com.ilumi.sdk

Class IlumiSDK

java.lang.Object
    com.ilumi.sdk.IlumiSDK

```
public class IlumiSDK
extends java.lang.Object
```

Created by ashish on 9/1/18.

Constructor Summary

Constructors

Constructor and Description
`IlumiSDK()`

Method Summary

All Methods | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method and Description
void	`commissionWithId(byte[] macAddress, int groupID, int nodeID, com.ilumi.sdk.callbacks.IsSuccessCallBack Completion)` Conduct the comissioning (pairing) process After comission, controller becomes the owner of iLumi device, no other controller can change configuraton on iLumi device until a
boolean	`connectIlumi(byte[] macAddress, com.ilumi.sdk.callbacks.IsSuccessCallBack callback)` Try to connect with ilumi.
boolean	`disconnectIlumi(byte[] macAddress)` Try to disconnect from ilumi
int	`getNetworkKey()` Get network key that has been set to SDK The value is not saved insdie SDK and will be lost whenever Application /SDK is restarted or reset
void	`injectAdvertisementPacket(byte[] macAddress, byte[] data)` Inject Group broadcast message to ilumi
boolean	`isIlumiConnected(byte[] macAddress)`

	Whether the ilumi currently connected with smartphone or through mesh connection
int	**meshIsConnected**(byte[] macAddress) Check how ilumi is connected.
void	**meshSendBroadcastMsg**(byte[] macAddress, byte[] data, int groupNodeId) Send broadcast message to a groupID or nodeID
void	**meshSendConnectedMsg**(byte[] macAddress, byte[] data, com.ilumi.sdk.callbacks.IsSuccessCallBack callback) Send connection based message to a node
void	**retrieveIlumis**() Start search any paired ilumi
void	**setNetworkKey**(int networkKey) Set 32bit netwok key which will be used for pairing and sending command between SDK and ilumi
void	**startSearchIlumi**(boolean pairing) Start search unpaired ilumi
void	**stopSearchIlumi**() Stop search ilumi

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

IlumiSDK

```
public IlumiSDK()
```

Method Detail

retrievellumis

```
public void retrieveIlumis()
```

Start search any paired ilumi

didFindiLumi call back will return founded paired ilumi

commissionWithId

```
public void commissionWithId(byte[] macAddress,
                             int groupID,
                             int nodeID,
                             com.ilumi.sdk.callbacks.IsSuccessCallBack Completion)
```

Conduct the comissioning (pairing) process After comission, controller becomes the owner of iLumi device, no other controller can change configuraton on iLumi device until a

Parameters:

`macAddress - MAC address of target iLumi device`

`groupID - Two byte current node ID number`

`nodeID - Two byte current group ID number`

`Completion - Callback to indicate whether API call is successful`

connectIlumi

```
public boolean connectIlumi(byte[] macAddress,
                            com.ilumi.sdk.callbacks.IsSuccessCallBack callback)
```

Try to connect with ilumi. Check if ilumi is already connected before calling this method. If ilumi is already connected, this method will fail and there won't be any callback to indicate that.

Parameters:

`macAddress - MAC address of target iLumi device`

Returns:

`Return true if the connect request has been send successfully`

getNetworkKey

```
public int getNetworkKey()
```

Get network key that has been set to SDK The value is not saved insdie SDK and will be lost whenever Application /SDK is restarted or reset

setNetworkKey

```
public void setNetworkKey(int networkKey)
```

Set 32bit netwok key which will be used for pairing and sending command between SDK and ilumi

Parameters:

`networkKey - set network key`

startSearchIlumi

```
public void startSearchIlumi(boolean pairing)
```

Start search unpaired ilumi

didFindiLumi call back will return founded unpaired ilumi

stopSearchIlumi

```
public void stopSearchIlumi()
```

Stop search ilumi

didFindiLumi call back will return founded unpaired ilumi

disconnectIlumi

```
public boolean disconnectIlumi(byte[] macAddress)
```

Try to disconnect from ilumi

Parameters:

`macAddress - MAC address of target iLumi device`

Returns:

`Return true if the disconnect request has been send successfully`

isIlumiConnected

```
public boolean isIlumiConnected(byte[] macAddress)
```

Whether the ilumi currently connected with smartphone or through mesh connection

Parameters:

`macAddress - MAC address of target iLumi device`

Returns:

`Return true if the ilumi is connected`

meshIsConnected

```
public int meshIsConnected(byte[] macAddress)
```

Check how ilumi is connected.

Parameters:

`macAddress -`

Returns:

`0 if not connected, 1 if directly connected or number >1 if connected via mesh. This number will represent the hop count`

injectAdvertisementPacket

```
public void injectAdvertisementPacket(byte[] macAddress,
                                      byte[] data)
```

Inject Group broadcast message to ilumi

Parameters:

`macAddress -`

`data -`

meshSendBroadcastMsg

```
public void meshSendBroadcastMsg(byte[] macAddress,
                                 byte[] data,
                                 int groupNodeId)
```

Send broadcast message to a groupID or nodeID

Parameters:

`macAddress - MAC address of proxy node device`

`data - message to be sent`

`groupNodeId - Each group ID or node ID of target`

meshSendConnectedMsg

```
public void meshSendConnectedMsg(byte[] macAddress,
                                 byte[] data,
                                 com.ilumi.sdk.callbacks.IsSuccessCallBack callback)
```

Send connection based message to a node

Parameters:

`macAddress - MAC address of target node device`

```
data - message to be sent

callback -
```