BluetoothPositioning

Generato da Doxygen 1.8.12

Indice

1 Indice dei namespace

1.1 Package

Questi sono i package e una loro breve descrizione (se disponibile):

2 Indice della gerarchia

2.1 Gerarchia delle classi

Questo elenco di ereditarietà è ordinato approssimativamente, ma non completamente, in ordine alfabetico:

3 Indice dei tipi composti

3.1 Elenco dei tipi composti

Queste sono le classi, le struct, le union e le interfacce con una loro breve descrizione:

4 Indice dei file

4.1 Elenco dei file

Questo è un elenco di tutti i file con una loro breve descrizione:

5 Documentazione dei namespace

5.1 Package it

Package

- package unibo

5.2 Package it.unibo

Package

- package torsello

5.3 Package it.unibo.torsello

Package

- package bluetoothpositioning

5.4 Package it.unibo.torsello.bluetoothpositioning

Package

- package activities
- package adapter
- package configuration
- package constant
- package distanceEstimation
- package examplesCamera
- package extra
- package fragment
- package kalmanFilter
- package model
- package observables
- package util

5.5 Package it.unibo.torsello.bluetoothpositioning.activities

Composti

- class ApplicationActivity
- class MainActivity

5.6 Package it.unibo.torsello.bluetoothpositioning.adapter

Composti

- class DeviceCardViewAdapter
- class StatePagerAdapter

5.7 Package it.unibo.torsello.bluetoothpositioning.configuration

Composti

- class MyArmaRssiFilter

5.8 Package it.unibo.torsello.bluetoothpositioning.constant

Composti

- class DeviceConstants
- class KFilterConstansts
- class SettingConstants

5.9 Package it.unibo.torsello.bluetoothpositioning.distanceEstimation

Composti

- class Estimation

5.10 Package it.unibo.torsello.bluetoothpositioning.examplesCamera

Composti

- class CamTestFragment
- class Preview
- class SaveImageTask

5.11 Package it.unibo.torsello.bluetoothpositioning.extra

Composti

- class FABBehavior

5.12 Package it.unibo.torsello.bluetoothpositioning.fragment

Package

- package devicesObservers
- package oldFragment
- package usbObservers

Composti

- class CameraFragment
- class DeviceDetailFragment
- class DeviceDetailInner2Fragmet
- class SettingsFragment

5.13 Package it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers

Composti

- class DeviceChartFragment
- class DeviceDetailInner1Fragment
- class DeviceListFragment

5.14 Package it.unibo.torsello.bluetoothpositioning.fragment.oldFragment

Composti

- class CompassFragment
- class CompassMagnoFragment
- class CountPassFragment

5.15 Package it.unibo.torsello.bluetoothpositioning.fragment.usbObservers

Composti

- class UsbMeasurementFragment

5.16 Package it.unibo.torsello.bluetoothpositioning.kalmanFilter

Composti

- class KalmanFilter
- class KFBuilder

5.17 Package it.unibo.torsello.bluetoothpositioning.model

Composti

- class Device

5.18 Package it.unibo.torsello.bluetoothpositioning.observables

Composti

- class DeviceObservable
- class UsbMeasurementObservable

5.19 Package it.unibo.torsello.bluetoothpositioning.util

Composti

- class CameraUtil
- class ChartUtil
- class UsbUtil
- class WalkDetectionUtil

6 Documentazione delle classi

6.1 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity

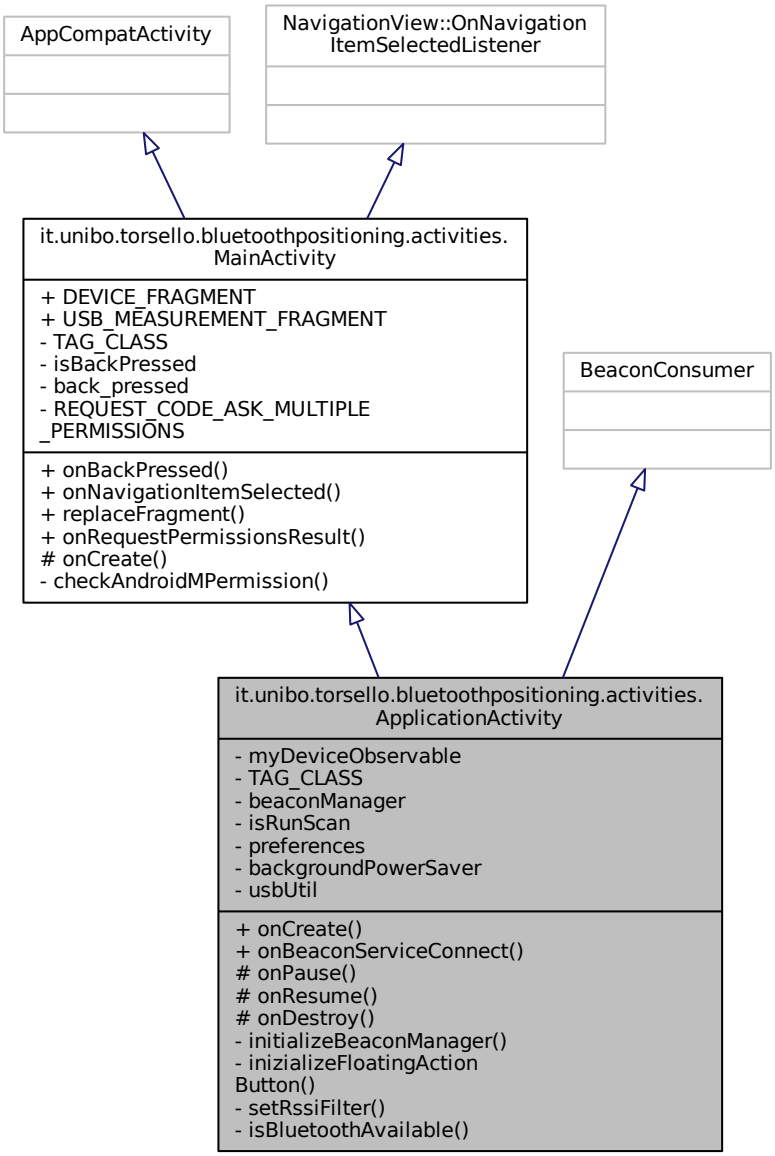
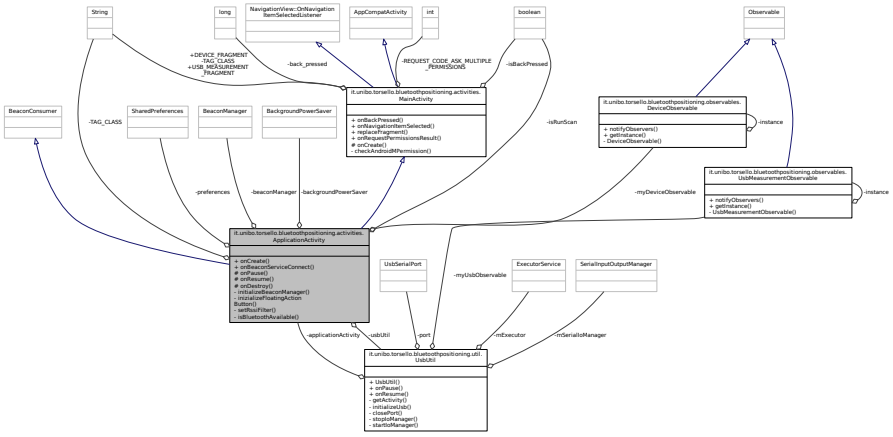


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity:



Membri pubblici

- void onCreate (Bundle savedInstanceState)
- void onBeaconServiceConnect ()

Membri protetti

- void onPause ()
- void onResume ()
- void onDestroy ()

Membri privati

- void initializeBeaconManager ()
- void inizializeFloatingActionButton ()
- void setRssiFilter ()
- boolean isBluetoothAvailable ()

Attributi privati

- DeviceObservable myDeviceObservable
- final String TAG_CLASS = getClass().getSimpleName()
- BeaconManager beaconManager
- boolean isRunScan = false
- SharedPreferences preferences
- BackgroundPowerSaver backgroundPowerSaver
- UsbUtil usbUtil

Altri membri ereditati

6.1.1 Descrizione dettagliata

Created by Federico Torsello. federico.torsello@studio.unibo.it

6.1.2 Documentazione delle funzioni membro

6.1.2.1 initializeBeaconManager()

void it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity.initializeBeacon↩
Manager ( ) [private]

```
66                                          {
67          beaconManager = BeaconManager.getInstanceForApplication(this);
68          beaconManager.bind(this);
69
70          // Save battery whenever the application is not visible.
71          // This reduces bluetooth power usage by about 60%
72          backgroundPowerSaver = new BackgroundPowerSaver(this);
73
74          Log.i("AltBeacon filter used:", BeaconManager.getRssiFilterImplClass().getSimpleName());
75
76          // for finding different type of beacon,
77          beaconManager.getBeaconParsers().clear();
78
79          // Alt beacon
80          beaconManager.getBeaconParsers().add(new BeaconParser()
81                  .setBeaconLayout(BeaconParser.ALTBEACON_LAYOUT));
82          // Detect the main identifier (UID) frame:
83          beaconManager.getBeaconParsers().add(new BeaconParser()
84                  .setBeaconLayout(BeaconParser.EDDYSTONE_UID_LAYOUT));
85          // Detect the telemetry (TLM) frame:
86          beaconManager.getBeaconParsers().add(new BeaconParser()
87                  .setBeaconLayout(BeaconParser.EDDYSTONE_TLM_LAYOUT));
88          // Detect the URL frame:
89          beaconManager.getBeaconParsers().add(new BeaconParser()
90                  .setBeaconLayout(BeaconParser.EDDYSTONE_URL_LAYOUT));
91          // Standard Apple iBeacon
92          beaconManager.getBeaconParsers().add(new BeaconParser()
93                  .setBeaconLayout(DeviceConstants.APPLE_BEACON_LAYOUT));
94          // Estimote Nearable
95          beaconManager.getBeaconParsers().add(new BeaconParser()
96                  .setBeaconLayout(DeviceConstants.ESTIMOTE_NEARABLE_LAYOUT));
97
98          beaconManager.setForegroundScanPeriod(250L);
99          beaconManager.setForegroundBetweenScanPeriod(0L);
100          beaconManager.setBackgroundScanPeriod(250L);
101          beaconManager.setBackgroundBetweenScanPeriod(0L);
102
103          beaconManager.setMaxTrackingAge(1000);
104      }
```

6.1.2.2 inizializeFloatingActionButton()

void it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity.inizializeFloating↩
ActionButton ( ) [private]

```
106                                                 {
107          final FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
108          assert fab != null;
109          Snackbar.make(fab, R.string.snackBar_start_scanning, Snackbar.LENGTH_LONG).show();
110          fab.setOnClickListener(new View.OnClickListener() {
111              @Override
112              public void onClick(View view) {
113
114                  if (isBluetoothAvailable()) {
115
116                      isRunScan = !isRunScan;
117                      Region region = new Region("RegionId", null, null, null);
118
119                      if (isRunScan) {
120                          fab.setImageResource(R.drawable.ic_bluetooth_searching_white_24dp);
121                          try {
122                              beaconManager.startRangingBeaconsInRegion(region);
123                          } catch (RemoteException e) {
124                              e.printStackTrace();
125                          }
126                          Snackbar.make(view, R.string.snackBar_scanning_enabled,
127                                  Snackbar.LENGTH_SHORT).show();
128                      } else {
```

```
129                         fab.setImageResource(R.drawable.ic_bluetooth_white_24dp);
130                         try {
131                             beaconManager.stopRangingBeaconsInRegion(region);
132                         } catch (RemoteException e) {
133                             e.printStackTrace();
134                         }
135                         Snackbar.make(view, R.string.snackBar_scanning_disabled,
136                                 Snackbar.LENGTH_INDEFINITE).show();
137                     }
138                 }
139             }
140         });
141     }
```

6.1.2.3 isBluetoothAvailable()

boolean it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity.isBluetooth↩
Available ( ) [private]

```
248                                         {
249
250         try {
251             if (!beaconManager.checkAvailability()) {
252
253                 final FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
254                 assert fab != null;
255
256                 new AlertDialog.Builder(this)
257                         .setTitle(R.string.dialog_bluetooth_title)
258                         .setMessage(R.string.dialog_bluetooth_text)
259                         .setPositiveButton(android.R.string.ok, null)
260                         .setOnDismissListener(new DialogInterface.OnDismissListener() {
261                             @Override
262                             public void onDismiss(DialogInterface dialog) {
263                                 fab.setImageResource(R.drawable.ic_bluetooth_white_24dp);
264                                 BluetoothAdapter.getDefaultAdapter().enable();
265                             }
266                         }).show();
267                 fab.setImageResource(R.drawable.ic_bluetooth_disabled_black_24dp);
268                 return false;
269             }
270         } catch (RuntimeException e) {
271             e.getStackTrace();
272         }
273         return true;
274     }
```

6.1.2.4 onBeaconServiceConnect()

void it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity.onBeaconService↩
Connect ( )

```
144                                       {
145
146         try {
147             beaconManager.updateScanPeriods();
148         } catch (RemoteException e) {
149             e.printStackTrace();
150         }
151
152         final List<Device> deviceList = new ArrayList<>();
153
154         beaconManager.addRangeNotifier(new RangeNotifier() {
155             @Override
156             public void didRangeBeaconsInRegion(final Collection<Beacon> beacons, Region region) {
157
158                 setRssiFilter();
159
160                 for (Beacon b : beacons) {
161
162                     // take from the list the device
163                     Device device = DeviceConstants.DEVICE_MAP.get(b.getBluetoothAddress());
164
165                     if (device != null) { // useful only if DEVICE_MAP is empty
166                         double processNoise = preferences.getFloat(SettingConstants.
```

```
      KALMAN_NOISE_VALUE, 0);
167                       device.setBeacon(b);
168                       device.updateDistance(processNoise);
169
170                       if (!deviceList.contains(device)) {
171                           deviceList.add(device);
172                       }
173                   }
174               }
175
176               new Thread(new Runnable() {
177                   @Override
178                   public void run() {
179                       runOnUiThread(new Runnable() {
180
181                           @Override
182                           public void run() {
183                               myDeviceObservable.
      notifyObservers(deviceList);
184                           }
185                       });
186                   }
187               }).start();
188           }
189       });
190    }
```

6.1.2.5 onCreate()

```
void it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity.onCreate (
            Bundle savedInstanceState )
```

```
52                                                  {
53        super.onCreate(savedInstanceState);
54
55        myDeviceObservable = DeviceObservable.getInstance();
56
57        preferences = getSharedPreferences(SettingConstants.SETTINGS_PREFERENCES, 0);
58
59        usbUtil = new UsbUtil(this);
60
61        initializeBeaconManager();
62
63        inizializeFloatingActionButton();
64    }
```

