分布式开发样例：带你玩转多设备

和佩佩

华为CBG资深研发工程师 5年开发经验
HarmonyOS Codelabs 负责人

目 录

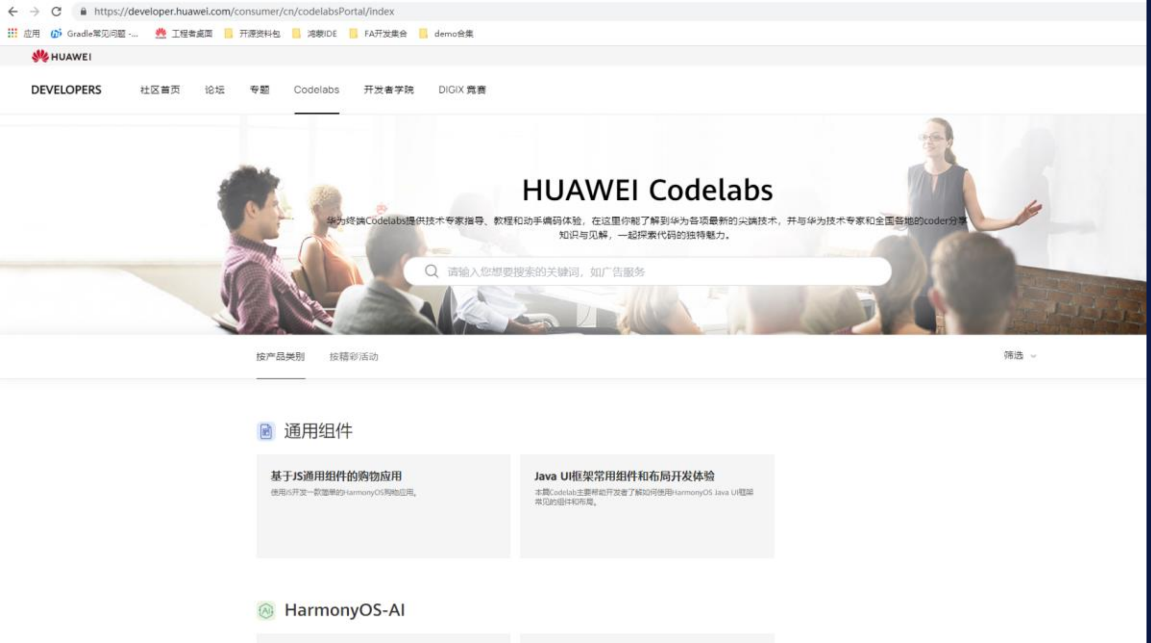
1. 概念：Demo、Sample、Cookbook、Codelab
2. 典型场景1：设备感知发现、数据迁移 （样例：分布式邮件编辑）
3. 典型场景2：设备感知发现、HarmonyOS IDL实现进程间通信 （样例：视频跨设备播放控制）
4. 样例代码的开源共建

Codelabs是完整应用拆解教程

教程	Cookbook 功能：列表/动画/图片	Codelab 场景、应用
代码	Demo	Sample

关于Codelabs

Codelabs概览



https://developer.huawei.com/consumer/cn/codelabsPortal/index

应用　Gradle常见问题 -...　工程者桌面　开源资料包　鸿蒙IDE　FA开发集合　demo合集

HUAWEI

DEVELOPERS　社区首页　论坛　专题　Codelabs　开发者学院　DIGIX 竞赛

HUAWEI Codelabs

华为终端Codelabs提供技术专家指导、教程和动手编码体验，在这里你能了解到华为各项最新的尖端技术，并与华为技术专家和全国各地的coder分享知识与见解，一起探索代码的独特魅力。

请输入您想要搜索的关键词，如广告服务

按产品类别　按精彩活动　筛选

通用组件

基于JS通用组件的购物应用

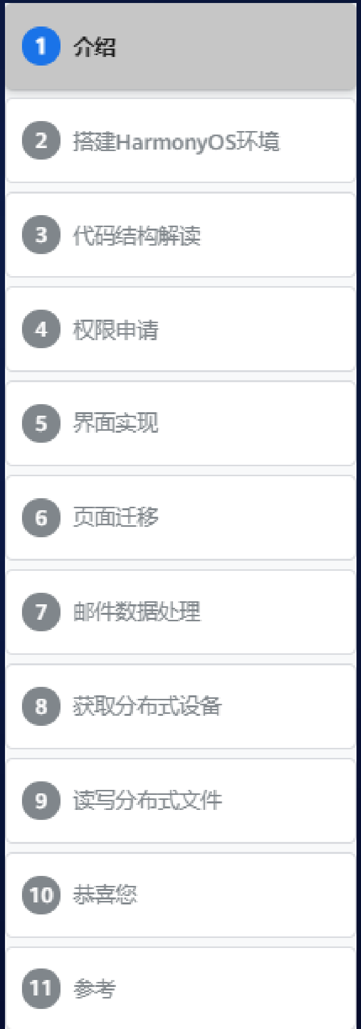
使用JS开发一款简单的HarmonyOS购物应用。

Java UI框架常用组件和布局开发体验

本篇Codelab主要帮助开发者了解如何使用HarmonyOS Java UI框架常见的组件和布局。

HarmonyOS-AI

拆解HAP应用程序、逐步解惑



1 介绍

2 搭建HarmonyOS环境

3 代码结构解读

4 权限申请

5 界面实现

6 页面迁移

7 邮件数据处理

8 获取分布式设备

9 读写分布式文件

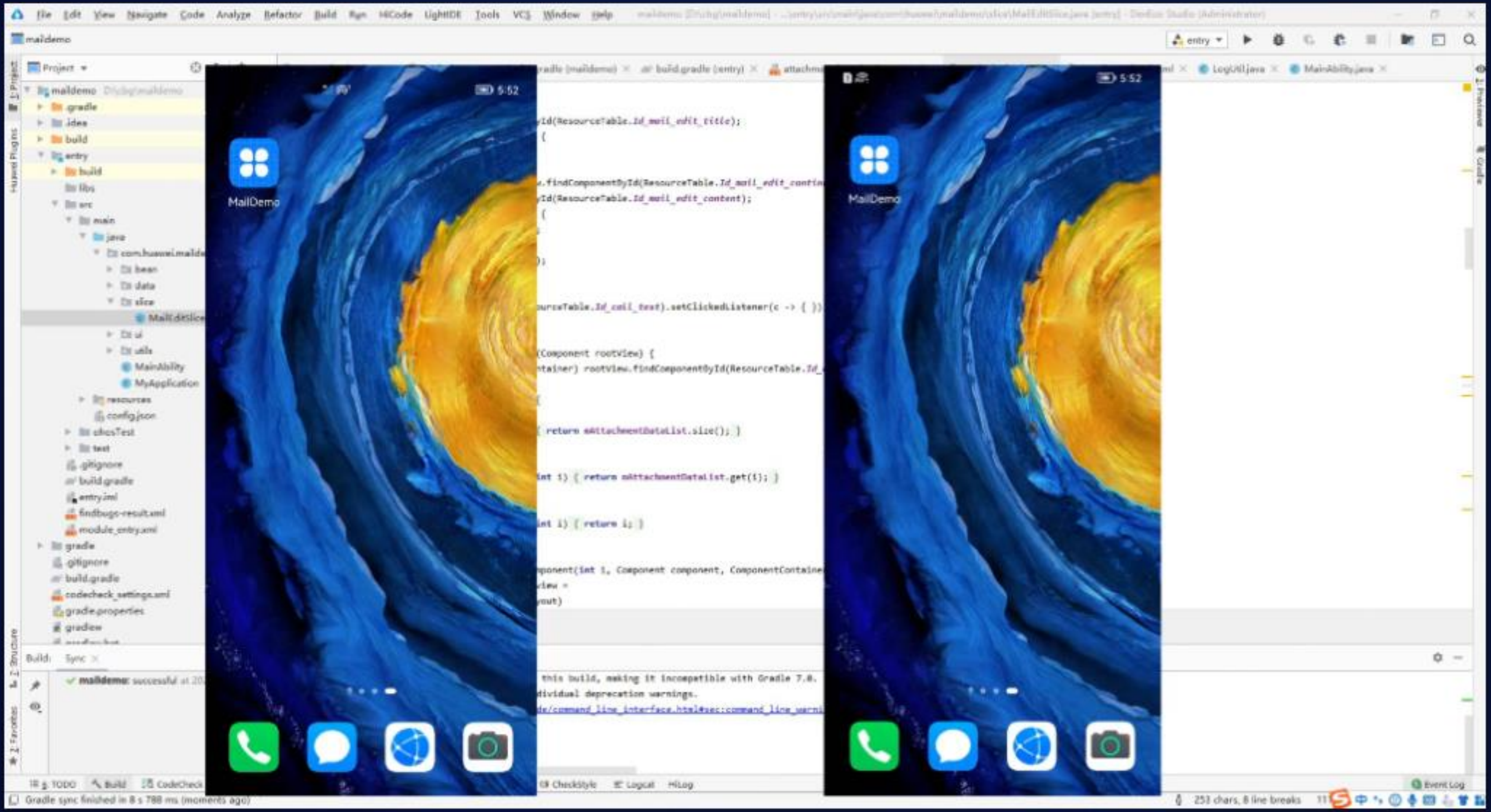
10 恭喜您

11 参考

目 录

1. 概念：Demo、Sample、Cookbook、Codelab
2. 典型场景1：设备感知、发现、数据迁移 （样例：分布式邮件编辑）
3. 典型场景2：用接口描述语言IDL实现进程间通信 （样例：视频跨设备播放控制）
4. 样例代码的开源共建

分布式邮件编辑效果



```
yId(ResourceTable.Id_mail_edit_title);
 {

.findComponentById(ResourceTable.Id_mail_edit_contin
yId(ResourceTable.Id_mail_edit_content);
 {

);

urceTable.Id_mail_test).setClickedListener(c -> { });

(Component rootView) {
tainer) rootView.findComponentById(ResourceTable.Id_
{
{ return mAttachmentDataList.size(); }

int i) { return mAttachmentDataList.get(i); }

int i) { return i; }

ponent(int i, Component component, ComponentContaine
view =
yout)
```

```
 this build, making it incompatible with Gradle 7.0.
dividual deprecation warnings.
de/command_line_interface.html#sec:command_line_warni
```

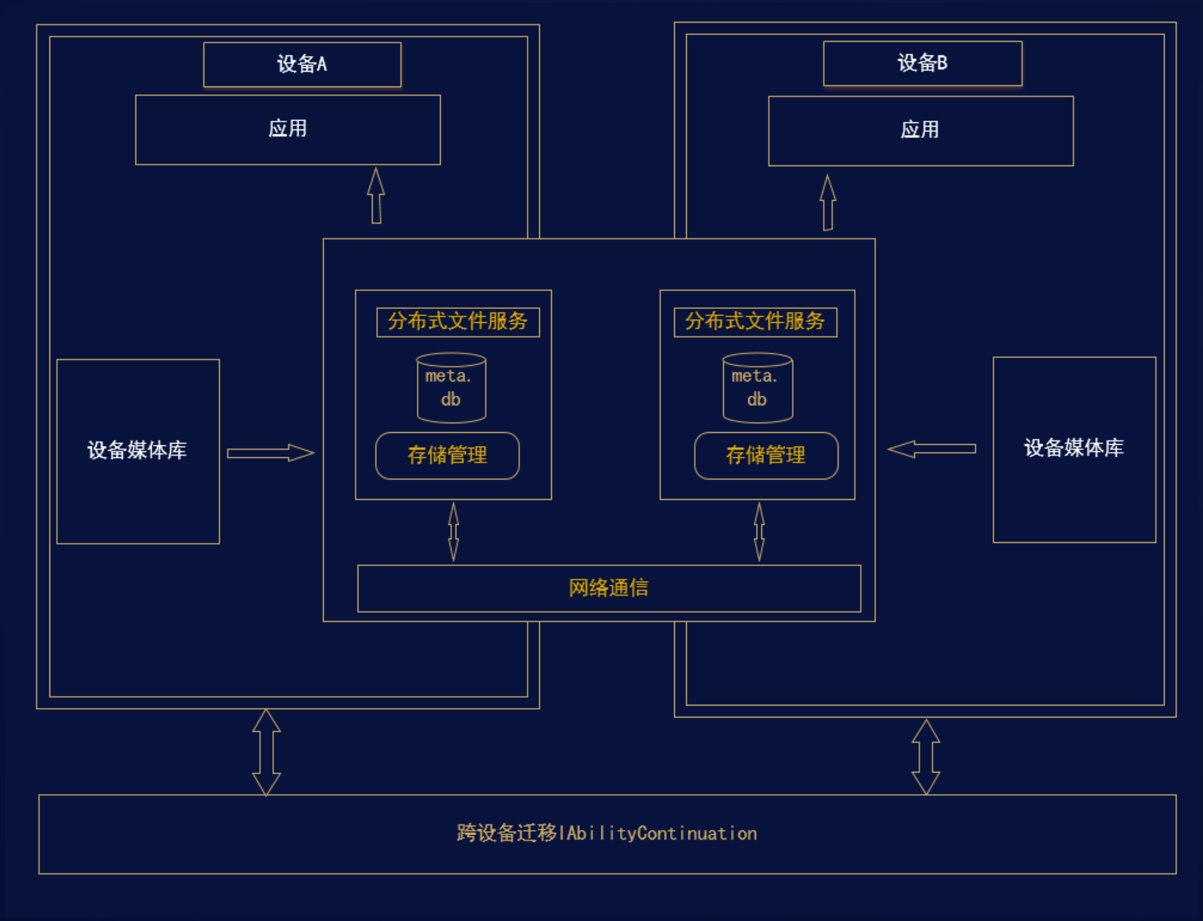
maildemo: successful at 20

Gradle sync finished in 8 s 788 ms (moments ago)

