

开发文档

开发版本

本应用基于 DevEco Studio 5.0.5 开发，使用 API12 的接口完成代码的编写。

代码流程

首先，使用 `PreferenceUtil` 读取 `User` 中是否有登录信息，如果没有登录信息，则设置初始页面为 `Login` 页面；如果已有登录信息，则设置初始页面为 `Index` 即主界面（备忘录详细页面）。

在登录页面（`LoginPage`）中：

首先确保已输入账号和密码，然后从 `PreferenceUtil` 中读取所有账号信息的数组，然后一一判断是否是已输入的账号密码。若正确，则使用 `router` 中的 `replaceUrl` 方法切换到 `Index` 页面（备忘录详细页面）；若不正确，则弹窗提示“账号或密码错误。”

同时可以点击立即注册进入注册页面。

在注册页面（`RegisterPage`）中：

首先确保账号和密码都已经输入且两次密码相同，再从 `PreferenceUtil` 中读取所有账号信息判断账号是否已经存在。若不存在，则注册成功并返回登录页面；若存在，则弹窗提醒“该账户已注册”。

在备忘录详细页面（`Index`）中：

点击退出会调用 `logOut` 函数，删除登录信息并使用 `router` 中的 `replaceUrl` 方法返回 `Login` 页面。

会遍历所有 `Memoltem`（即单条备忘录），并依次输出成 `Column` 组件，在点击组件时，使用 `router` 中的 `pushUrl` 方法转到 `MemorandumDetDetailsPage` 页面。

右下角有一个添加组件，点击同样使用使用 `router` 中的 `pushUrl` 方法转到 `AddMemorandumPage` 页面。

在添加备忘录界面（`AddMemorandumPage`）中：

先确保备忘录名称、内容、时间不为空，然后将时间部分通过 `formatDate` 进行处理（处理为 `new Date()` 可以处理的形式），读入自带的 `new Date` 函数进行处理（分开年月日时分），并分别写入变量并尝试添加至 `Memoltem[]` 中。

在详细备忘录界面（`MemorandumDetDetailsPage`）中：

使用 `new Date` 获取当前的时间，并将其和备忘录的截止时间都转化为毫秒单位作差，差值再通过计算转换为 `xx 天 xx 小时 xx 分钟 xx 秒`。

接口和函数说明

1. Memoltem

单个备忘录的类。

变量包括 id、名称、内容、截止时间和相对时间。

函数有一个 loadData，用于返回一个含有 20 个空 Memoltem 的数组。

用于单个备忘录的表示与记录。

2. UserModel

用户类。

变量包括用户名、id、密码。

用于保存和记录每个用户。

PS: 前两个类中的 id 是用于数据隔离的关键，在读取时：

```
let ararys = await PreferenceUtil.readModel<MemoItem[]>(userModel.id) ?? []  
this.memoList = ararys.reverse()  
this.creatTimer()
```

是根据用户 id 来判断这个备忘录是否属于这个用户，如果不属于则不会进行展示。

3. PreferenceUtil

操作用户首选项，用于读写数据。

4. Emitter

使用：

```
creatTimer() {  
  this.timeNumber = setInterval(() => {  
    emitter.emit("time-event", null)  
  }, 1000)  
}  
  
onPageShow(): void {  
  emitter.on("time-event", () => {  
    this.loadData()  
  });  
}
```

用于在备忘录详细页面每秒更新一次数据。

5. Router

用于进行页面之间的切换

6. `promptAction`

用于实现弹窗功能