePadding = "6dp"
        android: gravity = "center"
        android: padding = "5dp"
        android: text = "待办"
        android: textColor = "@color/navigator_color" />
< LinearLayout
    android: gravity = "center_horizontal"
    android: orientation = "vertical"
    android: layout_width = "0dp"
    android: layout_weight = "1"
    android: layout_height = "110dp">
    < ImageView
        android: id = "@ + id/rbAdd"
        android: layout_width = "55dp"
```

```

        android: layout_height = "55dp"
        android: src = "@mipmap/comui_tab_post" />
    < TextView
        android: textColor = "@color/black"
        android: text = "发布"
        android: padding = "5dp"
        android: layout_width = "wrap_content"
        android: layout_height = "wrap_content" />
    </ LinearLayout >
    < RadioButton
        android: id = "@+id/rb_message"
        android: layout_width = "0dp"
        android: layout_height = "match_parent"
        android: layout_weight = "1"
        android: background = "@null"
        android: button = "@null"
        android: drawablePadding = "6dp"
        android: gravity = "center"
        android: padding = "5dp"
        android: text = "动态"
        android: textColor = "@color/navigator_color" />
    < RadioButton
        android: id = "@+id/rb_me"
        android: layout_width = "0dp"
        android: layout_height = "match_parent"
        android: layout_weight = "1"
        android: background = "@null"
        android: button = "@null"
        android: drawablePadding = "6dp"
        android: gravity = "center"
        android: padding = "5dp"
        android: text = "我的"
        android: textColor = "@color/navigator_color" />
    </ RadioGroup >
</ LinearLayout >
```

以上代码中,使用帧布局 FrameLayout 来加载要被替换的 Fragment,在帧布局中使用 android: layout_weight = "1",使其占满整个空间,通过底部的 RadioButton 来切换 Fragment。

2. 编写首页页面 Controller

在 com. qianfeng. mis. ui. sale 包下新建 WorkActivity,该 Activity 为主 Activity。具体代码如下所示。

```

public class WorkActivity extends BaseActivity {
...
    private WorkMainFragment mWorkMainFragment;
    private NewsFragment mNewsFragment1;
    private NewsFragment mNewsFragment2;
    private NewsFragment mNewsFragment3;
    private Fragment mContent;
```

```
@Override
protected int setLayoutResId() {
    return R.layout.activity_work;
}
@Override
public void initView() {
    initListener();
    mRbHome.setChecked(true); //默认选中工作台
    Drawable dbHome = getResources().getDrawable(R.drawable.selector_home);
    dbHome.setBounds(0, 0, UIUtils.dip2Px(this, 20), UIUtils.dip2Px(this, 20));
    mRbHome.setCompoundDrawables(null, dbHome, null, null);
    Drawable dbPond = getResources().getDrawable(R.drawable.selector_pond);
    dbPond.setBounds(0, 0, UIUtils.dip2Px(this, 20), UIUtils.dip2Px(this, 20));
    mRbPond.setCompoundDrawables(null, dbPond, null, null);
    Drawable dbMsg = getResources().getDrawable(R.drawable.selector_message);
    dbMsg.setBounds(0, 0, UIUtils.dip2Px(this, 20), UIUtils.dip2Px(this, 20));
    mRbMessage.setCompoundDrawables(null, dbMsg, null, null);
    Drawable dbMe = getResources().getDrawable(R.drawable.selector_person);
    dbMe.setBounds(0, 0, UIUtils.dip2Px(this, 20), UIUtils.dip2Px(this, 20));
    mRbMe.setCompoundDrawables(null, dbMe, null, null);
    initFragment();
}
private void initFragment() {
    FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();
    if (mWorkMainFragment != null && mWorkMainFragment.isAdded()) {
        transaction.remove(mWorkMainFragment);
    }
    if (mNewsFragment1 != null && mNewsFragment1.isAdded()) {
        transaction.remove(mNewsFragment1);
    }
    if (mNewsFragment2 != null && mNewsFragment2.isAdded()) {
        transaction.remove(mNewsFragment2);
    }
    if (mNewsFragment3 != null && mNewsFragment3.isAdded()) {
        transaction.remove(mNewsFragment3);
    }
    transaction.commitAllowingStateLoss();
    mWorkMainFragment = null;
    mNewsFragment1 = null;
    mNewsFragment2 = null;
    mNewsFragment3 = null;
    mRbHome.performClick();
}
private void initListener() {
    mRadioGroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(RadioGroup group, int checkedId) {
            switch (checkedId) {
                case R.id.rb_home:
                    switchContent(mRbHome);
                    break;
                case R.id.rb_pond:
```