6.1.2.6 onDestroy()

```
void it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity.onDestroy ( )  [protected]
```

```
239                                 {
240        if (beaconManager.isBound(this)) {
241            beaconManager.unbind(this);
242            backgroundPowerSaver.onActivityDestroyed(this);
243        }
244
245        super.onDestroy();
246    }
```

6.1.2.7 onPause()

```
void it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity.onPause ( )  [protected]
```

```
212                               {
213        if (beaconManager.isBound(this)) {
214            beaconManager.setBackgroundMode(true);
215            backgroundPowerSaver.onActivityPaused(this);
216        }
217
218        usbUtil.onPause();
219
220        super.onPause();
221    }
```

6.1.2.8 onResume()

void it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity.onResume ( ) [protected]

```
224                                {
225         super.onResume();
226 
227         if (beaconManager.isBound(this)) {
228             beaconManager.setBackgroundMode(false);
229             backgroundPowerSaver.onActivityResumed(this);
230         }
231 
232         isBluetoothAvailable();
233 
234         usbUtil.onResume();
235 
236     }
```

6.1.2.9 setRssiFilter()

void it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity.setRssiFilter ( ) [private]

```
192                                  {
193 
194         int sorting = preferences.getInt(SettingConstants.FILTER_RSSI, 0);
195         switch (sorting) {
196             case 0:
197             case R.id.radioButton_no_rssi_filtering:
198                 MyArmaRssiFilter.enableArmaFilter(false);
199                 BeaconManager.setRssiFilterImplClass(MyArmaRssiFilter.class);
200                 break;
201             case R.id.radioButton_arma_rssi_filter:
202                 MyArmaRssiFilter.enableArmaFilter(true);
203                 BeaconManager.setRssiFilterImplClass(MyArmaRssiFilter.class);
204                 break;
205             case R.id.radioButton_average_rssi_filter:
206                 BeaconManager.setRssiFilterImplClass(RunningAverageRssiFilter.class);
207                 break;
208         }
209     }
```

6.1.3 Documentazione dei membri dato

6.1.3.1 backgroundPowerSaver

BackgroundPowerSaver it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity.↩
backgroundPowerSaver [private]

6.1.3.2 beaconManager

BeaconManager it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity.beacon↩
Manager [private]

6.1.3.3 isRunScan

boolean it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity.isRunScan =
false [private]

6.1.3.4 myDeviceObservable

DeviceObservable it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity.my↩
DeviceObservable [private]

6.1.3.5 preferences

```
SharedPreferences it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity.↩
preferences  [private]
```

6.1.3.6 TAG_CLASS

```
final String it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity.TAG_CLASS =
getClass().getSimpleName()  [private]
```

6.1.3.7 usbUtil

```
UsbUtil it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity.usbUtil  [private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- ApplicationActivity.java

6.2 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.fragment.CameraFragment

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.fragment.CameraFragment

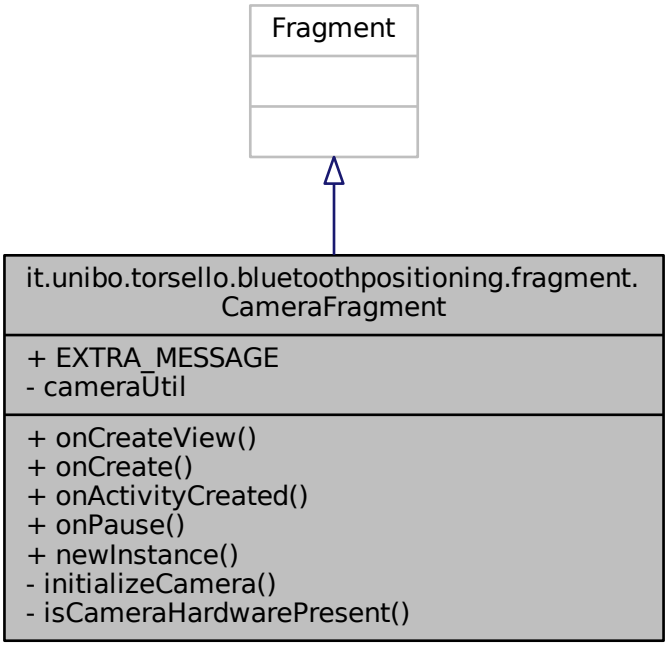
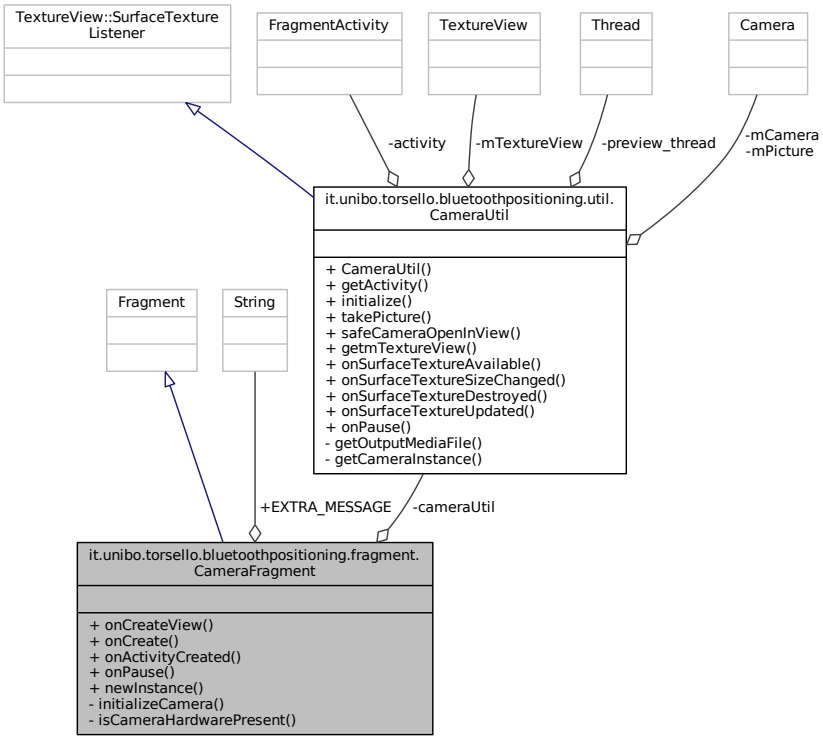


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.CameraFragment:



Membri pubblici

- View onCreateView (LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
- void onCreate (@Nullable Bundle savedInstanceState)
- void onActivityCreated (@Nullable Bundle savedInstanceState)
- void onPause ()

Membri pubblici statici

- static CameraFragment newInstance ()

Attributi pubblici statici

- static final String EXTRA_MESSAGE = "EXTRA_MESSAGE"

Membri privati

- void initializeCamera (View root)
- boolean isCameraHardwarePresent ()

Attributi privati

- CameraUtil cameraUtil

6.2.1 Descrizione dettagliata

Created by federico on 28/09/16.

6.2.2 Documentazione delle funzioni membro

6.2.2.1 initializeCamera()

```
void it.unibo.torsello.bluetoothpositioning.fragment.CameraFragment.initializeCamera (
            View root )  [private]
```

```
44                                          {
45
46         if (cameraUtil != null) {
47             final TextureView mTextureView = cameraUtil.
    getmTextureView();
48
49             // programmatically add camera preview
50             FrameLayout preview = (FrameLayout) root.findViewById(R.id.camera_preview);
51             preview.addView(mTextureView);
52             preview.setOnClickListener(new View.OnClickListener() {
53                 @Override
54                 public void onClick(View v) {
55                     // Restart the camera preview.
56                     cameraUtil.safeCameraOpenInView(mTextureView.
    getSurfaceTexture());
57                 }
58             });
59         } else {
60             Toast.makeText(getActivity(), "No camera onCameraListener this device", Toast.LENGTH_LONG)
61                     .show();
62         }
63     }
```

6.2.2.2 isCameraHardwarePresent()

```
boolean it.unibo.torsello.bluetoothpositioning.fragment.CameraFragment.isCameraHardwarePresent
( )  [private]
```

Check if this device has a camera

```
96                                          {
97         return getActivity().getPackageManager()
98                 .hasSystemFeature(PackageManager.FEATURE_CAMERA);
99     }
```

6.2.2.3 newInstance()

```
static CameraFragment it.unibo.torsello.bluetoothpositioning.fragment.CameraFragment.new↩
Instance ( )  [static]
```

```
27                                          {
28         CameraFragment fragment = new CameraFragment();
29         Bundle args = new Bundle();
30         args.putString(EXTRA_MESSAGE, "Camera");
31         fragment.setArguments(args);
32         return fragment;
33     }
```

6.2.2.4 onActivityCreated()

```
void it.unibo.torsello.bluetoothpositioning.fragment.CameraFragment.onActivityCreated (
            @Nullable Bundle savedInstanceState )
```

```
75                                                                          {
76          super.onActivityCreated(savedInstanceState);
77
78          getActivity().findViewById(R.id.fab_camera).setOnClickListener(new View.OnClickListener() {
79              @Override
80              public void onClick(View v) {
81                  cameraUtil.takePicture();
82              }
83          });
84
85      }
```

6.2.2.5 onCreate()

```
void it.unibo.torsello.bluetoothpositioning.fragment.CameraFragment.onCreate (
            @Nullable Bundle savedInstanceState )
```

```
66                                                          {
67          super.onCreate(savedInstanceState);
68
69          cameraUtil = new CameraUtil(getActivity());
70          cameraUtil.initialize();
71
72      }
```

6.2.2.6 onCreateView()

```
View it.unibo.torsello.bluetoothpositioning.fragment.CameraFragment.onCreateView (
            LayoutInflater inflater,
            ViewGroup container,
            Bundle savedInstanceState )
```

```
36                                                                                                  {
37          View root = inflater.inflate(R.layout.fragment_camera, container, false);
38
39          initializeCamera(root);
40
41          return root;
42      }
```

6.2.2.7 onPause()

```
void it.unibo.torsello.bluetoothpositioning.fragment.CameraFragment.onPause ( )
```

```
88                          {
89          cameraUtil.onPause();
90          super.onPause();
91      }
```

6.2.3 Documentazione dei membri dato

6.2.3.1 cameraUtil

```
CameraUtil it.unibo.torsello.bluetoothpositioning.fragment.CameraFragment.cameraUtil  [private]
```

6.2.3.2 EXTRA_MESSAGE

```
final String it.unibo.torsello.bluetoothpositioning.fragment.CameraFragment.EXTRA_MESSAGE =
"EXTRA_MESSAGE"  [static]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- CameraFragment.java

6.3 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.util.CameraUtil

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.util.CameraUtil

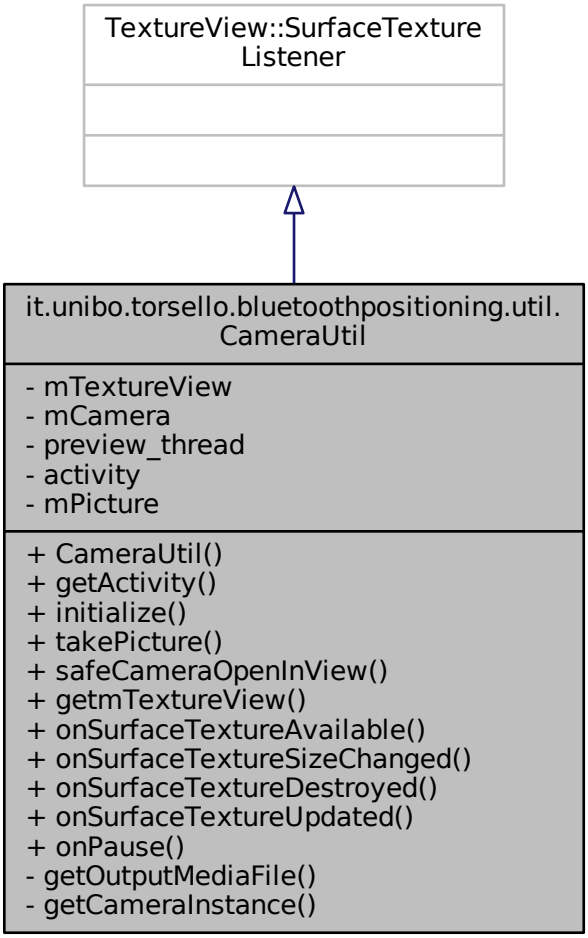
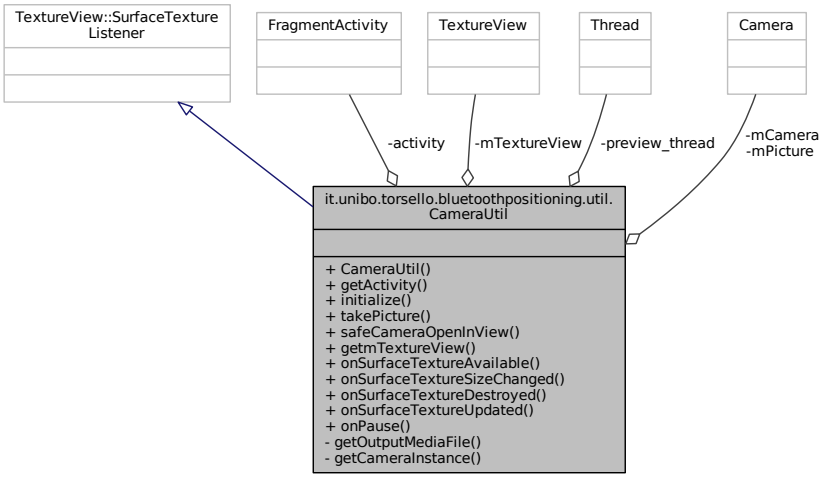


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.util.CameraUtil:



Membri pubblici

- CameraUtil (FragmentActivity fragmentActivity)
- FragmentActivity getActivity ()
- void initialize ()
- void takePicture ()
- void safeCameraOpenInView (SurfaceTexture surface)
- TextureView getmTextureView ()
- void onSurfaceTextureAvailable (final SurfaceTexture surface, int width, int height)
- void onSurfaceTextureSizeChanged (SurfaceTexture surface, int width, int height)
- boolean onSurfaceTextureDestroyed (SurfaceTexture surface)
- void onSurfaceTextureUpdated (SurfaceTexture surface)
- void onPause ()

Membri privati

- File getOutputMediaFile ()

Membri privati statici

- static Camera getCameraInstance ()

Attributi privati

- TextureView mTextureView
- Camera mCamera
- Thread preview_thread
- FragmentActivity activity
- Camera.PictureCallback mPicture

6.3.1 Descrizione dettagliata

Created by Federico Torsello. federico.torsello@studio.unibo.it

6.3.2 Documentazione dei costruttori e dei distruttori

6.3.2.1 CameraUtil()

```
it.unibo.torsello.bluetoothpositioning.util.CameraUtil.CameraUtil (
            FragmentActivity fragmentActivity )
```

```
52                                                      {
53         this.activity = fragmentActivity;
54     }
```

6.3.3 Documentazione delle funzioni membro

6.3.3.1 getActivity()

```
FragmentActivity it.unibo.torsello.bluetoothpositioning.util.CameraUtil.getActivity ( )
```

```
56                                          {
57         return activity;
58     }
```

6.3.3.2 getCameraInstance()

```
static Camera it.unibo.torsello.bluetoothpositioning.util.CameraUtil.getCameraInstance ( )
[static], [private]
```

A safe way to get an instance of the CameraUtil object.

```
63                                              {
64
65         Camera c = null;
66
67         try {
68             c = Camera.open(); // attempt to get a CameraUtil instance
69         } catch (RuntimeException e) {
70             // CameraUtil is not available (in use or does not exist)
71             e.getStackTrace();
72         }
73
74         return c; // returns null if camera is unavailable
75     }
```

6.3.3.3 getmTextureView()

```
TextureView it.unibo.torsello.bluetoothpositioning.util.CameraUtil.getmTextureView ( )
```

```
157                                         {
158         return mTextureView;
159     }
```

6.3.3.4 getOutputMediaFile()

```
File it.unibo.torsello.bluetoothpositioning.util.CameraUtil.getOutputMediaFile ( )  [private]
```

Used to return the camera File output.

