

# KBSW180106 SLAMWARE SDK API 参考(Android)

本页提供了基于Android操作系统开发SLAMWARE SDK的API参考。

---

## 本页内容

- SDK内容
- 创建项目工程
  - 新建工程
  - Hello World
- 概览
  - 包
  - 类列表
- com.slamtec.slamware包
  - AbstractSlamwarePlatform 类
- com.slamtec.slamware.action包
  - IAction接口
  - IMoveAction IAction接口
  - ISweepMoveAction类
  - ISweepMoveAction IAction接口
  - path类
  - ActionStatus枚举
  - MoveDirection 枚举
- com.slamtec.slamware.discovery包
  - AbstractDiscover.BleConfigureListener 接口
  - AbstractDiscover 类
  - AbstractDiscover.DiscoveryListener 类
  - BleDevice 类
  - Device类
  - DeviceManager 类
  - MdnsDevice 类
  - AbstractDiscover.DiscoverStatus 枚举
  - DiscoveryMode 枚举
- com.slamtec.slamware.FirmwareUpdate包
  - FirmwareUpdateInfo类
  - FirmwareUpdateProgress类
- com.slamtec.slamware.geometry包
  - Line类
  - PointF 类
  - Size 类
- com.slamtec.slamware.robot包
  - HealthInfo类
  - HealthInfo.BaseError类
  - LaserPoint 类
  - LaserScan 类
  - Location 类
  - Map 类
  - NetworkMode类
  - Pose 类
  - Rotation 类
  - SystemParameters 类
  - SleepMode枚举
  - DockingStatus枚举
  - PowerStatus类
  - GridMap类
  - CompositeMap类
  - MapKind枚举
  - MapType枚举
  - MapLayer类
  - PoseMap类
  - PointMap类
  - RecoverLocalizationMovement枚举
  - RecoverLocalizationOptions类
  - SensorType枚举
  - SlamcoreShutdownParam类
  - SleepMode 枚举

- RestartMode枚举
- LineMap类
- PointsMap类
- ImpactSensorInfo 类
- ImpactSensorType枚举
- ImpactSensorValue类
- com.slamtec.slamware.exceptions包
  - AlreadyConnectedException类
  - ConnectionFailException类
  - ConnectionLostException类
  - ConnectionTimeOutException类
  - InvalidArgumentException类
  - OperationFailException类
  - OutOfResourceException类
  - ParseInvalidException类
  - PathFindFailException类
  - RequestFailException类
  - RequestTimeOutException类
  - UnauthorizedRequestException类
  - UnsupportedCommandException类
- com.slamtec.slamware.message包
  - DepthCameraFrame类
- com.slamtec.slamware.sdp包
  - Action类
  - CompositeMapHelper

#### 开发环境要求

基于Android 操作系统对Slamware SDK进行应用开发，您的开发环境需满足如下条件：

- 您的计算机应当安装Android 4.3或以上版本。本文档将基于Android 4.3进行介绍。

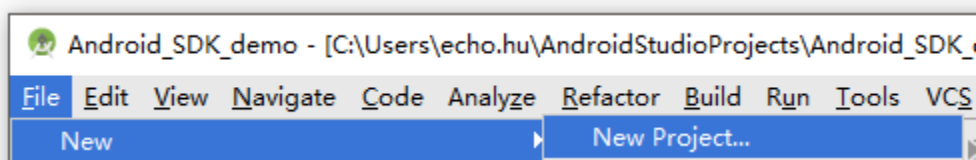
## SDK内容

- Slamware Android SDK主要包括slamtec\_agent.jar文件和armeabi文件夹。使用该SDK可以开发Android应用程序。

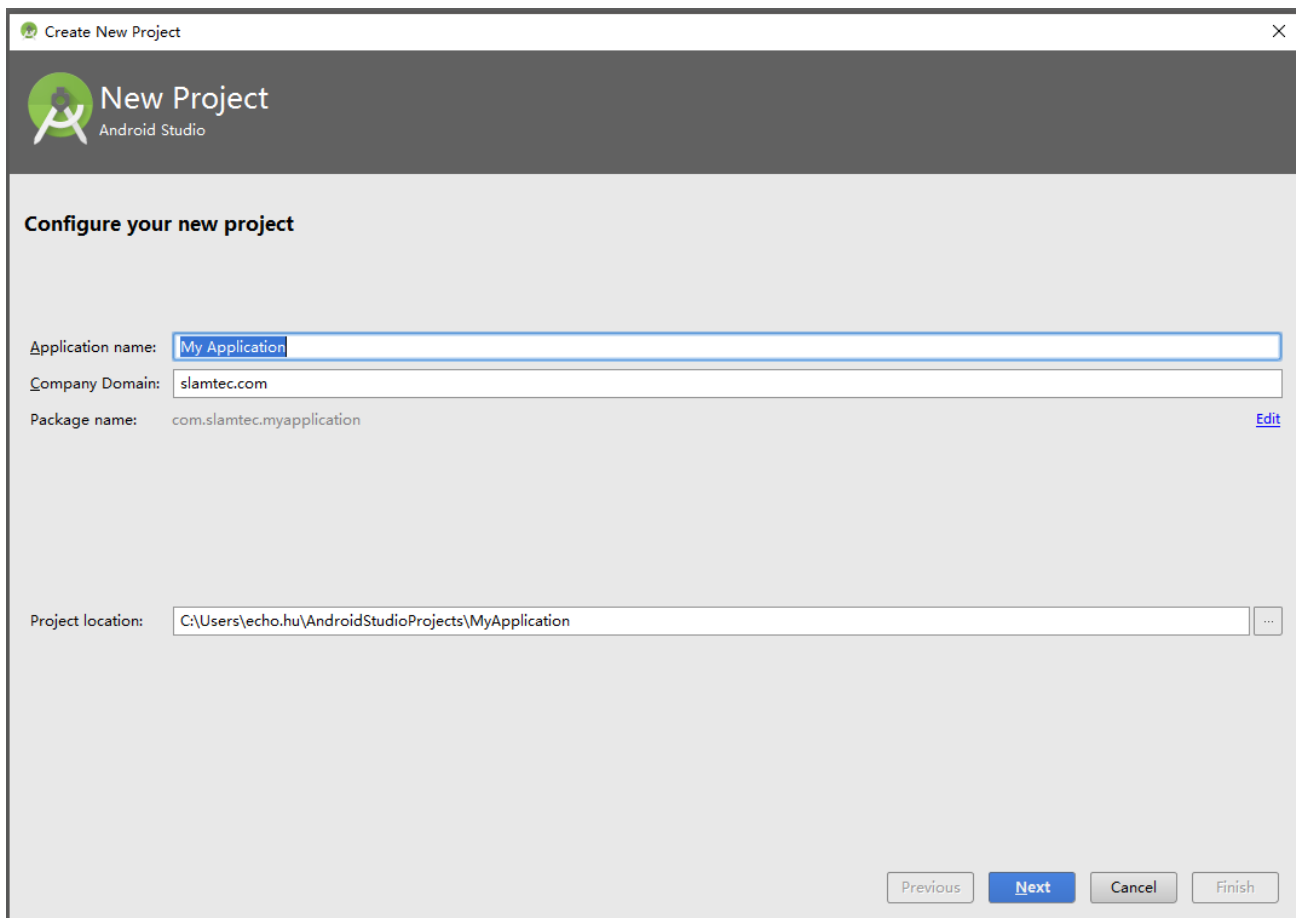
## 创建项目工程

### 新建工程

1. 打开Android Studio并点击File>New>New Project来新建一个项目（未曾建立过其他项目或者Android Studio 刚下载安装请点击Start a new Android Studio Project来新建项目）；



2. 为新建项目命名，选择域名，选择存放位置，然后点击下一步；



3. 选择所编译的应用程序将要运行的目标设备，然后点击Next；

## Target Android Devices

### Select the form factors your app will run on

Different platforms may require separate SDKs

Phone and Tablet

Minimum SDK

Lower API levels target more devices, but have fewer features available.

By targeting API 18 and later, your app will run on approximately **76.9%** of the devices that are active on the Google Play Store.

[Help me choose](#)

Wear

Minimum SDK

TV

Minimum SDK

Android Auto

Glass

Minimum SDK

⌘ Looking for SDKs available for download...