```
        switchContent(mRbPond);
        break;
    case R.id.rb_message:
        switchContent(mRbMessage);
        break;
    case R.id.rb_me:
        switchContent(mRbMe);
        break;
    default:
        break;
    }

}

mRbAdd.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
    }
});

}

public void switchContent(View view) {
    Fragment fragment;
    if (view == mRbHome) {
        if (mWorkMainFragment == null) {
            mWorkMainFragment = new WorkMainFragment();
        }
        fragment = mWorkMainFragment;
    } else if (view == mRbPond) {
        if (mNewsFragment1 == null) {
            mNewsFragment1 = new NewsFragment();
        }
        fragment = mNewsFragment1;
    } else if (view == mRbMessage) {
        if (mNewsFragment2 == null) {
            mNewsFragment2 = new NewsFragment();
        }
        fragment = mNewsFragment2;
    } else if (view == mRbMe) {
        if (mNewsFragment3 == null) {
            mNewsFragment3 = new NewsFragment();
        }
        fragment = mNewsFragment3;
    } else {
        return;
    }
    FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();
    if (mContent == null) {
        transaction.add(mFlMains.getId(), fragment).commit();
        mContent = fragment;
    }
}
```

```

        if (mContent != fragment) {
            if (!fragment.isAdded()) {
                transaction.hide(mContent).add(mFlMains.getId(), fragment).commitAllowingStateLoss();
            } else {
                transaction.hide(mContent).show(fragment).commitAllowingStateLoss();
            }
            mContent = fragment;
        }
    }
}

```

在 WorkActivity 类中，在 initView() 方法中，首先调用 initListener() 方法对 RadioGroup 进行事件监听，通过 mRbHome.setChecked(true) 方法默认选中工作台 Fragment。在 initFragment() 方法中对 Fragment 进行初始化操作，在 switchContent(View view) 方法中显示选中的 Fragment。

3.3.2 工作台页面

1. 编写工作台页面 View

编写代码，实现效果如图 1.5 所示。

```

<?xml version = "1.0" encoding = "utf - 8"?>
< android. support. v4. widget. NestedScrollView xmlns: android = "http://schemas. android. com/
apk/res/android"
    xmlns: app = "http://schemas. android. com/apk/res - auto"
    android: id = "@ + id/refreshLayout"
    android: layout_width = "match_parent"
    android: layout_height = "match_parent"
    android: background = "# fff"
    android: orientation = "vertical">
    < LinearLayout
        android: layout_width = "match_parent"
        android: layout_height = "match_parent"
        android: gravity = "center_horizontal"
        android: orientation = "vertical">
        < include layout = "@layout/title" />
        < com. qianfeng. mis. view. BannerLayout
            xmlns: app = "http://schemas. android. com/apk/res - auto"
            android: id = "@ + id/bannerLayout"
            android: layout_width = "match_parent"
            android: layout_height = "@dimen/px340"
            app: autoPlayDuration = "4000"
            app: indicatorMargin = "5dp"
            app: indicatorPosition = "rightBottom"
            app: indicatorShape = "oval"
            app: indicatorSpace = "3dp"
            app: isAutoPlay = "true"
        </ BannerLayout>
    </ LinearLayout>
</ NestedScrollView>

```

