ApiReference

ActivityResultFacade

- setResultBoolean
- setResultBooleanArray
- setResultByte
- setResultByteArray
- setResultChar
- setResultCharArray
- setResultDouble
- setResultDoubleArray
- setResultFloat
- setResultFloatArray
- setResultInteger
- setResultIntegerArray
- setResultLong
- setResultLongArray
- setResultSerializable
- setResultShort
- setResultShortArray
- setResultString
- setResultStringArray

AlarmManagerFacade

- cancelRepeating
- scheduleAbsolute
- scheduleInexactRepeating
- scheduleRelative
- scheduleRepeating

AndroidFacade

- getClipboard
- getInput
- getIntent
- getPackageVersion
- getPackageVersionCode
- getPassword
- log
- makeToast
- notify
- requiredVersion
- sendEmail
- setClipboard

- startActivity
- startActivityForResult
- vibrate

ApplicationManagerFacade

- forceStopPackage
- getLaunchableApplications
- getRunningPackages
- launch

BatteryManagerFacade

- batteryCheckPresent
- batteryGetHealth
- batteryGetLevel
- batteryGetPlugType
- batteryGetStatus
- batteryGetTechnology
- batteryGetTemperature
- batteryGetVoltage
- batteryStartMonitoring
- batteryStopMonitoring
- readBatteryData

BluetoothFacade Requires API Level 5.

- bluetoothAccept
- bluetoothConnect
- bluetoothGetConnectedDeviceName
- bluetoothMakeDiscoverable
- bluetoothRead
- bluetoothReadLine
- bluetoothReadReady
- bluetoothStop
- bluetoothWrite
- checkBluetoothState
- toggleBluetoothState

CameraFacade

- cameraCapturePicture
- cameraInteractiveCapturePicture

CommonIntentsFacade

- pick
- scanBarcode
- search
- view
- viewContacts
- viewHtml
- viewMap

ConditionManagerFacade

- onRingerSilent

ContactsFacade

- contactsGet
- contactsGetAttributes
- contactsGetById
- contactsGetCount
- contactsGetIds
- pickContact
- pickPhone

EventFacade

- postEvent
- receiveEvent
- waitForEvent

LocationFacade

- geocode
- getLastKnownLocation
- readLocation
- startLocating
- stopLocating

MediaRecorderFacade

- recorderCaptureVideo
- recorderStartMicrophone
- recorderStop
- startInteractiveVideoRecording

PhoneFacade

- checkNetworkRoaming

- getCellLocation
- getDeviceId
- getDeviceSoftwareVersion
- getLine1Number
- getNeighboringCellInfo
- getNetworkOperator
- getNetworkOperatorName
- getNetworkType
- getPhoneType
- getSimCountryIso
- getSimOperator
- getSimOperatorName
- getSimSerialNumber
- getSimState
- getSubscriberId
- getVoiceMailAlphaTag
- getVoiceMailNumber
- phoneCall
- phoneCallNumber
- phoneDial
- phoneDialNumber
- readPhoneState
- startTrackingPhoneState
- stopTrackingPhoneState

PulseGeneratorFacade

- pulseGeneratorGetHz
- pulseGeneratorGetHzSamples
- pulseGeneratorGetLeftPulseMs
- pulseGeneratorGetLeftPulsePercent
- pulseGeneratorGetLeftPulseSamples
- pulseGeneratorGetRightPulseMs
- pulseGeneratorGetRightPulsePercent
- pulseGeneratorGetRightPulseSamples
- pulseGeneratorIsPlaying
- pulseGeneratorRun
- pulseGeneratorSetHzPercent
- pulseGeneratorSetLeftPulsePercent
- pulseGeneratorSetRightPulsePercent
- pulseGeneratorStop
- pulseGeneratorToggleInverted
- pulseGeneratorTogglePlayback
- pulseGenteratorSetHz

SensorManagerFacade

- readSensors
- sensorsGetAccuracy
- sensorsGetLight
- sensorsReadAccelerometer
- sensorsReadMagnetometer
- sensorsReadOrientation
- startSensing
- stopSensing

SettingsFacade

- checkAirplaneMode
- checkRingerSilentMode
- checkScreenOn
- getMaxMediaVolume
- getMaxRingerVolume
- getMediaVolume
- getRingerVolume
- getScreenBrightness
- getScreenTimeout
- setMediaVolume
- setRingerVolume
- setScreenBrightness
- setScreenTimeout
- toggleAirplaneMode
- toggleRingerSilentMode

SignalStrengthFacade Requires API Level 7.

- readSignalStrengths
- startTrackingSignalStrengths
- stopTrackingSignalStrengths

SmsFacade

- smsDeleteMessage
- smsGetAttributes
- smsGetMessageById
- smsGetMessageCount
- smsGetMessageIds
- smsGetMessages
- smsMarkMessageRead
- smsSend

SpeechRecognitionFacade

- recognizeSpeech

TextToSpeechFacade Requires API Level 4.