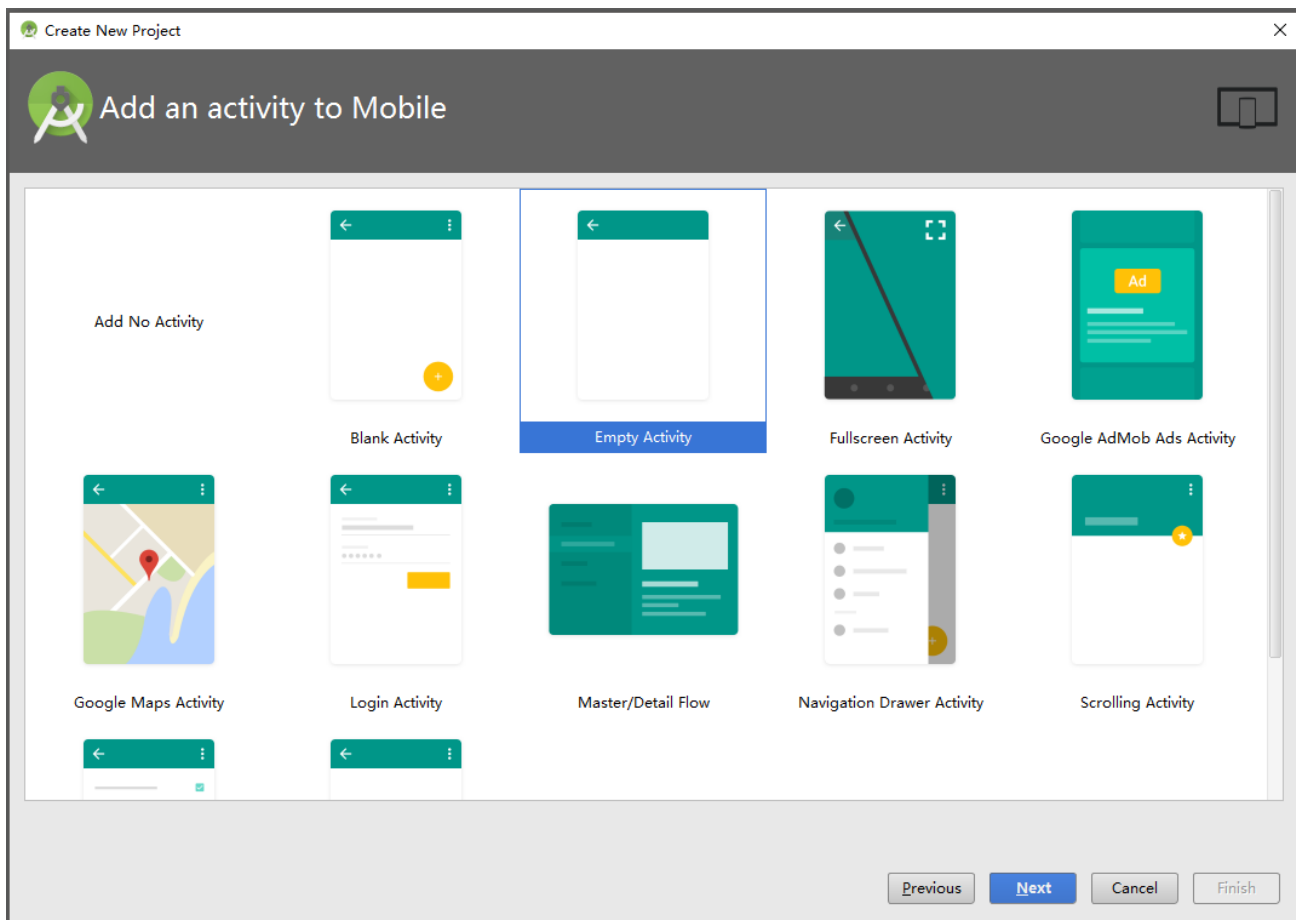
Previous

Next

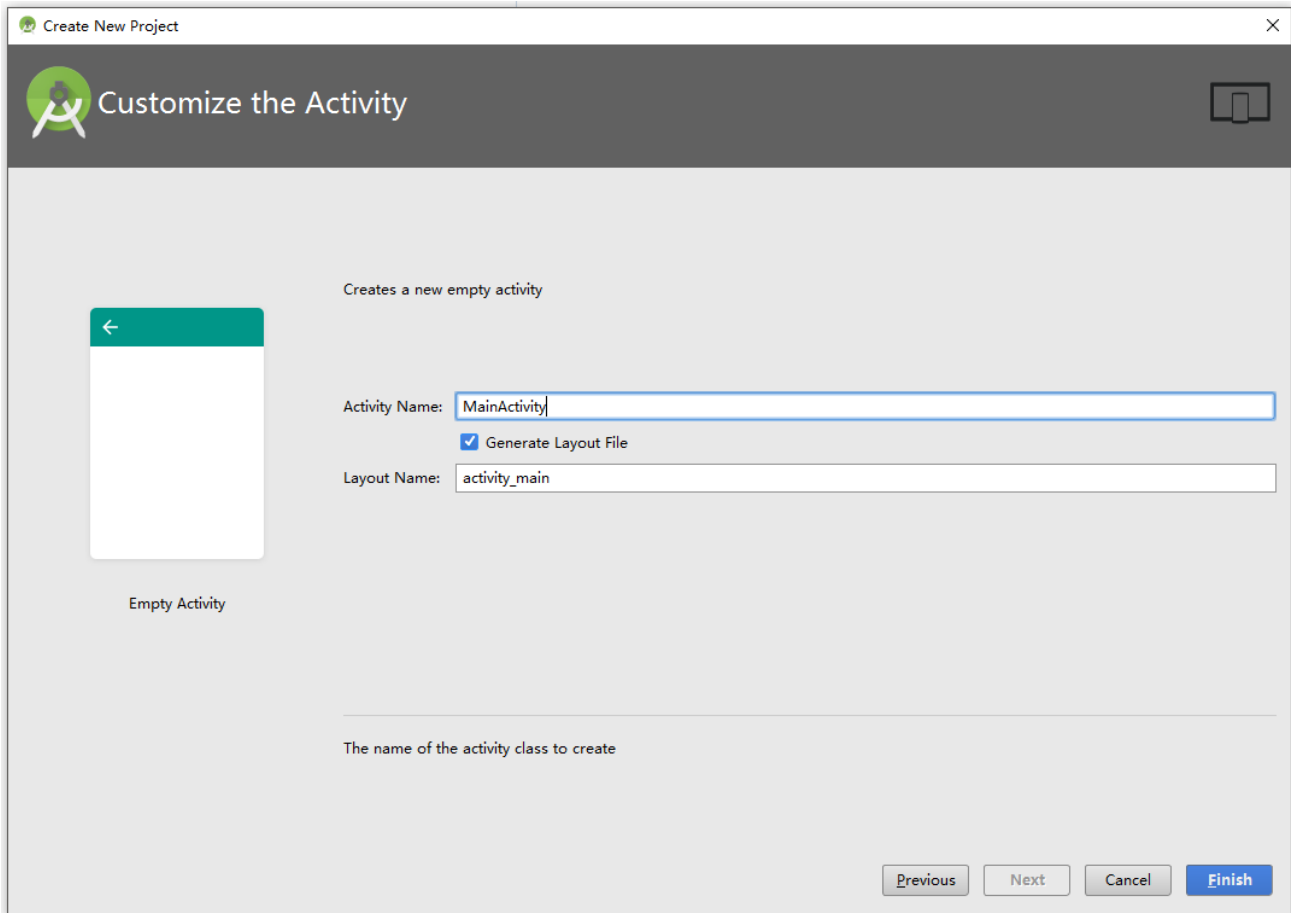
Cancel

Finish

4. 为该设备的应用程序选择一个Activity，此处选择Empty Activity，然后点击Next;