```

        app: scrollDuration = "900"
        app: selectedIndicatorColor = "# 222222"
        app: selectedIndicatorHeight = "4dp"
        app: selectedIndicatorWidth = "4dp"
        app: unSelectedIndicatorColor = "# 999"
        app: unSelectedIndicatorHeight = "4dp"
        app: unSelectedIndicatorWidth = "4dp">
    </com.qianfeng.mis.view.BannerLayout >
    < RelativeLayout
        android: layout_width = "match_parent"
        android: layout_height = "wrap_content">
        < android.support.v7.widget.RecyclerView
            android: id = "@ + id/icon_selected"
            android: layout_width = "match_parent"
            android: layout_height = "wrap_content"
            android: layout_margin = "@dimen/px20"
            android: background = "@drawable/stoke_ele2e4"
            android: overScrollMode = "never" />
    </RelativeLayout >
    < LinearLayout
        android: layout_width = "match_parent"
        android: layout_height = "wrap_content"
        android: layout_margin = "@dimen/px20"
        android: background = "@drawable/stoke_f9ebc5"
        android: orientation = "vertical">
        < TextView
            style = "@style/textview_333_30"
            android: layout_marginLeft = "@dimen/px30"
            android: layout_marginTop = "@dimen/px30"
            android: text = "产品中心"
            android: textColor = "# F00"
            android: textSize = "@dimen/px34" />
        < android.support.v7.widget.RecyclerView
            android: id = "@ + id/rv_chapin"
            android: layout_width = "match_parent"
            android: layout_height = "wrap_content"
            android: overScrollMode = "never" />
    </LinearLayout >
</LinearLayout >
</android.support.v4.widget.NestedScrollView>

```

在该页面,由于有的手机屏幕较小,需要一个支持嵌套滑动 NestedScrollView 包裹整个布局。NestedScrollView 与 ScrollView 比较类似,其作用就是作为控件父布局,从而具备(嵌套)滑动功能。通过 include 标签可以将相同的布局引入,从而避免每次编写重复代码。使用自定义的 BannerLayout 实现轮播图效果。各个功能导航按钮与产品中心都是使用 RecyclerView 实现。

2. 编写工作台页面 Controller

在 com.qianfeng.mis.ui.sale 包下新建 WorkMainFragment,该 Fragment 用于工作台

首页的展示。具体代码如下所示。

```
public class WorkMainFragment extends BaseFragment {  
    ...  
    private int[] iconList = {  
        R.drawable.caishen_1, R.drawable.caishen_2, R.drawable.caishen_3, R.drawable.  
caishen_4, R.drawable.caishen_5, R.drawable.caishen_6,  
        R.drawable.caishen_7, R.drawable.caishen_8, R.drawable.caishen_9, R.drawable.  
caishen_10, R.drawable.caishen_11, R.drawable.caishen_12,  
    };  
    private String[] nameList = {  
        "客户资料", "任务/跟进", "销售机会", "报价记录", "合同/订单", "回款记录",  
        "客户公共池", "产品信息", "数据审核", "费用报销", "库存出货", "更多"};  
  
    private int[] mIconbenner = {  
        R.mipmap.banner_1, R.mipmap.banner_2,  
        R.mipmap.banner_3};  
    int[] mChapinlist = {  
        R.mipmap.hm_zd240, R.mipmap.hm_zd350a,  
        R.mipmap.hm_zd350d, R.mipmap.hm_zd350e600,};  
    String[] mChapinnameList1 = {  
        "移动产业", "千锋教育",  
        "项目研发", "创业孵化"};  
    @Override  
    protected int setLayoutResId() {  
        return R.layout.fragment_workmain;  
    }  
    @Override  
    public void initView() {  
        mTitle.setText("工作台(" + SharedPreferenceScan.getInstance(getActivity())  
().getStringValue("fname") + ")");  
        //轮播图  
        initBannerData();  
        //导航图标  
        setIcon();  
        //产品中心  
        LinearLayoutManager linearLayoutManager = new LinearLayoutManager(getContext());  
        linearLayoutManager.setOrientation(LinearLayoutManager.HORIZONTAL);  
        mRvChapin.setLayoutManager(linearLayoutManager);  
        List<IconInfo> list1 = new ArrayList<>();  
        for (int i = 0; i < mChapinlist.length; i++) {  
            IconInfo iconInfo = new IconInfo();  
            iconInfo.setImage(mChapinlist[i]);  
            iconInfo.setName(mChapinnameList1[i]);  
            list1.add(iconInfo);  
        }  
        ChapinMainAdapter chapinMainAdapter = new ChapinMainAdapter(list1);  
        mRvChapin.setAdapter(chapinMainAdapter);  
    }  
    //初始化轮播图
```