- ttsIsSpeaking
- ttsSpeak

ToneGeneratorFacade

- generateDtmfTones

UiFacade

- addContextMenuItem
- addOptionsMenuItem
- clearContextMenu
- clearOptionsMenu
- dialogCreateAlert
- dialogCreateDatePicker
- dialogCreateHorizontalProgress
- dialogCreateInput
- dialogCreatePassword
- dialogCreateSeekBar
- dialogCreateSpinnerProgress
- dialogCreateTimePicker
- dialogDismiss
- dialogGetInput
- dialogGetPassword
- dialogGetResponse
- dialogGetSelectedItems
- dialogSetCurrentProgress
- dialogSetItems
- dialogSetMaxProgress
- dialogSetMultiChoiceItems
- dialogSetNegativeButtonText
- dialogSetNeutralButtonText
- dialogSetPositiveButtonText
- dialogSetSingleChoiceItems
- dialogShow
- webViewShow

WakeLockFacade

- wakeLockAcquireBright
- wakeLockAcquireDim
- wakeLockAcquireFull

- wakeLockAcquirePartial
- wakeLockRelease

WifiFacade

- checkWifiState
- toggleWifiState
- wifiDisconnect
- wifiGetConnectionInfo
- wifiGetScanResults
- wifiLockAcquireFull
- wifiLockAcquireScanOnly
- wifiLockRelease
- wifiReassociate
- wifiReconnect
- wifiStartScan

addContextMenuItem¶

```
addContextMenuItem(
 String label: label for this menu item,
 String event: event that will be generated on menu item click,
 Object eventData[optional])

Adds a new item to context menu.
```

addOptionsMenuItem¶

```
addOptionsMenuItem(
 String label: label for this menu item,
 String event: event that will be generated on menu item click,
 Object eventData[optional],
 String iconName[optional]: Android system menu icon, see

http://developer.android.com/reference/android/R.drawable.html)

Adds a new item to options menu.
```

batteryCheckPresent¶

```
batteryCheckPresent()

Returns the most recently received battery presence data

Requires API Level 5.
```

batteryGetHealth¶

```
batteryGetHealth()
```

Returns the most recently received battery health data:
1 - unknown;
2 - good;
3 - overheat;
4 - dead;
5 - over voltage;
6 - unspecified failure;

batteryGetLevel¶

```
batteryGetLevel()
```

Returns the most recently received battery level (percentage)

Requires API Level 5.

batteryGetPlugType¶

```
batteryGetPlugType()
```

Returns the most recently received plug type data:
-1 - unknown;
0 - unplugged;
1 - power source is an AC charger;
2 - power source is a USB port;

batteryGetStatus¶

```
batteryGetStatus()
```

Returns the most recently received battery status data:
1 - unknown;
2 - charging;
3 - discharging;
4 - not charging;
5 - full;

batteryGetTechnology¶

```
batteryGetTechnology()
```

Returns the most recently received battery technology data

Requires API Level 5.

batteryGetTemperature¶

```
batteryGetTemperature()
```

Returns the most recently received battery temperature

Requires API Level 5.

batteryGetVoltage¶

```
batteryGetVoltage()

Returns the most recently received battery voltage

Requires API Level 5.
```

batteryStartMonitoring¶

```
batteryStartMonitoring()

Starts tracking battery state.

Generates "battery" events.
```

batteryStopMonitoring¶

```
batteryStopMonitoring()

Stops tracking battery state.
```

bluetoothAccept¶

```
bluetoothAccept(
 String uuid[optional, default 457807c0-4897-11df-9879-0800200c9a66])

Listens for and accepts a Bluetooth connection. Blocks until the connection
is
established or fails.

Requires API Level 5.
```

bluetoothConnect¶

```
bluetoothConnect(
 String uuid[optional, default 457807c0-4897-11df-9879-0800200c9a66]: The
UUID
passed here must match the UUID used by the server device.,
 String address[optional]: The user will be presented with a list of
discovered
devices to choose from if an address is not provided.)

Connect to a device over Bluetooth. Blocks until the connection is
established
or fails.

Returns:
  True if the connection was established successfully.

Requires API Level 5.
```

bluetoothGetConnectedDeviceName¶

```
bluetoothGetConnectedDeviceName()
```

Returns the name of the connected device.

Requires API Level 5.

bluetoothMakeDiscoverable¶

```
bluetoothMakeDiscoverable(
 Integer duration[optional, default 300]: period of time, in seconds, during

which the device should be discoverable)
```

Requests that the device be discoverable for Bluetooth connections.

Requires API Level 5.

bluetoothRead¶

```
bluetoothRead(
 Integer bufferSize[optional, default 4096])
```

Read up to bufferSize bytes.

Requires API Level 5.

bluetoothReadLine¶

```
bluetoothReadLine()
```

Read the next line.

Requires API Level 5.

bluetoothReadReady¶

```
bluetoothReadReady()
```

Returns True if the next read is guaranteed not to block.

Requires API Level 5.

bluetoothStop¶

```
bluetoothStop()
```

Stops Bluetooth connection.

Requires API Level 5.

bluetoothWrite¶

```
bluetoothWrite(
 String bytes)
```

Sends bytes over the currently open Bluetooth connection.

Requires API Level 5.

cameraCapturePicture¶

```
cameraCapturePicture(
 String path,
 Boolean useAutoFocus[optional, default true])
```

Take a picture and save it to the specified path.

Returns:
A map of Booleans autoFocus and takePicture where True indicates success.

cameraInteractiveCapturePicture¶

```
cameraInteractiveCapturePicture(
 String path)
```

Starts the image capture application to take a picture and saves it to the specified path.

cancelRepeating¶

```
cancelRepeating(
 String script)
```

cancels all scheduled regular executions of a given script

checkAirplaneMode¶

```
checkAirplaneMode()
```

Checks the airplane mode setting.

Returns:
True if airplane mode is enabled.

checkBluetoothState¶

```
checkBluetoothState()
```

Checks Bluetooth state.

Returns:

  True if Bluetooth is enabled.