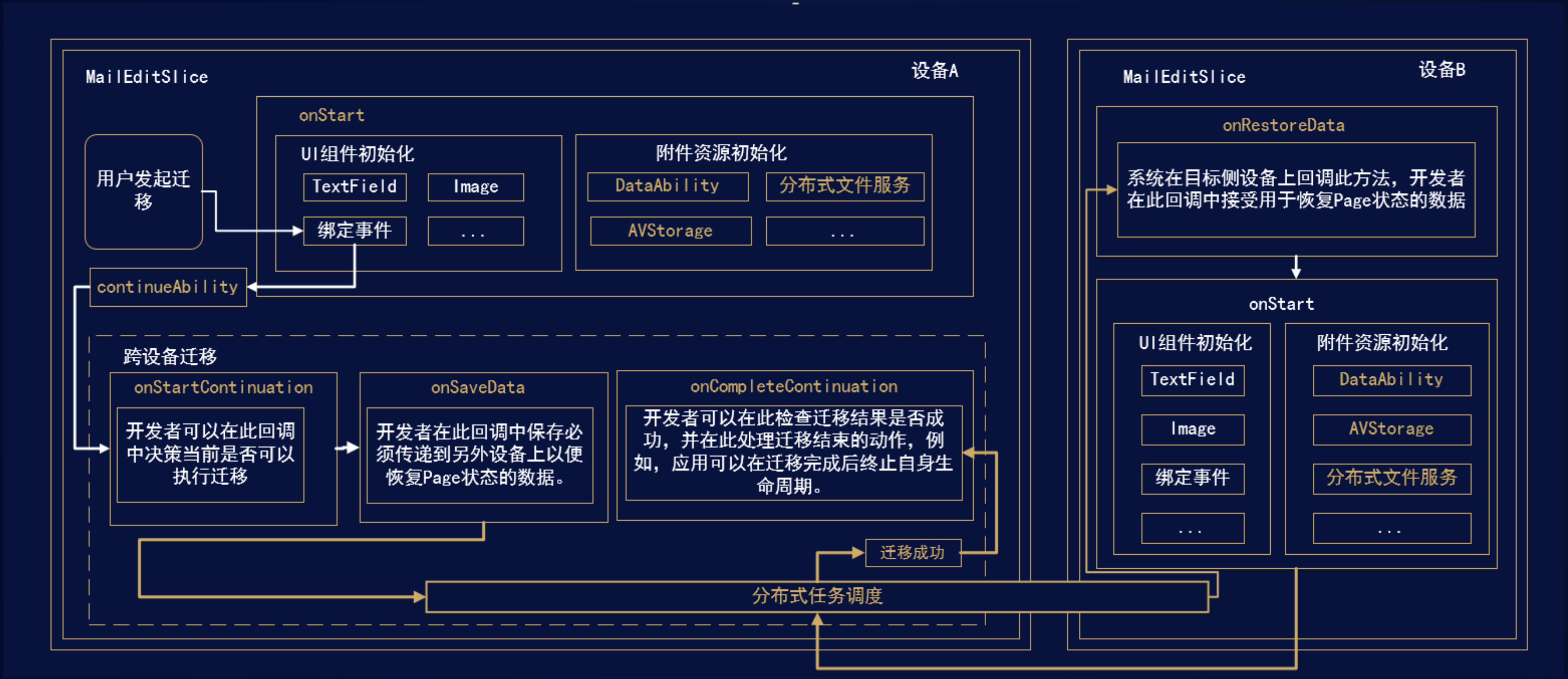
253 chars, 8 line breaks

玩转分布式之分布式邮件编辑

逻辑图



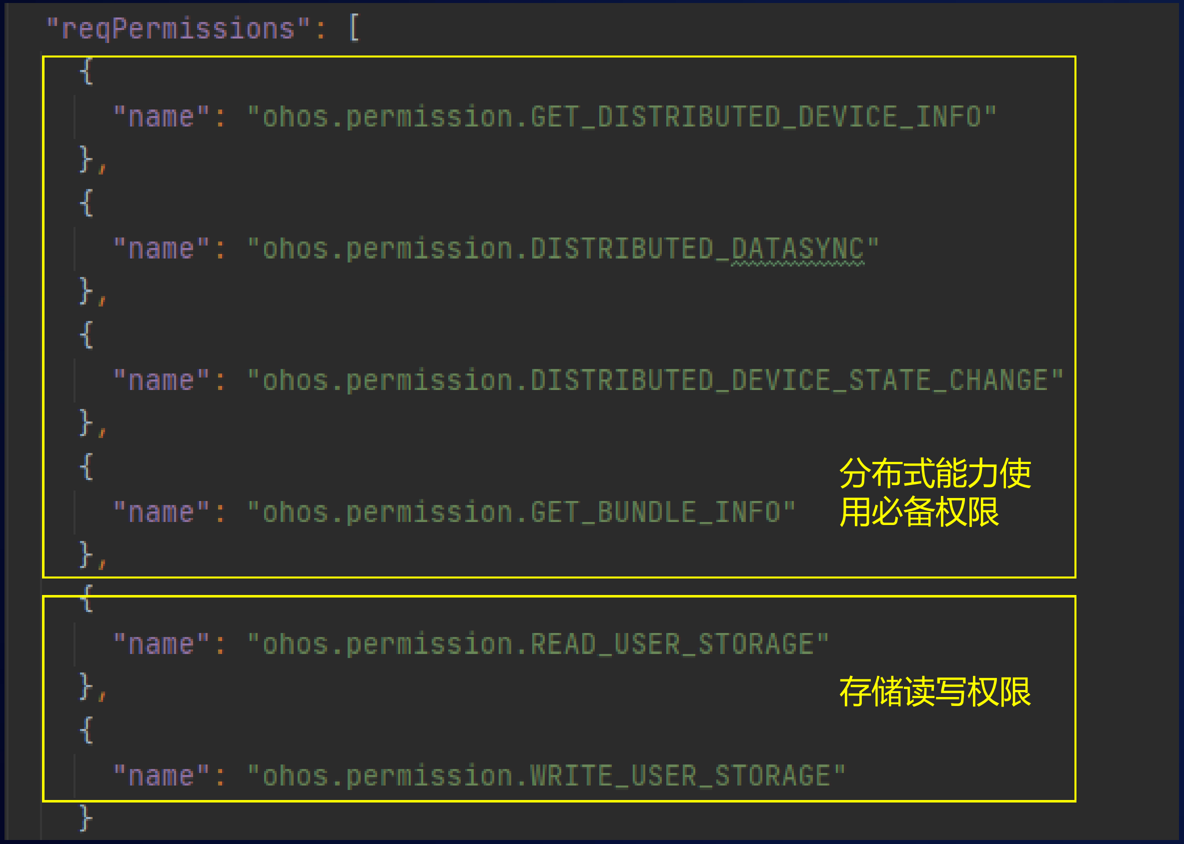
分布式邮件编辑流程图



MailEditSlice
设备A
onStart
用户发起迁移
UI组件初始化
TextField
Image
绑定事件
...
附件资源初始化
DataAbility
分布式文件服务
AVStorage
...
continueAbility
跨设备迁移
onStartContinuation
开发者可以在此回调中决策当前是否可以执行迁移
onSaveData
开发者在此回调中保存必须传递到另外设备上以便恢复Page状态的数据。
onCompleteContinuation
开发者可以在此检查迁移结果是否成功，并在此处理迁移结束的动作，例如，应用可以在迁移完成后终止自身生命周期。
迁移成功
分布式任务调度
MailEditSlice
设备B
onRestoreData
系统在目标侧设备上回调此方法，开发者在此回调中接受用于恢复Page状态的数据
onStart
UI组件初始化
TextField
Image
绑定事件
...
附件资源初始化
DataAbility
AVStorage
分布式文件服务
...

分布式邮件编辑开发步骤详解

Step1:权限声明



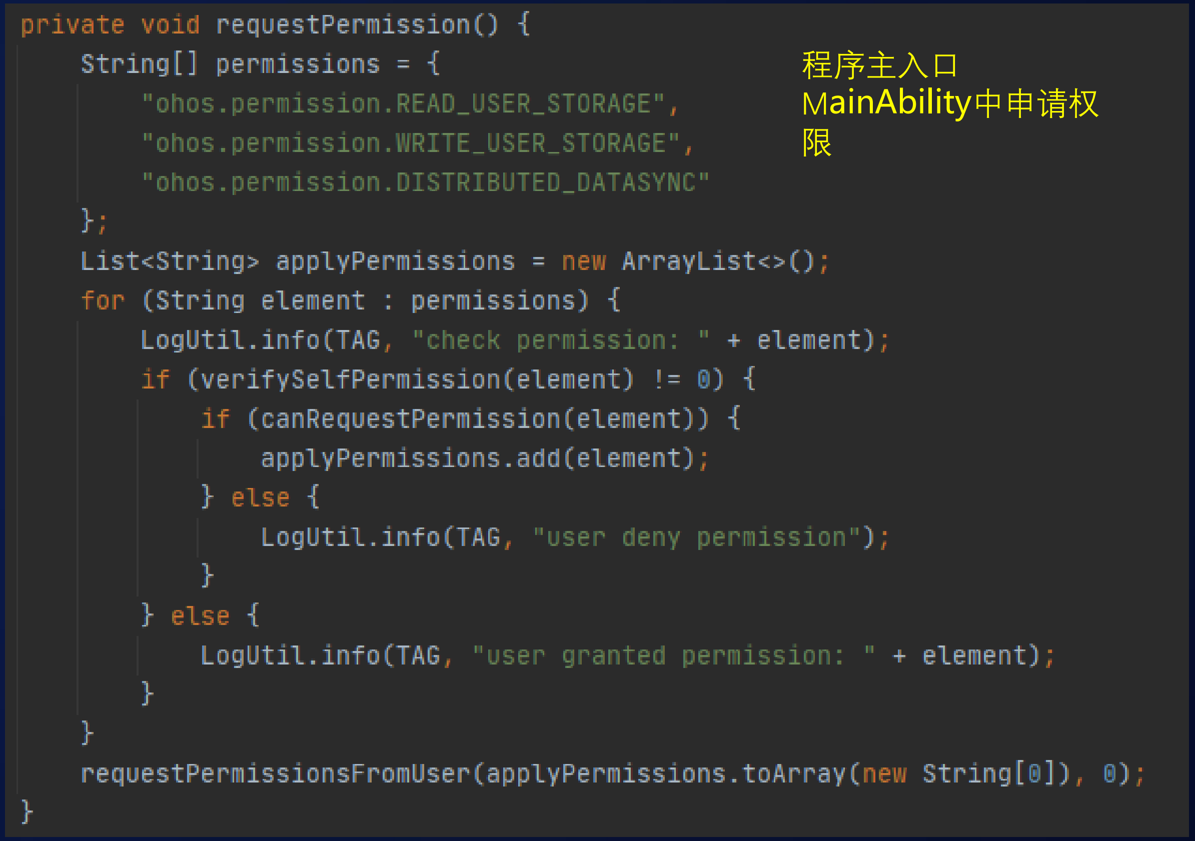
```
"reqPermissions": [
  {
    "name": "ohos.permission.GET_DISTRIBUTED_DEVICE_INFO"
  },
  {
    "name": "ohos.permission.DISTRIBUTED_DATASYNC"
  },
  {
    "name": "ohos.permission.DISTRIBUTED_DEVICE_STATE_CHANGE"
  },
  {
    "name": "ohos.permission.GET_BUNDLE_INFO"
  },
  {
    "name": "ohos.permission.READ_USER_STORAGE"
  },
  {
    "name": "ohos.permission.WRITE_USER_STORAGE"
  }
```