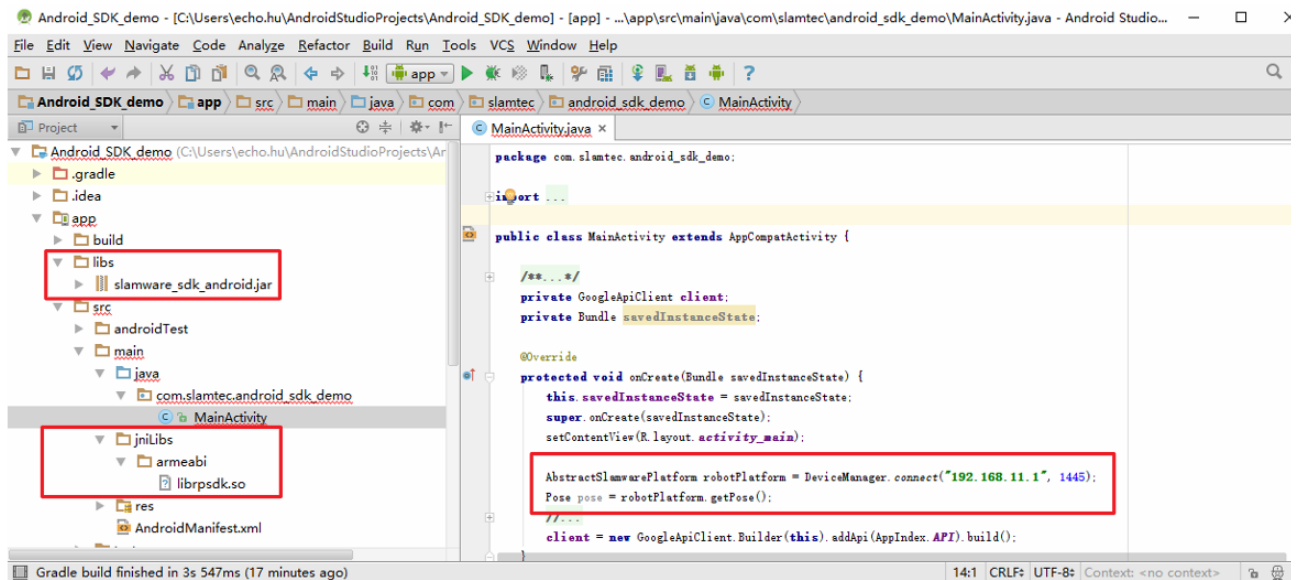


5. 进一步设置该Activity并点击完成。



## Hello World

- 在上述步骤完成后，将下载的SDK文件夹中的文件slamtec\_agent.jar和文件夹armeabi复制到下图所示文件夹中，并在右侧添加两行代码。



- 让测试手机连接上开发的SDP，运行该程序出现界面无崩溃，则代表成功。

### 注意

- 测试的手机需是ARM处理器的，x86的将无法运行
- 不支持模拟器运行

# 概览

## 包

包名	描述
<code>com.slamtec slamware</code>	包含有主要与slamware相关的类。
<code>com.slamtec slamware.action</code>	包含有与运动方式相关的类。
<code>com.slamtec slamware.discovery</code>	包含有与发现设备相关的类。
<code>com.slamtec slamware.FirmwareUpdate</code>	包含有与固件更新相关的类。
<code>com.slamtec slamware.geometry</code>	包含有与geometry相关的类。
<code>com.slamtec slamware.robot</code>	包含有与robot相关的类。
<code>com.slamtec slamware.exceptions</code>	包含有与异常有关的类。
<code>com.slamtec slamware.message</code>	包含有与message有关的类。
<code>com.slamtec slamware.sdp</code>	包含有与SDP有关的类。

## 类列表

类名	描述
<code>AbstractDiscover</code>	类, 表示abstract discover interface。
<code>AbstractDiscover.BleConfigureListener</code>	类, 表示BleConfigureListener。
<code>AbstractDiscover.DiscoverStatus</code>	类, 表示DiscoverStatus。
<code>AbstractDiscover.DiscoveryListener</code>	类, 表示DiscoveryListener。
<code>AbstractSlamwarePlatform</code>	类, 定义统一的接口用于和SLAMWARE设备通讯。
<code>ActionStatus</code>	枚举, 列举动作状态。
<code>BleDevice</code>	类, 表示BleDevice。
<code>CompositeMapHelper</code>	
<code>Device</code>	类, 表示设备。
<code>DeviceManager</code>	类, 表示管理设备的管理程序。
<code>DepthCameraFrame</code>	类, 表示深度摄像头。

DiscoveryMode	枚举，列举发现机器人的模式。
FirmwareUpdateInfo	类，表示固件更新信息。
FirmwareUpdateProgress	类，表示固件更新进程。
HealthInfo	类，表示健康状况。
HealthInfo.BaseError	类，表示底盘健康状况和错误信息。
IAction	接口，表示机器人动作。
IMoveAction	接口，表示移动动作。
ImpactSensorInfo	类，碰撞传感器信息
ImpactSensorType	枚举，表示碰撞传感器类型。
ImpactSensorValue	类，表示碰撞传感器数据值。
ISweepMoveAction	接口，表示清扫动作。
LaserPoint	类，表示激光扫描点。
LaserScan	类，表示一次激光扫描。
Line	类，表示线。
Location	类，表示机器人在3D空间中的位置。
Map(Robot)	类，表示机器人所在的位置地图。
Map(SDP)	类，表示SDP所在的位置地图。
MapKind	枚举，列举地图种类。
MapLayer	类，表示地图层。
MapType	枚举，列举地图数据类型。
MdnsDevice	类，MdnsDevice。
MoveDirection	类，列举人工控制机器人时的方向指令。
NetworkMode	类，表示网络模式。
Path	类，表示路径。
PointF	类，表示2D浮点小数数据类型。
PointMap	类，表示点地图。



Pose	类，表示机器人姿态。
PoseMap	类，表示PoseMap。
RecoverLocalizationMovement	枚举，列举当前重新定位时发生的位置变化。
RecoverLocalizationOptions	类，表示重新定位需要的参数。
RestartMode	枚举，列举重启模式。
Rotation	类，表示旋转。
SensorType	枚举，列举机器人传感器类型。
Size	类，表示整数型的Size类别。
SlamwareCorePlatform	类，Abstract Slamware Platform类的子类。
SlamwareSdpPlatform	类，表示SDP扩展平台。
SlamcoreShutdownParam	类，表示底盘重启参数。
SleepMode	枚举，列举当前底盘的休眠模式。
SystemParameters	类，表示系统参数。

## com.slamtec.slamware包

### AbstractSlamwarePlatform 类

Abstract Slamware Platform 类，定义统一的接口用于和SLAMWARE设备通讯。

已知子类：SlamwareCorePlatform

#### 构造器

`AbstractSlamwarePlatform()`

AbstractSlamwarePlatform是一个抽象类，不能直接构造该类的对象。请使用DeviceManager连接设备并获取AbstractSlamwarePlatform的对象。

#### 方法

`addWall(Line wall)`

添加虚拟墙至SLAMWARE。

#### 参数

参数名称	描述
wall	需要添加的虚拟墙

`addWalls(java.util.List<Line> walls)`

添加多个虚拟墙至SLAMWARE。

### 参数

参数名称	描述
walls	需要添加的多个虚拟墙。

`clearMap()`

清除当前地图。

`clearWallById(int id)`

移除指定的虚拟墙。(id 不是设定的id, 是通过AbstractSlamwarePlatform.getLines(ArtifactUsage var1)返回的line id。)

### 参数

参数名称	描述
id	需要移除的虚拟墙id

`clearWalls()`

从Slamware清除所有虚拟墙。

`configureNetwork(int mode, java.util.HashMap<java.lang.String, java.lang.String> options)`

配置Slamware Core的网络工作模式。数据返回类型布尔值boolean。

### 参数

参数名称	描述
mode	网络模式
options	选项

### 说明

当前支持的工作模式主要有：

模式名称	描述
AP模式	Slamware Core本身作为一个WiFi热点, 当用户设备通过Wifi或者有线网络连接该WiFi热点时, 会通过DHCP获得一个IP地址, 而后通过192.168.1 Core出厂的预置模式。
Station模式	Slamware Core本身最为一个WiFi设备, 连接到其他的WiFi热点上。同时Slamware Core会自动成为无线网桥, 为High Speed Bus上的设备提供网络访问。
Disable模式	Slamware Core关闭自身的无线访问功能, 只能通过有限网络访问, 其IP地址、网关和DNS服务器均由API调用的参数决定。

### 代码示例

- 将Slamware Core配置成AP模式

```
Platform.configureNetwork(NetworkMode.NetworkModeAp, new HashMap<String, String>());
```

- 让Slamware Core连上名字为Slamtec的AP

```
HashMap<String, String> options = newHashMap<String, String>();
options.put("ssid", "Slamtec"); options.put("password", "Password");
platform.configureNetwork(NetworkMode.NetworkModeStation, options);
```

- 将Slamware Core的IP地址配置为192.168.12.13，默认网关为192.168.12.1，DNS服务器为：114.114.114.114：

```
HashMap<String, String> options = newHashMap<String, String>();
options.put("ip", "192.168.12.13"); options.put("mask":
"255.255.255.0"); options.put("gateway": "192.168.12.1");
options.put("dns": "114.114.114.114");
platform.configureNetwork(NetworkMode.NetworkModeDisabled, options);
```

disconnect()

断开与平台之间的连接。

getAvailableMaps()

获取SLAMWARE中可用的地图类型。返回值为地图类型列表。

getBatteryIsCharging()

获取电池是否在充电。返回值为布尔值，表明电池是否在充电。

getBatteryPercentage()

获取电池剩余电量百分比（从0~100）。返回值为电池电量百分比。

getCurrentAction()

获取机器人当前的动作。返回值为机器人当前正在进行的移动动作。

getDCIsConnected()

获取机器人是否连接到电源插座。返回值为布尔值，表明机器人是否连接到充电器。

getDeviceId()

获取设备的UUID。返回值数据类型为string。

getFirmwareUpdateInfo()

请参考FirmwareUpdateInfo类。

getFirmwareUpdateProgress()

请参考FirmwareUpdateProgress类。

getHardwareVersion()

获取设备硬件版本。返回值数据类型为string。

getKnownArea(MapType type)

获取已探索到的地图区域。返回值为已探索的地图区域。

参数：type – 地图数据类型

getKnownArea(MapType type, MapKind kind)