Requires API Level 5.

checkNetworkRoaming¶

checkNetworkRoaming()

Returns true if the device is considered roaming on the current network, for GSM purposes.

checkRingerSilentMode¶

checkRingerSilentMode()

Checks the ringer silent mode setting.

Returns:
  True if ringer silent mode is enabled.

checkScreenOn¶

checkScreenOn()

Checks if the screen is on or off (requires API level 7).

Returns:
  True if the screen is currently on.

checkWifiState¶

checkWifiState()

Checks Wifi state.

Returns:
  True if Wifi is enabled.

clearContextMenu¶

clearContextMenu()

Removes all items previously added to context menu.

clearOptionsMenu¶

clearOptionsMenu()

Removes all items previously added to options menu.

contactsGet¶

```
contactsGet(
 JSONArray attributes[optional])
```

Returns a List of all contacts.

```
Returns:
  a List of contacts as Maps
```

contactsGetAttributes¶

```
contactsGetAttributes()
```

Returns a List of all possible attributes for contacts.

contactsGetById¶

```
contactsGetById(
 Integer id,
  JSONArray attributes[optional])
```

Returns contacts by ID.

contactsGetCount¶

```
contactsGetCount()
```

Returns the number of contacts.

contactsGetIds¶

```
contactsGetIds()
```

Returns a List of all contact IDs.

dialogCreateAlert¶

```
dialogCreateAlert(
 String title[optional],
  String message[optional])
```

Create alert dialog.

dialogCreateDatePicker¶

```
dialogCreateDatePicker(
 Integer year[optional, default 1970],
 Integer month[optional, default 1],
 Integer day[optional, default 1])
```

Create date picker dialog.

dialogCreateHorizontalProgress¶

```
dialogCreateHorizontalProgress(
 String title[optional],
 String message[optional],
 Integer maximum progress[optional, default 100])
```

Create a horizontal progress dialog.

dialogCreateInput¶

```
dialogCreateInput(
 String title[optional, default Value]: title of the input box,
 String message[optional, default Please enter value:]: message to display

above the input box,
 String defaultText[optional]: text to insert into the input box)
```

Create a text input dialog.

dialogCreatePassword¶

```
dialogCreatePassword(
 String title[optional, default Password]: title of the input box,
 String message[optional, default Please enter password:]: message to display

above the input box)
```

Create a password input dialog.

dialogCreateSeekBar¶

```
dialogCreateSeekBar(
 Integer starting value[optional, default 50],
 Integer maximum value[optional, default 100],
 String title,
  String message)
```

Create seek bar dialog.

dialogCreateSpinnerProgress¶

```
dialogCreateSpinnerProgress(
 String title[optional],
 String message[optional],
 Integer maximum progress[optional, default 100])
```

Create a spinner progress dialog.

dialogCreateTimePicker¶

```
dialogCreateTimePicker(
 Integer hour[optional, default 0],
 Integer minute[optional, default 0],
 Boolean is24hour[optional, default false]: Use 24 hour clock)
```

Create time picker dialog.

dialogDismiss¶

```
dialogDismiss()
```

Dismiss dialog.

dialogGetInput¶

```
dialogGetInput(
 String title[optional, default Value]: title of the input box,
 String message[optional, default Please enter value:]: message to display

above the input box,
 String defaultText[optional]: text to insert into the input box)
```

Queries the user for a text input.

dialogGetPassword¶

```
dialogGetPassword(
 String title[optional, default Password]: title of the password box,
 String message[optional, default Please enter password:]: message to display

above the input box)
```

Queries the user for a password.

dialogGetResponse¶

```
dialogGetResponse()
```

Returns dialog response.

dialogGetSelectedItems¶

```
dialogGetSelectedItems()
```

This method provides list of items user selected.

Returns:
  Selected items

dialogSetCurrentProgress¶

```
dialogSetCurrentProgress(
 Integer current)
```

Set progress dialog current value.

dialogSetItems¶

```
dialogSetItems(
  JSONArray items)
```

Set alert dialog list items.

dialogSetMaxProgress¶

```
dialogSetMaxProgress(
  Integer max)
```

Set progress dialog maximum value.

dialogSetMultiChoiceItems¶

```
dialogSetMultiChoiceItems(
 JSONArray items,
 JSONArray selected[optional]: list of selected items)
```

Set dialog multiple choice items and selection.

dialogSetNegativeButtonText¶

```
dialogSetNegativeButtonText(
  String text)
```

Set alert dialog button text.

dialogSetNeutralButtonText¶

```
dialogSetNeutralButtonText(
  String text)
```

Set alert dialog button text.

dialogSetPositiveButtonText¶

```
dialogSetPositiveButtonText(
 String text)
```

Set alert dialog positive button text.

dialogSetSingleChoiceItems¶

```
dialogSetSingleChoiceItems(
 JSONArray items,
 Integer selected[optional, default 0]: selected item index)
```

Set dialog single choice items and selected item.

dialogShow¶

```
dialogShow()
```

Show dialog.

forceStopPackage¶

```
forceStopPackage(
 String packageName: name of package)
```

Force stops a package.

generateDtmfTones¶

```
generateDtmfTones(
 String phoneNumber,
 Integer toneDuration[optional, default 100]: duration of each tone in

milliseconds)
```

Generate DTMF tones for the given phone number.

geocode¶

```
geocode(
 Double latitude,
 Double longitude,
 Integer maxResults[optional, default 1]: maximum number of results)
```

Returns a list of addresses for the given latitude and longitude.