Restituisce

```
128                                     {
129
130         File mediaStorageDir = new File(Environment.getExternalStoragePublicDirectory(
131                 Environment.DIRECTORY_PICTURES), getActivity().getString(R.string.app_name));
132
133         if (!mediaStorageDir.exists()) {
134             if (!mediaStorageDir.mkdirs()) {
135                 Log.i("CameraUtil Guide", "Required media storage does not exist");
136                 return null;
137             }
138         }
139
140         // Create a media file name
141         String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss").format(new Date());
142         File mediaFile = new File(mediaStorageDir.getPath() + File.separator +
143                 "IMG_" + timeStamp + ".jpg");
144
145         new AlertDialog.Builder(getActivity())
146                 .setTitle("Success!")
147                 .setMessage("Your picture has been saved!")
148                 .setPositiveButton(android.R.string.ok, new DialogInterface.OnClickListener() {
149                     public void onClick(DialogInterface dialog, int id) {
150                         dialog.dismiss();
151                     }
152                 }).show();
153
154         return mediaFile;
155     }
```

6.3.3.5 initialize()

```
void it.unibo.torsello.bluetoothpositioning.util.CameraUtil.initialize ( )
```

```
77                              {
78         if (mCamera == null) {
79             mCamera = getCameraInstance();
80         }
81
82         mTextureView = new TextureView(getActivity());
83         mTextureView.setSurfaceTextureListener(this);
84
85     }
```

6.3.3.6 onPause()

```
void it.unibo.torsello.bluetoothpositioning.util.CameraUtil.onPause ( )
```

```
195                             {
196         if (mCamera != null) {
197             if (preview_thread != null && !preview_thread.isInterrupted()) {
198                 preview_thread.interrupt();
199             }
200             mCamera.stopPreview();
201             mCamera.release();
202             mCamera = null;
203         }
204     }
```

6.3.3.7 onSurfaceTextureAvailable()

```
void it.unibo.torsello.bluetoothpositioning.util.CameraUtil.onSurfaceTextureAvailable (
              final SurfaceTexture surface,
              int width,
              int height )
```

```
162                                                                                                 {
163
164         if (surface == null) {
165             // preview surface does not exist
166             return;
167         }
168
169         // Restart the camera preview.
170         safeCameraOpenInView(surface);
171     }
```

6.3.3.8 onSurfaceTextureDestroyed()

```
boolean it.unibo.torsello.bluetoothpositioning.util.CameraUtil.onSurfaceTextureDestroyed (
              SurfaceTexture surface )
```

```
178                                                                 {
179         if (mCamera != null) {
180             mCamera.stopPreview();
181             if (!preview_thread.isInterrupted()) {
182                 preview_thread.interrupt();
183             }
184             mCamera.release();
185             mCamera = null;
186         }
187         return true;
188     }
```

6.3.3.9 onSurfaceTextureSizeChanged()

```
void it.unibo.torsello.bluetoothpositioning.util.CameraUtil.onSurfaceTextureSizeChanged (
              SurfaceTexture surface,
              int width,
              int height )
```

```
174                                                                                             {
175     }
```

6.3.3.10 onSurfaceTextureUpdated()

```
void it.unibo.torsello.bluetoothpositioning.util.CameraUtil.onSurfaceTextureUpdated (
              SurfaceTexture surface )
```

```
191                                                         {
192         // Invoked every time there's a new CameraUtil preview frame
193     }
```

6.3.3.11 safeCameraOpenInView()

```
void it.unibo.torsello.bluetoothpositioning.util.CameraUtil.safeCameraOpenInView (
            SurfaceTexture surface )
```

```
99                                                          {
100
101         if (mCamera != null) {
102
103             if (preview_thread != null)
104                 preview_thread.interrupt();
105
106             preview_thread = new Thread(new Runnable() {
107                 @Override
108                 public void run() {
109                     mCamera.startPreview();
110                 }
111             });
112
113             preview_thread.start();
114
115             try {
116                 mCamera.setPreviewTexture(surface);
117             } catch (IOException ioe) {
118                 ioe.getStackTrace();
119             }
120         }
121     }
```

6.3.3.12 takePicture()

```
void it.unibo.torsello.bluetoothpositioning.util.CameraUtil.takePicture ( )
```

Picture Callback for handling a picture capture and saving it out to a file.

```
90                                {
91
92         if (mCamera != null) {
93             // get an image from the camera
94             mCamera.takePicture(null, null, mPicture);
95         }
96
97     }
```

6.3.4 Documentazione dei membri dato

6.3.4.1 activity

```
FragmentActivity it.unibo.torsello.bluetoothpositioning.util.CameraUtil.activity  [private]
```

6.3.4.2 mCamera

```
Camera it.unibo.torsello.bluetoothpositioning.util.CameraUtil.mCamera  [private]
```

6.3.4.3 mPicture

```
Camera.PictureCallback it.unibo.torsello.bluetoothpositioning.util.CameraUtil.mPicture  [private]
```

Valore iniziale:

```
= new Camera.PictureCallback() {

        @Override
        public void onPictureTaken(byte[] data, Camera camera) {

            File pictureFile = getOutputMediaFile();
            if (pictureFile == null) {
                Snackbar.make(activity.findViewById(R.id.fab),
                        "Image retrieval failed.", Snackbar.LENGTH_SHORT);
            } else {
                try {
                    FileOutputStream fos = new FileOutputStream(pictureFile);
                    fos.write(data);
                    fos.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }
```

6.3.4.4 mTextureView

```
TextureView it.unibo.torsello.bluetoothpositioning.util.CameraUtil.mTextureView  [private]
```

6.3.4.5 preview_thread

```
Thread it.unibo.torsello.bluetoothpositioning.util.CameraUtil.preview_thread  [private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- CameraUtil.java

6.4 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.examplesCamera.CamTestFragment

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.examplesCamera.CamTestFragment

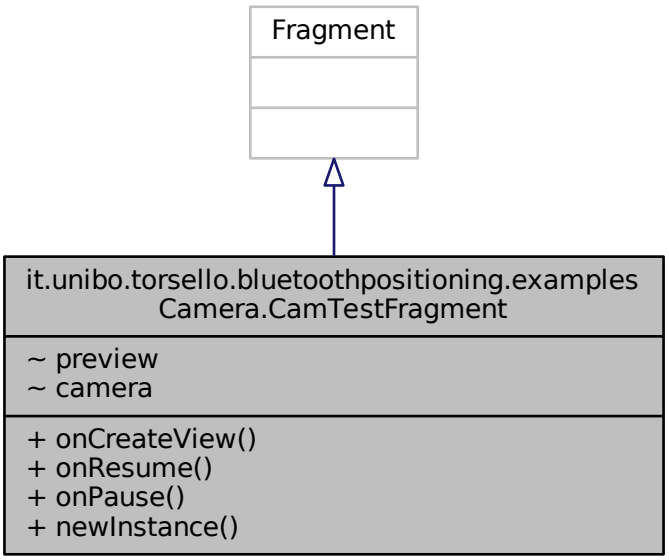
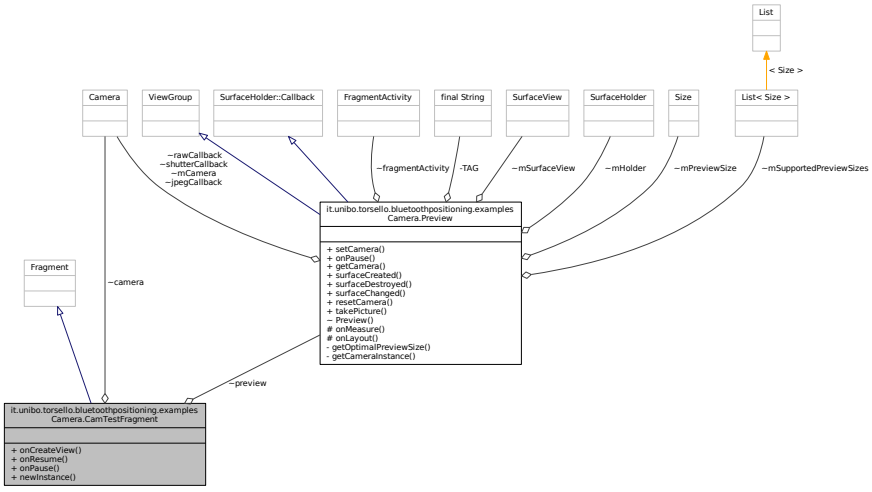


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.examplesCamera.CamTestFragment:



Membri pubblici

- View onCreateView (LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
- void onResume ()
- void onPause ()

Membri pubblici statici

- static CamTestFragment newInstance ()

Attributi con visibilità di package

- Preview preview
- Camera camera

6.4.1 Documentazione delle funzioni membro

6.4.1.1 newInstance()

```
static CamTestFragment it.unibo.torsello.bluetoothpositioning.examplesCamera.CamTestFragment.↩
newInstance ( )  [static]
```

```
23                                                  {
24         return new CamTestFragment();
25     }
```

6.4.1.2 onCreateView()

```
View it.unibo.torsello.bluetoothpositioning.examplesCamera.CamTestFragment.onCreateView (
              LayoutInflater inflater,
              ViewGroup container,
              Bundle savedInstanceState )
```

```
28                                                                                                  {
29
30         View root = inflater.inflate(R.layout.example, container, false);
31         preview = new Preview(getActivity(), (SurfaceView) root.findViewById(R.id.surfaceView));
32         ((FrameLayout) root.findViewById(R.id.layout)).addView(preview);
33         preview.setKeepScreenOn(true);
34         preview.setOnClickListener(new OnClickListener() {
35
36             @Override
37             public void onClick(View arg0) {
38                 preview.takePicture();
39             }
40         });
41         preview.setOnLongClickListener(new View.OnLongClickListener() {
42             @Override
43             public boolean onLongClick(View arg0) {
44
45                 camera.autoFocus(new Camera.AutoFocusCallback() {
46                     @Override
47                     public void onAutoFocus(boolean success, Camera arg1) {
48                         if (success) {
49                             preview.takePicture();
50                         }
51                     }
52                 });
53                 return true;
54             }
55         });
56         return root;
57     }
```

6.4.1.3 onPause()

```
void it.unibo.torsello.bluetoothpositioning.examplesCamera.CamTestFragment.onPause ( )
```

```
67                          {
68         preview.onPause();
69         super.onPause();
70     }
```

6.4.1.4 onResume()

```
void it.unibo.torsello.bluetoothpositioning.examplesCamera.CamTestFragment.onResume ( )
```

```
60                          {
61         super.onResume();
62         preview.setCamera(getActivity());
63         camera = preview.getCamera();
64     }
```

6.4.2 Documentazione dei membri dato

6.4.2.1 camera

```
Camera it.unibo.torsello.bluetoothpositioning.examplesCamera.CamTestFragment.camera  [package]
```

6.4.2.2 preview

```
Preview it.unibo.torsello.bluetoothpositioning.examplesCamera.CamTestFragment.preview  [package]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- CamTestFragment.java

6.5 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.util.ChartUtil

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.util.ChartUtil

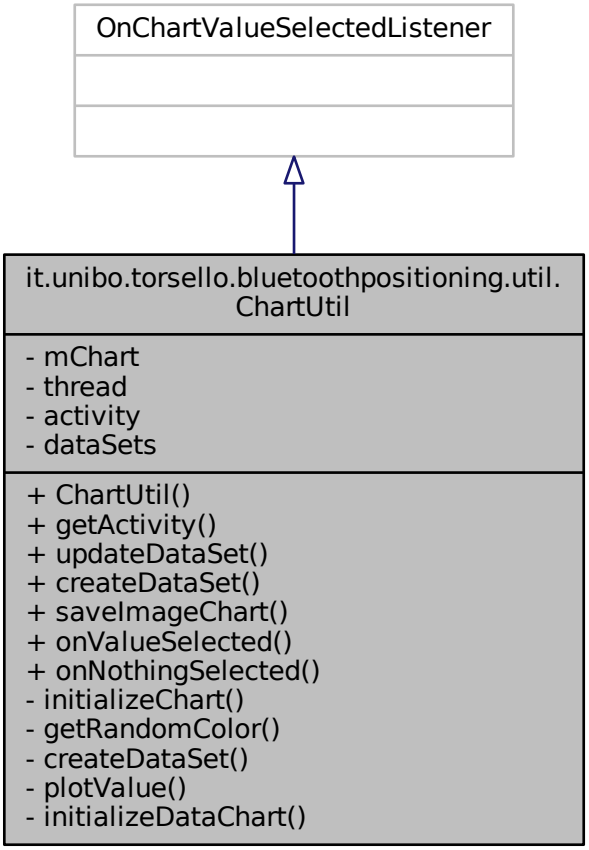
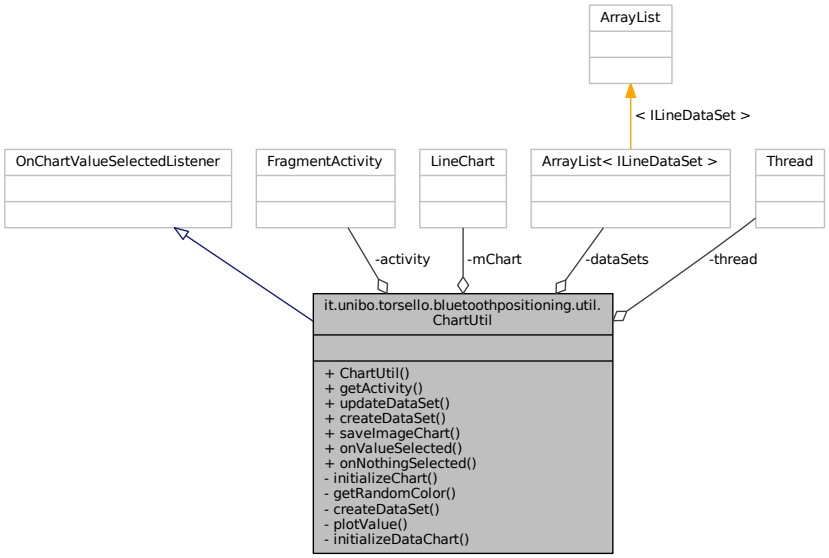