分布式能力使用必备权限

存储读写权限

config.json文件

权限申请

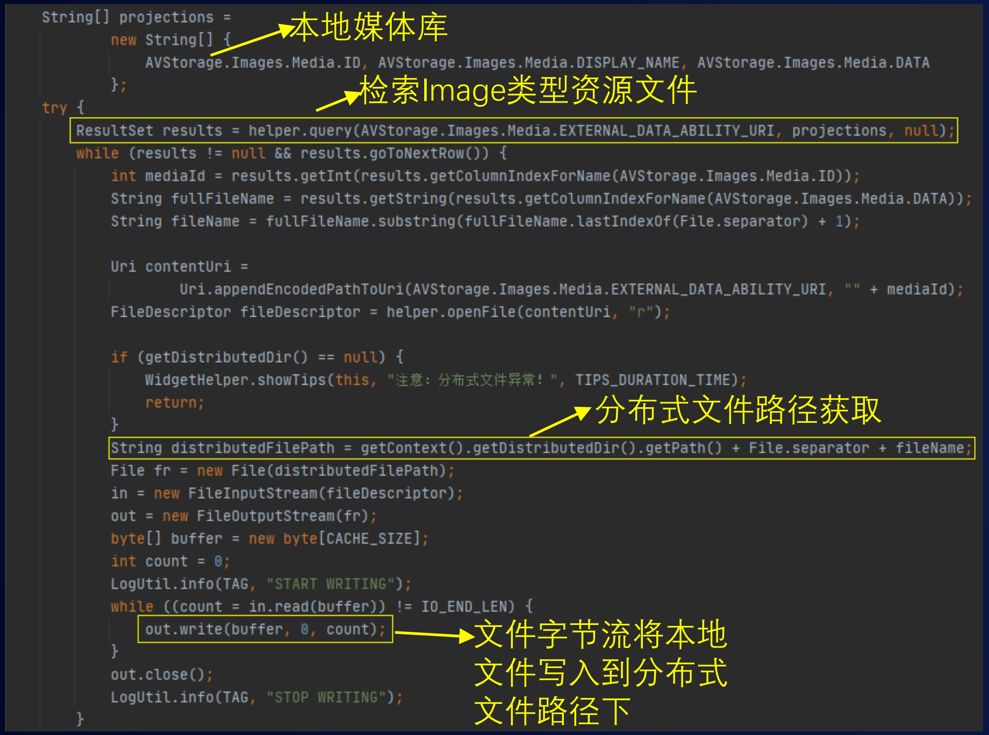


```
private void requestPermission() {
    String[] permissions = {
        "ohos.permission.READ_USER_STORAGE",
        "ohos.permission.WRITE_USER_STORAGE",
        "ohos.permission.DISTRIBUTED_DATASYNC"
    };
    List<String> applyPermissions = new ArrayList<>();
    for (String element : permissions) {
        LogUtil.info(TAG, "check permission: " + element);
        if (verifySelfPermission(element) != 0) {
            if (canRequestPermission(element)) {
                applyPermissions.add(element);
            } else {
                LogUtil.info(TAG, "user deny permission");
            }
        } else {
            LogUtil.info(TAG, "user granted permission: " + element);
        }
    }
    requestPermissionsFromUser(applyPermissions.toArray(new String[0]), 0);
}
```

程序主入口MainAbility中申请权限

分布式邮件编辑开发步骤详解

Step2:分布式文件获取



```java
String[] projections =
        new String[] {
            AVStorage.Images.Media.ID, AVStorage.Images.Media.DISPLAY_NAME, AVStorage.Images.Media.DATA
        };
try {
    ResultSet results = helper.query(AVStorage.Images.Media.EXTERNAL_DATA_ABILITY_URI, projections, null);
    while (results != null && results.goToNextRow()) {
        int mediaId = results.getInt(results.getColumnIndexForName(AVStorage.Images.Media.ID));
        String fullFileName = results.getString(results.getColumnIndexForName(AVStorage.Images.Media.DATA));
        String fileName = fullFileName.substring(fullFileName.lastIndexOf(File.separator) + 1);

        Uri contentUri =
                Uri.appendEncodedPathToUri(AVStorage.Images.Media.EXTERNAL_DATA_ABILITY_URI, "" + mediaId);
        FileDescriptor fileDescriptor = helper.openFile(contentUri, "r");

        if (getDistributedDir() == null) {
            WidgetHelper.showTips(this, "注意：分布式文件异常！", TIPS_DURATION_TIME);
            return;
        }
        String distributedFilePath = getContext().getDistributedDir().getPath() + File.separator + fileName;
        File fr = new File(distributedFilePath);
        in = new FileInputStream(fileDescriptor);
        out = new FileOutputStream(fr);
        byte[] buffer = new byte[CACHE_SIZE];
        int count = 0;
        LogUtil.info(TAG, "START WRITING");
        while ((count = in.read(buffer)) != IO_END_LEN) {
            out.write(buffer, 0, count);
        }
        out.close();
        LogUtil.info(TAG, "STOP WRITING");
    }
```

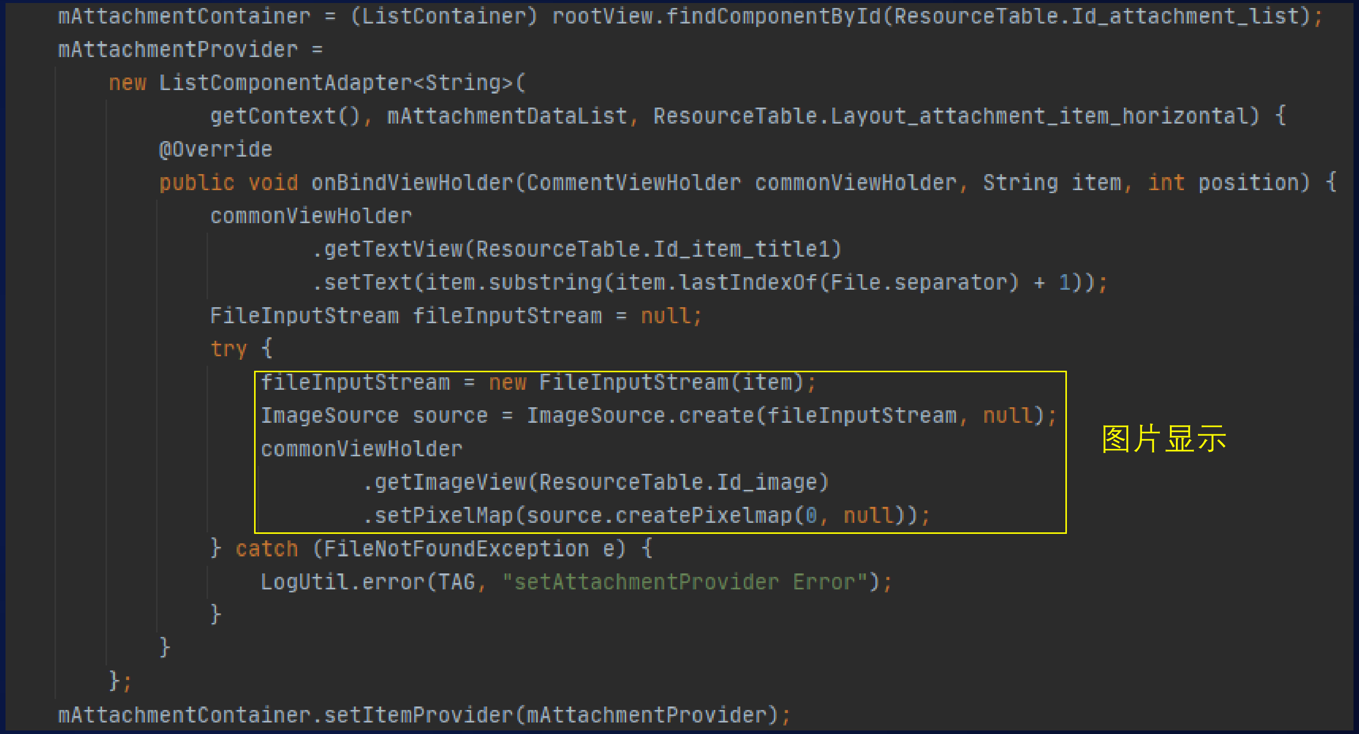
本地媒体库

检索Image类型资源文件

分布式文件路径获取

文件字节流将本地
文件写入到分布式
文件路径下

附件文件展示



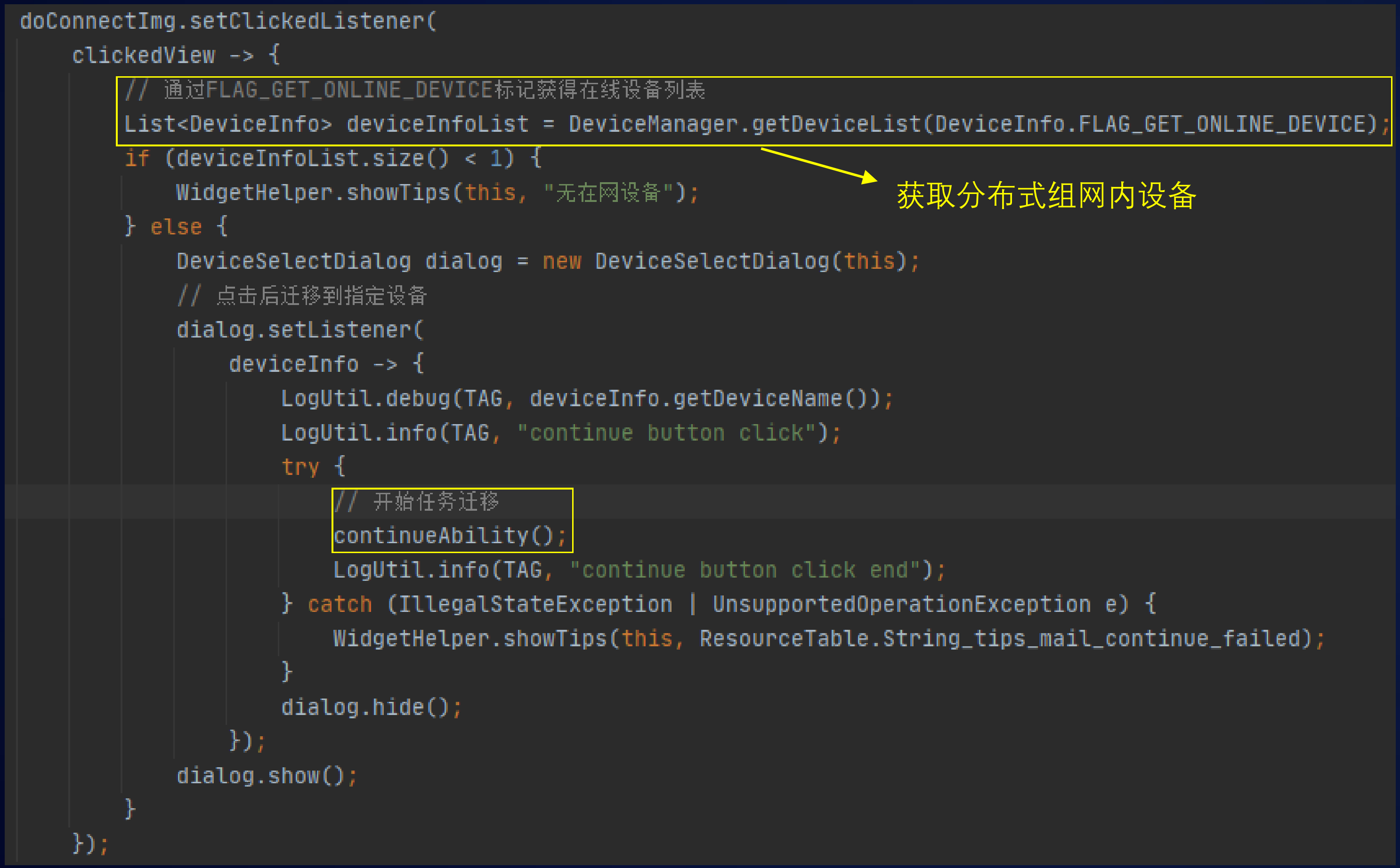
```java
mAttachmentContainer = (ListContainer) rootView.findComponentById(ResourceTable.Id_attachment_list);
mAttachmentProvider =
    new ListComponentAdapter<String>(
            getContext(), mAttachmentDataList, ResourceTable.Layout_attachment_item_horizontal) {
        @Override
        public void onBindViewHolder(CommentViewHolder commonViewHolder, String item, int position) {
            commonViewHolder
                    .getTextView(ResourceTable.Id_item_title1)
                    .setText(item.substring(item.lastIndexOf(File.separator) + 1));
            FileInputStream fileInputStream = null;
            try {
                fileInputStream = new FileInputStream(item);
                ImageSource source = ImageSource.create(fileInputStream, null);
                commonViewHolder
                        .getImageView(ResourceTable.Id_image)
                        .setPixelMap(source.createPixelmap(0, null));
            } catch (FileNotFoundException e) {
                LogUtil.error(TAG, "setAttachmentProvider Error");
            }
        }
    };
mAttachmentContainer.setItemProvider(mAttachmentProvider);
```