Returns:
  A list of addresses.

getCellLocation¶

```
getCellLocation()
```

Returns the current cell location.

getClipboard¶

```
getClipboard()
```

Read text from the clipboard.

Returns:
  The text in the clipboard.

getDeviceId¶

```
getDeviceId()
```

Returns the unique device ID, for example, the IMEI for GSM and the MEID for CDMA phones. Return null if device ID is not available.

getDeviceSoftwareVersion¶

```
getDeviceSoftwareVersion()
```

Returns the software version number for the device, for example, the IMEI/SV for GSM phones. Return null if the software version is not available.

getInput¶

```
getInput(
 String title[optional, default SL4A Input]: title of the input box,
 String message[optional, default Please enter value:]: message to display

above the input box)
```

Queries the user for a text input.

Deprecated! Please use dialogGetInput instead.

getIntent¶

```
getIntent()
```

Returns the intent that launched the script.

getLastKnownLocation¶

```
getLastKnownLocation()
```

Returns the last known location of the device.

Returns:
  A map of location information by provider.

getLaunchableApplications¶

getLaunchableApplications()

Returns a list of all launchable application class names.

getLine1Number¶

getLine1Number()

Returns the phone number string for line 1, for example, the MSISDN for a GSM phone. Return null if it is unavailable.

getMaxMediaVolume¶

getMaxMediaVolume()

Returns the maximum media volume.

getMaxRingerVolume¶

getMaxRingerVolume()

Returns the maximum ringer volume.

getMediaVolume¶

getMediaVolume()

Returns the current media volume.

getNeighboringCellInfo¶

getNeighboringCellInfo()

Returns the neighboring cell information of the device.

getNetworkOperator¶

getNetworkOperator()

Returns the numeric name (MCC+MNC) of current registered operator.

getNetworkOperatorName¶

getNetworkOperatorName()

Returns the alphabetic name of current registered operator.

getNetworkType¶

```
getNetworkType()
```

Returns a the radio technology (network type) currently in use on the device.

getPackageVersion¶

```
getPackageVersion(
  String packageName)
```

Returns package version name.

getPackageVersionCode¶

```
getPackageVersionCode(
  String packageName)
```

Returns package version code.

getPassword¶

```
getPassword(
 String title[optional, default SL4A Password Input]: title of the input box,
 String message[optional, default Please enter password:]: message to display

above the input box)
```

Queries the user for a password.

Deprecated! Please use dialogGetPassword instead.

getPhoneType¶

```
getPhoneType()
```

Returns the device phone type.

getRingerVolume¶

```
getRingerVolume()
```

Returns the current ringer volume.

getRunningPackages¶

```
getRunningPackages()
```

Returns a list of packages running activities or services.

```
Returns:
  List of packages running activities.
```

getScreenBrightness¶

getScreenBrightness()

Returns the screen backlight brightness.

Returns:
  the current screen brightness between 0 and 255

getScreenTimeout¶

getScreenTimeout()

Returns the current screen timeout in seconds.

Returns:
  the current screen timeout in seconds.

getSimCountryIso¶

getSimCountryIso()

Returns the ISO country code equivalent for the SIM provider's country code.

getSimOperator¶

getSimOperator()

Returns the MCC+MNC (mobile country code + mobile network code) of the provider of the SIM. 5 or 6 decimal digits.

getSimOperatorName¶

getSimOperatorName()

Returns the Service Provider Name (SPN).

getSimSerialNumber¶

getSimSerialNumber()

Returns the serial number of the SIM, if applicable. Return null if it is

unavailable.

getSimState¶

getSimState()

Returns the state of the device SIM card.

getSubscriberId¶

```
getSubscriberId()
```

Returns the unique subscriber ID, for example, the IMSI for a GSM phone. Return null if it is unavailable.

getVoiceMailAlphaTag¶

```
getVoiceMailAlphaTag()
```

Retrieves the alphabetic identifier associated with the voice mail number.

getVoiceMailNumber¶

```
getVoiceMailNumber()
```

Returns the voice mail number. Return null if it is unavailable.

launch¶

```
launch(
  String className)
```

Start activity with the given class name.

log¶

```
log(
  String message)
```

Writes message to logcat.

makeToast¶

```
makeToast(
  String message)
```

Displays a short-duration Toast notification.

notify¶

```
notify(
 String title: title,
 String message)
```

Displays a notification that will be canceled when the user clicks on it.

onRingerSilent¶

```
onRingerSilent(
 String script: The path to a script to execute when the ringer mode
changes.)
```

Schedules a script for execution when the ringer volume is set to silent.

phoneCall¶

```
phoneCall(
  String uri)
```

Calls a contact/phone number by URI.

phoneCallNumber¶

```
phoneCallNumber(
  String phone number)
```

Calls a phone number.

phoneDial¶

```
phoneDial(
  String uri)
```

Dials a contact/phone number by URI.

phoneDialNumber¶

```
phoneDialNumber(
  String phone number)
```

Dials a phone number.

pick¶

```
pick(
 String uri)
```

Display content to be picked by URI (e.g. contacts)

```
Returns:
  A map of result values.
```

pickContact¶

```
pickContact()
```

Displays a list of contacts to pick from.

```
Returns:
  A map of result values.
```

pickPhone¶

```
pickPhone()
```

Displays a list of phone numbers to pick from.