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.util.ChartUtil:



Membri pubblici

- ChartUtil (FragmentActivity fragmentActivity, LineChart chart)
- FragmentActivity getActivity ()
- void updateDataSet (final ArrayList< Double > doubleArrayList)
- ArrayList< ILineDataSet > createDataSet (ArrayList< String > args)
- void saveImageChart ()
- void onValueSelected (Entry e, Highlight h)
- void onNothingSelected ()

Membri privati

- void initializeChart ()
- int getRandomColor ()
- LineDataSet createDataSet (String nameDataSet, int color)
- void plotValue (LineData data, int index, Double value)
- void initializeDataChart (ArrayList< ILineDataSet > dataSets)

Attributi privati

- LineChart mChart
- Thread thread
- FragmentActivity activity
- ArrayList< ILineDataSet > dataSets

6.5.1 Descrizione dettagliata

Created by Federico Torsello. federico.torsello@studio.unibo.it

6.5.2 Documentazione dei costruttori e dei distruttori

6.5.2.1 ChartUtil()

```
it.unibo.torsello.bluetoothpositioning.util.ChartUtil.ChartUtil (
            FragmentActivity fragmentActivity,
            LineChart chart )
```

```
39                                                                          {
40          this.activity = fragmentActivity;
41          this.mChart = chart;
42          initializeChart();
43      }
```

6.5.3 Documentazione delle funzioni membro

6.5.3.1 createDataSet() [1/2]

```
ArrayList<ILineDataSet> it.unibo.torsello.bluetoothpositioning.util.ChartUtil.createDataSet (
            ArrayList< String > args )
```

```
128                                                                         {
129         // create a dataset and give it a type
130
131         for (String s : args) {
132             if (s != null) {
133                 if (s.equals(getActivity().getString(R.string.chart_arduino))) {
134                     dataSets.add(createDataSet(s, Color.RED));
135                 } else {
136                     dataSets.add(createDataSet(s,
    getRandomColor()));
137                 }
138             }
139         }
140
141         return dataSets;
142     }
```

6.5.3.2 createDataSet() [2/2]

```
LineDataSet it.unibo.torsello.bluetoothpositioning.util.ChartUtil.createDataSet (
            String nameDataSet,
            int color )  [private]
```

```
153                                                                     {
154         LineDataSet set = new LineDataSet(null, nameDataSet);
155         set.setColor(color);
156         return set;
157     }
```

6.5.3.3 getActivity()

```
FragmentActivity it.unibo.torsello.bluetoothpositioning.util.ChartUtil.getActivity ( )
```

```
45                                      {
46          return activity;
47      }
```

6.5.3.4 getRandomColor()

```
int it.unibo.torsello.bluetoothpositioning.util.ChartUtil.getRandomColor ( )  [private]
```

```
144                                 {
145         Random rnd = new Random();
146         int color = 0;
147         while (color == 0) {
148             color = Color.argb(255, rnd.nextInt(255), rnd.nextInt(255), rnd.nextInt(255));
149         }
150         return color;
151     }
```

6.5.3.5 initializeChart()

```
void it.unibo.torsello.bluetoothpositioning.util.ChartUtil.initializeChart ( )  [private]
```

```
49                                  {
50         dataSets = new ArrayList<ILineDataSet>();
51
52         mChart.setOnChartValueSelectedListener(this);
53
54         // no description text
55         mChart.setDescription("");
56         mChart.setNoDataTextDescription("You need to provide data for the chart.");
57
58         mChart.setDrawGridBackground(true);
59
60         // if disabled, scaling can be done on x- and y-axis separately
61         mChart.setPinchZoom(true);
62
63         // set an alternative background color
64         mChart.setBackgroundColor(Color.LTGRAY);
65
66         Typeface mTfLight = Typeface.createFromAsset(getActivity().getAssets(), "
      OpenSans-Light.ttf");
67         Typeface mTfBold = Typeface.createFromAsset(getActivity().getAssets(), "
      OpenSans-Bold.ttf");
68
69         // get the legend (only possible after setting data)
70         Legend l = mChart.getLegend();
71 //         l.setPosition(Legend.LegendPosition.RIGHT_OF_CHART);
72 //         l.setOrientation(Legend.LegendOrientation.VERTICAL);
73         l.setXEntrySpace(7f);
74         l.setYEntrySpace(7f);
75
76         XAxis xl = mChart.getXAxis();
77         xl.setTypeface(mTfLight);
78         xl.setGridColor(Color.LTGRAY);
79         xl.setTextColor(Color.WHITE);
80
81         YAxis leftAxis = mChart.getAxisLeft();
82         leftAxis.setTypeface(mTfLight);
83         leftAxis.setTextColor(Color.WHITE);
84
85         YAxis rightAxis = mChart.getAxisRight();
86         rightAxis.setTypeface(mTfBold);
87
88     }
```

6.5.3.6 initializeDataChart()

```
void it.unibo.torsello.bluetoothpositioning.util.ChartUtil.initializeDataChart (
             ArrayList< ILineDataSet > dataSets )  [private]
```

```
177                                                                 {
178
179         // create a data object with the datasets
180         LineData lineData = new LineData(dataSets);
181         lineData.setValueTextColor(Color.RED);
182         lineData.setValueTextSize(9f);
183         lineData.setValueFormatter(new DefaultValueFormatter(2));
184
185         // set data
186         mChart.setData(lineData);
187     }
```

6.5.3.7 onNothingSelected()

```
void it.unibo.torsello.bluetoothpositioning.util.ChartUtil.onNothingSelected ( )
```

```
199                                     {
200         Log.i("Nothing selected", "Nothing selected.");
201     }
```

6.5.3.8 onValueSelected()

```
void it.unibo.torsello.bluetoothpositioning.util.ChartUtil.onValueSelected (
              Entry e,
              Highlight h )
```

```
194                                                     {
195         Log.i("Entry selected", e.toString());
196     }
```

6.5.3.9 plotValue()

```
void it.unibo.torsello.bluetoothpositioning.util.ChartUtil.plotValue (
              LineData data,
              int index,
              Double value )  [private]
```

```
159                                                                 {
160
161         ILineDataSet set = data.getDataSetByIndex(index);
162
163         set.addEntry(new Entry(set.getEntryCount(), value.floatValue()));
164
165         data.notifyDataChanged();
166
167         // let the chart know it's data has changed
168         mChart.notifyDataSetChanged();
169
170         // limit the number of visible entries
171         mChart.setVisibleXRangeMaximum(10);
172
173         // move to the latest entry
174         mChart.moveViewToX(data.getEntryCount());
175     }
```

6.5.3.10 saveImageChart()

```
void it.unibo.torsello.bluetoothpositioning.util.ChartUtil.saveImageChart ( )
```

```
189                                   {
190         mChart.saveToGallery("prova", 100);
191     }
```

6.5.3.11 updateDataSet()

```
void it.unibo.torsello.bluetoothpositioning.util.ChartUtil.updateDataSet (
            final ArrayList< Double > doubleArrayList )
```

```
90                                                                          {
91         if (thread != null)
92             thread.interrupt();
93
94         thread = new Thread(new Runnable() {
95             @Override
96             public void run() {
97                 if (getActivity() != null) {
98
99                     getActivity().runOnUiThread(new Runnable() {
100                        @Override
101                        public void run() {
102
103                            LineData data = mChart.getData();
104
105                            if (data == null) {
106                                if (dataSets != null) {
107                                    initializeDataChart(
    dataSets);
108                                } else {
109                                    throw new Error("Error: dataSet is null!!!");
110                                }
111                            } else {
112                                if (data.getDataSetCount() > 0) {
113
114                                    for (int i = 0; i < doubleArrayList.size(); i++) {
115                                        plotValue(data, i, doubleArrayList.get(i));
116                                    }
117                                }
118                            }
119                        }
120                    });
121                }
122            }
123        });
124
125        thread.start();
126    }
```

6.5.4 Documentazione dei membri dato

6.5.4.1 activity

```
FragmentActivity it.unibo.torsello.bluetoothpositioning.util.ChartUtil.activity  [private]
```

6.5.4.2 dataSets

```
ArrayList<ILineDataSet> it.unibo.torsello.bluetoothpositioning.util.ChartUtil.dataSets  [private]
```

6.5.4.3 mChart

```
LineChart it.unibo.torsello.bluetoothpositioning.util.ChartUtil.mChart  [private]
```

6.5.4.4 thread

```
Thread it.unibo.torsello.bluetoothpositioning.util.ChartUtil.thread  [private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- ChartUtil.java

6.6 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassFragment

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassFragment

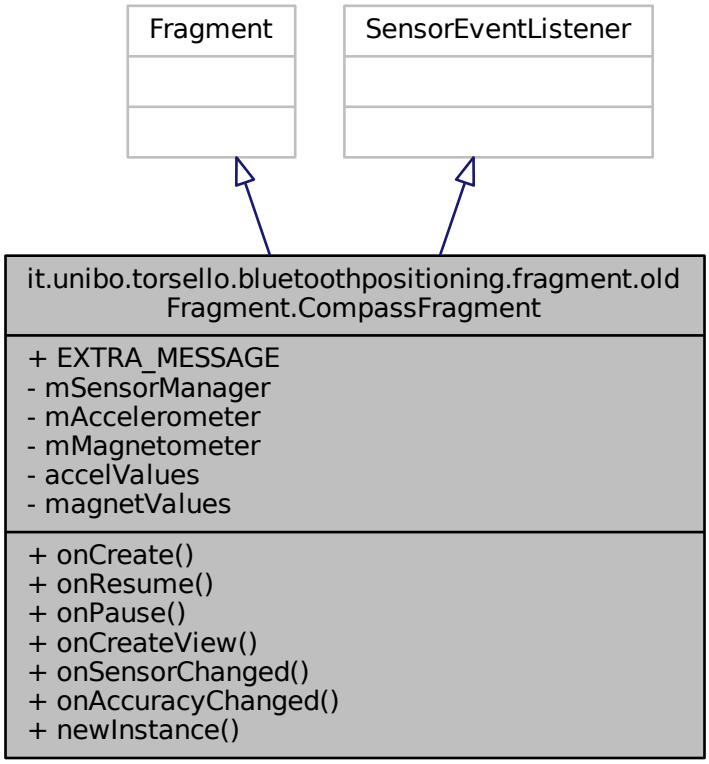
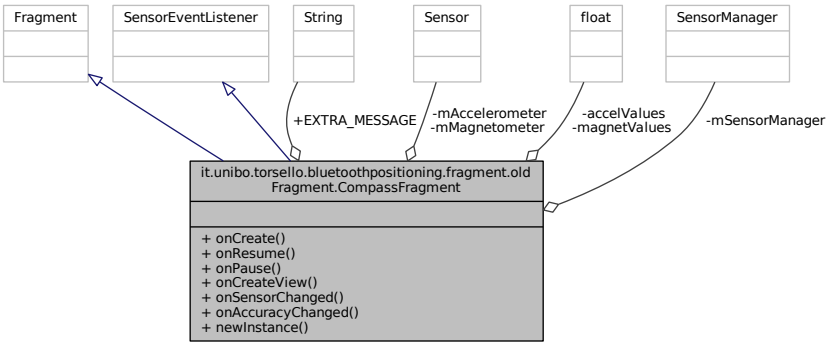


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassFragment:



Membri pubblici

- void onCreate (Bundle savedInstanceState)
- void onResume ()
- void onPause ()
- View onCreateView (LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
- void onSensorChanged (SensorEvent event)
- void onAccuracyChanged (Sensor sensor, int accuracy)

Membri pubblici statici

- static CompassFragment newInstance (String message)

Attributi pubblici statici

- static final String EXTRA_MESSAGE = "EXTRA_MESSAGE"

Attributi privati

- SensorManager mSensorManager
- Sensor mAccelerometer
- Sensor mMagnetometer
- float accelValues [ ]
- float magnetValues [ ]

6.6.1 Descrizione dettagliata

Created by Federico Torsello. federico.torsello@studio.unibo.it

6.6.2 Documentazione delle funzioni membro

6.6.2.1 newInstance()

```
static CompassFragment it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.Compass↩
Fragment.newInstance (
            String message )  [static]
```

```
32                                                          {
33         CompassFragment fragment = new CompassFragment();
34         Bundle args = new Bundle();
35         args.putString(EXTRA_MESSAGE, message);
36         fragment.setArguments(args);
37         return fragment;
38     }
```

6.6.2.2 onAccuracyChanged()

```
void it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassFragment.onAccuracy↩
Changed (
            Sensor sensor,
            int accuracy )
```

```
118                                                                 {
119     }
```

6.6.2.3 onCreate()

```
void it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassFragment.onCreate (
            Bundle savedInstanceState )
```

```
41                                                  {
42         super.onCreate(savedInstanceState);
43         mSensorManager = (SensorManager) getActivity().getSystemService(Context.
       SENSOR_SERVICE);
44         mAccelerometer = mSensorManager.getDefaultSensor(Sensor.
       TYPE_ACCELEROMETER);
45         mMagnetometer = mSensorManager.getDefaultSensor(Sensor.
       TYPE_MAGNETIC_FIELD);
46      }
```

6.6.2.4 onCreateView()

```
View it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassFragment.onCreateView
(
            LayoutInflater inflater,
            ViewGroup container,
            Bundle savedInstanceState )
```

```
78                                                      {
79         return inflater.inflate(R.layout.fragment_compass, container, false);
80     }
```

6.6.2.5 onPause()

```
void it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassFragment.onPause ( )
```

```
70                          {
71         super.onPause();
72         mSensorManager.unregisterListener(this, mAccelerometer);
73         mSensorManager.unregisterListener(this, mMagnetometer);
74     }
```

6.6.2.6 onResume()

```
void it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassFragment.onResume ( )
```

```
49                              {
50          super.onResume();
51          if (mAccelerometer != null) {
52              if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {
53                  mSensorManager.registerListener(this,
      mAccelerometer, SensorManager.SENSOR_DELAY_FASTEST,
54                          SensorManager.SENSOR_STATUS_ACCURACY_HIGH);
55              } else {
56                  mSensorManager.registerListener(this,
      mAccelerometer, SensorManager.SENSOR_DELAY_FASTEST);
57              }
58          }
59
60          if (mMagnetometer != null) {
61              if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {
62                  mSensorManager.registerListener(this,
      mMagnetometer, SensorManager.SENSOR_DELAY_FASTEST,
63                          SensorManager.SENSOR_STATUS_ACCURACY_HIGH);
64              } else {
65                  mSensorManager.registerListener(this,
      mMagnetometer, SensorManager.SENSOR_DELAY_FASTEST);
66              }
67          }
68      }
```

6.6.2.7 onSensorChanged()

```
void it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassFragment.onSensor↩
Changed (
              SensorEvent event )
```

