



02-13 flutter+mvp

Posted on 2020-11-23 23:39 肖无情 阅读(64) 评论(0) 编辑 收藏 举报

1、

```
abstract class CLoginPrenter{

    void loginWithShopCode(String shopCode);

}

abstract class CLoginView {
    void loginSuccess();
}

abstract class CLoginModel extends IModel{
    void toLogin( Map<String,dynamic> params, {ResponseSuccess success,ResponseFail fail});
}
```

p:

```
class LoginPresenter implements CLoginPrenter{
    CLoginModel cLoginModel;
    CLoginView cLoginView;

    LoginPresenter(CLoginView cLoginView){
        this.cLoginView = cLoginView;
        cLoginModel = LoginModel();
    }

    @override
    void loginWithShopCode(String shopCode) {

        cLoginModel.toLogin(Map(),success: (data){

        },fail: (code,msg){

        }
        );
    }

}
```

m:

```
typedef ResponseSuccess (dynamic data);
typedef ResponseFail (int code,String msg);

class LoginModel implements CLoginModel{

    @override
    void toLogin(Map<String, dynamic> params,{ResponseSuccess success,ResponseFail fail}) {

        if(true){
            success("data");
        }
        else{
            fail(1,"data");
        }

    }

}
```

v

```
class _FirstLoginPageState extends State<FirstLoginPage> implements CLoginView {

    LoginPresenter loginPresenter ;

    @override
    void initState() {
        super.initState();
        loginPresenter = LoginPresenter(this);
    }

    @override
    void loginSuccess() {

    }

}
```

方式2:

```

abstract class CLoginPrenter{

    void loginWithShopCode(String shopCode);

}

abstract class CLoginView {
    void loginSuccess();

}

abstract class CLoginModel extends IModel{
    void toLogin( Map<String,dynamic> params);
}

class ResponseCallBack {

    responseSuccess(dynamic data, {String msg}){}
    //登录/注册失败
    responseFailed(int code , String msg){}

}

```

p:

```

class LoginPresenter implements CLoginPrenter,ResponseCallBack{
    CLoginModel cLoginModel;
    CLoginView cLoginView;

    LoginPresenter(CLoginView cLoginView){
        this.cLoginView = cLoginView;
        cLoginModel = LoginModel(this);
    }

    @override
    void loginWithShopCode(String shopCode) {

        cLoginModel.toLogin(Map());
    }

    @override
    responseFailed(int code, String msg) {
        throw UnimplementedError();
    }

    @override
    responseSuccess(code, {String msg}) {
        throw UnimplementedError();
    }
}

```

m:

```

class LoginModel implements CLoginModel{

    ResponseCallBack callBack;

    LoginModel(ResponseCallBack callBack){
        this.callBack = callBack;
    }

    @override
    void toLogin(Map<String, dynamic> params) {

        if(true){
            callBack.responseSuccess("data");
        }
        else{
            callBack.responseFailed(1,"data");
        }

    }

}

```

v:同上

方式3:

```

abstract class CLoginPrenter{

    void loginWithShopCode(String shopCode);

}

abstract class CLoginView {
    void loginSuccess();

}

abstract class CLoginModel extends IModel{
    void toLogin( Map<String,dynamic> params,{ResponseCallBack responseCallBack});
}

```

```
    }

    typedef ResponseSuccess (dynamic data);
    typedef ResponseFailed (int code,String msg);
    class ResponseCallBack {

        ResponseSuccess responseSuccess ;
        //登录/注册失败
        ResponseFailed responseFailed;

        ResponseCallBack({
            this.responseSuccess,
            this.responseFailed,
        });
    }
}
```

```
class LoginPresenter implements CLoginPrenter{
    CLoginModel cLoginModel;
    CLoginView cLoginView;

    LoginPresenter(CLoginView cLoginView){
        this.cLoginView = cLoginView;
        cLoginModel = LoginModel();
    }

    @override
    void loginWithShopCode(String shopCode) {

        cLoginModel.toLogin(Map(),
            responseCallBack: ResponseCallBack(
                responseSuccess: (data){

                },responseFailed: (code,msg){

                })
        );
    }
}
```

```
class LoginModel implements CLoginModel{

    @override
    void toLogin(Map<String, dynamic> params,{ResponseCallBack responseCallBack}) {

        if(true){
            responseCallBack.responseSuccess("data");
        }
        else{
            responseCallBack.responseFailed(1,"data");
        }
    }

}
```

分类: flutter

标签: flutter

好文要顶

关注我

收藏该文

肖无情

粉丝 - 2

关注 - 5

+加关注

0

推荐

0

反对

« 上一篇: 02-07 flutter mixin
» 下一篇: 02-08 flutter InheritedWidget

[刷新评论](#) [刷新页面](#) [返回顶部](#)

登录后才能查看或发表评论, 立即 [登录](#) 或者 [逛逛](#) 博客园首页

【推荐】阿里云新人特惠, 爆款云服务器2核4G低至0.46元/天

编辑推荐:

- gRPC 入门与实操 (.NET 篇)
- dotnet 代码优化 聊聊逻辑圈复杂度
- 一个棘手的生产问题，但是我写出来之后，就是你的了
- 你可能不知道的容器镜像安全实践
- .Net 6 使用 Consul 实现服务注册与发现

阅读排行:

- Redux与前端表格施展 “组合拳”， 实现大屏展示应用的交互增强
- gRPC入门与实操(.NET篇)
- 博客园主题修改分享 - 过年篇
- 如何优雅地校验后端接口数据，不做前端背锅侠
- 产品与研发相处之道