Returns:
  The selected phone number.

postEvent¶

```
postEvent(
  String name,
  String data)
```

Post an event to the event queue.

pulseGeneratorGetHz¶

```
pulseGeneratorGetHz()
```

Wraps PulseGenerator getHz().

pulseGeneratorGetHzSamples¶

```
pulseGeneratorGetHzSamples()
```

Wraps PulseGenerator getHzSamples().

pulseGeneratorGetLeftPulseMs¶

```
pulseGeneratorGetLeftPulseMs()
```

Wraps PulseGenerator getLeftPulseMs().

pulseGeneratorGetLeftPulsePercent¶

```
pulseGeneratorGetLeftPulsePercent()
```

Wraps PulseGenerator getLeftPulsePercent().

pulseGeneratorGetLeftPulseSamples¶

```
pulseGeneratorGetLeftPulseSamples()
```

Wraps PulseGenerator getLeftPulseSamples().

pulseGeneratorGetRightPulseMs¶

```
pulseGeneratorGetRightPulseMs()
```

Wraps PulseGenerator getRightPulseMs().

pulseGeneratorGetRightPulsePercent¶

```
pulseGeneratorGetRightPulsePercent()
```

Wraps PulseGenerator getRightPulsePercent().

pulseGeneratorGetRightPulseSamples¶

```
pulseGeneratorGetRightPulseSamples()
```

Wraps PulseGenerator getRightPulseSamples().

pulseGeneratorIsPlaying¶

```
pulseGeneratorIsPlaying()
```

Wraps PulseGenerator isPlaying().

pulseGeneratorRun¶

```
pulseGeneratorRun()
```

Wraps PulseGenerator run().

pulseGeneratorSetHzPercent¶

```
pulseGeneratorSetHzPercent(
 Integer percent)
```

Wraps PulseGenerator setHzPercent().

pulseGeneratorSetLeftPulsePercent¶

```
pulseGeneratorSetLeftPulsePercent(
 Integer percent)
```

Wraps PulseGenerator setLeftPulsePercent().

pulseGeneratorSetRightPulsePercent¶

```
pulseGeneratorSetRightPulsePercent(
 Integer percent)
```

Wraps PulseGenerator setRightPulsePercent().

pulseGeneratorStop¶

```
pulseGeneratorStop()
```

Wraps PulseGenerator stop().

pulseGeneratorToggleInverted¶

```
pulseGeneratorToggleInverted()
```

Wraps PulseGenerator toggleInverted().

pulseGeneratorTogglePlayback¶

```
pulseGeneratorTogglePlayback()
```

Wraps PulseGenerator togglePlayback().

pulseGenteratorSetHz¶

```
pulseGenteratorSetHz(
  Float hz)
```

Wraps PulseGenerator setHz().

readBatteryData¶

```
readBatteryData()
```

Returns the most recently recorded battery data.

readLocation¶

```
readLocation()
```

Returns the current location as indicated by all available providers.

Returns:
  A map of location information by provider.

readPhoneState¶

```
readPhoneState()
```

Returns the current phone state and incoming number.

Returns:
  A Map of "state" and "incomingNumber"

readSensors¶

```
readSensors()

Returns the most recently recorded sensor data.
```

readSignalStrengths¶

```
readSignalStrengths()

Returns the current signal strengths.

Returns:
  A map of "gsm_signal_strength"

Requires API Level 7.
```

receiveEvent¶

```
receiveEvent()

Receives the most recent event (i.e. location or sensor update, etc.) and

removes it from the event buffer (which stores up to 1024 most recent
events).

Returns:
  Map of event properties.
```

recognizeSpeech¶

```
recognizeSpeech(
 String prompt[optional]: text prompt to show to the user when asking them to

speak,
 String language[optional]: language override to inform the recognizer that
it
should expect speech in a language different than the one set in the

java.util.Locale.getDefault(),
 String languageModel[optional]: informs the recognizer which speech model to

prefer (see android.speech.RecognizeIntent))

Recognizes user's speech and returns the most likely result.

Returns:
  An empty string in case the speech cannot be recongnized.
```

recorderCaptureVideo¶

```
recorderCaptureVideo(
 String targetPath,
 Double duration[optional],
 Boolean recordAudio[optional, default true])
```

```
Records video (and optionally audio) from the camera and saves it to the
given
location.
Duration specifies the maximum duration of the recording session.
If duration is not provided this method will return immediately and the

recording will only be stopped
when recorderStop is called or when a scripts exits.
Otherwise it will block for the time period equal to the duration argument.
```

recorderStartMicrophone¶

```
recorderStartMicrophone(
 String targetPath)

Records audio from the microphone and saves it to the given location.
```

recorderStop¶

```
recorderStop()

Stops a previously started recording.
```

requiredVersion¶

```
requiredVersion(
 Integer requiredVersion)

Checks if version of SL4A is greater than or equal to the specified version.
```

scanBarcode¶

```
scanBarcode()

Starts the barcode scanner.

Returns:
  A Map representation of the result Intent.
```

scheduleAbsolute¶

```
scheduleAbsolute(
 String script: script to execute,
 Double time: time of invocation, in seconds since epoch,
 Boolean wakeup[optional, default true]: whether or not to wake up the device

if asleep)

schedules one-time execution of a script
```

scheduleInexactRepeating¶

```
scheduleInexactRepeating(
 Double interval: the interval between invocations, in seconds,
 String script: the script to execute,
 Boolean wakeup[optional, default true]: whether or not to wakeup the device
if
asleep)
```

schedules a script for (inexact) regular execution - saves battery in comparison to scheduleRepeating

scheduleRelative¶

```
scheduleRelative(
 String script: script to execute,
 Double secondsFromNow: after what time to execute the script,
 Boolean wakeup[optional, default true]: whether or not to wake up the device

if asleep)
```

schedules one-time execution of a script, a given number of seconds from now

scheduleRepeating¶

```
scheduleRepeating(
 Double interval: interval between invocations, in seconds,
 String script: script to execute,
 Double firstExecutionTime[optional]: first time to execute script, in
seconds
since epoch,
 Boolean wakeup[optional, default true]: whether or not to wake up the device

if asleep)
```

scheudles a script for (exact) regular execution

search¶

```
search(
  String query)
```

Starts a search for the given query.