```
83                                                  {
84          if (event.sensor == mAccelerometer) {
85              accelValues = event.values;
86          } else if (event.sensor == mMagnetometer) {
87              magnetValues = event.values;
88          }
89
90          if (accelValues != null && magnetValues != null) {
91              float[] rotationMatrix = new float[16];
92              float[] orientation = new float[16];
93
94              SensorManager.getRotationMatrix(rotationMatrix, null, accelValues,
      magnetValues);
95              SensorManager.getOrientation(rotationMatrix, orientation);
96
97              float azimuthDegree = (float) Math.toDegrees(-orientation[0]);
98
99              RotateAnimation rotateAnimation = new RotateAnimation(
100                     azimuthDegree, azimuthDegree,
101                     Animation.RELATIVE_TO_SELF, 0.5f,
102                     Animation.RELATIVE_TO_SELF, 0.5f);
103             rotateAnimation.setDuration(250);
104             rotateAnimation.setFillAfter(true);
105
106
107             try {
108                 ImageView mPointer = (ImageView) getActivity().findViewById(R.id.pointer);
109                 mPointer.startAnimation(rotateAnimation);
110             } catch (NullPointerException e) {
111                 e.getStackTrace();
112             }
113         }
114
115     }
```

6.6.3 Documentazione dei membri dato

6.6.3.1 accelValues

```
float it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassFragment.accel↩
Values[]  [private]
```

6.6.3.2 EXTRA_MESSAGE

```
final String it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassFragment.EXT↩
RA_MESSAGE = "EXTRA_MESSAGE"  [static]
```

6.6.3.3 mAccelerometer

```
Sensor it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassFragment.mAccelerometer
[private]
```

6.6.3.4 magnetValues

```
float it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassFragment.magnet↩
Values[]  [private]
```

6.6.3.5 mMagnetometer

```
Sensor it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassFragment.mMagnetometer
[private]
```

6.6.3.6 mSensorManager

```
SensorManager it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassFragment.m↩
SensorManager  [private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- CompassFragment.java

6.7 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassMagnoFragment

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassMagnoFragment

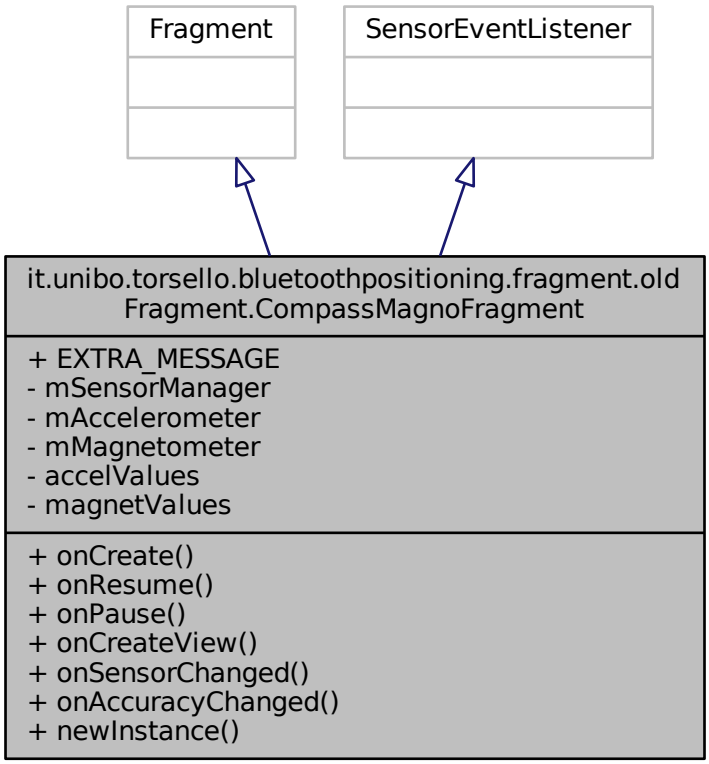
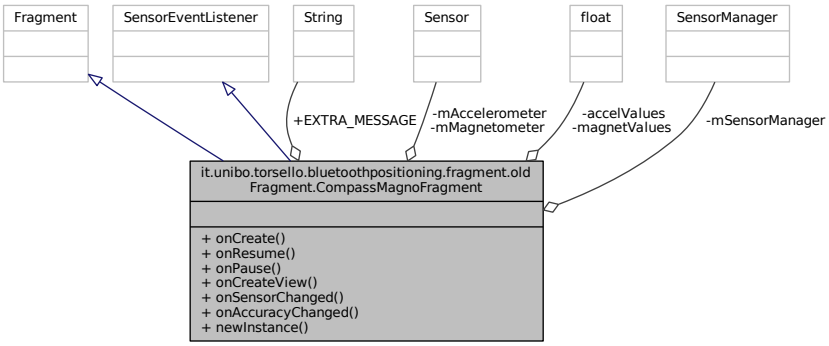


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassMagnoFragment:



Membri pubblici

- void onCreate (@Nullable Bundle savedInstanceState)
- void onResume ()
- void onPause ()
- View onCreateView (LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
- void onSensorChanged (SensorEvent event)
- void onAccuracyChanged (Sensor sensor, int accuracy)

Membri pubblici statici

- static CompassMagnoFragment newInstance (String message)

Attributi pubblici statici

- static final String EXTRA_MESSAGE = "EXTRA_MESSAGE"

Attributi privati

- SensorManager mSensorManager
- Sensor mAccelerometer
- Sensor mMagnetometer
- float accelValues [ ]
- float magnetValues [ ]

6.7.1 Descrizione dettagliata

Created by Federico Torsello. federico.torsello@studio.unibo.it

6.7.2 Documentazione delle funzioni membro

6.7.2.1 newInstance()

```
static CompassMagnoFragment it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.↩
CompassMagnoFragment.newInstance (
            String message )  [static]
```

```
30                                                                  {
31         CompassMagnoFragment fragment = new CompassMagnoFragment();
32         Bundle args = new Bundle();
33         args.putString(EXTRA_MESSAGE, message);
34         fragment.setArguments(args);
35         return fragment;
36     }
```

6.7.2.2 onAccuracyChanged()

```
void it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassMagnoFragment.on↩
AccuracyChanged (
            Sensor sensor,
            int accuracy )
```

```
126                                                                 {
127
128     }
```

6.7.2.3 onCreate()

```
void it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassMagnoFragment.onCreate
(
            @Nullable Bundle savedInstanceState )
```

```
40                                                                  {
41         super.onCreate(savedInstanceState);
42         setHasOptionsMenu(true);
43         mSensorManager = (SensorManager) getActivity().getSystemService(Context.
      SENSOR_SERVICE);
44         mAccelerometer = mSensorManager.getDefaultSensor(Sensor.
      TYPE_ACCELEROMETER);
45         mMagnetometer = mSensorManager.getDefaultSensor(Sensor.
      TYPE_MAGNETIC_FIELD);
46     }
```

6.7.2.4 onCreateView()

```
View it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassMagnoFragment.on↩
CreateView (
            LayoutInflater inflater,
            ViewGroup container,
            Bundle savedInstanceState )
```

```
80                                                  {
81         return inflater.inflate(R.layout.fragment_compass_text, container, false);
82     }
```

6.7.2.5 onPause()

```
void it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassMagnoFragment.onPause
( )
```

```
72                          {
73         super.onPause();
74         mSensorManager.unregisterListener(this, mAccelerometer);
75         mSensorManager.unregisterListener(this, mMagnetometer);
76     }
```

6.7.2.6 onResume()

void it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassMagnoFragment.onResume ( )

```
49                              {
50          super.onResume();
51          if (mAccelerometer != null) {
52              if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {
53                  mSensorManager.registerListener(this,
    mAccelerometer, SensorManager.SENSOR_DELAY_FASTEST,
54                          SensorManager.SENSOR_STATUS_ACCURACY_HIGH);
55              } else {
56                  mSensorManager.registerListener(this,
    mAccelerometer, SensorManager.SENSOR_DELAY_FASTEST);
57              }
58          }
59
60          if (mMagnetometer != null) {
61              if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {
62                  mSensorManager.registerListener(this,
    mMagnetometer, SensorManager.SENSOR_DELAY_FASTEST,
63                          SensorManager.SENSOR_STATUS_ACCURACY_HIGH);
64              } else {
65                  mSensorManager.registerListener(this,
    mMagnetometer, SensorManager.SENSOR_DELAY_FASTEST);
66              }
67          }
68      }
```

6.7.2.7 onSensorChanged()

void it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassMagnoFragment.on↩ SensorChanged (SensorEvent *event*)

```
85                                                  {
86
87          if (event.sensor == mAccelerometer) {
88              accelValues = event.values;
89          } else if (event.sensor == mMagnetometer) {
90              magnetValues = event.values;
91          }
92
93
94          if (accelValues != null && magnetValues != null) {
95              float[] rotation = new float[16];
96              float[] orientation = new float[16];
97
98              SensorManager.getRotationMatrix(rotation, null, accelValues,
    magnetValues);
99              SensorManager.getOrientation(rotation, orientation);
100
101
102              float azimuthDegree = (float) (Math.toDegrees(orientation[0]) + 360) % 360;
103              float orientationDegree = Math.round(azimuthDegree);
104
105              String compassOrientation;
106              if (orientationDegree >= 0 && orientationDegree < 90) {
107                  compassOrientation = "N";
108              } else if (orientationDegree >= 90 && orientationDegree < 180) {
109                  compassOrientation = "E";
110              } else if (orientationDegree >= 180 && orientationDegree < 270) {
111                  compassOrientation = "S";
112              } else {
113                  compassOrientation = "W";
114              }
115
116 //              try {
117 //                  TextView messageTextView = (TextView) getActivity().findViewById(R.id.compass);
118 //                  messageTextView.setText(compassOrientation);
119 //              }catch (NullPointerException e){
120 //                  e.getStackTrace();
121 //              }
122          }
123      }
```

6.7.3 Documentazione dei membri dato

6.7.3.1 accelValues

```
float it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassMagnoFragment.accel↩
Values[]  [private]
```

6.7.3.2 EXTRA_MESSAGE

```
final String it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassMagnoFragment.↩
EXTRA_MESSAGE = "EXTRA_MESSAGE"  [static]
```

6.7.3.3 mAccelerometer

```
Sensor it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassMagnoFragment.m↩
Accelerometer  [private]
```

6.7.3.4 magnetValues

```
float it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassMagnoFragment.↩
magnetValues[]  [private]
```

6.7.3.5 mMagnetometer

```
Sensor it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassMagnoFragment.m↩
Magnetometer  [private]
```

6.7.3.6 mSensorManager

```
SensorManager it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassMagnoFragment.↩
mSensorManager  [private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- CompassMagnoFragment.java

6.8 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CountPass↩ Fragment

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CountPassFragment

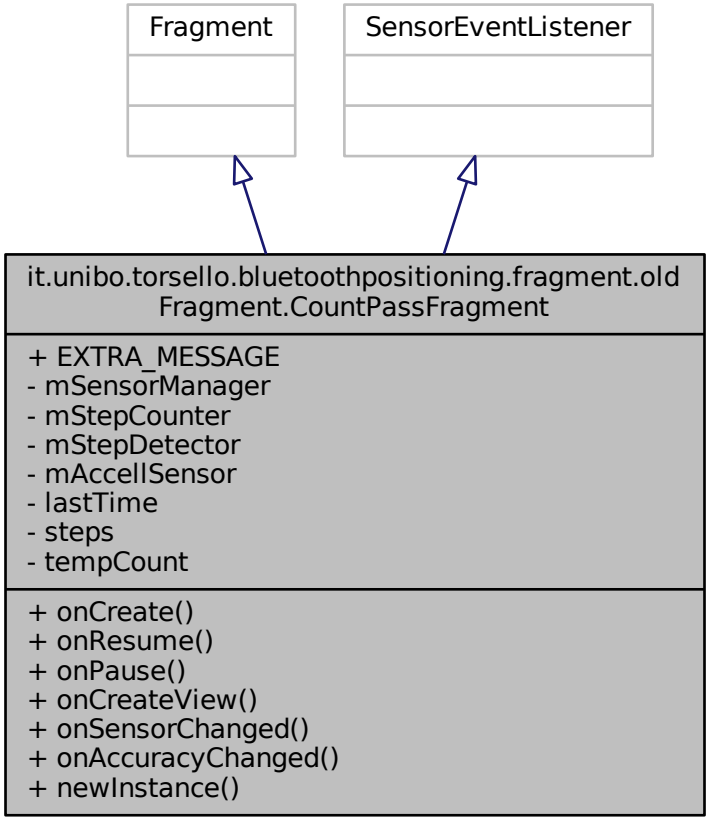
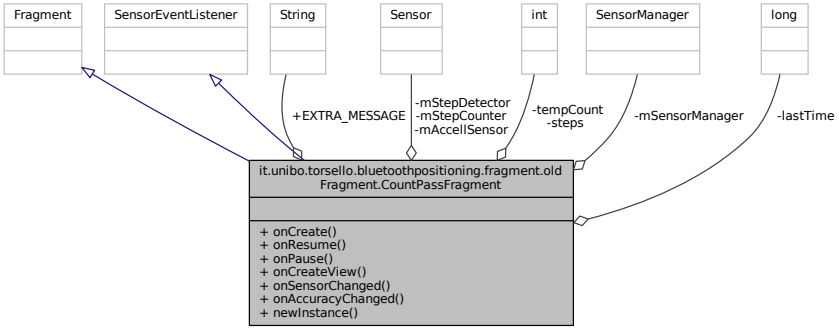


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CountPass↩ Fragment:



Membri pubblici

- void onCreate (Bundle savedInstanceState)
- void onResume ()
- void onPause ()
- View onCreateView (LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
- void onSensorChanged (SensorEvent event)
- void onAccuracyChanged (Sensor sensor, int accuracy)

Membri pubblici statici

- static CountPassFragment newInstance (String message)

Attributi pubblici statici

- static final String EXTRA_MESSAGE = "EXTRA_MESSAGE"

Attributi privati

- SensorManager mSensorManager
- Sensor mStepCounter
- Sensor mStepDetector
- Sensor mAccellSensor
- long lastTime = 0L
- int steps = 0
- int tempCount

6.8.1 Descrizione dettagliata

Created by Federico Torsello. federico.torsello@studio.unibo.it

6.8.2 Documentazione delle funzioni membro

6.8.2.1 newInstance()

```
static CountPassFragment it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.Count↩
PassFragment.newInstance (
            String message )  [static]
```

```
34                                                          {
35         CountPassFragment fragment = new CountPassFragment();
36         Bundle args = new Bundle();
37         args.putString(EXTRA_MESSAGE, message);
38         fragment.setArguments(args);
39         return fragment;
40     }
```

6.8.2.2 onAccuracyChanged()

```
void it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CountPassFragment.onAccuracy↩
Changed (
            Sensor sensor,
            int accuracy )
```

```
145                                                         {
146     }
```

6.8.2.3 onCreate()

```
void it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CountPassFragment.onCreate (
            Bundle savedInstanceState )
```