图片显示

MailEditSlice.java文件

分布式邮件编辑开发步骤详解

Step3:获取分布式设备并迁移

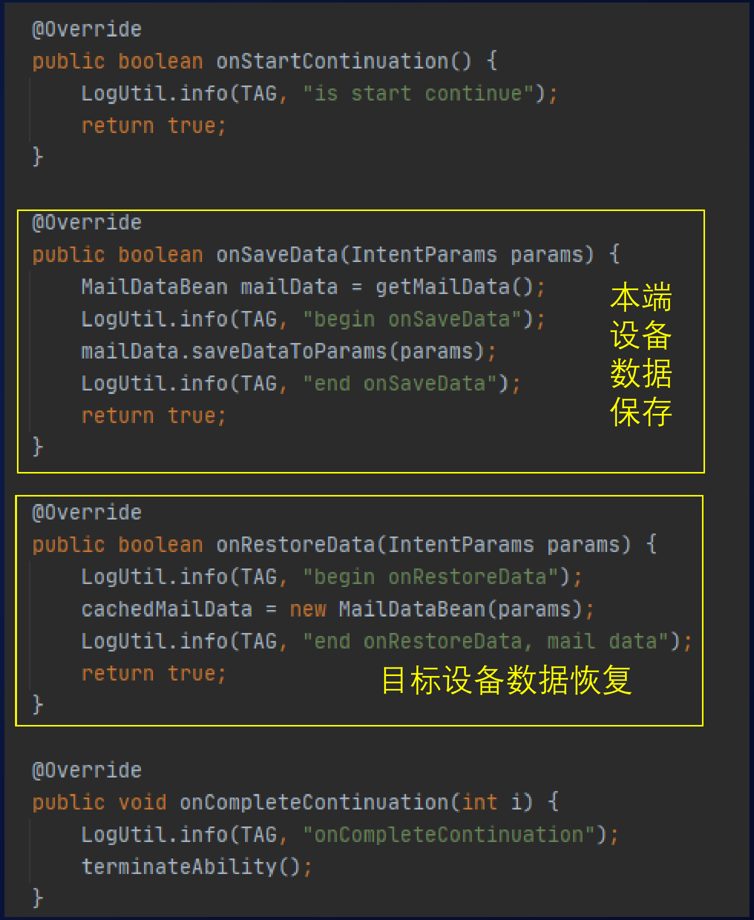


```
doConnectImg.setClickedListener(
    clickedView -> {
        // 通过FLAG_GET_ONLINE_DEVICE标记获得在线设备列表
        List<DeviceInfo> deviceInfoList = DeviceManager.getDeviceList(DeviceInfo.FLAG_GET_ONLINE_DEVICE);
        if (deviceInfoList.size() < 1) {
            WidgetHelper.showTips(this, "无在网设备");
        } else {
            DeviceSelectDialog dialog = new DeviceSelectDialog(this);
            // 点击后迁移到指定设备
            dialog.setListener(
                deviceInfo -> {
                    LogUtil.debug(TAG, deviceInfo.getDeviceName());
                    LogUtil.info(TAG, "continue button click");
                    try {
                        // 开始任务迁移
                        continueAbility();
                        LogUtil.info(TAG, "continue button click end");
                    } catch (IllegalStateException | UnsupportedOperationException e) {
                        WidgetHelper.showTips(this, ResourceTable.String_tips_mail_continue_failed);
                    }
                    dialog.hide();
                });
            dialog.show();
        }
    });
```

获取分布式组网内设备

MailEditSlice.java文件

IAbilityContinuation接口实现



```
@Override
public boolean onStartContinuation() {
    LogUtil.info(TAG, "is start continue");
    return true;
}

@Override
public boolean onSaveData(IntentParams params) {
    MailDataBean mailData = getMailData();
    LogUtil.info(TAG, "begin onSaveData");
    mailData.saveDataToParams(params);
    LogUtil.info(TAG, "end onSaveData");
    return true;
}

@Override
public boolean onRestoreData(IntentParams params) {
    LogUtil.info(TAG, "begin onRestoreData");
    cachedMailData = new MailDataBean(params);
    LogUtil.info(TAG, "end onRestoreData, mail data");
    return true;
}

@Override
public void onCompleteContinuation(int i) {
    LogUtil.info(TAG, "onCompleteContinuation");
    terminateAbility();
}
```

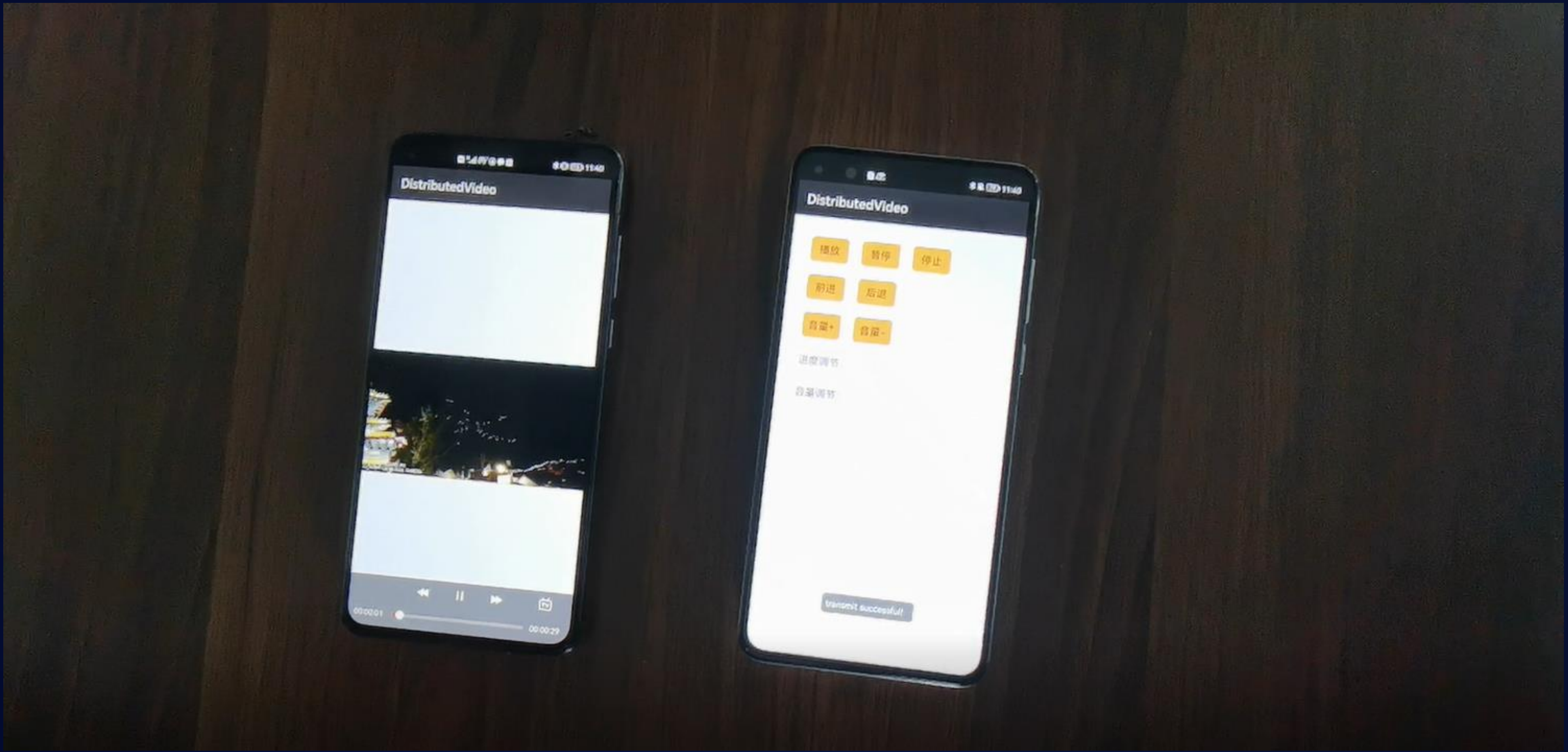
本端设备数据保存

目标设备数据恢复

目 录

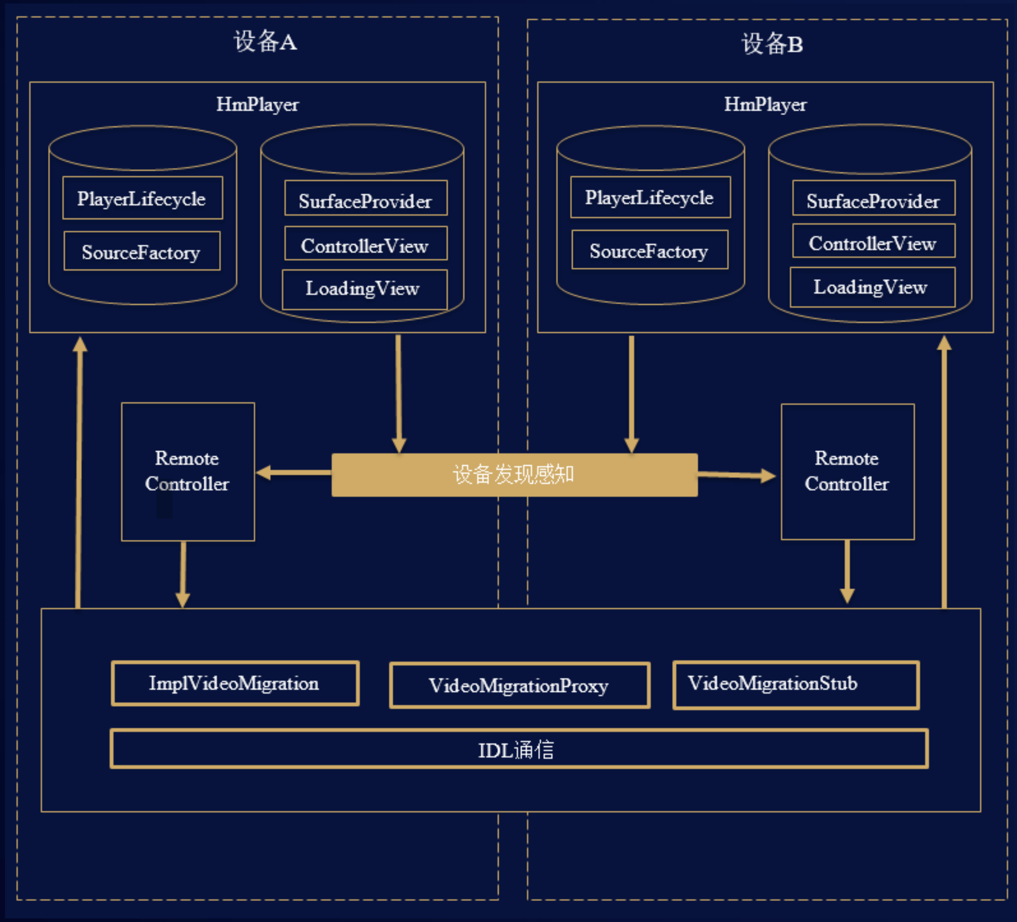
1. 概念：Demo、Sample、Cookbook、Codelab
2. 典型场景1：设备感知、发现、数据迁移 （样例：分布式邮件编辑）
3. 典型场景2：用接口描述语言IDL实现进程间通信 （样例：视频跨设备播放控制）
4. 样例代码的开源共建