获取已探索到的地图区域。返回值为已探索的地图区域。

参数

参数名称	描述
------	----

type	地图数据类型
kind	地图种类

`getLaserScan()`

获取最新的LASER扫描。返回值最新的LASER扫描。

`getLocalizationQuality()`

获取定位可信度。

`getLocation()`

获取机器人在上述地图坐标系统中的坐标。返回值为机器人位置信息。

`getManufacturerId()`

获取设备制造商id。返回值数据类型为int。

`getManufacturerName()`

获取设备制造商名称。返回值数据类型为string。

`getMap(MapType type, MapKind kind, android.graphics.RectF area)`

从SLAMWARE获取地图数据。返回值为局部地图对象。

## 参数

参数名称	描述
type	地图数据类型
kind	地图种类
area	地图区域

`getMapLocalization()`

获取SLAMWARE是否在进行定位。返回值为布尔值表明SLAMWARE是否在进行定位。

`getMapUpdate()`

获取SLAMWARE是否更新地图。返回值为布尔值表明SLAMWARE是否更新地图。

`getModelId()`

获取设备型号id。返回数据类型为int。

`getModelName()`

获取设备型号名称。返回数据类型为string。

`getNetworkStatus()`

获取网络状况。返回数据类型为string。

`getPose()`

获取机器人在上述坐标系统中的姿态（包含位置信息和角度信息）。返回值为机器人的姿态。

`getRobotHealth()`

获取机器人健康状态。返回数据为健康状况。

`getSDKVersion()`

获取SLAMWARE SDK版本。返回值为标明SLAMWARESDK版本的字符串。

`getSlamwareVersion()`

获取SLAMWARE版本。返回值为标明SLAMWARE版本的字符串。

`getSoftwareVersion()`

获取设备的软件版本。返回值数据类型为字符串。

```
getSystemParameter(java.lang.String param)
```

获取系统参数。返回值为当前参数的值。

#### 参数

参数名称	描述
param	获取的参数

```
getWalls()
```

获取当前存在的虚拟墙。返回值为当前存在的虚拟墙列表。

```
goHome()
```

使机器人返回充电桩（注意：该方法仅适用于支持自动回充功能的机器人版本）。

```
moveBy(MoveDirection direction)
```

人工控制机器人的移动。返回值为执行该项操作的移动动作。

（注意：在此状态下，机器人不会进行避障）。需要反复调用该函数来保持机器人的移动状态并调用MoveAction.cancel()函数来及时使机器人停止移动，否则机器人会在持续调用moveBy函数一段时期后停止移动。

#### 参数

参数名称	描述
direction	期望机器人移动的方向

```
moveTo(java.util.List<Location> locations)
```

使机器人移动到一系列指定位置。返回值为执行该项操作的移动动作。

#### 参数

参数名称	描述
locations	机器人前往指定位置经过的一系列节点。

```
moveTo(java.util.List<Location> locations, boolean appending)
```

使机器人移动到一系列的指定位置。返回值为执行该项操作的移动动作。

#### 参数

参数名称	描述
locations	机器人前往指定位置经过的一系列节点。
appending	布尔值，用于决定SLAMWARE是清除当前任务建立新的点还是将新的点添加到已有的节点列表中。

```
moveTo(java.util.List<Location> locations, boolean appending, boolean isMilestone)
```

使机器人移动到一系列的指定位置。返回值为执行该项操作的移动动作。

#### 参数

参数名称	描述
locations	机器人前往指定位置经过的一系列节点。

appending	布尔值，用于决定SLAMWARE是清除当前任务建立新的点还是将新的点添加到已有的节点列表中。
isMilestone	布尔值，用于决定SLAMWARE是规划路径到一系列节点还是直接前往。当这个参数为true时，机器人会将上述点视作关键点，通过路径

`moveTo(Location location)`

使机器人移动到指定地点。返回值为执行该项操作的移动动作。

### 参数

参数名称	描述
location	机器人将要到达的点。

`moveTo(Location location, boolean appending)`

使机器人移动到指定地点。返回值为执行该项操作的移动动作。

### 参数

参数名称	描述
location	机器人将要到达的点。
appending	布尔值，用于决定SLAMWARE是清除当前任务建立新的点还是将新的点添加到已有的节点列表中。

`moveTo(Location location, boolean appending, boolean isMilestone)`

使机器人移动到指定地点。返回值为执行该项操作的移动动作。

### 参数

参数名称	描述
location	机器人将要到达的点。
appending	布尔值，用于决定SLAMWARE是清除当前任务建立新的点还是将新的点添加到已有的节点列表中。
isMilestone	布尔值，用于决定SLAMWARE是规划路径到该节点还是直接前往。当这个参数为true时，机器人会将上述点视作关键点，通过路径搜

`restartModule()`

重启SLAMWARE模块。

`restartModule(RestartMode mode)`

重启SLAMWARE模块。

### 参数

参数名称	描述
mode	重启SLAMWARE模块的模式

`rotate(Rotation rotation)`

使机器人旋转一个角度（不同的角度）。返回值为执行该项操作的移动动作。

### 参数

参数名称	描述
rotation	机器人被要求旋转的弧度。

`rotateTo(Rotation orientation)`

使机器人旋转到特定的朝向。返回值为执行该项操作的移动动作。

#### 参数

参数名称	描述
orientation	要求的姿态

`searchPath(Location location)`

在地图中寻找使机器人从当前位置移动到指定目标位置的路径。返回值为一条从机器人的当前位置到指定目标位置的路径。

#### 参数

参数名称	描述
location	目标位置

`setMap(Map map)`

上传地图数据到SLAMWARE。（注意：应与setPose配套使用并确保地图未处于更新或定位状态。在设置地图时候，不可以调用操作机器人的方法比如moveto。）

#### 参数

参数名称	描述
map	地图

`setMap(Map map, MapType type)`

上传地图数据到SLAMWARE。（注意：应与setPose配套使用并确保地图未处于更新或定位状态。）

#### 参数

参数名称	描述
map	地图
type	地图数据类型

`setMap(Map map, MapType type, MapKind kind)`

上传地图数据到SLAMWARE。（注意：应与setPose配套使用并确保地图未处于更新或定位状态。）

#### 参数

参数名称	描述
map	地图
type	地图数据类型
kind	地图种类

`setMapLocalization(boolean v)`

获取是否启用定位功能。

#### 参数

---

参数名称	描述
v	布尔值，表明SLAMWARE是否该进行定位。

`setMapUpdate(boolean v)`

获取是否更新地图。

### 参数

参数名称	描述
v	布尔值，表明SLAMWARE是否应该更新地图。

`setPose(Pose pose)`

设置机器人的姿态。

### 参数

参数名称	描述
pose	机器人的新姿态

`setSystemParameter(java.lang.String param, java.lang.String value)`

设置系统参数。

### 参数

参数名称	描述
param	将要进行设置的参数。
value	将要设置的值。

`startFirmwareUpdate()`

获取是否开始固件更新。返回值为boolean表示是否开始更新固件。

`startSweep()`

使机器人开始清扫（注意：该方法仅适用于扫地机版本的SLAMWARE Core）。返回值为执行该项操作的清扫移动动作。

`sweepSpot(Location location)`

使机器人开始定点清扫（注意：该方法仅适用于扫地机版本的SLAMWARE Core）。返回值为执行该项操作的清扫移动动作。

## ArtifactUsage枚举

表示虚拟墙或者虚拟轨道。

### 枚举常量

`ArtifactUsageVirtualWall`

虚拟墙，用于限制设备的活动范围。

`ArtifactUsageVirtualTrack`

虚拟轨道，用于为设备提供预设的路线，设备将尽可能按照虚拟轨道移动。

`List<Line> getLines(ArtifactUsage usage)`

获取所有虚拟墙或虚拟轨道的信息。返回值为一个line的列表。



## 参数

参数名称	描述
usage	虚拟墙或者虚拟轨道

```
boolean addLine(ArtifactUsage usage, Line line)
```

添加虚拟墙或虚拟轨道。

```
boolean addLines(ArtifactUsage usage, List<Line> lines)
```

添加多条虚拟墙或虚拟轨道。id必须是Arraylist<Line> lines = AbstractSlamwarePlatform.getLines(ArtifactUsage usage);

```
boolean removeLineById(ArtifactUsage usage, int id)
```

移除指定的虚拟墙或虚拟轨道。

```
boolean clearLines(ArtifactUsage usage)
```

移除所有虚拟墙或虚拟轨道。

```
PowerStatus getPowerStatus()
```

获取电源相关的信息。

```
void wakeUp()
```

从休眠状态唤醒设备。

```
IMoveAction recoverLocalization(RectF area)
```

当机器人被抱起时可调用此接口进行重定位。

如果area区域为空时，机器人将在当前20\*20的区域进行从定位，否则机器人将在指定区域进行重新定位。

```
PointPDF getAuxLocation()
```

根据当前辅助定位设备的信息获取当前机器人的大致位置信息

当机器人被抱起时，可以先调用getAuxLocation()获取机器人当前大致位置信息，然后调用recoverLocalization传入包含大致位置信息的区域，使机器人尝试重定位。

## 代码示例

```
auto locationPdf = platform.getAuxLocation();
Location location = locationPdf.location;
float distant = locationPdf.circular_error_probability;
area((location.x() - distant), (location.y() - distant), 2*distant, 2*distant);RectangleF
auto act = platform.recoverLocalization(area);
act.waitUntilDone();
```

```
CompositeMap getCompositeMap()
```