```
43                                                  {
44         super.onCreate(savedInstanceState);
45
46         mSensorManager = (SensorManager) getActivity().getSystemService(Context.
       SENSOR_SERVICE);
47
48         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {
49             mStepCounter = mSensorManager.getDefaultSensor(Sensor.
       TYPE_STEP_COUNTER);
50             mStepDetector = mSensorManager.getDefaultSensor(Sensor.
       TYPE_STEP_DETECTOR);
51         }
52
53         if (mStepCounter == null && mStepDetector == null) {
54             mAccellSensor = mSensorManager.getDefaultSensor(Sensor.
       TYPE_ACCELEROMETER);
55         }
56
57     }
```

6.8.2.4 onCreateView()

```
View it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CountPassFragment.onCreate↩
View (
            LayoutInflater inflater,
            ViewGroup container,
            Bundle savedInstanceState )
```

```
97                                                                                          {
98         return inflater.inflate(R.layout.fragment_count_pass, container, false);
99     }
```

6.8.2.5 onPause()

```
void it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CountPassFragment.onPause ( )
```

```
86                          {
87         super.onPause();
88         if (mStepCounter != null && mStepDetector != null) {
89             mSensorManager.unregisterListener(this, mStepCounter);
90             mSensorManager.unregisterListener(this, mStepDetector);
91         } else {
92             mSensorManager.unregisterListener(this, mAccellSensor);
93         }
94     }
```

6.8.2.6 onResume()

void it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CountPassFragment.onResume (
)

```
60                          {
61         super.onResume();
62         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {
63             if (mStepCounter != null) {
64                 mSensorManager.registerListener(this, mStepCounter, SensorManager
     .SENSOR_DELAY_FASTEST,
65                         SensorManager.SENSOR_STATUS_ACCURACY_HIGH);
66 //                mSensorManager.registerListener(this, mStepCounter, SensorManager.SENSOR_DELAY_FASTEST);
67             }
68
69             if (mStepDetector != null) {
70                 mSensorManager.registerListener(this,
     mStepDetector, SensorManager.SENSOR_DELAY_FASTEST,
71                         SensorManager.SENSOR_STATUS_ACCURACY_HIGH);
72 //                mSensorManager.registerListener(this, mStepDetector, SensorManager.SENSOR_DELAY_FASTEST);
73             }
74         }
75
76         if (mStepCounter == null && mStepDetector == null) {
77             if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {
78                 mSensorManager.registerListener(this,
     mAccellSensor, SensorManager.SENSOR_DELAY_FASTEST,
79                         SensorManager.SENSOR_STATUS_ACCURACY_HIGH);
80             } else {
81                 mSensorManager.registerListener(this,
     mAccellSensor, SensorManager.SENSOR_DELAY_FASTEST);
82             }
83         }
84     }
```

6.8.2.7 onSensorChanged()

void it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CountPassFragment.onSensor↩
Changed (
SensorEvent *event*)

```
102                                                 {
103 //        if (event.sensor == mStepCounter) {
104 //            stepCounter = event.values;
105 //            Log.d("stepCount" , String.valueOf(stepCounter[0]));
106 //        } else if (event.sensor == mStepDetector) {
107 //            stepDetector = event.values;
108 //            Log.d("stepDetector" , String.valueOf(stepDetector[0]));
109 //        }
110
111         switch (event.sensor.getType()) {
112             case Sensor.TYPE_STEP_DETECTOR:
113                 Log.d("stepCount", String.valueOf(event.values[0]));
114                 break;
115             case Sensor.TYPE_STEP_COUNTER:
116                 Log.d("stepDetector", String.valueOf(event.values[0]));
117                 break;
118             case Sensor.TYPE_ACCELEROMETER: {
119                 float x = event.values[0];
120                 float y = event.values[1];
121                 float z = event.values[2];
122
123                 float accelationSquareRoot = Math.abs((x * x + y * y + z * z)
124                         / (SensorManager.GRAVITY_EARTH * SensorManager.GRAVITY_EARTH)) - 1.0f;
125                 long actualTime = System.currentTimeMillis();
126
127                 if (actualTime - lastTime > 300) {
128                     if (accelationSquareRoot < -0.45f) {
129                         steps++;
130                         tempCount++;
131                     }
132                     lastTime = actualTime;
133                 }
134
135                 TextView textView = (TextView) getActivity().findViewById(R.id.countPassText);
136                 textView.setText(String.valueOf(steps));
137             }
138             break;
139         }
140
141
142     }
```

6.8.3 Documentazione dei membri dato

6.8.3.1 EXTRA_MESSAGE

```
final String it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CountPassFragment.E↩
XTRA_MESSAGE = "EXTRA_MESSAGE"  [static]
```

6.8.3.2 lastTime

```
long it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CountPassFragment.lastTime =
0L  [private]
```

6.8.3.3 mAccellSensor

```
Sensor it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CountPassFragment.mAccell↩
Sensor  [private]
```

6.8.3.4 mSensorManager

```
SensorManager it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CountPassFragment.↩
mSensorManager  [private]
```

6.8.3.5 mStepCounter

```
Sensor it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CountPassFragment.mStep↩
Counter  [private]
```

6.8.3.6 mStepDetector

```
Sensor it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CountPassFragment.mStep↩
Detector  [private]
```

6.8.3.7 steps

```
int it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CountPassFragment.steps = 0
[private]
```

6.8.3.8 tempCount

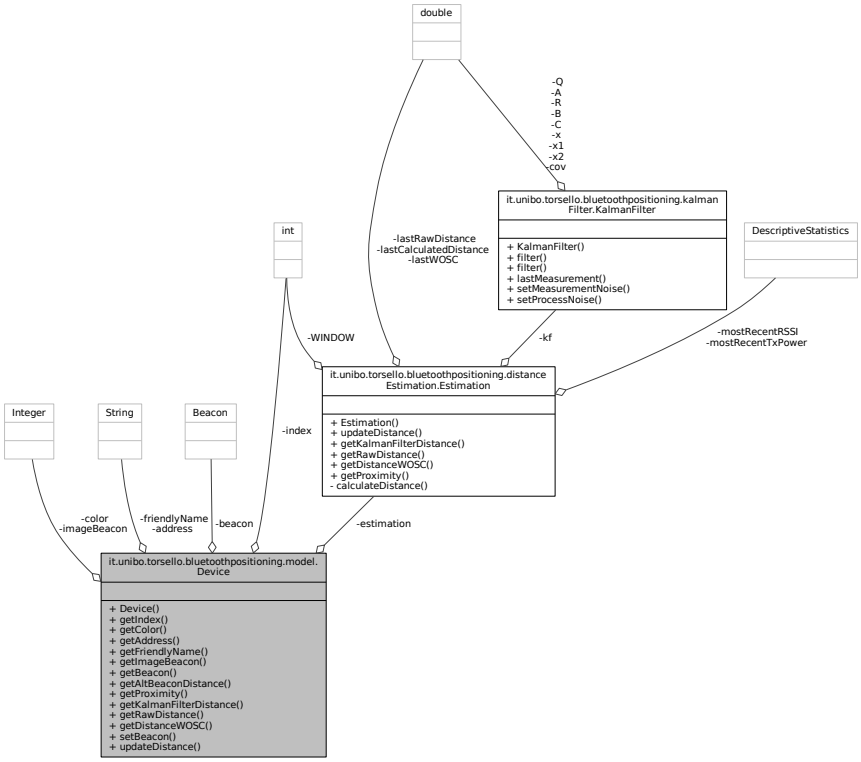
```
int it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CountPassFragment.tempCount
[private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- CountPassFragment.java

6.9 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.model.Device

Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.model.Device:



Membri pubblici

- Device (int index, String address, String friendlyName, Integer color, Integer imageBeacon)
- int getIndex ()
- Integer getColor ()
- String getAddress ()
- String getFriendlyName ()
- Integer getImageBeacon ()
- Beacon getBeacon ()
- double getAltBeaconDistance ()
- String getProximity ()
- double getKalmanFilterDistance ()
- double getRawDistance ()
- double getDistanceWOSC ()
- void setBeacon (Beacon beacon)
- void updateDistance (double processNoise)

Attributi privati

- Estimation estimation
- String address
- String friendlyName
- Beacon beacon
- Integer imageBeacon
- Integer color
- int index

6.9.1 Descrizione dettagliata

Created by Federico Torsello. federico.torsello@studio.unibo.it

6.9.2 Documentazione dei costruttori e dei distruttori

6.9.2.1 Device()

```
it.unibo.torsello.bluetoothpositioning.model.Device.Device (
            int index,
            String address,
            String friendlyName,
            Integer color,
            Integer imageBeacon )
```

```
21                                                                                  {
22          this.index = index;
23          this.address = address;
24          this.friendlyName = friendlyName;
25          this.estimation = new Estimation();
26          this.imageBeacon = imageBeacon;
27          this.color = color;
28      }
```

6.9.3 Documentazione delle funzioni membro

6.9.3.1 getAddress()

```
String it.unibo.torsello.bluetoothpositioning.model.Device.getAddress ( )
```

```
38                                {
39          return this.address;
40      }
```

6.9.3.2 getAltBeaconDistance()

```
double it.unibo.torsello.bluetoothpositioning.model.Device.getAltBeaconDistance ( )
```

```
54                                        {
55          return beacon.getDistance();
56      }
```

6.9.3.3 getBeacon()

```
Beacon it.unibo.torsello.bluetoothpositioning.model.Device.getBeacon ( )
```

```
50                               {
51          return beacon;
52      }
```

6.9.3.4 getColor()

```
Integer it.unibo.torsello.bluetoothpositioning.model.Device.getColor ( )
```

```
34                              {
35          return color;
36      }
```

6.9.3.5 getDistanceWOSC()

```
double it.unibo.torsello.bluetoothpositioning.model.Device.getDistanceWOSC ( )
```

```
71                                  {
72          return estimation.getDistanceWOSC();
73      }
```

6.9.3.6 getFriendlyName()

```
String it.unibo.torsello.bluetoothpositioning.model.Device.getFriendlyName ( )
```

```
42                                  {
43          return friendlyName;
44      }
```

6.9.3.7 getImageBeacon()

```
Integer it.unibo.torsello.bluetoothpositioning.model.Device.getImageBeacon ( )
```

```
46                                  {
47          return imageBeacon;
48      }
```

6.9.3.8 getIndex()

```
int it.unibo.torsello.bluetoothpositioning.model.Device.getIndex ( )
```

```
30                          {
31          return index;
32      }
```

6.9.3.9 getKalmanFilterDistance()

```
double it.unibo.torsello.bluetoothpositioning.model.Device.getKalmanFilterDistance ( )
```

```
63                                      {
64          return estimation.getKalmanFilterDistance();
65      }
```

6.9.3.10 getProximity()

```
String it.unibo.torsello.bluetoothpositioning.model.Device.getProximity ( )
```

```
58                                  {
59         return estimation.getProximity();
60     }
```

6.9.3.11 getRawDistance()

```
double it.unibo.torsello.bluetoothpositioning.model.Device.getRawDistance ( )
```

```
67                                     {
68         return estimation.getRawDistance();
69     }
```

6.9.3.12 setBeacon()

```
void it.unibo.torsello.bluetoothpositioning.model.Device.setBeacon (
              Beacon beacon )
```

```
75                                          {
76         this.beacon = beacon;
77     }
```

6.9.3.13 updateDistance()

```
void it.unibo.torsello.bluetoothpositioning.model.Device.updateDistance (
              double processNoise )
```

```
79                                                      {
80         if (beacon != null) {
81             estimation.updateDistance(beacon, processNoise);
82         }
83     }
```

6.9.4 Documentazione dei membri dato

6.9.4.1 address

```
String it.unibo.torsello.bluetoothpositioning.model.Device.address  [private]
```

6.9.4.2 beacon

```
Beacon it.unibo.torsello.bluetoothpositioning.model.Device.beacon  [private]
```

6.9.4.3 color

```
Integer it.unibo.torsello.bluetoothpositioning.model.Device.color  [private]
```

6.9.4.4 estimation

```
Estimation it.unibo.torsello.bluetoothpositioning.model.Device.estimation  [private]
```

6.9.4.5 friendlyName

```
String it.unibo.torsello.bluetoothpositioning.model.Device.friendlyName  [private]
```

6.9.4.6 imageBeacon

```
Integer it.unibo.torsello.bluetoothpositioning.model.Device.imageBeacon  [private]
```

6.9.4.7 index

```
int it.unibo.torsello.bluetoothpositioning.model.Device.index  [private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- Device.java

6.10 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter

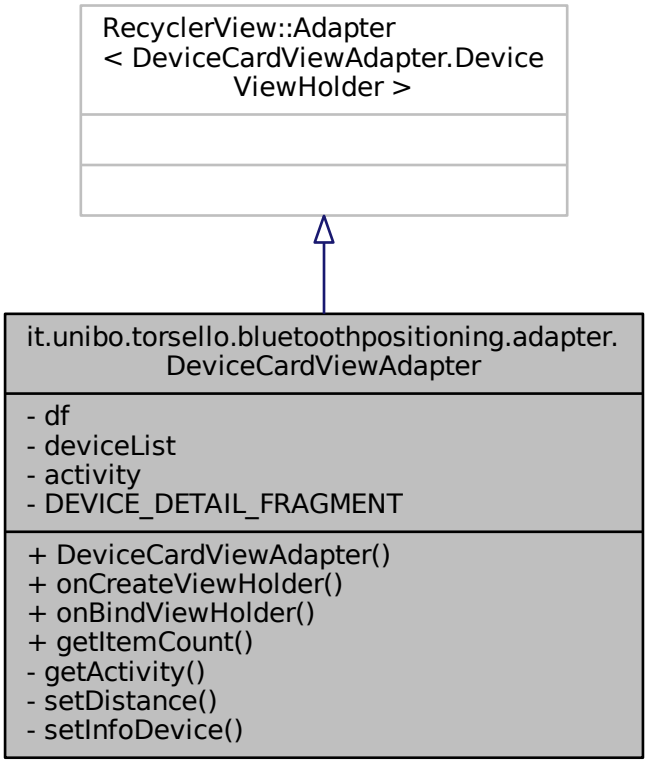
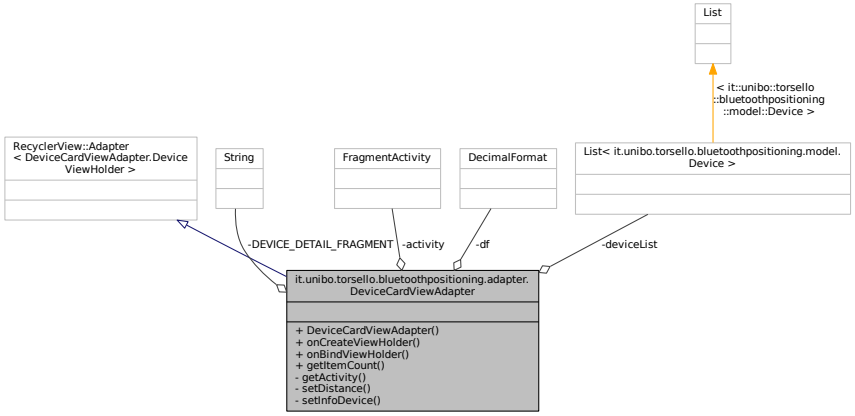


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter:



Composti

- class DeviceViewHolder

Membri pubblici

- DeviceCardViewAdapter (final FragmentActivity fragmentActivity, List< Device > deviceList)
- DeviceViewHolder onCreateViewHolder (ViewGroup parent, int viewType)
- void onBindViewHolder (DeviceViewHolder holder, final int position)
- int getItemCount ()

Membri privati

- FragmentActivity getActivity ()
- void setDistance (DeviceViewHolder holder, Device device)
- void setInfoDevice (DeviceViewHolder holder, Beacon beacon)

Attributi privati

- DecimalFormat df
- List< Device > deviceList
- FragmentActivity activity

Attributi privati statici

- static final String DEVICE_DETAIL_FRAGMENT = "device detail"

6.10.1 Descrizione dettagliata

Created by Federico Torsello. federico.torsello@studio.unibo.it

6.10.2 Documentazione dei costruttori e dei distruttori

6.10.2.1 DeviceCardViewAdapter()