sendEmail¶

```
sendEmail(
 String to: A comma separated list of recipients.,
 String subject,
 String body,
 String attachmentUri[optional])
```

Launches an activity that sends an e-mail message to a given recipient.

sensorsGetAccuracy¶

```
sensorsGetAccuracy()
```

Returns the most recently received accuracy value.

sensorsGetLight¶

```
sensorsGetLight()
```

Returns the most recently received light value.

sensorsReadAccelerometer¶

```
sensorsReadAccelerometer()
```

Returns the most recently received accelerometer values.

```
Returns:
  a List of Floats [(acceleration on the) X axis, Y axis, Z axis].
```

sensorsReadMagnetometer¶

```
sensorsReadMagnetometer()
```

Returns the most recently received magnetic field values.

```
Returns:
 a List of Floats [(magnetic field value for) X axis, Y axis, Z axis].
```

sensorsReadOrientation¶

```
sensorsReadOrientation()
```

Returns the most recently received orientation values.

```
Returns:
  a List of Doubles [azimuth, pitch, roll].
```

setClipboard¶

```
setClipboard(
  String text)
```

Put text in the clipboard.

setMediaVolume¶

```
setMediaVolume(
  Integer volume)
```

Sets the media volume.

setResultBoolean¶

```
setResultBoolean(
 Integer resultCode: The result code to propagate back to the originating

activity, often RESULT_CANCELED (0) or RESULT_OK (-1),
 Boolean resultValue)
```

Sets the result of a script execution. Whenever the script APK is called via

startActivityForResult(), the resulting intent will contain SCRIPT_RESULT extra with the given value.

setResultBooleanArray¶

```
setResultBooleanArray(
 Integer resultCode: The result code to propagate back to the originating

activity, often RESULT_CANCELED (0) or RESULT_OK (-1),
 Boolean[] resultValue)
```

Sets the result of a script execution. Whenever the script APK is called via

startActivityForResult(), the resulting intent will contain SCRIPT_RESULT extra with the given value.

setResultByte¶

```
setResultByte(
 Integer resultCode: The result code to propagate back to the originating

activity, often RESULT_CANCELED (0) or RESULT_OK (-1),
 Byte resultValue)
```

Sets the result of a script execution. Whenever the script APK is called via

startActivityForResult(), the resulting intent will contain SCRIPT_RESULT extra with the given value.

setResultByteArray¶

```
setResultByteArray(
 Integer resultCode: The result code to propagate back to the originating

activity, often RESULT_CANCELED (0) or RESULT_OK (-1),
 Byte[] resultValue)
```

Sets the result of a script execution. Whenever the script APK is called via

startActivityForResult(), the resulting intent will contain SCRIPT_RESULT extra
with the given value.

setResultChar¶

```
setResultChar(
 Integer resultCode: The result code to propagate back to the originating

activity, often RESULT_CANCELED (0) or RESULT_OK (-1),
 Character resultValue)
```

Sets the result of a script execution. Whenever the script APK is called via

startActivityForResult(), the resulting intent will contain SCRIPT_RESULT extra
with the given value.

setResultCharArray¶

```
setResultCharArray(
 Integer resultCode: The result code to propagate back to the originating

activity, often RESULT_CANCELED (0) or RESULT_OK (-1),
 Character[] resultValue)
```

Sets the result of a script execution. Whenever the script APK is called via

startActivityForResult(), the resulting intent will contain SCRIPT_RESULT extra
with the given value.

setResultDouble¶

```
setResultDouble(
 Integer resultCode: The result code to propagate back to the originating

activity, often RESULT_CANCELED (0) or RESULT_OK (-1),
 Double resultValue)
```

Sets the result of a script execution. Whenever the script APK is called via

startActivityForResult(), the resulting intent will contain SCRIPT_RESULT extra
with the given value.

setResultDoubleArray¶

```
setResultDoubleArray(
 Integer resultCode: The result code to propagate back to the originating
activity, often RESULT_CANCELED (0) or RESULT_OK (-1),
 Double[] resultValue)

Sets the result of a script execution. Whenever the script APK is called via

startActivityForResult(), the resulting intent will contain SCRIPT_RESULT
extra
with the given value.
```

setResultFloat¶

```
setResultFloat(
 Integer resultCode: The result code to propagate back to the originating

activity, often RESULT_CANCELED (0) or RESULT_OK (-1),
 Float resultValue)

Sets the result of a script execution. Whenever the script APK is called via

startActivityForResult(), the resulting intent will contain SCRIPT_RESULT
extra
with the given value.
```

setResultFloatArray¶

```
setResultFloatArray(
 Integer resultCode: The result code to propagate back to the originating

activity, often RESULT_CANCELED (0) or RESULT_OK (-1),
 Float[] resultValue)

Sets the result of a script execution. Whenever the script APK is called via

startActivityForResult(), the resulting intent will contain SCRIPT_RESULT
extra
with the given value.
```