分布式视频播放控制效果

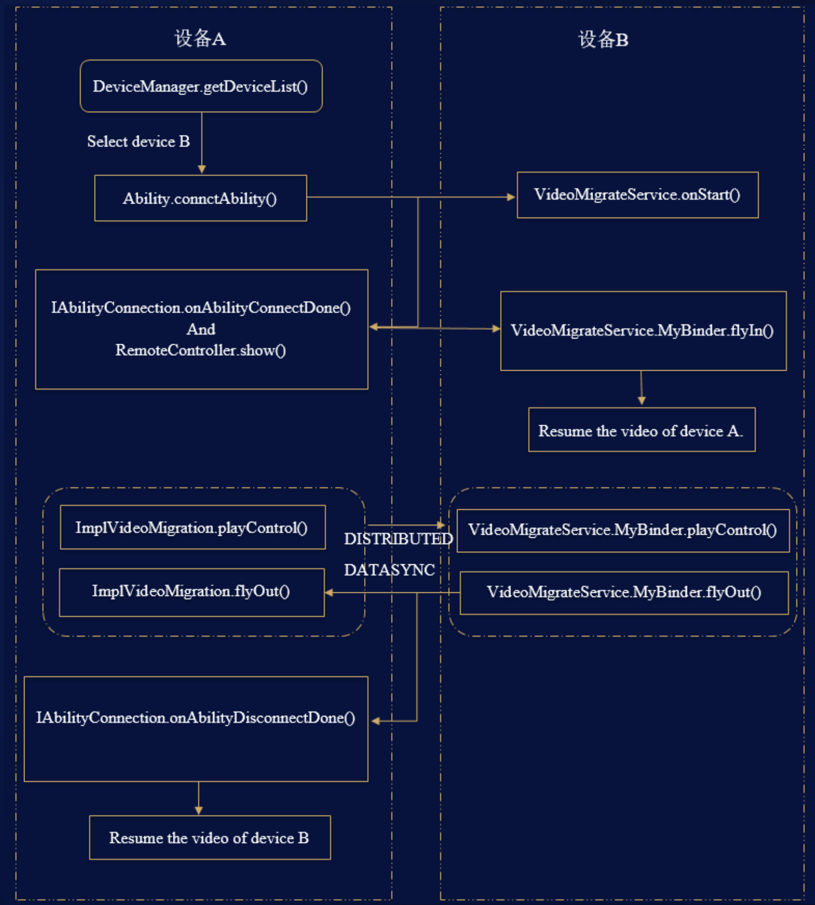


玩转分布式之分布式视频播放控制

逻辑图

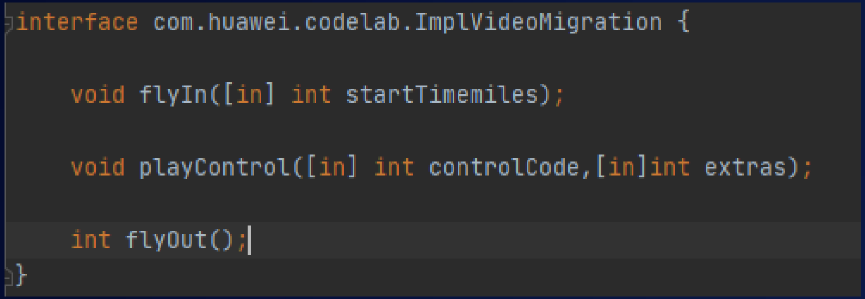


流程图

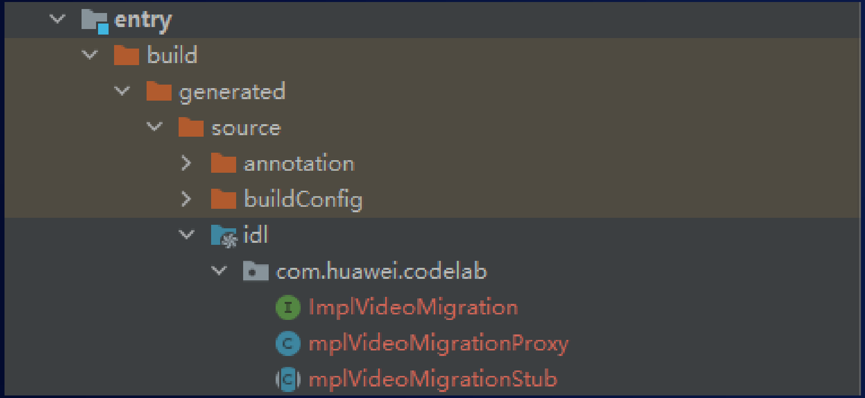

设备A
设备B
DeviceManager.getDeviceList()
Select device B
Ability.connctAbility()
VideoMigrateService.onStart()
IAbilityConnection.onAbilityConnectDone()
And
RemoteController.show()
VideoMigrateService.MyBinder.flyIn()
Resume the video of device A.
ImplVideoMigration.playControl()
DISTRIBUTED
DATASYNC
VideoMigrateService.MyBinder.playControl()
ImplVideoMigration.flyOut()
VideoMigrateService.MyBinder.flyOut()
IAbilityConnection.onAbilityDisconnectDone()
Resume the video of device B

分布式视频播放开发步骤详解

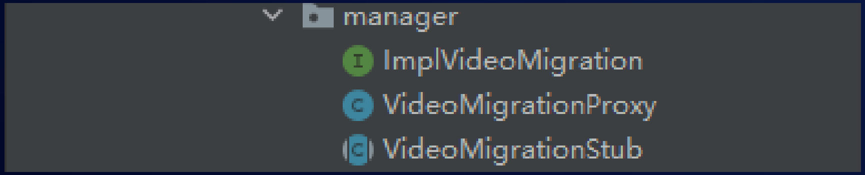
Step1:IDL接口编写



```
interface com.huawei.codelab.ImplVideoMigration {

    void flyIn([in] int startTimemiles);

    void playControl([in] int controlCode,[in]int extras);

    int flyOut();
}
```

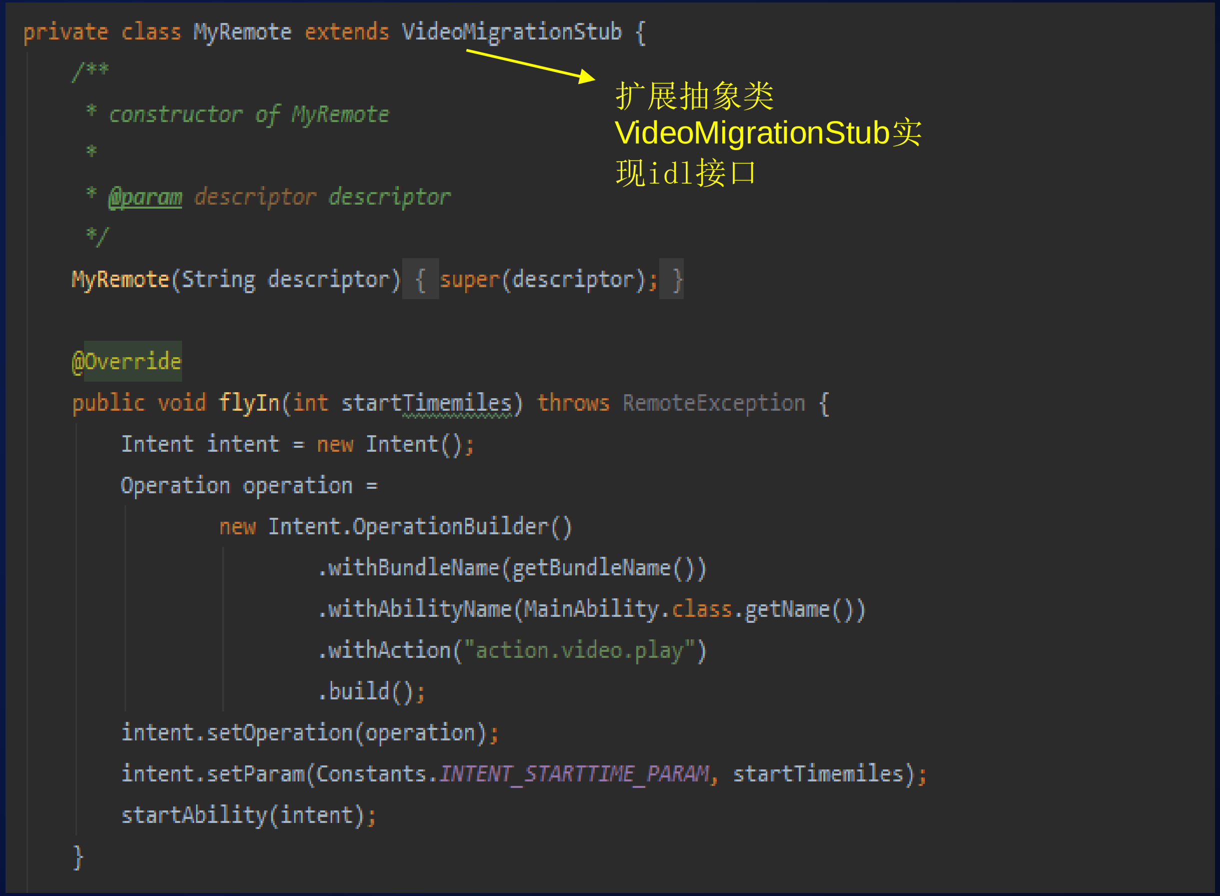


```
entry
  build
    generated
      source
        annotation
        buildConfig
        idl
          com.huawei.codelab
            ImplVideoMigration
            mplVideoMigrationProxy
            mplVideoMigrationStub
```



```
manager
  ImplVideoMigration
  VideoMigrationProxy
  VideoMigrationStub
```

