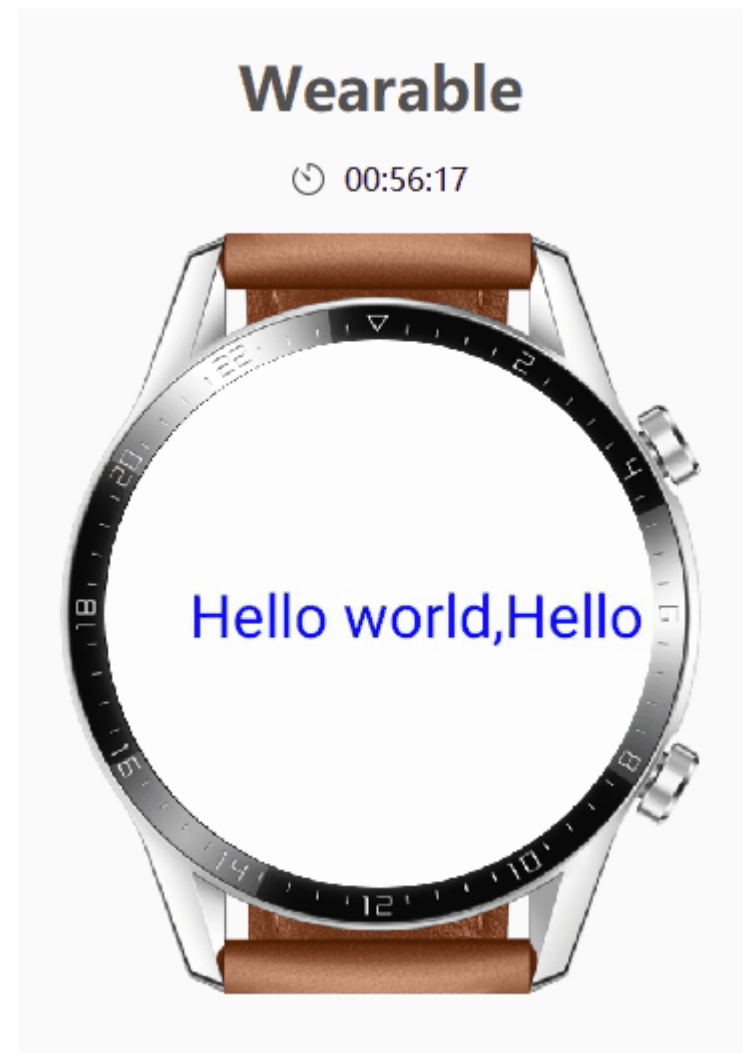


Harmony OS 入门系列课程 <快速上手>

快速掌握鸿蒙系统应用开发基础操作技巧

第 5 讲：创建自己的定位程序



章节目录

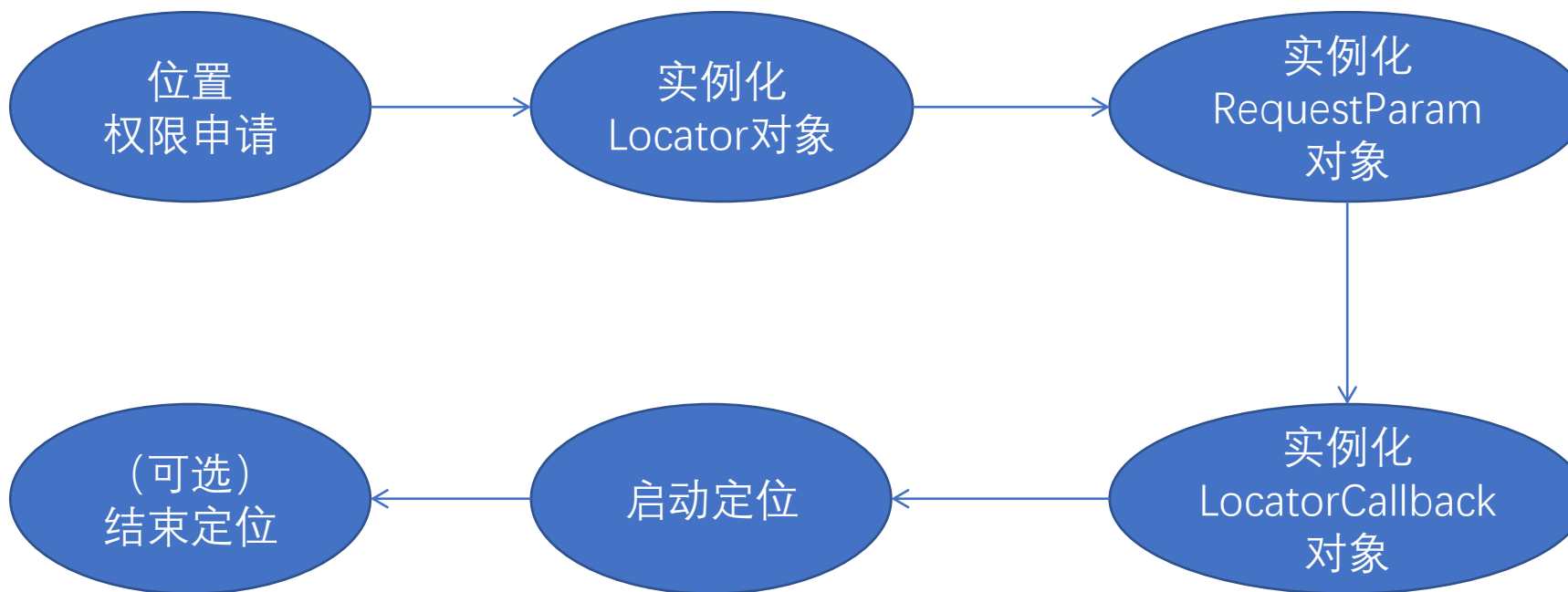
- 场景介绍
- 权限申请
- 接口说明
- 代码详解
- 内容小结
- 练习巩固

课程目标

- 掌握常用组件和布局的开发;
- 掌握鸿蒙应用权限申请和配置;
- 掌握鸿蒙位置服务的相关接口;
- 掌握（逆）地理编码转化;

- 系统使用位置坐标标示设备的位置，并用多种定位技术提供服务，如GNSS定位、基站定位、WLAN/蓝牙定位（基站定位、WLAN/蓝牙定位后续统称“网络定位技术”）。
- 无论用户设备在室内或是户外，都可以准确地确定设备位置。
- **对于用户：**系统的位置能力可以提供实时准确的位置数据。
- **对于开发者：**设计基于位置体验的服务，提高用户体验，使应用的使用体验更贴近每个用户。
- 场景举例：
 - 驾车导航
 - 记录运动轨迹
 -

开发步骤



权限申请

- 应用在使用系统能力前，需要检查是否已经获取用户授权访问设备位置信息。如未获得授权，可以向用户申请需要的位置权限。
- 权限可以分为数据和能力。
 - 数据包含：
 - 个人数据（如照片、通讯录、日历、位置等）
 - 设备数据（如设备标识、相机、麦克风等）
 - 应用数据
 -
 - 能力包括：
 - 设备能力（如打电话、发短信、联网等）
 - 应用能力（如弹出悬浮框、创建快捷方式等）
 -

敏感权限

敏感权限的申请需要按照动态申请流程向用户申请授权。
部分敏感权限如下：

权限分类名称	权限名	说明
位置	ohos.permission.LOCATION	允许应用在 前台 运行时获取位置信息。如果应用在后台运行时也要获取位置信息，则需要同时申请ohos.permission.LOCATION_IN_BACKGROUND权限。
	ohos.permission.LOCATION_IN_BACKGROUND	允许应用在 后台 运行时获取位置信息，需要同时申请ohos.permission.LOCATION权限。
相机	ohos.permission.CAMERA	允许应用使用相机拍摄照片和录制视频。
麦克风	ohos.permission.MICROPHONE	允许应用使用麦克风进行录音。
日历	ohos.permission.READ_CALENDAR	允许应用读取日历信息。
	ohos.permission.WRITE_CALENDAR	允许应用在设备上添加、移除或修改日历活动。
...