```
private void initBannerData() {
    if (mIconbenner != null && mIconbenner.length > 0) {
        List<Integer> urls = new ArrayList<>();
        for (int i = 0; i < mIconbenner.length; i++) {
            urls.add(mIconbenner[i]);
            if (mIconbenner.length > 1) {
                mBannerLayout.setAutoPlay(true);
            } else {
                mBannerLayout.setAutoPlay(false);
            }
        }
        if (urls != null && urls.size() > 0) {
            mBannerLayout.setViewRes(urls);
        }
        mBannerLayout.setOnBannerItemClickListener(new BannerLayout.
        OnBannerItemClickListener() {
            @Override
            public void onItemClick(int position) {
            }
        });
    }
}
//设置导航图标
private void setIcon() {
    mIconSelected.setLayoutManager(new GridLayoutManager(getActivity(), 4));
    List<IconInfo> list = new ArrayList<>();
    for (int i = 0; i < iconList.length; i++) {
        IconInfo iconInfo = new IconInfo();
        iconInfo.setImage(iconList[i]);
        iconInfo.setName(nameList[i]);
        list.add(iconInfo);
    }
    IconMainAdapter icon_main_adapter = new IconMainAdapter(list);
    mIconSelected.setAdapter(icon_main_adapter);
    icon_main_adapter.setOnItemClickListener(new BaseQuickAdapter.OnItemClickListener() {
        @Override
        public void onItemClick(BaseQuickAdapter adapter, View view, int position) {
            if (list.get(position).getName().equals("客户资料")) {
                Intent intent = new Intent(getActivity()
                    , CustomerListActivity.class);
                startActivity(intent);
            } else if (list.get(position).getName().equals("任务/跟进")) {
                Intent intent = new Intent(getActivity(),
                    CustomGenjinActivity.class);
                startActivity(intent);
            } else if (list.get(position).getName().equals("销售机会")) {
                Intent intent = new Intent(getActivity(),
                    CustomXiaoShouJiHuiActivity.class);
                startActivity(intent);
            }
        }
    });
}
```

```
        } else if (list.get(position).getName().equals("产品信息")) {
            Intent intent = new Intent(getActivity(),
                ProductInformationListActivity.class);
            startActivity(intent);
        } else if (list.get(position).getName().equals("报价记录")) {
            Intent intent = new Intent(getActivity(),
                ProductOfferByIdListActivity.class);
            startActivity(intent);
        } else if (list.get(position).getName().equals("合同/订单")) {
            Intent intent = new Intent(getActivity(),
                ContractOrderListActivity.class);
            startActivity(intent);
        } else if (list.get(position).getName().equals("客户公共池")) {
            Intent intent = new Intent(getActivity(),
                PublicCustomerActivity.class);
            startActivity(intent);
        } else if (list.get(position).getName().equals("回款记录")) {
            Intent intent = new Intent(getActivity(),
                HuiKuanJiHuaActivity.class);
            startActivity(intent);
        } else if (list.get(position).getName().equals("数据审核")) {
            Intent intent = new Intent(getActivity(),
                DataAuditListActivity.class);
            startActivity(intent);
        } else if (list.get(position).getName().equals("费用报销")) {
            Intent intent = new Intent(getActivity(),
                ReimbursementListActivity.class);
            startActivity(intent);
        } else if (list.get(position).getName().equals("库存出货")) {
            Intent intent = new Intent(getActivity(),
                StockActivity.class);
            startActivity(intent);
        }
    }
});
```

在 WorkMainFragment 类中,首先对各个控件和相关数据进行初始化,在 initView()方法中调用 initBannerData()方法初始化轮播图,调用 setIcon()方法初始化导航图标,分别给各个图标设置单击事件,然后设置产品中心的 adapter。