获取当前地图的所有信息。

```
void setCompositeMap(CompositeMap map, Pose pose)
```

设置当前的地图信息。

```
Pose getHomePose()
```

获取充电座位置。如果找不到充电座则返回(0,0)。

```
List<ImpactSensorInfo> getSensors()
```

获取到机器人所有的传感器信息，返回值是list集合。

```
HashMap<Integer, ImpactSensorValue> getSensorValues()
```

获取到机器人所有的传感器信息，返回值是hashMap集合。

ImpactSensorValue getSensorValue(Integer var1)

根据key 获取到机器人对应的传感器信息。

IMoveAction moveTo(Location location, MoveOption option, float yaw)

使机器人移动到指定的位置。location 指定的位置，moveOption 是移动时候的参数，可以为：

参数名称	数据类型	描述
appending	boolean	用于决定SLAMWARE是清除当前任务建立新的点还是将新的点添加到已有的节点列表中。
milestone	boolean	用于决定SLAMWARE是规划路径到一系列节点还是直接前往。当这个参数为true时，机器人会将上述点视为
noSmooth	boolean	暂时没有开放。
keyPoints	boolean	设置是否走虚拟轨道。
precise	boolean	机器人移动的时候精确到点。
withYaw	boolean	是否让机器人停下来的时候旋转。
yaw	float	机器人停下来的时候旋转到一定的角度。角度范围？
returnUnreachableDirectly	boolean	为true时，当机器人规划路径失败后，机器人不进行旋转重新规划。
trackWithOA	boolean	trackWithOA 为true时,机器人走虚拟轨道时候，也会进行避障，避障后继续优先走虚拟轨道。(如果不走虚拟
speedRatio	Double	机器人行走的速度,范围是0到1,(只用于moveBy)。

IMoveAction moveTo(java.util.List locations, MoveOption option, float yaw)

参考上面的注释，只是移动到很多点。

publishDepthCamFrame(int sensorId,DepthCameraFrame depthCameraFrame)

将深度摄像头的的数据传递给底盘，其中sensorId 是通过底盘配置的sensorId, depthCameraFrame是深度摄像头数据对象。

## com. slamtec slamware action包

### IAction接口

接口，表示机器人的动作。提供接口来执行该动作。

已知子接口：IMoveAction, ISweepMoveAction

### 方法

cancel()

取消本次操作。

getActionName()

获取动作名称。返回值为动作名称（string）。

getProgress()

获取动作进程（0~1）。返回值为动作进程（double）。

getStatus()

获取动作状态。返回值为动作状态。

waitUntilDone()

等待动作完成。返回值为完成的动作结果。

```
boolean isEmpty()
```

检查当前动作是否为空。

```
int getActionId()
```

获取当前动作id。

```
String getReason()
```

获取当前action结束后返回的字符串，成功后会返回不为null且长度为0的字符串，失败后会返回说明失败原因的字符串，如“failed”表示action失败，“aborted”表示action被打断(主动调用cancel或在action未结束时启动新的action)，对于moveTo接口返回的action，“unreachable”表示目标点不可达(搜索路径失败)。

## IMoveAction IAction接口

继承自 IAction

已知超接口：IAction

已知子接口：ISweepMoveAction

接口：表示移动动作。

### 方法

```
getRemainingMilestones()
```

获取剩余里程碑。返回值为剩余的里程碑。

```
getRemainingPath()
```

获取到下一里程碑的剩余路径。返回值为到下一里程碑的剩余路径。

继承自com.slamtec.slamware.action.IAction接口的方法如下：

cancel, getActionName, getProgress, getStatus, waitUntilDone

## ISweepMoveAction类

```
pause()
```

让当前机器人的动作暂停。

```
resume()
```

重新恢复当前机器人的动作。

## ISweepMoveAction IAction接口

继承自 IMoveAction

已知超接口：IAction, IMoveAction

接口，表示清扫时的移动动作。

### 方法

继承自com.slamtec.slamware.action.IMoveAction接口的方法如下：

```
getRemainingMilestones, getRemainingPath
```

继承自com.slamtec.slamware.action.IAction接口的方法如下：

cancel, getActionName, getProgress, getStatus, waitUntilDone

## path类

类，表示一条路径。

### 构造器

`Path()`

创建对象Path。

`Path(Path path)`

创建参数为path的对象Path。

`Path(java.util.Vector<Location> points)`

创建参数为points的对象Path。

### 方法

`getPoints()`

获取点。

`setPoints(java.util.Vector<Location> points)`

设置点。

## ActionStatus枚举

`public enum ActionStatus`, 继承自 `java.lang.Enum<ActionStatus>`, 表示动作的状态。

### 枚举常量

`WAITING_FOR_START`

动作已创建但未开始。

`RUNNING`

动作正在进行。

`FINISHED`

动作成功完成。

`PAUSED`

动作已暂停。

`STOPPED`

动作已停止。

`ERROR`

动作遇到错误。

## MoveDirection 枚举

`public enum MoveDirection`, 继承自 `java.lang.Enum<MoveDirection>`, 人工控制机器人时要求机器人移动的方向。

## 枚举常量

FORWARD

当前移动动作作为向前。

BACKWARD

当前移动动作作为向后。

TURN\_RIGHT

当前移动动作作为向右。

TURN\_LEFT

当前移动动作作为向左。

## com.slamtec.slamware.discovery包

### AbstractDiscover.BleConfigureListener 接口

封闭类：AbstractDiscover

#### 方法

onConfigureSuccess()

配置成功。

onConfigureFailure(java.lang.String error)

配置失败。

#### String error枚举常量

BLE\_CONFIG\_ERROR\_BLE\_DISCONNECTED

断开连接

BLE\_CONFIG\_ERROR\_WRITTING\_FAIL

配置失败

BLE\_CONFIG\_ERROR\_SSID\_REQUIRED

SSID无效

BLE\_CONFIG\_ERROR\_UNABLE\_CONNECT\_WIFI

无法连接至WiFi

BLE\_CONFIG\_ERROR\_INVALID\_PWD

密码错误

### AbstractDiscover 类

public abstract class AbstractDiscover, abstract discover 接口。

已知子类：DeviceManager

#### 嵌套类

AbstractDiscover.BleConfigureListener

AbstractDiscover.DiscoverStatus

AbstractDiscover.DiscoveryListener

## 构造器

AbstractDiscover()

创建对象AbstractDiscover。

## 方法

getMode()

获取模式。

setListener(AbstractDiscover.DiscoveryListener listener)

设置listener。

getStatus(DiscoveryMode mode)

获取状态。

start(DiscoveryMode mode)

开始。

stop(DiscoveryMode mode)

结束。

## AbstractDiscover.DiscoveryListener 类

封闭类：AbstractDiscover

## 构造器

DiscoveryListener()

创建对象DiscoveryListener。

## 方法

onStartDiscovery(AbstractDiscover discover)

Discovery扫描开始的回调函数。

onStopDiscovery(AbstractDiscover discover)

Discovery扫描停止的回调函数。

onDiscoveryError(AbstractDiscover discover, java.lang.String error)

Discovery错误的回调函数。

onDeviceFound(AbstractDiscover discover, Device device)

发现设备的回调函数。

## BleDevice 类

继承自 Device

## 构造器

BleDevice(BluetoothDevice device)

创建对象BleDevice。

## 方法

getDevice()

获取设备device。

canBeFoundWith(DiscoveryMode mode)

在discovery模式开启的状态下能否发现设备。

由Device 类中的canBeFoundWith指定

## Device类

public abstract class Device, 表示一台设备。

已知子类：

BleDevice, MdnsDevice

## 构造器

Device()

创建对象Device。

## 方法

getManufactureId()

获取Manufacture id。

setManufactureId(int manufactureId)

设置manufacture id。

getModelId()

获取mode id。

setModelId(int modelId)

设置mode id。

getManufactureName()

获取manufacture name。

setManufactureName(java.lang.String manufactureName)

设置manufacture name。

getModelName()

获取mode name。

setModelName(java.lang.String modelName)

设置mode name。

getHardwareVersion()

获取 hard ware version。

setHardwareVersion(int hardwareVersion)

设置hardware version。

getSoftwareVersion()

获取 software version。

```
setSoftwareVersion(int softwareVersion)
```

设置software version。

```
getSerialNumber()
```

获取serial number。

```
setSerialNumber(java.lang.String serialNumber)
```

设置serial number。

```
canBeFoundWith(DiscoveryMode mode)
```

在discovery模式开启的状态下能否发现设备。

```
getDeviceId()
```

获取device id。

```
setDeviceId(java.util.UUID deviceId)
```

设置device id。

```
getDeviceName()
```

获取device name。

```
setDeviceName(java.lang.String deviceName)
```

设置device name。

## DeviceManager 类

继承自 AbstractDiscover

The manager to manage devices。

Nested classes/interfaces inherited from class

com.slamtec.slamware.discovery.AbstractDiscover:

AbstractDiscover.BleConfigureListener,

AbstractDiscover.DiscoverStatus,

AbstractDiscover.DiscoveryListener

### 构造器

```
DeviceManager(Context context)
```

创建对象DeivceManager。

### 方法

```
connect(java.lang.String host, int port)
```

直接连接到SLAMWARE Core（该方法常用于Android设备通过高速总线直接连接到SLAMWARE Core）。返回值为连接到的平台。

### 参数

参数名称	描述
host	设备主机（通常是192.168.11.1）
port	端口

```
connect(Device device)
```



连接到指定的基于SLAMWARE的设备。返回值为连接到的设备。

### 参数

参数名称	描述
device	准备连接的设备

```
pair(Device device, java.lang.String wifiSSID, java.lang.String wifiPassword, AbstractDiscover.BleConfigureListener listener)
```

通过SSID和密码匹配SLAMWARE 设备。

### 参数

参数名称	描述
device	准备配对的设备
wifiSSID	WiFi的SSID
wifiPassword	WiFi的密码
listener	配置监听器

```
setListener(AbstractDiscover.DiscoveryListener listener)
```

由AbstractDiscover 类中的setListener指定

```
getStatus(DiscoveryMode mode)
```

由AbstractDiscover类中的getStatus指定。

```
start(DiscoveryMode mode)
```

由AbstractDiscover类中的start指定

```
stop(DiscoveryMode mode)
```

由AbstractDiscover类中的stop指定

```
getMode()
```

由AbstractDiscover类中的getMode指定

## MdnsDevice 类

继承自 Device

### 构造器

```
MdnsDevice(java.lang.String addr, int port)
```

创建对象 MdnsDevice。

### 方法

```
getAddr()
```

获取地址。

```
getPort()
```

获取端口。

```
canBeFoundWith(DiscoveryMode mode)
```

由Device类中的canBeFoundWith指定

## AbstractDiscover.DiscoverStatus 枚举

封闭类

AbstractDiscover

### 枚举常量

STOPPED

停止。

WORKING

工作。

ERROR

错误。

## DiscoveryMode 枚举

表明机器人如何被发现。

### 枚举常量

BLE

BLE 模式。

MDNS

mdns

## com.slamtec.slamware.FirmwareUpdate包

### FirmwareUpdateInfo类

public class FirmwareUpdateInfo表示固件更新信息。

#### 构造器

```
FirmwareUpdateInfo(java.lang.String current, java.lang.String latest, java.lang.String releaseDate,  
java.lang.String brief)
```

创建对象FirmwareUpdateInfo且参数为current/latest/releasedDate/brief。

#### 方法

```
getBrief()
```

获取简明信息, 数据类型为字符串string。

```
getCurrent()
```

获取当前固件版本, 数据类型为字符串string。

```
getLatest()
```

获取最新固件版本, 数据类型为字符串string。

```
getReleaseDate()
```

获取固件发布日期,数据类型为字符串string。

## FirmwareUpdateProgress类

public class FirmwareUpdateProgress表示固件更新进程。

### 构造器

```
FirmwareUpdateProgress(int currentStep, int totalStep, int currentStepProgress, java.lang.String currentStepName)
```

创建对象FirmwareUpdateProgress且参数为currentStep/totalStep/currentStepProgress/currentStepName。

### 方法

```
getCurrentStep()
```

获取当前步骤,数据类型为int。

```
getCurrentStepName()
```

获取当前步骤名称,数据类型为字符串string。

```
getCurrentStepProgress()
```

获取当前步骤进程,数据类型为int。

```
getTotalStep()
```

获取所有步骤,数据类型为int。

## com.slamtec.slamware.geometry包

### Line类

public class Line,表示一条线。

### 构造器

```
Line(int segmentId, PointF startPoint, PointF endPoint)
```

创建对象Line且segment id, start point和end point为指定值。

```
Line(int segmentId, float startX, float startY, float endX, float endY)
```

创建对象Line 且segment id, startX, startY, endX, endy 为指定值。

```
Line(Line line)
```

创建对象Line 且以Line为参数。

```
Line(PointF startP, PointF endP)
```

创建对象Line 且startP和endP为指定值。

### 方法

```
getStartPoint()
```

获取起始点。

```
setStartPoint(PointF startPointF)
```

设置起始点。

```
getEndPoint()
```

获取结束点。

```
setEndPoint(PointF endPoint)
```

设置结束点。

```
getStartX()
```

获取start x。

```
getStartY()
```

获取star y。

```
getEndX()
```

获取end x。

```
getEndY()
```

获取end y。

```
getSegmentId()
```

获取segment id。

```
setSegmentId(int segmentId)
```

设置Segment id。

## PointF 类

public class PointF,表示一个2d浮点数据类型。

### 构造器

```
PointF()
```

创建对象PointF。

```
PointF(float x, float y)
```

创建对象PointF且x, y为指定值。

```
PointF(PointF rhs)
```

创建对象PointF且以PointF为参数。

### 方法

```
getX()
```

获取X。

```
setX(float x)
```

设置X。

```
getY()
```

获取Y。

```
setY(float y)
```

设置Y。

## Size 类

public class Size, 表示整数型的size数据类型。

## 构造器

Size()

创建对象Size。

Size(int width, int height)

创建对象Size且width和height为指定值。

Size(Size rhs)

创建对象Size且以Size为参数。

## 方法

getWidth()

获取width。

setWidth(int width)

设置width。

getHeight()

获取height。

setHeight(int height)

设置height。

# com.slamtec.slamware.robot包

## HealthInfo类

public class HealthInfo, 表示机器人的健康状况。

## 嵌套类

HealthInfo.BaseError

## 构造器

HealthInfo()

创建对象HealthInfo()。

HealthInfo(boolean warning, boolean error, boolean fatal, java.util.ArrayList<HealthInfo.BaseError> errors)

创建对象HealthInfo(), 参数为warning/error/fatal/错误列表。

## 方法

getErrors()

获取错误信息。返回值为错误信息列表。

isError()

获取是否是错误信息。返回数据类型为Boolean。

isFatal()

获取是否是致命错误信息。返回数据类型为Boolean。

`isWarning()`

获取是否是警告信息。返回数据类型为Boolean。

`setError(boolean error)`

是否设置错误信息。

#### 参数

参数名称	描述
error	将要处理的错误信息

`setErrors(java.util.ArrayList<HealthInfo.BaseError> errors)`

设置为错误信息列表。

#### 参数

参数名称	描述
errors	错误信息列表

`setFatal(boolean fatal)`

设置为致命错误信息。

#### 参数

参数名称	描述
fatal	致命错误

`setWarning(boolean warning)`

设置为警告。

#### 参数

参数名称	描述
warning	警告类错误信息

## HealthInfo.BaseError类

封闭类：HealthInfo

### Fields

`BaseErrorLevelHealthy`  
机器人底盘健康错误等级

`BaseErrorComponentMotion`  
机器人底盘运动错误。

`BaseErrorComponentPower`  
机器人底盘充电错误。

`BaseErrorComponentSensor`

机器人底盘传感器错误。

BaseErrorComponentSystem

机器人底盘系统错误。

BaseErrorComponentUnknown

机器人底盘未知错误。

BaseErrorComponentUser

机器人底盘用户错误。

BaseErrorLevelError

底盘错误等级为错误。

BaseErrorLevelFatal

底盘错误等级为致命错误。

BaseErrorLevelUnknown

底盘错误等级为未知错误。

BaseErrorLevelWarn

底盘错误等级为警告错误。

BaseComponentErrorTypeUnknown

BaseComponentErrorTypeUser

BaseComponentErrorTypeSystemNone

( )

BaseComponentErrorTypeSystemEmergencyStop

BaseComponentErrorTypeSystemTemperatureHigh

BaseComponentErrorTypeSystemTemperatureLow

BaseComponentErrorTypeSystemWatchDogOverFlow

BaseComponentErrorTypeSystemCtrlBusDisconnected

BaseComponentErrorTypeSystemSlamwareRebooted

slamware

BaseComponentErrorTypePowerNone

( )

BaseComponentErrorTypePowerControllerDown

BaseComponentErrorTypePowerPowerLow

BaseComponentErrorTypePowerOverCurrent

BaseComponentErrorTypeMotionNone

()

BaseComponentErrorTypeMotionControllerDown

BaseComponentErrorTypeMotionMotorAlarm

BaseComponentErrorTypeMotionMotorDown

BaseComponentErrorTypeMotionOdometryDown

BaseComponentErrorTypeMotionBrushStall

()

BaseComponentErrorTypeMotionBlowerStall

BaseComponentErrorTypeSensorNone

BaseComponentErrorTypeSensorControllerDown

BaseComponentErrorTypeSensorBumperDown

BaseComponentErrorTypeSensorCliffDown

BaseComponentErrorTypeSensorSonarDown

BaseComponentErrorTypeSensorDustbinBlock

()



BaseComponentErrorTypeSensorDustbinGone

()

BaseComponentErrorTypeSensorWallIrDown

()

BaseComponentErrorTypeSensorMagTapeTriggered

BaseComponentErrorTypeSensorMagSelfTestFailed

## 构造器

BaseError()

创建对象BaseError ()。

BaseError(int id, int errorCode, int errorLevel, int errorComponent, int componentErrorCode, java.lang.String errorMessage)