非敏感权限

非敏感权限不涉及用户的敏感数据或危险操作，仅需在config.json中声明，应用安装后即被授权。
部分非敏感权限如下：

权限名	说明
ohos.permission.GET_NETWORK_INFO	允许应用获取数据网络信息。
ohos.permission.GET_WIFI_INFO	允许获取WLAN信息。
ohos.permission.SET_NETWORK_INFO	允许应用控制数据网络。
ohos.permission.USE_BLUETOOTH	允许应用查看蓝牙的配置。
ohos.permission.SYSTEM_FLOAT_WINDOW	提供显示悬浮窗的能力。
ohos.permission.MULTIMODAL_INTERACTIVE	允许应用订阅语音或手势事件。
.....

权限申请

- 开发者需要在config.json文件中的“reqPermissions”字段中声明所需要的权限。
- 权限申请格式采用数组格式，可支持同时申请多个权限，权限个数最多不能超过1024个。
- 使用设备的位置能力，需要用户进行**确认**并**主动开启**位置开关。
- 系统提供的定位权限有：
 - ✓ **ohos.permission.LOCATION**
 - ✓ **ohos.permission.LOCATION_IN_BACKGROUND**
- 对于位置敏感的应用业务，建议获取设备实时位置信息。如果不需要设备实时位置信息，并且希望尽可能的节省耗电，开发者可以考虑获取最近的历史位置。

位置权限申请代码示例

开发者在应用config.json文件中申请位置权限，示例代码如下：

```
{
  "reqPermissions": [{
    "name": "ohos.permission.LOCATION",
    "reason": "$string:reason_description",
    "usedScene": {
      "ability": ["com.myapplication.LocationAbility"],
      "when": "inuse"
    }, {
      ...
    }
  ]
}
```

权限申请

reqPermissions权限申请字段说明:

键	值说明	类型	取值范围	默认值	规则约束
name	必须，填写需要使用的权限名称。	字符串	自定义	无	未填写时，解析失败。
reason	可选，当申请的权限为user_grant权限时此字段必填。 描述申请权限的原因。	字符串	显示文字长度不能超过256个字节	空	user_grant权限必填，否则不允许在应用市场上架。 需做多语种适配。
usedScene	可选，当申请的权限为user_grant权限时此字段必填。 描述权限使用的场景和时机。 场景类型有：ability、when（调用时机）。可配置多个ability。	ability：字符串数组 when：字符串	ability：ability的名称 when：inuse（使用时）、always（始终）	ability：空 when：inuse	user_grant权限必填ability，可选填when。

接口说明

开发者可以调用HarmonyOS位置相关接口，获取设备实时位置，或者最近的历史位置。
获取设备的位置信息，相关接口说明如下：

接口名	功能描述
Locator(Context context)	创建Locator实例对象。
RequestParam(int scenario)	根据定位场景类型创建定位请求的RequestParam对象。
onLocationReport(Location location)	获取定位结果。
startLocating(RequestParam request, LocatorCallback callback)	向系统发起定位请求。
requestOnce(RequestParam request, LocatorCallback callback)	向系统发起单次定位请求。
stopLocating(LocatorCallback callback)	结束定位。
getCachedLocation()	获取系统缓存的位置信息。

Locator对象

实例化Locator对象，所有与基础定位能力相关的功能API，都通过Locator提供。

*入参context需要提供当前应用程序的AbilityInfo信息，便于系统管理应用的定位请求。

```
Locator locator = new Locator(context);
```

RequestParam对象

实例化RequestParam对象，设置提供何种类型的位置服务，以及位置结果上报的频率。

以导航场景为例，实例化方式如下：

```
RequestParam requestParam = new RequestParam(RequestParam.SCENE_NAVIGATION);
```