setResultInteger¶

```
setResultInteger(
 Integer resultCode: The result code to propagate back to the originating

activity, often RESULT_CANCELED (0) or RESULT_OK (-1),
 Integer resultValue)

Sets the result of a script execution. Whenever the script APK is called via

startActivityForResult(), the resulting intent will contain SCRIPT_RESULT
extra
with the given value.
```

setResultIntegerArray¶

```
setResultIntegerArray(
 Integer resultCode: The result code to propagate back to the originating

activity, often RESULT_CANCELED (0) or RESULT_OK (-1),
 Integer[] resultValue)

Sets the result of a script execution. Whenever the script APK is called via

startActivityForResult(), the resulting intent will contain SCRIPT_RESULT
extra
with the given value.
```

setResultLong¶

```
setResultLong(
 Integer resultCode: The result code to propagate back to the originating

activity, often RESULT_CANCELED (0) or RESULT_OK (-1),
 Long resultValue)

Sets the result of a script execution. Whenever the script APK is called via

startActivityForResult(), the resulting intent will contain SCRIPT_RESULT
extra
with the given value.
```

setResultLongArray¶

```
setResultLongArray(
 Integer resultCode: The result code to propagate back to the originating

activity, often RESULT_CANCELED (0) or RESULT_OK (-1),
 Long[] resultValue)

Sets the result of a script execution. Whenever the script APK is called via

startActivityForResult(), the resulting intent will contain SCRIPT_RESULT
extra
with the given value.
```

setResultSerializable¶

```
setResultSerializable(
 Integer resultCode: The result code to propagate back to the originating

activity, often RESULT_CANCELED (0) or RESULT_OK (-1),
 Serializable resultValue)

Sets the result of a script execution. Whenever the script APK is called via

startActivityForResult(), the resulting intent will contain SCRIPT_RESULT
extra
with the given value.
```

setResultShort¶

```
setResultShort(
 Integer resultCode: The result code to propagate back to the originating

activity, often RESULT_CANCELED (0) or RESULT_OK (-1),
 Short resultValue)

Sets the result of a script execution. Whenever the script APK is called via

startActivityForResult(), the resulting intent will contain SCRIPT_RESULT
extra
with the given value.
```

setResultShortArray¶

```
setResultShortArray(
 Integer resultCode: The result code to propagate back to the originating

activity, often RESULT_CANCELED (0) or RESULT_OK (-1),
 Short[] resultValue)

Sets the result of a script execution. Whenever the script APK is called via

startActivityForResult(), the resulting intent will contain SCRIPT_RESULT
extra
with the given value.
```

setResultString¶

```
setResultString(
 Integer resultCode: The result code to propagate back to the originating

activity, often RESULT_CANCELED (0) or RESULT_OK (-1),
 String resultValue)

Sets the result of a script execution. Whenever the script APK is called via

startActivityForResult(), the resulting intent will contain SCRIPT_RESULT
extra
with the given value.
```

setResultStringArray¶

```
setResultStringArray(
 Integer resultCode: The result code to propagate back to the originating

activity, often RESULT_CANCELED (0) or RESULT_OK (-1),
 String[] resultValue)

Sets the result of a script execution. Whenever the script APK is called via

startActivityForResult(), the resulting intent will contain SCRIPT_RESULT
```

```
extra
with the given value.
```

setRingerVolume¶

```
setRingerVolume(
  Integer volume)

Sets the ringer volume.
```

setScreenBrightness¶

```
setScreenBrightness(
 Integer value: brightness value between 0 and 255)

Sets the the screen backlight brightness.

Returns:
  the original screen brightness.
```

setScreenTimeout¶

```
setScreenTimeout(
 Integer value)

Sets the screen timeout to this number of seconds.

Returns:
  The original screen timeout.
```

smsDeleteMessage¶

```
smsDeleteMessage(
 Integer id)

Deletes a message.

Returns:
  True if the message was deleted
```

smsGetAttributes¶

```
smsGetAttributes()

Returns a List of all possible message attributes.
```

smsGetMessageById¶

```
smsGetMessageById(
 Integer id: message ID,
 JSONArray attributes[optional])
```

```
Returns message attributes.
```

smsGetMessageCount¶

```
smsGetMessageCount(
 Boolean unreadOnly,
 String folder[optional, default inbox])

Returns the number of messages.
```

smsGetMessageIds¶

```
smsGetMessageIds(
 Boolean unreadOnly,
 String folder[optional, default inbox])

Returns a List of all message IDs.
```

smsGetMessages¶

```
smsGetMessages(
 Boolean unreadOnly,
 String folder[optional, default inbox],
 JSONArray attributes[optional])

Returns a List of all messages.

Returns:
  a List of messages as Maps
```

smsMarkMessageRead¶

```
smsMarkMessageRead(
 JSONArray ids: List of message IDs to mark as read.,
 Boolean read)

Marks messages as read.

Returns:
  number of messages marked read
```

smsSend¶

```
smsSend(
 String destinationAddress: typically a phone number,
 String text)

Sends an SMS.
```

startActivity¶

```
startActivity(
 String action,
 String uri[optional],
 String type[optional]: MIME type/subtype of the URI,
 JSONObject extras[optional]: a Map of extras to add to the Intent,
 Boolean wait[optional]: block until the user exits the started activity)
```

Starts an activity.

startActivityForResult¶

```
startActivityForResult(
 String action,
 String uri[optional],
 String type[optional]: MIME type/subtype of the URI,
 JSONObject extras[optional]: a Map of extras to add to the Intent)
```

Starts an activity and returns the result.