创建对象BaseError ()。

## 参数

参数名称	描述
error code	错误代码
error level	错误等级
error component	错误组件
componentErrorCode	组件错误代码
errorMessage	错误信息

## 方法

getComponentErrorCode()

获取组件错误代码。返回数据类型为int。

getErrorCode()

获取错误代码。返回数据类型为int。

getErrorComponent()

获取错误组件。返回数据类型为int。

getErrorLevel()

获取错误等级。返回数据类型为int。

getErrorMessage()

获取错误信息。返回数据类型为string。

getId()

获取错误id。返回数据类型为int。

setComponentErrorCode(int componentErrorCode)

设置组件错误代码。返回值为组件错误代码。数据类型为int。

```
setErrorCode(int errorCode)
```

设置错误代码。返回值为错误代码。数据类型为int。

```
setErrorComponent(int errorComponent)
```

设置错误组件。返回值为错误组件。数据类型为int。

```
setErrorLevel(int errorLevel)
```

设置错误等级。返回值为错误等级。数据类型为int。

```
setErrorMessage(java.lang.String errorMessage)
```

设置错误信息。返回值为错误信息。数据类型string。

```
setId(int id)
```

设置Id。返回值为Id。数据类型为int。

## LaserPoint 类

public class LaserPoint, 表示一个激光扫描点。

### 构造器

```
LaserPoint()
```

创建对象LaserPoint。

```
LaserPoint(float distance, float angle)
```

创建对象LaserPoint且距离和角度为指定值。

```
LaserPoint(float distance, float angle, boolean valid)
```

创建对象LaserPoint且距离, 角度和有效性为指定值。

```
LaserPoint(LaserPoint rhs)
```

创建对象LaserPoint且以LaserPoint为参数。

### 方法

```
getDistance()
```

获取距离。

```
setDistance(float distance)
```

设置距离。

```
getAngle()
```

获取角度。

```
setAngle(float angle)
```

设置角度。

```
isValid()
```

测量是否有效。

```
setValid(boolean valid)
```

将值设为有效。

## LaserScan 类

public class LaserScan, 表示激光扫描。

## 构造器

LaserScan()

创建对象LaserScan。

LaserScan(java.util.Vector<LaserPoint> laserPoints)

创建对象LaserScan且以LaserPoints为参数。

LaserScan(java.util.Vector<LaserPoint> laserPointsPose pose)

创建对象LaserScan且以LaserPointPose为参数。

LaserScan(LaserScan rhs)

创建对象LaserScan且以LaserScan为参数。

## 方法

getLaserPoints()

获取激光点。

setLaserPoints(java.util.Vector<LaserPoint> laserPoints)

设置激光点。

getPose()

获取姿态。

setPose(Pose pose)

设置姿态。

## Location 类

public class Location, 表示3d空间中机器人的位置。

## 构造器

Location()

创建对象Location。

Location(float x, float y, float z)

创建对象Location且x, y, z为指定值。

Location(Location rhs)

创建对象Location且以Location为参数。

## 方法

distanceTo(Location that)

获取到Location的距离。

getX()

获取X值。

setX(float v)

设置X值。

getY()

获取Y值。

setY(float v)

设置Y值。

getZ()

获取Z值。

setZ(float v)

设置Z值。

## Map 类

public class Map, 表示地图。

### 构造器

Map(PointF origin, Size dimension, PointF resolution, long timestamp, byte[] data)

创建对象Map。

getOrigin()

获取origin。

setOrigin(PointF origin)

设置origin。

getDimension()

获取dimension。

setDimension(Size dimension)

设置dimensions。

getResolution()

获取resolution。

setResolution(PointF resolution)

设置resolution。

getTimestamp()

获取time stamp。

setTimestamp(long timestamp)

设置time stamp。

getMapArea()

获取map area。

getData()

获取data。

setData(byte[] data)

设置data。

## NetworkMode类

## Fields

NetworkModeAP

网络模式为AP。

NetworkModeStation

网络模式为Station。

NetworkModeWifiDisabled

Wifi禁用的网络模式。

## 构造器

NetworkMode()

创建对象NetworkMode()。

## Pose 类

public class Pose, 表示机器人姿态。

## 构造器

Pose()

创建对象Pose。

Pose(Location loc, Rotation rot)

创建对象Pose且loc和rot为指定值。

Pose(float x, float y, float z, float yaw, float roll, float pitch)

创建对象Pose且x, y, z, yaw, roll, pitch为指定值。

Pose(Pose rhs)

创建对象Pose且以Pose为参数。

## 方法

Location getLocation()

获取Location。

setLocation(Location location)

设置Location。

getRotation()

获取rotation。

setRotation(Rotation rotation)

设置rotation。

getX()

获取X。

setX(float v)

设置X。

getY()

获取Y。

`setY(float v)`

设置Y。

`getZ()`

获取Z。

`setZ(float v)`

设置Z。

`getYaw()`

获取yaw。

`setYaw(float v)`

设置yaw。

`getRoll()`

获取roll。

`setRoll(float v)`

设置roll。

`getPitch()`

获取pitch。

`setPitch(float v)`

设置pitch。

## Rotation 类

`public class Rotation。`

### 构造器

`Rotation()`

创建对象Rotation。

`Rotation(float yaw)`

创建对象Rotation且yaw为指定值。

`Rotation(float yaw, float pitch, float roll)`

创建对象Rotation且yaw, pitch, roll为指定值。

`Rotation(Rotation rhs)`

创建对象Rotation且以Rotation为参数。

### 方法

`getYaw()`

获取yaw。

`setYaw(float yaw)`

设置Yaw。

`getRoll()`

获取roll。

`setRoll(float roll)`

设置roll。

```
getPitch()
```

获取pitch。

```
setPitch(float pitch)
```

设置pitch。

## SystemParameters 类

### 构造器

```
SystemParameters()
```

创建对象SystemParameters。

### Fields

```
SYSPARAM_ROBOT_SPEED
```

机器人速度。

```
SYSVAL_ROBOT_SPEED_HIGH
```

机器人高速度。

```
SYSVAL_ROBOT_SPEED_MEDIUM
```

机器人中速度。

```
SYSVAL_ROBOT_SPEED_LOW
```

机器人低速度。

## SleepMode枚举

枚举常量。

```
Unknown
```

目标Slamware固件不支持该命令。

```
Awake
```

醒着的模式。（当前雷达处于工作状态,机器人可以立刻工作）。

```
WakingUp
```

正在唤醒模式。(当调用wakeup()方法后,就处于当前状态)。

```
Asleep
```

睡眠模式。（当前雷达处于休眠状态,机器人不会立刻工作,调用wakeup()方法可以将雷达处于WakingUp状态,然后处于Awake）。

## DockingStatus枚举

枚举常量。

```
Unknown
```

Docking状态未知（目标Slamware固件不支持）。

```
OnDock
```

设备在充电座上。

```
NotOnDock
```

设备不在充电座上。

## PowerStatus类

### 构造器

```
public PowerStatus(boolean isDCConnected, DockingStatus dockingStatus, boolean isCharging,int batteryPercentage, SleepMode sleepMode)
```