定位场景

定位场景类型说明：

场景名称	常量定义	说明
导航场景	SCENE_NAVIGATION	适用于在户外定位设备实时位置的场景，如车载、步行导航； 主要使用GNSS定位技术提供定位服务； 使用系统网络定位技术，提升用户体验； 默认以最小1秒间隔上报定位结果。
轨迹跟踪场景	SCENE_TRAJECTORY_TRACKING	适用于记录用户位置轨迹的场景，如运动类应用记录轨迹功能； 主要使用GNSS定位技术提供定位服务； 默认以最小1秒间隔上报定位结果。
出行约车场景	SCENE_CAR_HAILING	适用于用户出行打车时定位当前位置的场景，如网约车类应用。 默认以最小1秒间隔上报定位结果。
生活服务场景	SCENE_DAILY_LIFE_SERVICE	适用于不需要定位用户精确位置的使用场景，如新闻资讯、网购、点餐类应用。 默认以最小1秒间隔上报定位结果。
无功耗场景	SCENE_NO_POWER	无功耗场景，适用于不需要主动启动定位业务。系统在响应其他应用启动定位业务并上报位置结果时，会同时向请求此场景的应用程序上报定位结果，当前的应用程序不产生定位功耗。 默认以最小1秒间隔上报定位结果。

RequestParam对象

如果定义的现有场景类型不能满足所需的开发场景，系统提供了基本的定位优先级策略类型。

后两个入参用于限定系统向应用上报定位结果的频率：

- 位置上报的最小时间间隔
- 位置上报的最小距离间隔

```
RequestParam requestParam = new RequestParam(RequestParam.PRIORITY_ACCURACY,0,0);
```


定位优先级

定位优先级策略类型说明：

策略类型	常量定义	说明
定位精度优先策略	PRIORITY_ACCURACY	主要以GNSS定位技术为主，在开阔场景下可以提供米级的定位精度，具体性能指标依赖用户设备的定位硬件能力，但在室内等强遮蔽定位场景下，无法提供准确的位置服务。
快速定位优先策略	PRIORITY_FAST_FIRST_FIX	同时使用GNSS定位、基站定位和WLAN、蓝牙定位技术，以便室内和户外场景下，通过此策略都可以获得位置结果，当各种定位技术都有提供位置结果时，系统会选择其中精度较好的结果返回给应用。因为对各种定位技术同时使用，对设备的硬件资源消耗较大，功耗也较大。
低功耗定位优先策略	PRIORITY_LOW_POWER	主要使用基站定位和WLAN、蓝牙定位技术，也可以同时提供室内和户外场景下的位置服务； 依赖周边基站、可见WLAN、蓝牙设备的分布情况，定位结果的精度波动范围较大，如果对定位结果精度要求不高，或者使用场景多在有基站、可见WLAN、蓝牙设备高密度分布的情况下，推荐使用，可以有效节省设备功耗。

LocatorCallback对象

实例化LocatorCallback对象，用于向系统提供位置上报的途径。

```
MyLocatorCallback locatorCallback = new MyLocatorCallback();

public class MyLocatorCallback implements LocatorCallback {
    @Override
    public void onLocationReport(Location location) {
    }

    @Override
    public void onStatusChanged(int type) {
    }

    @Override
    public void onErrorReport(int type) {
    }
}
```

启动定位

通过如下方式启动定位：

```
locator.startLocating(requestParam, locatorCallback);
```

如果应用不需要持续获取位置结果，可以使用如下方式启动定位，系统会上报一次实时定位结果后，自动结束应用的定位请求。应用不需要执行结束定位。

```
locator.requestOnce(requestParam, locatorCallback);
```

