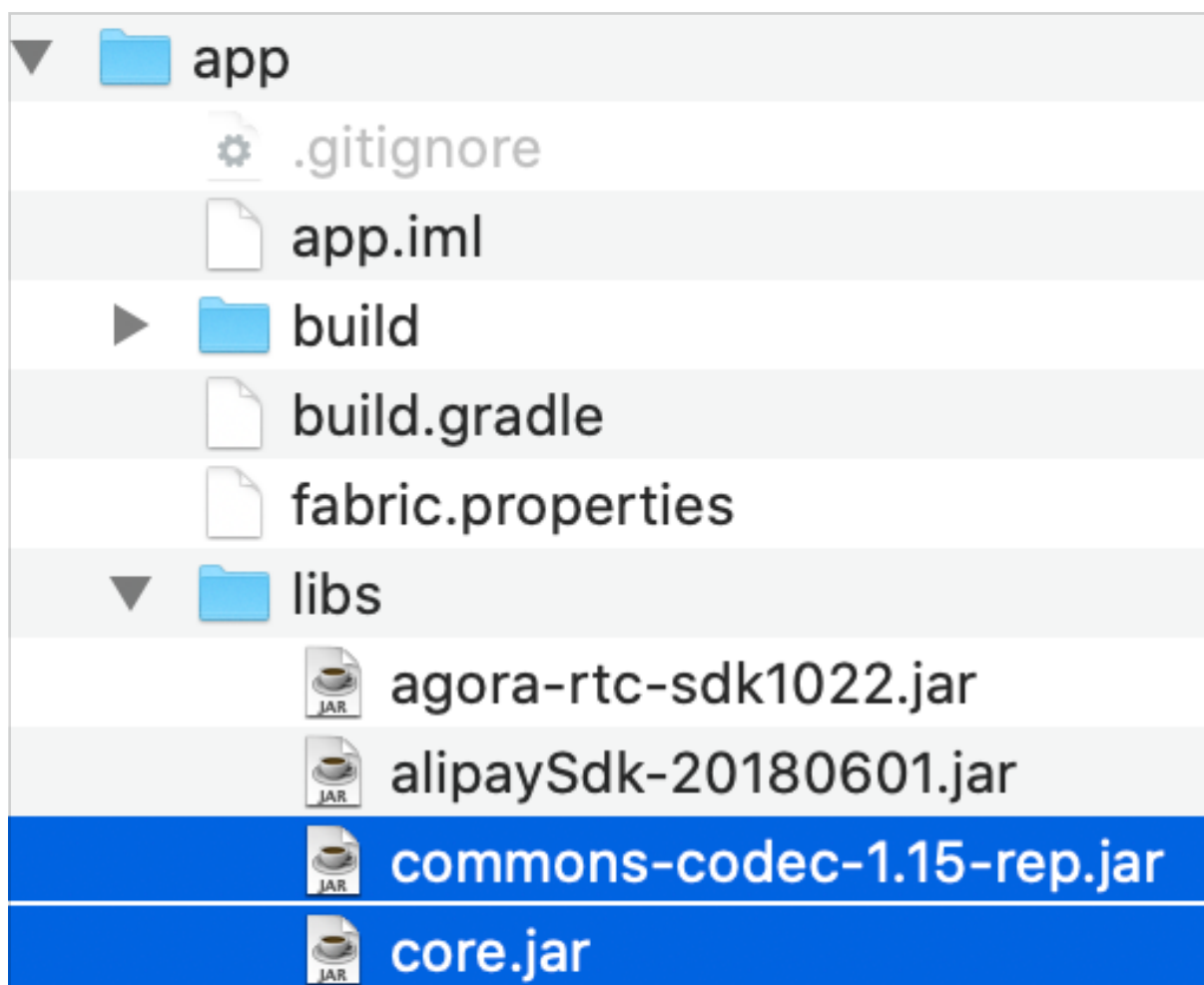


安卓集成方法

集成方法

1、将jar包导入lib



2、配置build.gradle

```
implementation "com.google.code.gson:gson:2.8.6"
implementation "com.squareup.okhttp3:okhttp:4.8.0"
implementation files('libs/core.jar')
implementation files('libs/commons-codec-1.15-rep.jar')
```

3、代码里调用

- Java:

```
AuthenticationClient client = new AuthenticationClient("your userpoolId");

client.registerByEmail(new RegisterByEmailInput("xxx@qq.com",
"123456")).enqueue(new
cn.authing.core.http.Callback<cn.authing.core.types.User>() {
    @Override
    public void onSuccess(cn.authing.core.types.User user) {

    }

    @Override
    public void onFailure(@Nullable GraphQLResponse.ErrorInfo errorInfo) {

    }
});
```

- Kotlin

```
val authenticationClient = AuthenticationClient("your userpoolId")

authenticationClient.registerByEmail(
    RegisterByEmailInput(
        "xxx@qq.com",
        "123456"
    )
).enqueue(object : cn.authing.core.http.Callback<User> {
    override fun onFailure(error: ErrorInfo?) {

    }

    override fun onSuccess(result: User) {

    }
})
```

```
})
```

微信登录

loginByWechat 所需四个参数均为微信返回的参数：

country | OPTIONAL | string | 微信返回给 APP 的 country

lang | OPTIONAL: string | 微信返回给 APP 的 lang

state | OPTIONAL | string | 微信返回给 APP 的 state

code | REQUIRED | string | 微信返回给 APP 的 code

Java:

```
val authenticationClient = AuthenticationClient("your userpoolId")

String code = "#returned code from wechat#";

authenticationClient.client.loginByWechat(code).enqueue(new Callback<User>() {
    @Override
    public void onSuccess(User result) {

    }

    @Override
    public void onFailure(@Nullable GraphQLResponse.ErrorInfo error) {

    }
});
```

Kotlin:

```
val authenticationClient = AuthenticationClient("your userpoolId")
```

```
val code = "#returned code from wechat#";

authenticationClient.loginByWechat(code).enqueue(object:
cn.authing.core.http.Callback<User> {
    override fun onFailure(error: ErrorInfo?) {

    }

    override fun onSuccess(result: User) {
        val user = result
    }
})
```

用户的自定义数据

- 获取当前用户的自定义数据列表

```
List<UserDefinedData> udv = client.udv().enqueue...
```

- 添加自定义数据

```
List<UserDefinedData> udv = client.setUdv("testKey", "testValue").enqueue...
```

- 移除自定义数据

```
List<UserDefinedData> udv = client.removeUdv("testKey").enqueue...
```