Returns:
  A Map representation of the result Intent.

startInteractiveVideoRecording¶

```
startInteractiveVideoRecording(
 String path)
```

Starts the video capture application to record a video and saves it to the specified path.

startLocating¶

```
startLocating(
 Integer minDistance[optional, default 60000]: minimum time between updates
in
milliseconds,
 Integer minUpdateDistance[optional, default 30]: minimum distance between
updates in meters)
```

Starts collecting location data.

Generates "location" events.

startSensing¶

```
startSensing(
 Integer sampleSize[optional, default 5]: number of samples for calculating
average readings)
```

Starts recording sensor data to be available for polling.

Generates "sensors" events.

startTrackingPhoneState¶

```
startTrackingPhoneState()
```

Starts tracking phone state.

Generates "phone" events.

startTrackingSignalStrengths¶

```
startTrackingSignalStrengths()
```

Starts tracking signal strengths.

Generates "signal_strengths" events.

Requires API Level 7.

stopLocating¶

```
stopLocating()
```

Stops collecting location data.

stopSensing¶

```
stopSensing()
```

Stops collecting sensor data.

stopTrackingPhoneState¶

```
stopTrackingPhoneState()
```

Stops tracking phone state.

stopTrackingSignalStrengths¶

```
stopTrackingSignalStrengths()
```

Stops tracking signal strength.

Requires API Level 7.

toggleAirplaneMode¶

```
toggleAirplaneMode(
 Boolean enabled[optional])
```

```
Toggles airplane mode on and off.

Returns:
  True if airplane mode is enabled.
```

toggleBluetoothState¶

```
toggleBluetoothState(
 Boolean enabled[optional],
 Boolean prompt[optional, default true]: Prompt the user to confirm changing

the Bluetooth state.)

Toggle Bluetooth on and off.

Returns:
  True if Bluetooth is enabled.

Requires API Level 5.
```

toggleRingerSilentMode¶

```
toggleRingerSilentMode(
 Boolean enabled[optional])

Toggles ringer silent mode on and off.

Returns:
  True if ringer silent mode is enabled.
```

toggleWifiState¶

```
toggleWifiState(
 Boolean enabled[optional])

Toggle Wifi on and off.

Returns:
  True if Wifi is enabled.
```

ttsIsSpeaking¶

```
ttsIsSpeaking()

Returns True if speech is currently in progress.

Requires API Level 4.
```

ttsSpeak¶

```
ttsSpeak(
  String message)
```

Speaks the provided message via TTS.

Requires API Level 4.

vibrate¶

```
vibrate(
 Integer duration[optional, default 300]: duration in milliseconds)
```

Vibrates the phone or a specified duration in milliseconds.

view¶

```
view(
 String uri,
 String type[optional]: MIME type/subtype of the URI,
 JSONObject extras[optional]: a Map of extras to add to the Intent)
```

Start activity with view action by URI (i.e. browser, contacts, etc.).

viewContacts¶

```
viewContacts()
```

Opens the list of contacts.

viewHtml¶

```
viewHtml(
 String path: the path to the HTML file)
```

Opens the browser to display a local HTML file.

viewMap¶

```
viewMap(
 String query, e.g. pizza, 123 My Street)
```

Opens a map search for query (e.g. pizza, 123 My Street).

waitForEvent¶

```
waitForEvent(
 String eventName,
 Integer timeout[optional]: the maximum time to wait)
```

Blocks until an event with the supplied name occurs. The returned event is not removed from the buffer.

```
Returns:
  Map of event properties.
```

wakeLockAcquireBright¶

```
wakeLockAcquireBright()
```

Acquires a bright wake lock (CPU on, screen bright).

wakeLockAcquireDim¶

```
wakeLockAcquireDim()
```

Acquires a dim wake lock (CPU on, screen dim).

wakeLockAcquireFull¶

```
wakeLockAcquireFull()
```

Acquires a full wake lock (CPU on, screen bright, keyboard bright).

wakeLockAcquirePartial¶

```
wakeLockAcquirePartial()
```

Acquires a partial wake lock (CPU on).

wakeLockRelease¶

```
wakeLockRelease()
```

Releases the wake lock.

webViewShow¶

```
webViewShow(
 String url,
 Boolean wait[optional]: block until the user exits the WebView)
```

Display a WebView with the given URL.

wifiDisconnect¶

```
wifiDisconnect()
```

Disconnects from the currently active access point.

```
Returns:
  True if the operation succeeded.
```

wifiGetConnectionInfo¶

```
wifiGetConnectionInfo()

Returns information about the currently active access point.
```

wifiGetScanResults¶

```
wifiGetScanResults()

Returns the list of access points found during the most recent Wifi scan.
```

wifiLockAcquireFull¶

```
wifiLockAcquireFull()

Acquires a full Wifi lock.
```

wifiLockAcquireScanOnly¶

```
wifiLockAcquireScanOnly()

Acquires a scan only Wifi lock.
```

wifiLockRelease¶

```
wifiLockRelease()

Releases a previously acquired Wifi lock.
```

wifiReassociate¶

```
wifiReassociate()

Reassociates with the currently active access point.

Returns:
  True if the operation succeeded.
```

wifiReconnect¶

```
wifiReconnect()

Reconnects to the currently active access point.

Returns:
  True if the operation succeeded.
```

wifiStartScan¶

wifiStartScan()

Starts a scan for Wifi access points.

Returns:
  True if the scan was initiated successfully.