结束定位

结束定位为可选操作。

```
locator.stopLocating(locatorCallback);
```

如果应用使用场景不需要实时的设备位置，可以获取系统缓存的最近一次历史定位结果。

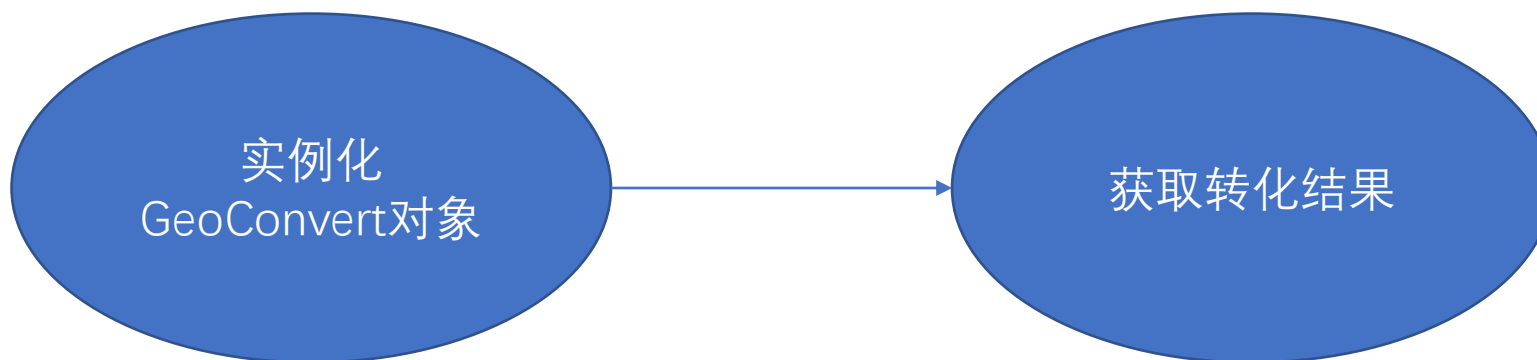
```
locator.getCachedLocation();
```

(逆) 地理编码转化

使用坐标描述一个位置，非常准确，但是并不直观，面向用户表达并不友好。

系统向开发者提供了地理编码转化能力（将坐标转化为地理编码信息），以及逆地理编码转化能力（将地理描述转化为具体坐标）。其中地理编码包含多个属性来描述位置，包括国家、行政区划、街道、门牌号、地址描述等等，这样的信息更便于用户理解。

开发流程：



接口说明

进行坐标和地理编码信息的相互转化，所使用的接口说明如下：

接口名	功能描述
GeoConvert()	创建GeoConvert实例对象。
GeoConvert(Locale locale)	根据自定义参数创建GeoConvert实例对象。
getAddressFromLocation(double latitude, double longitude, int maxItems)	根据指定的经纬度坐标获取地理位置信息。
getAddressFromLocationName(String description, int maxItems)	根据地理位置信息获取相匹配的包含坐标数据的地址列表。
getAddressFromLocationName(String description, double minLatitude, double minLongitude, double maxLatitude, double maxLongitude, int maxItems)	根据指定的位置信息和地理区域获取相匹配的包含坐标数据的地址列表。

代码详解

实例化GeoConvert对象

```
GeoConvert geoConvert = new GeoConvert();
```

坐标转化地理位置信息

```
geoConvert.getAddressFromLocation(纬度值, 经度值, 1);
```

位置描述转化坐标

```
geoConvert.getAddressFromLocationName("北京大兴国际机场", 1);
```

如果需要查询的位置描述可能出现多地重名的请求，可设置一个经纬度范围

```
geoConvert.getAddressFromLocationName("北京大兴国际机场", 纬度下限, 经度下限, 纬度上限, 经度上限, 1);
```

内容小结

- 鸿蒙应用权限申请
- 鸿蒙位置服务开发流程
- （逆）地理编码转化流程

练习巩固

问题1：以下哪项不属于敏感权限？（ ）

- A. ohos.permission.LOCATION
- B. ohos.permission.NFC_TAG
- C. ohos.permission.CAMERA
- D. ohos.permission.READ_MEDIA

问题2：如果应用不需要持续获取位置结果，可以使用“`locator.requestOnce(requestParam, locatorCallback)`”启动定位，系统会上报一次实时定位结果后，自动结束应用的定位请求。应用不需要执行结束定位。

- A. 正确
- B. 错误

思考挑战

编写程序获取当前设备位置，并计算出当前位置到北京天安门距离。

THANKS

更多学习视频，关注宅客学院.....