继承IDL文件



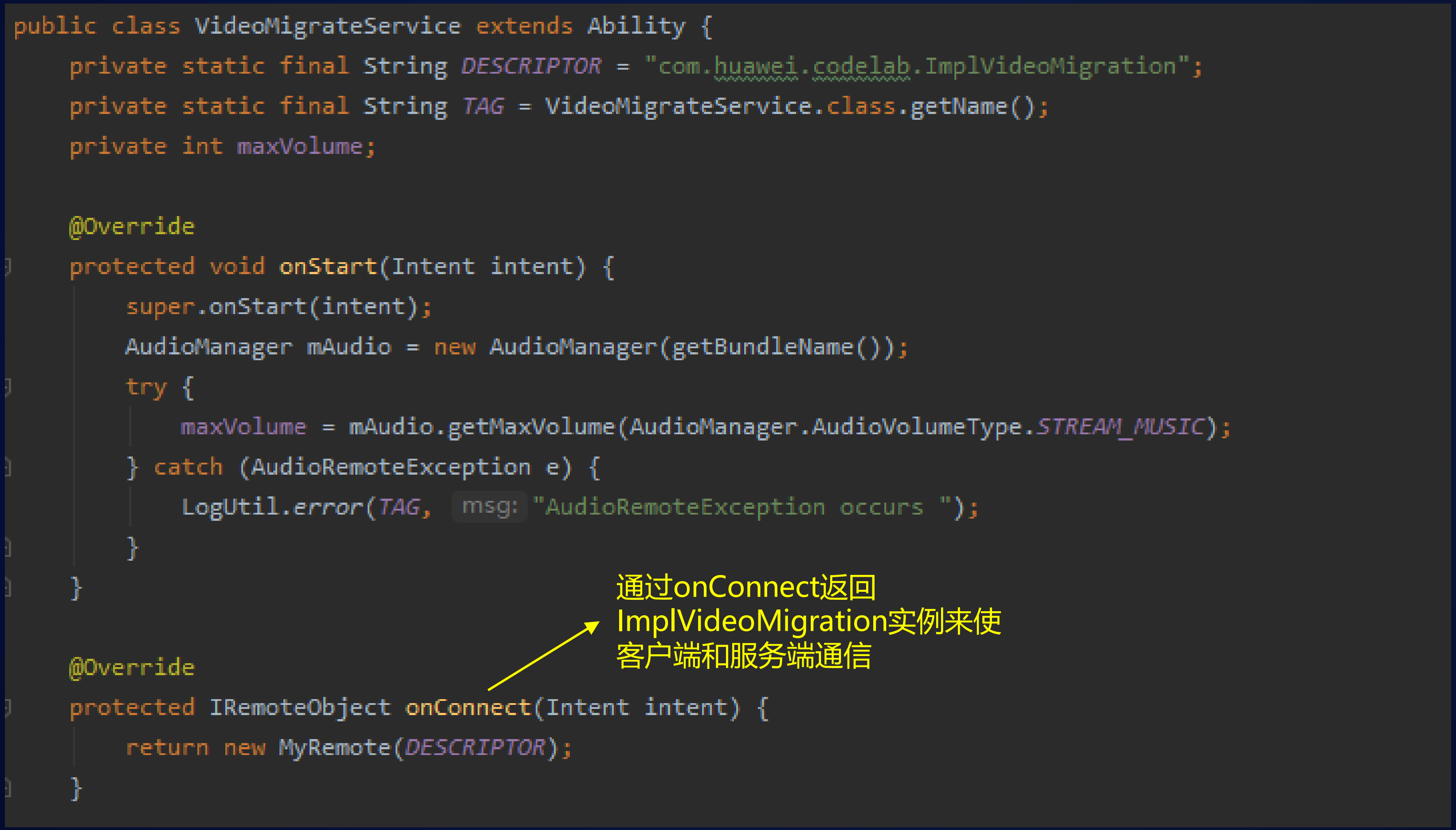
```
private class MyRemote extends VideoMigrationStub {
    /**
     * constructor of MyRemote
     *
     * @param descriptor descriptor
     */
    MyRemote(String descriptor) { super(descriptor); }

    @Override
    public void flyIn(int startTimemiles) throws RemoteException {
        Intent intent = new Intent();
        Operation operation =
                new Intent.OperationBuilder()
                        .withBundleName(getBundleName())
                        .withAbilityName(MainAbility.class.getName())
                        .withAction("action.video.play")
                        .build();
        intent.setOperation(operation);
        intent.setParam(Constants.INTENT_STARTTIME_PARAM, startTimemiles);
        startAbility(intent);
    }
```

扩展抽象类VideoMigrationStub实现idl接口

VideoMigrationService.java文件

分布式邮件编辑开发步骤详解

Step2:服务端向客户端公开接口

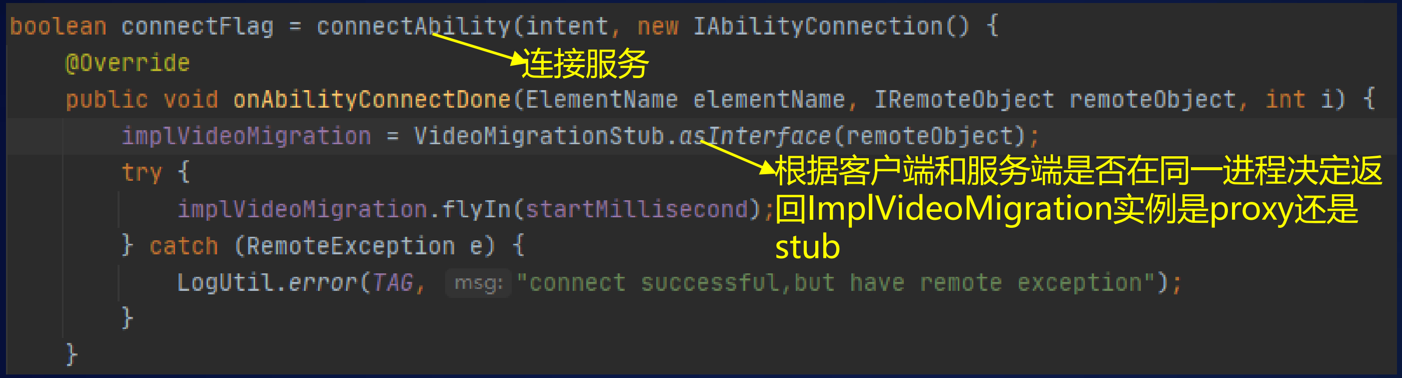


```
public class VideoMigrateService extends Ability {
    private static final String DESCRIPTOR = "com.huawei.codelab.ImplVideoMigration";
    private static final String TAG = VideoMigrateService.class.getName();
    private int maxVolume;

    @Override
    protected void onStart(Intent intent) {
        super.onStart(intent);
        AudioManager mAudio = new AudioManager(getBundleName());
        try {
            maxVolume = mAudio.getMaxVolume(AudioManager.AudioVolumeType.STREAM_MUSIC);
        } catch (AudioRemoteException e) {
            LogUtil.error(TAG,  msg: "AudioRemoteException occurs ");
        }
    }

    @Override
    protected IRemoteObject onConnect(Intent intent) {
        return new MyRemote(DESCRIPTOR);
    }
```

通过onConnect返回ImplVideoMigration实例来使客户端和服务端通信

分布式邮件编辑开发步骤详解

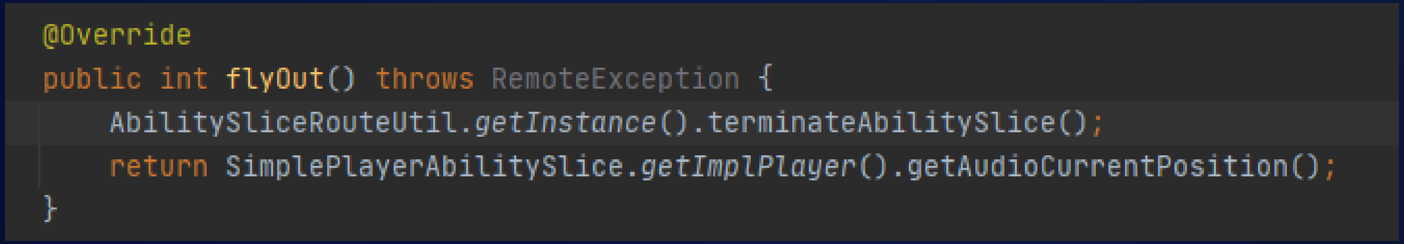
Step3:客户端连接服务实现视频跨设备协同



```java
boolean connectFlag = connectAbility(intent, new IAbilityConnection() {
    @Override
    public void onAbilityConnectDone(ElementName elementName, IRemoteObject remoteObject, int i) {
        implVideoMigration = VideoMigrationStub.asInterface(remoteObject);
        try {
            implVideoMigration.flyIn(startMillisecond);
        } catch (RemoteException e) {
            LogUtil.error(TAG,  msg: "connect successful,but have remote exception");
        }
    }
```

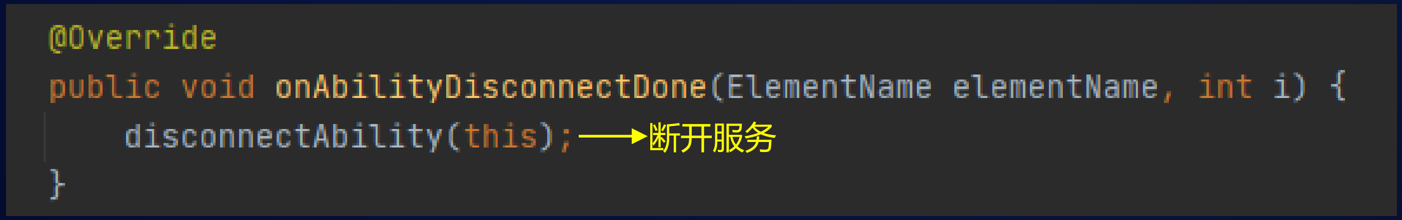
连接服务

根据客户端和服务端是否在同一进程决定返回ImplVideoMigration实例是proxy还是stub



```java
@Override
public int flyOut() throws RemoteException {
    AbilitySliceRouteUtil.getInstance().terminateAbilitySlice();
    return SimplePlayerAbilitySlice.getImplPlayer().getAudioCurrentPosition();
}
```

客服务端执行退出操作

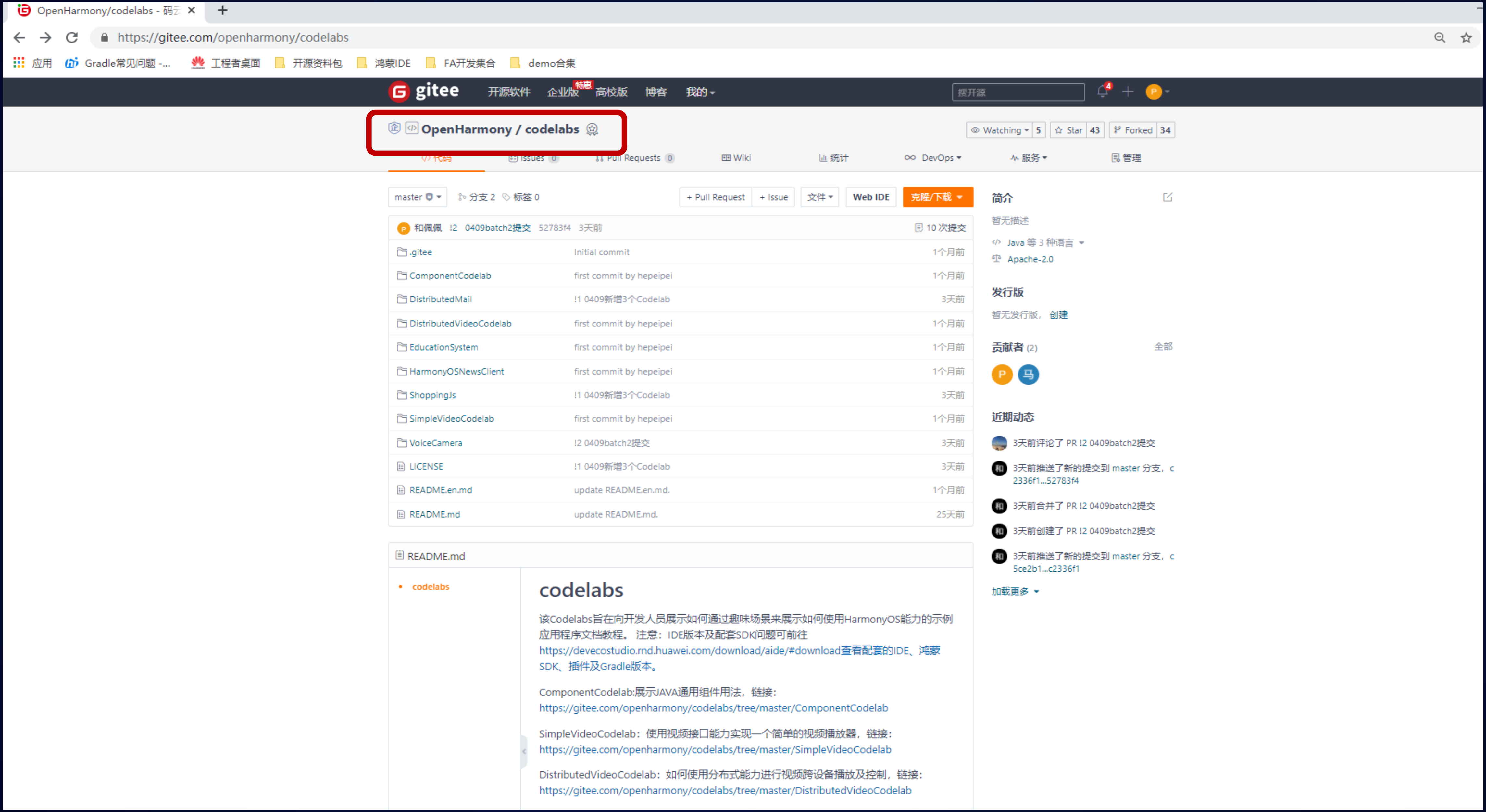


```java
@Override
public void onAbilityDisconnectDone(ElementName elementName, int i) {
    disconnectAbility(this);
}
```