```
it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceCardViewAdapter (
            final FragmentActivity fragmentActivity,
            List< Device > deviceList )
```

```
36                                                                                  {
37
38          this.deviceList = new ArrayList<>();
39          this.deviceList = deviceList;
40          this.activity = fragmentActivity;
41
42          df = new DecimalFormat("0.00", DecimalFormatSymbols.getInstance());
43      }
```

6.10.3 Documentazione delle funzioni membro

6.10.3.1 getActivity()

```
FragmentActivity it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.get↩
Activity ( )  [private]
```

```
45                                          {
46          return activity;
47      }
```

6.10.3.2 getItemCount()

```
int it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.getItemCount ( )
```

```
217                                 {
218          return deviceList.size();
219      }
```

6.10.3.3 onBindViewHolder()

```
void it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.onBindViewHolder (
            DeviceViewHolder holder,
            final int position )
```

```
58                                                                              {
59
60          final Beacon beacon = deviceList.get(position).getBeacon();
61          final Device device = deviceList.get(position);
62
63          setInfoDevice(holder, beacon);
64
65          setDistance(holder, device);
66
67          final Integer imageBeacon = device.getImageBeacon();
68          if (imageBeacon != null) {
69              holder.imageView.setImageResource(imageBeacon);
70          } else {
71              holder.imageView.setImageResource(R.drawable.beacon_unknown);
72          }
73
74          holder.rssiTextView.setText(String.format("%sdb", beacon.getTxPower()));
75
```

```
76          holder.txPowerTextView.setText(String.format("%sdb", beacon.getRssi()));
77
78          final String friendlyName = device.getFriendlyName();
79          if (friendlyName != null) {
80              holder.friendlyNameTextView.setText(friendlyName);
81          } else {
82              holder.friendlyNameTextView.setText(android.R.string.unknownName);
83          }
84
85          final String bluetoothName = beacon.getBluetoothName();
86          if (bluetoothName != null) {
87              holder.defaultNameTextView.setText(bluetoothName);
88          } else {
89              holder.defaultNameTextView.setText(android.R.string.unknownName);
90          }
91
92          final String macAddress = beacon.getBluetoothAddress();
93          if (macAddress != null) {
94              holder.macTextView.setText(macAddress);
95          } else {
96              holder.macTextView.setText(android.R.string.unknownName);
97          }
98
99          final String proximity = device.getProximity();
100          if (proximity != null) {
101              holder.proximityTextView.setText(proximity);
102          } else {
103              holder.proximityTextView.setText(android.R.string.unknownName);
104          }
105
106          final Integer color = device.getColor();
107          if (color != null) {
108              holder.colorTextView.setText(color);
109          } else {
110              holder.colorTextView.setText(android.R.string.unknownName);
111          }
112
113          holder.view.setOnClickListener(new View.OnClickListener() {
114              @Override
115              public void onClick(View v) {
116                  final String deviceDetailName;
117                  if (device.getFriendlyName() != null) {
118                      deviceDetailName = device.getFriendlyName();
119                  } else {
120                      deviceDetailName = device.getAddress();
121                  }
122
123                  new Thread(new Runnable() {
124                      @Override
125                      public void run() {
126                          getActivity().runOnUiThread(new Runnable() {
127                              @Override
128                              public void run() {
129
130                                  Fragment currentFrag = getActivity().getSupportFragmentManager()
131                                          .findFragmentByTag(
    DEVICE_DETAIL_FRAGMENT);
132
133                                  if (currentFrag == null) {
134                                      getActivity().getSupportFragmentManager().beginTransaction()
135                                              .replace(R.id.contentMainLayout,
136                                                      DeviceDetailFragment.newInstance(deviceDetailName),
137                                                      DEVICE_DETAIL_FRAGMENT)
138                                              .commit();
139                                  }
140                              }
141                          });
142                      }
143                  }).start();
144              }
145          });
146      }
```

6.10.3.4 onCreateViewHolder()

```
DeviceViewHolder it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.on↩
CreateViewHolder (
            ViewGroup parent,
            int viewType )

50                                                                                  {
```

```
51
52          View root = LayoutInflater.from(parent.getContext())
53                  .inflate(R.layout.list_items, parent, false);
54          return new DeviceViewHolder(root);
55      }
```

6.10.3.5 setDistance()

```
void it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.setDistance (
              DeviceViewHolder holder,
              Device device )  [private]
```

```
148                                                                     {
149         holder.altbeaconDistanceTextView.setText(String.format("%sm", df.format(device.
    getAltBeaconDistance())));
150
151         holder.standardRawDistanceTextView.setText(String.format("%sm", df.format(device.getRawDistance()
    )));
152
153         holder.kalmanFilterDistanceTextView.setText(String.format("%s", df.format(device.
    getKalmanFilterDistance())));
154       }
```

6.10.3.6 setInfoDevice()

```
void it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.setInfoDevice (
              DeviceViewHolder holder,
              Beacon beacon )  [private]
```

```
156                                                                       {
157         if (beacon.getServiceUuid() == 0xfeaa) {
158
159             holder.visibilityUUIDLinearLayout.setVisibility(View.GONE);
160             holder.visibilityNameSpaceLinearLayout.setVisibility(View.VISIBLE);
161
162             if (beacon.getBeaconTypeCode() == 0x00) {
163                 // Eddystone-UID
164                 if (beacon.getId1() != null) {
165                     holder.nameSpaceTextView.setText(beacon.getId1().toString());
166                 } else {
167                     holder.nameSpaceTextView.setText(android.R.string.unknownName);
168                 }
169
170                 if (beacon.getId2() != null) {
171                     holder.instanceTextView.setText(beacon.getId2().toString());
172                 } else {
173                     holder.instanceTextView.setText(android.R.string.unknownName);
174                 }
175
176             } else if (beacon.getBeaconTypeCode() == 0x10) {
177                 // Eddystone-URL
178                 // String url = UrlBeaconUrlCompressor.uncompress(beacon.getId1().toByteArray());
179             } else if (beacon.getBeaconTypeCode() == 0x20) {
180                 if (!beacon.getExtraDataFields().isEmpty()) {
181                     // Eddystone-TLM
182                 }
183             }
184         } else if (beacon.getServiceUuid() == 0xbeac) {
185             // AltBeacon
186         } else if (beacon.getBeaconTypeCode() == 0x0215) { //533 in dec)
187
188             holder.visibilityUUIDLinearLayout.setVisibility(View.VISIBLE);
189             holder.visibilityNameSpaceLinearLayout.setVisibility(View.GONE);
190
191             // AppleIBeacon
192             if (beacon.getId1() != null) {
193                 holder.uuidTextView.setText(beacon.getId1().toString());
194             } else {
195                 holder.uuidTextView.setText(android.R.string.unknownName);
196             }
197
198             if (beacon.getId2() != null) {
199                 holder.majorTextView.setText(beacon.getId2().toString());
200             } else {
```

```
201                 holder.majorTextView.setText(android.R.string.unknownName);
202             }
203
204             if (beacon.getId3() != null) {
205                 holder.minorTextView.setText(beacon.getId3().toString());
206             } else {
207                 holder.minorTextView.setText(android.R.string.unknownName);
208             }
209
210         } else if (beacon.getBeaconTypeCode() == 0x0101) {
211             // EstimoteNearable
212         }
213
214     }
```

6.10.4 Documentazione dei membri dato

6.10.4.1 activity

```
FragmentActivity it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.activity
[private]
```

6.10.4.2 DEVICE_DETAIL_FRAGMENT

```
final String it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DEVICE_DET↩
AIL_FRAGMENT = "device detail"  [static], [private]
```

6.10.4.3 deviceList

```
List<Device> it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.deviceList
[private]
```

6.10.4.4 df

```
DecimalFormat it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.df  [private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- DeviceCardViewAdapter.java

6.11 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceChartFragment

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceChartFragment

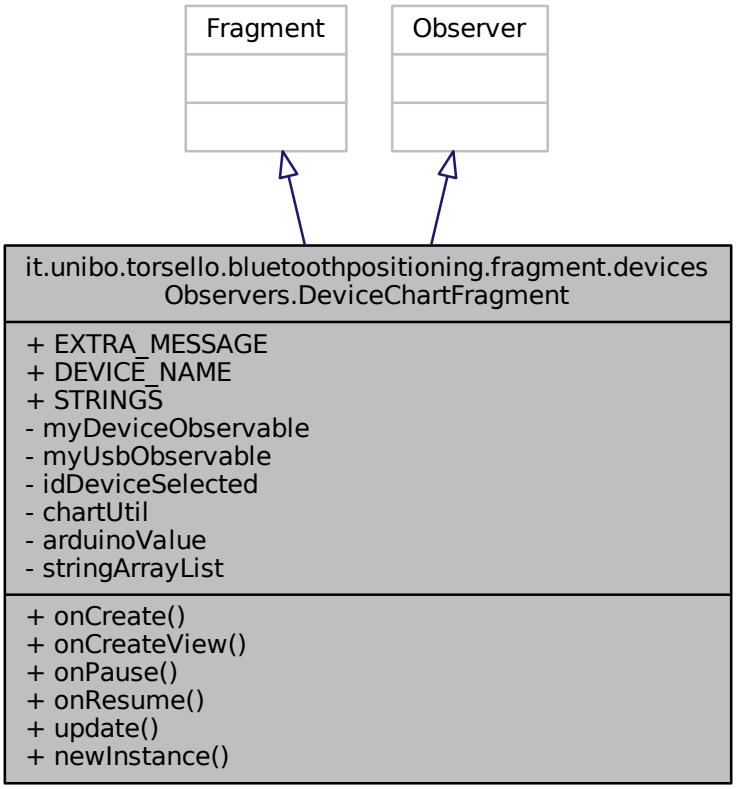
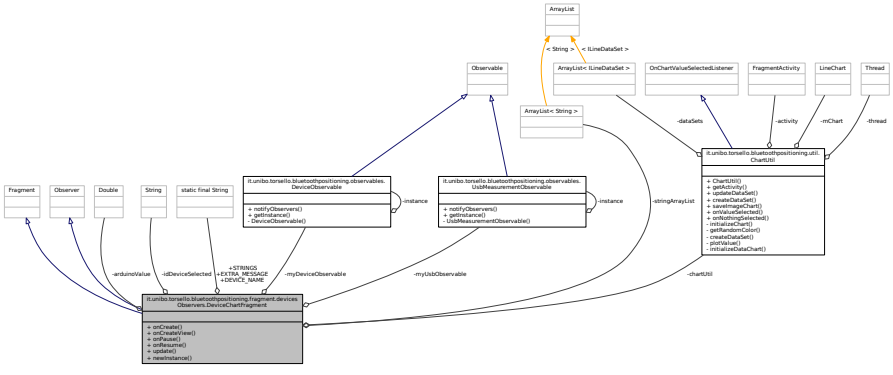


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceChartFragment:



Membri pubblici

- void onCreate (Bundle savedInstanceState)
- View onCreateView (LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
- void onPause ()
- void onResume ()
- void update (Observable o, Object arg)

Membri pubblici statici

- static DeviceChartFragment newInstance (String message, String deviceName, ArrayList< String > strings)

Attributi pubblici statici

- static final String EXTRA_MESSAGE = "EXTRA_MESSAGE"
- static final String DEVICE_NAME = "DEVICE_NAME"
- static final String STRINGS = "STRINGS"

Attributi privati

- DeviceObservable myDeviceObservable
- UsbMeasurementObservable myUsbObservable
- String idDeviceSelected
- ChartUtil chartUtil
- Double arduinoValue = 0D
- ArrayList< String > stringArrayList

6.11.1 Documentazione delle funzioni membro

6.11.1.1 newInstance()

```
static DeviceChartFragment it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.↩
DeviceChartFragment.newInstance (
            String message,
            String deviceName,
            ArrayList< String > strings )  [static]
```

```
40                                                                          {
41         DeviceChartFragment fragment = new DeviceChartFragment();
42         Bundle args = new Bundle();
43         args.putString(EXTRA_MESSAGE, message);
44         args.putString(DEVICE_NAME, deviceName);
45         args.putStringArrayList(STRINGS, strings);
46         fragment.setArguments(args);
47         return fragment;
48     }
```

6.11.1.2 onCreate()

```
void it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceChartFragment.on↩
Create (
            Bundle savedInstanceState )
```

```
51                                                  {
52         super.onCreate(savedInstanceState);
53
54         myDeviceObservable = DeviceObservable.getInstance();
55         myUsbObservable = UsbMeasurementObservable.getInstance();
56
57         idDeviceSelected = getArguments().getString(DEVICE_NAME);
58         stringArrayList = getArguments().getStringArrayList(
      STRINGS);
59     }
```

6.11.1.3 onCreateView()

```
View it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceChartFragment.on↩
CreateView (
            LayoutInflater inflater,
            ViewGroup container,
            Bundle savedInstanceState )
```

```
62                                                                                                  {
63
64         View root = inflater.inflate(R.layout.fragment_device_chart, container, false);
65
66         LineChart lineChart = (LineChart) root.findViewById(R.id.chart);
67
68         // add the charts
69         chartUtil = new ChartUtil(getActivity(), lineChart);
70
71         return root;
72     }
```

6.11.1.4 onPause()

```
void it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceChartFragment.on↩
Pause ( )
```

```
75                          {
76         myDeviceObservable.deleteObserver(this);
77         myUsbObservable.deleteObserver(this);
78         super.onPause();
79     }
```

6.11.1.5 onResume()

```
void it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceChartFragment.on↩
Resume ( )
```

```
82                           {
83         super.onResume();
84         myDeviceObservable.addObserver(this);
85         myUsbObservable.addObserver(this);
86     }
```