### 方法

```
public boolean isDCConnected()
```

直流电源连接。

```
public DockingStatus getDockingStatus()
```

Docking状态。

```
public boolean isCharging()
```

设备是否处于充电状态。

```
public int getBatteryPercentage()
```

电池电量（从0~100）。

```
public SleepMode getSleepMode()
```

设备休眠模式。

## GridMap类

### 方法

```
public Location getOrigin()
```

获取原点位置。

```
public void setOrigin(Location origin)
```

设置原点位置。

```
public Size getDimension()
```

获取grid地图尺寸。

```
public void setDimension(Size dimension)
```

设置grid地图尺寸。

```
public PointF getResolution()
```

获取grid地图分辨率。

```
public void setResolution(PointF resolution)
```

设置grid地图分辨率。

```
public byte[] getMapData()
```

获取grid地图信息。

```
public void setMapData(byte[] mapData)
```

设置grid地图信息。



```
public void clear()
```

清除grid地图信息。

## CompositeMap类

### 构造器

```
public CompositeMap()
```

默认构造器。

```
public CompositeMap(CompositeMap another)
```

赋值构造器。

### 方法

```
public MapMetaData getMetaData()
```

获取Composite map属性信息。

```
public void setMetaData(MapMetaData data)
```

设置Composite map属性信息。

```
public ArrayList<MapLayer> getMaps()
```

获取Composite map地图信息。

```
public void setMaps(ArrayList<MapLayer> maps)
```

设置Composite map地图信息。

## MapKind枚举

### 枚举常量

```
EXPLORE_MAP
```

利用SLAM算法创建的地图。

```
SWEEP_MAP
```

清扫操作使用的地图（仅适用于扫地机版本机器人）。

```
UWB_MAP
```

机器人根据UWB技术获取位置，形成的地图。

## MapType枚举

### 枚举常量

```
BITMAP_8BIT
```

位图，每个像素是一个带符号的8bit integer。

## MapLayer类

地图层。

## 构造器

`MapLayer()`

创建MapLayer对象。

## 方法

`getMetaData()`

获取到地图数据。

`getName()`

获取到地图名称。

`getUsage()`

获取到地图用法。

`clear()`

清除所有数据。

## PoseMap类

继承自MapLayer。

## 构造器

`PoseMap()`

创造PoseMap对象。

## 方法

`getPoses()`

获取到当前Poses map集合。

`setPoses()`

设置当前Poses map集合数据。

## PointMap类

## 构造器

`PointMap()`

创建PointMap对象。

## 方法

`PointMap(java.util.List points, long timestamp)`

根据points集合和timeStamp 创建对象。

`getMapArea()`

获取到当前地图区域的rectF。

`getPoints()`

获取到所有的点，返回值类型是集合。

`setPoints()`

设置所有的点。

## RecoverLocalizationMovement枚举

根据当前机器人行为，进行的重定位的枚举。

`Unknown`

未知当前机器人行为（一般不用）。

`NoMove`

当前机器人不移动。

`RotateOnly`

当前机器人仅仅进行旋转时。

`Any`

当前机器人进行任意行为。

## RecoverLocalizationOptions类

重新定位选项。

### 构造器

`RecoverLocalizationOptions()`

创建`RecoverLocalizationOptions`对象。

### 方法

`getMaxRecoverTimeInMilliseconds()`

获取从新定位的最大时间。

`setMaxRecoverTimeInMilliseconds(Integer maxRecoverMS)`

设置从新定位的最大时间。

`setRecoverMovementType(RecoverLocalizationMovement movementType)`

设置从新定位需要传入的参数，该参数是当前发生的movement。

`RecoverLocalizationMovement getRecoverMovementType()`

获取从新定位时，当前机器人发生的movement。

## SensorType枚举

机器人传感器类型。

`Bumper`

碰撞传感器。

`Cliff`

防跌落传感器。

`Sonar`

超声波传感器。

DepthCamera

深度摄像头传感器。

WallSensor

沿墙传感器。

## SlamcoreShutdownParam类

底盘重启参数类。

### 构造器

SlamcoreShutdownParam()

创建SlamcoreShutdownParam对象。

### 方法

int getRestartTimeIntervalMinute()

获取到底盘多少分钟重启。

setRestartTimeIntervalMinute(int restartTimeIntervalMinute)

设置底盘重启时间参数。

## SleepMode 枚举

当前底盘休眠模式。

Unknown

未知模式。

Awake

醒着的模式。（当前雷达处于工作状态,机器人可以立刻工作）。

WakingUp

正在唤醒模式。（当调用wakeup()方法后，就处于当前状态）。

Asleep

睡眠模式。（当前雷达处于休眠状态,机器人不会立刻工作,调用wakeup()方法可以将雷达处于WakingUp状态,然后处于Awake）。

## RestartMode枚举

### 枚举常量

SOFT

软重启模块，仅重启模块中的核心部分。

HARD

硬重启模块，整个模块重新启动，可能会花费较长的时间。

## LineMap类

## 方法

```
public HashMap<String, CompositeLine> getLines()
```

获取line map数据。

```
public void setLines(HashMap<String, CompositeLine> lines)
```

设置line map数据。

```
public void clear()
```

清除line map数据。

## PointsMap类

```
public ArrayList<PointPDF> getPoints()
```

获取points map数据。

```
public void setPoints(ArrayList<PointPDF> points)
```

设置points map数据。

```
public void clear()
```

清除points map数据。

## ImpactSensorInfo 类

### 构造器

```
ImpactSensorInfo(int sensorId, Pose pose, ImpactSensorType type, SensorType coreSensorType, float refreshFreq)
```

创建ImpactSensorInfo对象。

## 方法

```
getSensorId()
```

获取传感器的id。

```
getPose()
```

获取当前传感器的位置。

```
getKind()
```

获取到传感器种类。

```
getType()
```

获取到传感器类型。

## ImpactSensorType枚举

```
Digital
```

数字类传感器。

```
Analog
```

模拟信号类传感器。

## ImpactSensorValue类

### 构造器

```
ImpactSensorValue(long time, float value)
```

创建对象参数是 time, value。

### 方法

```
getTime()
```

获取时间。

```
getValue()
```

获取值。

## com.slamtec.slamware.exceptions包

### AlreadyConnectedException类

已连接到机器，异常信息为“Already Connected to the device”。

### ConnectionFailException类

连接机器人失败，异常信息为“Connection Failed”。

### ConnectionLostException类

连接丢失，异常信息为“Connection Lost”。

### ConnectionTimeOutException类

连接超时，异常信息为“Connection Time Out”。

### InvalidArgumentException类

参数无效，异常信息为“Invalid Argument”。

### OperationFailException类

操作失败，异常信息为“Operation Failed”。

### OutOfResourceException类

资源不足，异常信息为“Out of Resource”。

### ParseInvalidException类

参数解析无效，异常信息为“Failed to parse device data”。

## PathFindFailException类

未能找到路径，异常信息为“Failed to find path”。

## RequestFailException类

请求失败，异常信息为“Request Failed”。

## RequestTimeoutException类

请求时间超时，异常信息为“Request Time Out”。

## UnauthorizedRequestException类

请求未授权，异常信息为“Unauthorized request”。

## UnsupportedCommandException类

命令不支持，异常信息为“Unsupported Command”。

# com.slamtec.slamware.message包

## DepthCameraFrame类

### 构造器

```
DepthCameraFrame()
```

创建DepthCameraFrame 对象。

### 方法

```
setMinValidDistance(float minValidDistance)
```

设置深度摄像头最小探测的距离，单位是米。

```
float getMinValidDistance()
```

获取深度摄像头最小探测的距离，单位是米。

```
setMaxValidDistance()
```

设置深度摄像头最大探测的距离，单位是米。

```
float getMaxValidDistance()
```

获取深度摄像头最大探测的距离，单位是米。

```
setMinFovPitch(float MinFovPitch)
```

设置深度摄像头垂直方向朝下最大探测仰度，单位是弧度。

```
float getMinFovPitch()
```

获取深度摄像头垂直方向朝下最大探测仰度，单位是弧度。

```
setMaxFovPitch(float MaxFovPitch)
```

设置深度摄像头垂直方向朝上最小探测的仰角，单位是弧度。

```
float getMaxFovPitch()
```

获取深度摄像头垂直方向朝上最小探测的仰角，单位是弧度。

```
setMinFovYaw(float MinFovYaw )
```

设置深度摄像头水平方向朝左最大探测的角度，单位是弧度。

```
float getMinFovYaw()
```

获取深度摄像头水平方向朝左最大探测的角度，单位是弧度。

```
setMaxFovYaw(float MaxFovYaw)
```

设置深度摄像头水平方向朝右最小探测的角度，单位是弧度。

```
float getMaxFovYaw()
```

获取深度摄像头水平方向朝右最小探测的角度，单位是弧度。

```
setCols(int cols)
```

设置深度摄像头数据多少列。

```
int getCols()
```

获取深度摄像头数据多少列。

```
setRows(int rows)
```

设置深度摄像头数据多少行。

```
int getRows()
```

获取深度摄像头数据多少行。

```
setData(ArrayList<Float> data)
```

设置深度摄像头采集到的数据，数据大小为行\*列（cols\*rows），数据的排列由左到右依次往下，float的单位是米。

```
ArrayList<float> getData()
```

获取深度摄像头采集到的数据，数据大小为行\*列（cols\*rows），数据的排列由左到右依次往下，float的单位是米。

## com.slamtec.slamware.sdp包

### Action类

继承自 IAction。

#### 方法

```
cancel()
```

取消本次操作。

```
getActionName()
```

获取动作名称。返回值为动作名称（string）。

```
getProgress()
```

获取动作进程（0~1）。返回值为动作进程（double）。

```
getStatus()
```

获取动作状态。返回值为动作状态。

```
waitUntilDone()
```



等待动作完成。返回值为完成的动作结果。

```
isEmpty()
```

判断当前action是否为空。

```
getActionId()
```

获取当前action的id。

```
String getActionName()
```

获取到当前action的名字。

```
String getReason()
```

获取当前action结束后返回的字符串，成功后会返回不为null且长度为0的字符串，失败后会返回失败原因的字符串。如：

```
"failed"action
```

```
"aborted"action(cancelactionaction)
```

```
"unreachable"()
```

```
"unhealthy",getRobotHealth()
```

```
"lifted",,"low_localization_quality"
```

```
moveToMoveOptionFlagKeyPointsflag,moveto:
```

```
getReason("blocked[lidar;wall;contact;depth_camera;sonar;cliff]","[]",",",(),).lidar:;wall:;contact:;depth_
```

```
camera:;sonar:cliff:
```

```
releaseCPointer()
```

释放所有指针资源。

## CompositeMapHelper

地图工具类。

### 构造器

```
CompositeMapHelper()
```

```
CompositeMapHelper
```

### 方法

```
loadFile(String var1)
```

根据路径加载地图。

```
saveFile(String var1, CompositeMap var2)
```

根据路径保存地图。