断开服务

目录

1. 概念：Demo、Sample、Cookbook、Codelab
2. 典型场景1：设备感知、发现、数据迁移 （样例：分布式邮件编辑）
3. 典型场景2：用接口描述语言IDL实现进程间通信 （样例：视频跨设备播放控制）
4. 样例代码的开源共建



OpenHarmony / codelabs

和佩佩 !2 0409batch2提交 52783f4 3天前 10 次提交

.gitee	Initial commit	1个月前
ComponentCodelab	first commit by hepeipei	1个月前
DistributedMail	!1 0409新增3个Codelab	3天前
DistributedVideoCodelab	first commit by hepeipei	1个月前
EducationSystem	first commit by hepeipei	1个月前
HarmonyOSNewsClient	first commit by hepeipei	1个月前
ShoppingJs	!1 0409新增3个Codelab	3天前
SimpleVideoCodelab	first commit by hepeipei	1个月前
VoiceCamera	!2 0409batch2提交	3天前
LICENSE	!1 0409新增3个Codelab	3天前
README.en.md	update README.en.md.	1个月前
README.md	update README.md.	25天前

简介

暂无描述

Java 等 3 种语言

Apache-2.0

发行版

暂无发行版， 创建

贡献者 (2) 全部

近期动态

3天前评论了 PR !2 0409batch2提交

3天前推送了新的提交到 master 分支，c 2336f1...52783f4

3天前合并了 PR !2 0409batch2提交

3天前创建了 PR !2 0409batch2提交

3天前推送了新的提交到 master 分支，c 5ce2b1...c2336f1

加载更多

README.md

- codelabs

codelabs

该Codelabs旨在向开发人员展示如何通过趣味场景来展示如何使用HarmonyOS能力的示例应用程序文档教程。 注意：IDE版本及配套SDK问题可前往https://devecostudio.rnd.huawei.com/download/aide/#download查看配套的IDE、鸿蒙SDK、插件及Gradle版本。

ComponentCodelab:展示JAVA通用组件用法，链接：https://gitee.com/openharmony/codelabs/tree/master/ComponentCodelab

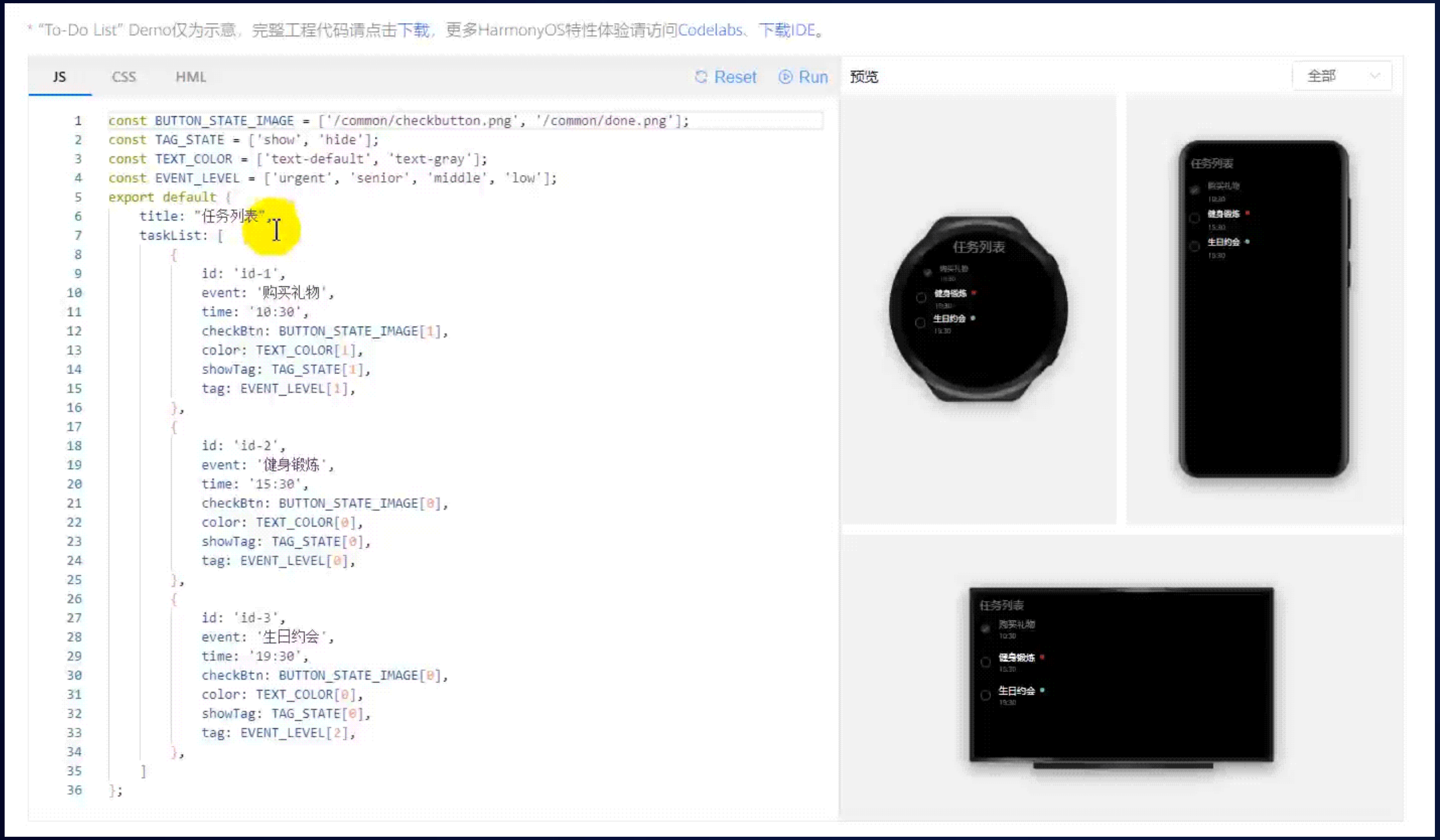
SimpleVideoCodelab：使用视频接口能力实现一个简单的视频播放器，链接：https://gitee.com/openharmony/codelabs/tree/master/SimpleVideoCodelab

DistributedVideoCodelab：如何使用分布式能力进行视频跨设备播放及控制，链接：https://gitee.com/openharmony/codelabs/tree/master/DistributedVideoCodelab

贡献与分享案例

1. 大二学生：深圳大学的张诏添 - 黑白翻棋、俄罗斯方块等游戏改装
2. 中科院研究所-条形扫码
3. 软通动力三方件开发Demo

扫码在线体验HarmonyOS应用开发



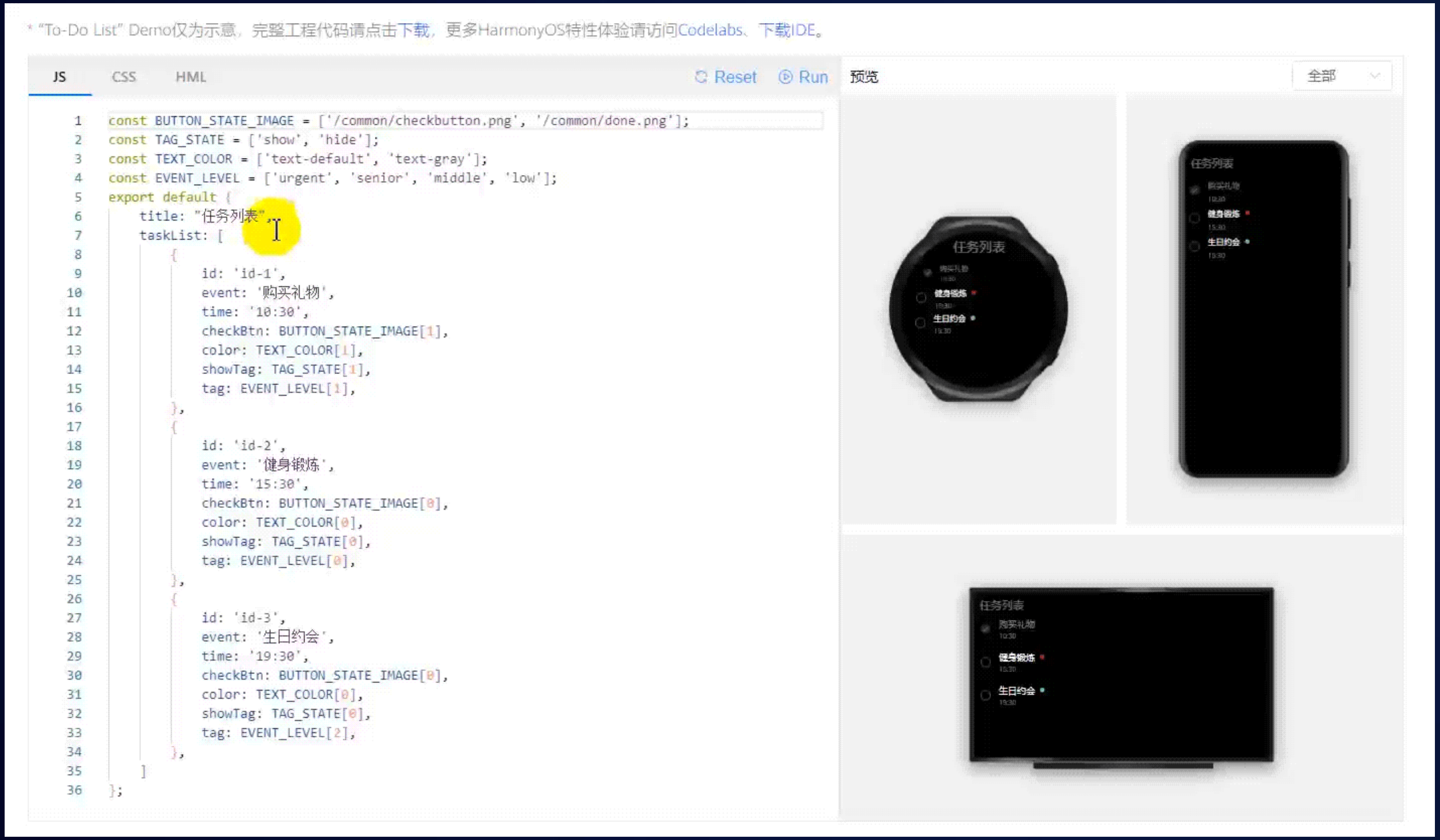
* "To-Do List" Demo仅为示意，完整工程代码请点击下载，更多HarmonyOS特性体验请访问Codelabs、下载IDE。

```javascript
const BUTTON_STATE_IMAGE = ['/common/checkbutton.png', '/common/done.png'];
const TAG_STATE = ['show', 'hide'];
const TEXT_COLOR = ['text-default', 'text-gray'];
const EVENT_LEVEL = ['urgent', 'senior', 'middle', 'low'];
export default {
    title: "任务列表",
    taskList: [
        {
            id: 'id-1',
            event: '购买礼物',
            time: '10:30',
            checkBtn: BUTTON_STATE_IMAGE[1],
            color: TEXT_COLOR[1],
            showTag: TAG_STATE[1],
            tag: EVENT_LEVEL[1],
        },
        {
            id: 'id-2',
            event: '健身锻炼',
            time: '15:30',
            checkBtn: BUTTON_STATE_IMAGE[0],
            color: TEXT_COLOR[0],
            showTag: TAG_STATE[0],
            tag: EVENT_LEVEL[0],
        },
        {
            id: 'id-3',
            event: '生日约会',
            time: '19:30',
            checkBtn: BUTTON_STATE_IMAGE[0],
            color: TEXT_COLOR[0],
            showTag: TAG_STATE[0],
            tag: EVENT_LEVEL[2],
        },
    ]
};
```

HarmonyOS应用开发在线体验

HarmonyOS开发者微信公众号

扫码在线体验HarmonyOS应用开发



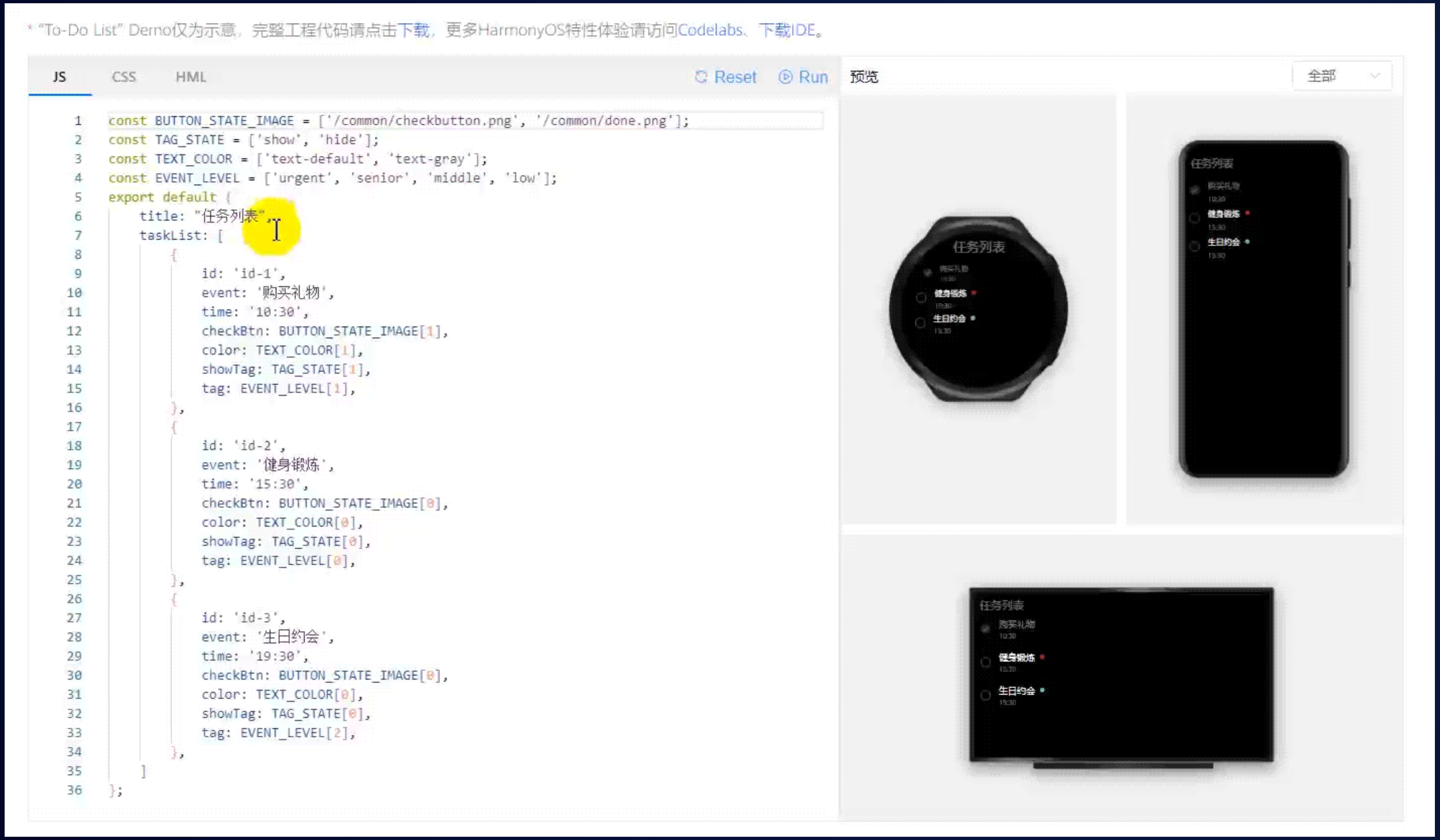
* “To-Do List” Demo仅为示意，完整工程代码请点击下载，更多HarmonyOS特性体验请访问Codelabs、下载IDE。

```
const BUTTON_STATE_IMAGE = ['/common/checkbutton.png', '/common/done.png'];
const TAG_STATE = ['show', 'hide'];
const TEXT_COLOR = ['text-default', 'text-gray'];
const EVENT_LEVEL = ['urgent', 'senior', 'middle', 'low'];
export default {
    title: "任务列表",
    taskList: [
        {
            id: 'id-1',
            event: '购买礼物',
            time: '10:30',
            checkBtn: BUTTON_STATE_IMAGE[1],
            color: TEXT_COLOR[1],
            showTag: TAG_STATE[1],
            tag: EVENT_LEVEL[1],
        },
        {
            id: 'id-2',
            event: '健身锻炼',
            time: '15:30',
            checkBtn: BUTTON_STATE_IMAGE[0],
            color: TEXT_COLOR[0],
            showTag: TAG_STATE[0],
            tag: EVENT_LEVEL[0],
        },
        {
            id: 'id-3',
            event: '生日约会',
            time: '19:30',
            checkBtn: BUTTON_STATE_IMAGE[0],
            color: TEXT_COLOR[0],
            showTag: TAG_STATE[0],
            tag: EVENT_LEVEL[2],
        },
    ]
};
```

HarmonyOS应用开发在线体验

HarmonyOS开发者微信公众号

扫码在线体验HarmonyOS应用开发



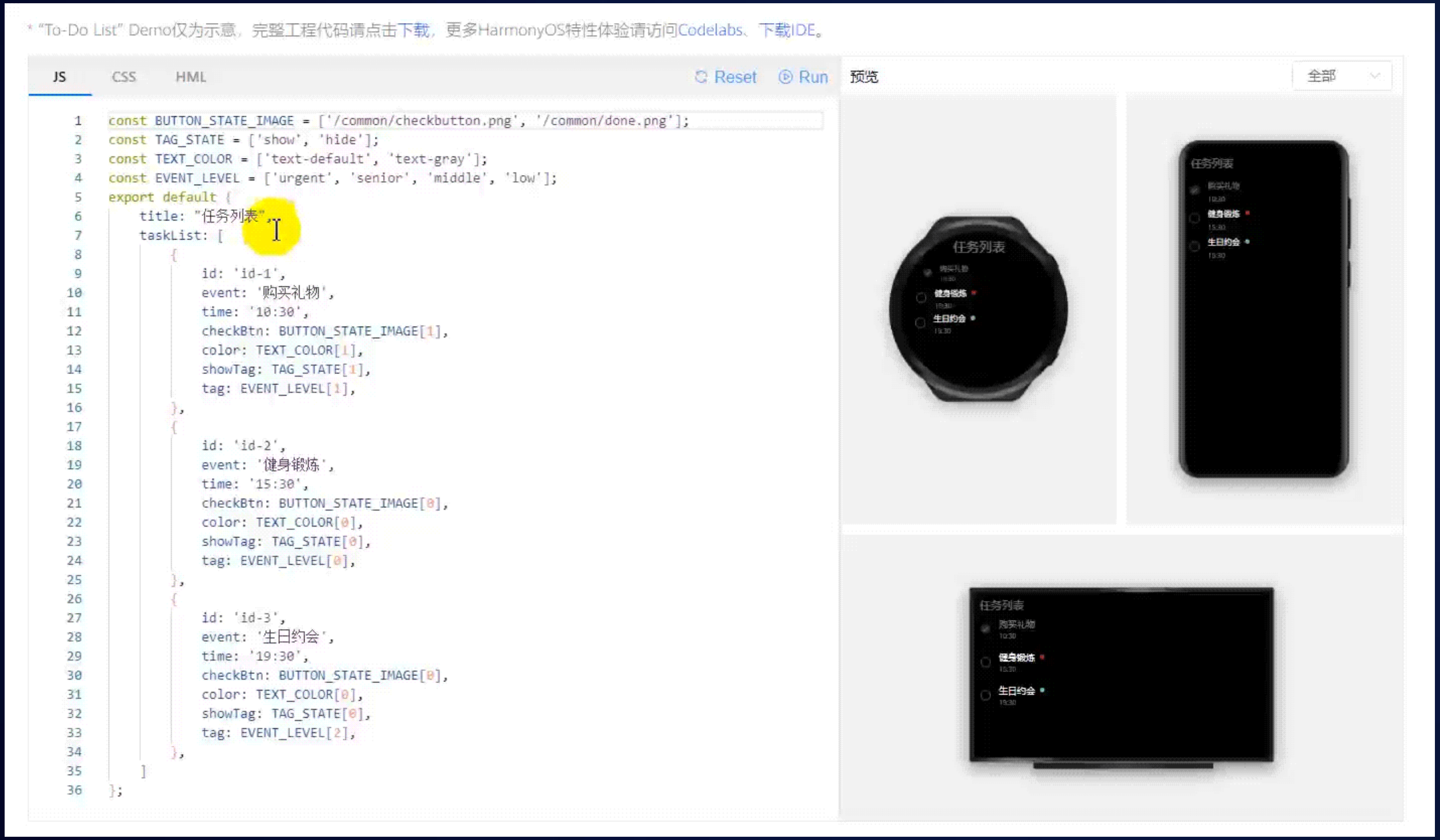
* “To-Do List” Demo仅为示意，完整工程代码请点击下载，更多HarmonyOS特性体验请访问Codelabs、下载IDE。

```
const BUTTON_STATE_IMAGE = ['/common/checkbutton.png', '/common/done.png'];
const TAG_STATE = ['show', 'hide'];
const TEXT_COLOR = ['text-default', 'text-gray'];
const EVENT_LEVEL = ['urgent', 'senior', 'middle', 'low'];
export default {
    title: "任务列表",
    taskList: [
        {
            id: 'id-1',
            event: '购买礼物',
            time: '10:30',
            checkBtn: BUTTON_STATE_IMAGE[1],
            color: TEXT_COLOR[1],
            showTag: TAG_STATE[1],
            tag: EVENT_LEVEL[1],
        },
        {
            id: 'id-2',
            event: '健身锻炼',
            time: '15:30',
            checkBtn: BUTTON_STATE_IMAGE[0],
            color: TEXT_COLOR[0],
            showTag: TAG_STATE[0],
            tag: EVENT_LEVEL[0],
        },
        {
            id: 'id-3',
            event: '生日约会',
            time: '19:30',
            checkBtn: BUTTON_STATE_IMAGE[0],
            color: TEXT_COLOR[0],
            showTag: TAG_STATE[0],
            tag: EVENT_LEVEL[2],
        },
    ]
};
```

HarmonyOS应用开发在线体验

HarmonyOS开发者微信公众号

扫码在线体验HarmonyOS应用开发



* “To-Do List” Demo仅为示意，完整工程代码请点击下载，更多HarmonyOS特性体验请访问Codelabs、下载IDE。

```
const BUTTON_STATE_IMAGE = ['/common/checkbutton.png', '/common/done.png'];
const TAG_STATE = ['show', 'hide'];
const TEXT_COLOR = ['text-default', 'text-gray'];
const EVENT_LEVEL = ['urgent', 'senior', 'middle', 'low'];
export default {
    title: "任务列表",
    taskList: [
        {
            id: 'id-1',
            event: '购买礼物',
            time: '10:30',
            checkBtn: BUTTON_STATE_IMAGE[1],
            color: TEXT_COLOR[1],
            showTag: TAG_STATE[1],
            tag: EVENT_LEVEL[1],
        },
        {
            id: 'id-2',
            event: '健身锻炼',
            time: '15:30',
            checkBtn: BUTTON_STATE_IMAGE[0],
            color: TEXT_COLOR[0],
            showTag: TAG_STATE[0],
            tag: EVENT_LEVEL[0],
        },
        {
            id: 'id-3',
            event: '生日约会',
            time: '19:30',
            checkBtn: BUTTON_STATE_IMAGE[0],
            color: TEXT_COLOR[0],
            showTag: TAG_STATE[0],
            tag: EVENT_LEVEL[2],
        },
    ]
};
```

HarmonyOS应用开发在线体验

HarmonyOS开发者微信公众号