6.11.1.6 update()

```
void it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceChartFragment.↩
update (
            Observable o,
            Object arg )
```

```
89                                                  {
90
91          if (o instanceof UsbMeasurementObservable) {
92              if (arg instanceof Double) {
93                  arduinoValue = (Double) arg;
94              }
95          }
96
97          if (o instanceof DeviceObservable) {
98              if (arg instanceof List) {
99
100                 List<Device> devices = (List<Device>) arg;
101
102                 for (Device deviceSelected : devices) {
103                     if (deviceSelected.getFriendlyName().equals(idDeviceSelected) ||
104                             deviceSelected.getAddress().equals(idDeviceSelected)) {
105
106                         if (chartUtil != null) {
107
108                             chartUtil.createDataSet(
      stringArrayList);
109
110                             ArrayList<Double> doubleArrayList = new ArrayList<>();
111
112                             for (String s : stringArrayList) {
113                                 if (s.equals(getString(R.string.chart_arduino))) {
114                                     doubleArrayList.add(arduinoValue);
115                                 }
116
117                                 if (s.equals(getString(R.string.chart_raw_distance))) {
118                                     doubleArrayList.add(deviceSelected.getRawDistance());
119                                 }
120
121                                 if (s.equals(getString(R.string.chart_altbeacon))) {
122                                     doubleArrayList.add(deviceSelected.getAltBeaconDistance());
123                                 }
124
125                                 if (s.equals(getString(R.string.chart_kalman_filter))) {
126                                     doubleArrayList.add(deviceSelected.getKalmanFilterDistance());
127                                 }
128                             }
129
130                             chartUtil.updateDataSet(doubleArrayList);
131                         }
132                     }
133                 }
134             }
135         }
136     }
```

6.11.2 Documentazione dei membri dato

6.11.2.1 arduinoValue

```
Double it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceChartFragment.↩
arduinoValue = 0D  [private]
```

6.11.2.2 chartUtil

```
ChartUtil it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceChartFragment.↩
chartUtil  [private]
```

6.11.2.3 DEVICE_NAME

```
final String it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceChart↩
Fragment.DEVICE_NAME = "DEVICE_NAME"  [static]
```

6.11.2.4 EXTRA_MESSAGE

```
final String it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceChart↩
Fragment.EXTRA_MESSAGE = "EXTRA_MESSAGE"  [static]
```

6.11.2.5 idDeviceSelected

```
String it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceChartFragment.↩
idDeviceSelected  [private]
```

6.11.2.6 myDeviceObservable

```
DeviceObservable it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.Device↩
ChartFragment.myDeviceObservable  [private]
```

6.11.2.7 myUsbObservable

```
UsbMeasurementObservable it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.↩
DeviceChartFragment.myUsbObservable  [private]
```

6.11.2.8 stringArrayList

```
ArrayList<String> it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.Device↩
ChartFragment.stringArrayList  [private]
```

6.11.2.9 STRINGS

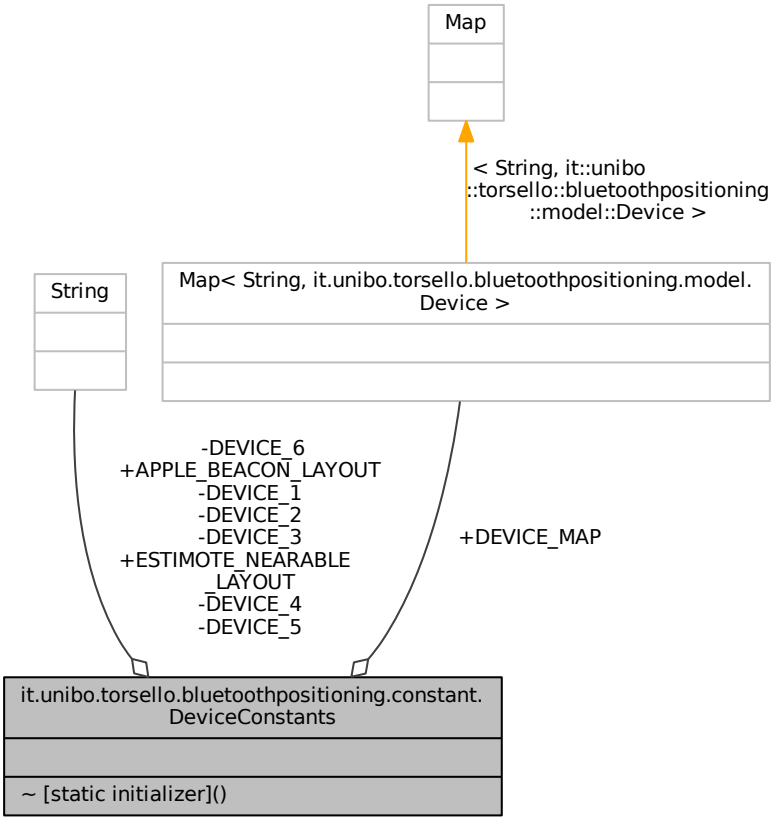
```
final String it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceChart↩
Fragment.STRINGS = "STRINGS"  [static]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- DeviceChartFragment.java

6.12 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.constant.DeviceConstants

Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.constant.DeviceConstants:



Attributi pubblici statici

- static final String APPLE_BEACON_LAYOUT = "m:2-3=0215,i:4-19,i:20-21,i:22-23,p:24-24"
- static final String ESTIMOTE_NEARABLE_LAYOUT
- static final Map< String, Device > DEVICE_MAP

Funzioni statiche con visibilità di package

- [static initializer]

Attributi privati statici

- static final String DEVICE_1 = "C1:9B:B0:B9:01:9E"
- static final String DEVICE_2 = "D1:BE:E2:E9:67:A6"
- static final String DEVICE_3 = "FA:6B:72:1E:EB:46"
- static final String DEVICE_4 = "D9:80:00:B7:16:78"
- static final String DEVICE_5 = "DB:F6:F5:0C:23:BF"
- static final String DEVICE_6 = "E7:E4:0E:F6:79:3F"

6.12.1 Descrizione dettagliata

Created by Federico Torsello. federico.torsello@studio.unibo.it

6.12.2 Documentazione delle funzioni membro

6.12.2.1 [static initializer]()

```
it.unibo.torsello.bluetoothpositioning.constant.DeviceConstants.[static initializer] ( )  [static],
[package]
```

6.12.3 Documentazione dei membri dato

6.12.3.1 APPLE_BEACON_LAYOUT

```
final String it.unibo.torsello.bluetoothpositioning.constant.DeviceConstants.APPLE_BEACON_LA↩
YOUT = "m:2-3=0215,i:4-19,i:20-21,i:22-23,p:24-24"  [static]
```

6.12.3.2 DEVICE_1

```
final String it.unibo.torsello.bluetoothpositioning.constant.DeviceConstants.DEVICE_1 = "C1↩
:9B:B0:B9:01:9E"  [static], [private]
```

6.12.3.3 DEVICE_2

```
final String it.unibo.torsello.bluetoothpositioning.constant.DeviceConstants.DEVICE_2 = "D1:↩
BE:E2:E9:67:A6"  [static], [private]
```

6.12.3.4 DEVICE_3

```
final String it.unibo.torsello.bluetoothpositioning.constant.DeviceConstants.DEVICE_3 = "FA↩
:6B:72:1E:EB:46"  [static], [private]
```

6.12.3.5 DEVICE_4

```
final String it.unibo.torsello.bluetoothpositioning.constant.DeviceConstants.DEVICE_4 = "D9↩
:80:00:B7:16:78"  [static], [private]
```

6.12.3.6 DEVICE_5

```
final String it.unibo.torsello.bluetoothpositioning.constant.DeviceConstants.DEVICE_5 = "DB:↩
F6:F5:0C:23:BF"  [static], [private]
```

6.12.3.7 DEVICE_6

```
final String it.unibo.torsello.bluetoothpositioning.constant.DeviceConstants.DEVICE_6 = "E7:↩
E4:0E:F6:79:3F"  [static], [private]
```

6.12.3.8 DEVICE_MAP

```
final Map<String, Device> it.unibo.torsello.bluetoothpositioning.constant.DeviceConstants.DE↩
VICE_MAP  [static]
```

6.12.3.9 ESTIMOTE_NEARABLE_LAYOUT

```
final String it.unibo.torsello.bluetoothpositioning.constant.DeviceConstants.ESTIMOTE_NEARAB↩
LE_LAYOUT  [static]
```

Valore iniziale:

```
= "m:1-2=0101,i:3-10,d:11-11,d:12-12," +
            "d:13-14,d:15-15,d:16-16,d:17-17,d:18-18,d:19-19,d:20-20, p:21-21"
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- DeviceConstants.java

6.13 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailFragment

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailFragment

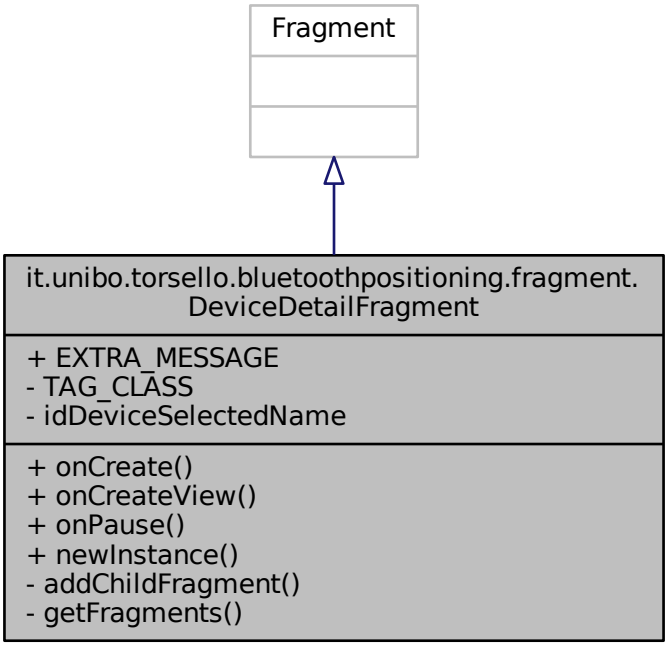
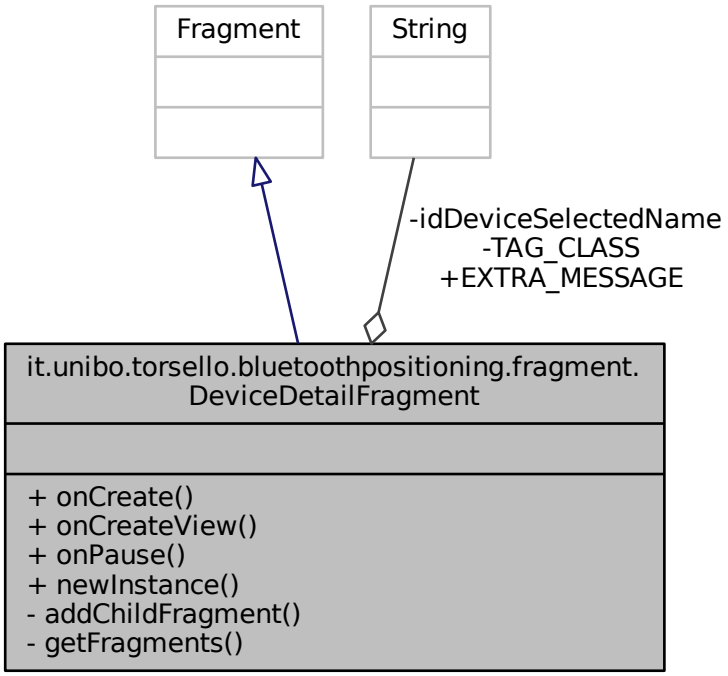


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailFragment:



Membri pubblici

- void onCreate (Bundle savedInstanceState)
- View onCreateView (LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
- void onPause ()

Membri pubblici statici

- static DeviceDetailFragment newInstance (String message)

Attributi pubblici statici

- static final String EXTRA_MESSAGE = "EXTRA_MESSAGE"

Membri privati

- void addChildFragment (View root)
- ArrayList< Fragment > getFragments ()

Attributi privati

- final String TAG_CLASS = getClass().getSimpleName()
- String idDeviceSelectedName

6.13.1 Descrizione dettagliata

Created by Federico Torsello. federico.torsello@studio.unibo.it

6.13.2 Documentazione delle funzioni membro

6.13.2.1 addChildFragment()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailFragment.addChildFragment (
            View root )  [private]
```

```
63                                          {
64
65          ViewPager mViewPager = (ViewPager) root.findViewById(R.id.view_pager);
66          StatePagerAdapter myPageAdapter = new StatePagerAdapter(getChildFragmentManager(),
    getFragments());
67          mViewPager.setAdapter(myPageAdapter);
68
69          TabLayout tabLayout = (TabLayout) root.findViewById(R.id.sliding_tabs);
70          tabLayout.setupWithViewPager(mViewPager);
71      }
```

6.13.2.2 getFragments()

```
ArrayList<Fragment> it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailFragment.get↩
Fragments ( )  [private]
```

```
73                                              {
74          ArrayList<Fragment> fragments = new ArrayList<>();
75          // fragment 0
76          fragments.add(DeviceDetailInner1Fragment.newInstance(
    idDeviceSelectedName));
77
78          // fragment 1
79          fragments.add(DeviceDetailInner2Fragmet.newInstance("Details",
    idDeviceSelectedName));
80
81          return fragments;
82      }
```

6.13.2.3 newInstance()

```
static DeviceDetailFragment it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetail↩
Fragment.newInstance (
            String message )  [static]
```

```
29                                                                  {
30          DeviceDetailFragment fragment = new DeviceDetailFragment();
31          Bundle args = new Bundle();
32          args.putString(EXTRA_MESSAGE, message);
33          fragment.setArguments(args);
34          return fragment;
35      }
```

6.13.2.4 onCreate()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailFragment.onCreate (
             Bundle savedInstanceState )
```

```
38                                                {
39         super.onCreate(savedInstanceState);
40
41         idDeviceSelectedName = getArguments().getString(
      EXTRA_MESSAGE);
42     }
```

6.13.2.5 onCreateView()

```
View it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailFragment.onCreateView (
             LayoutInflater inflater,
             ViewGroup container,
             Bundle savedInstanceState )
```

```
45                                                                                                {
46         View root = inflater.inflate(R.layout.fragment_device_detail, container, false);
47
48         getActivity().findViewById(R.id.toolbar).setVisibility(View.GONE);
49
50         ((CollapsingToolbarLayout) root.findViewById(R.id.collapsing_toolbar)).setTitle(
      idDeviceSelectedName);
51
52         addChildFragment(root);
53
54         return root;
55     }
```

6.13.2.6 onPause()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailFragment.onPause ( )
```

```
58                          {
59         getActivity().findViewById(R.id.toolbar).setVisibility(View.VISIBLE);
60         super.onPause();
61     }
```

6.13.3 Documentazione dei membri dato

6.13.3.1 EXTRA_MESSAGE

```
final String it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailFragment.EXTRA_MESS↩
AGE = "EXTRA_MESSAGE"  [static]
```

6.13.3.2 idDeviceSelectedName

```
String it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailFragment.idDeviceSelected↩
Name  [private]
```

6.13.3.3 TAG_CLASS

```
final String it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailFragment.TAG_CLASS =
getClass().getSimpleName()  [private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- DeviceDetailFragment.java

6.14 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceDetailInner1Fragment

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceDetailInner1Fragment

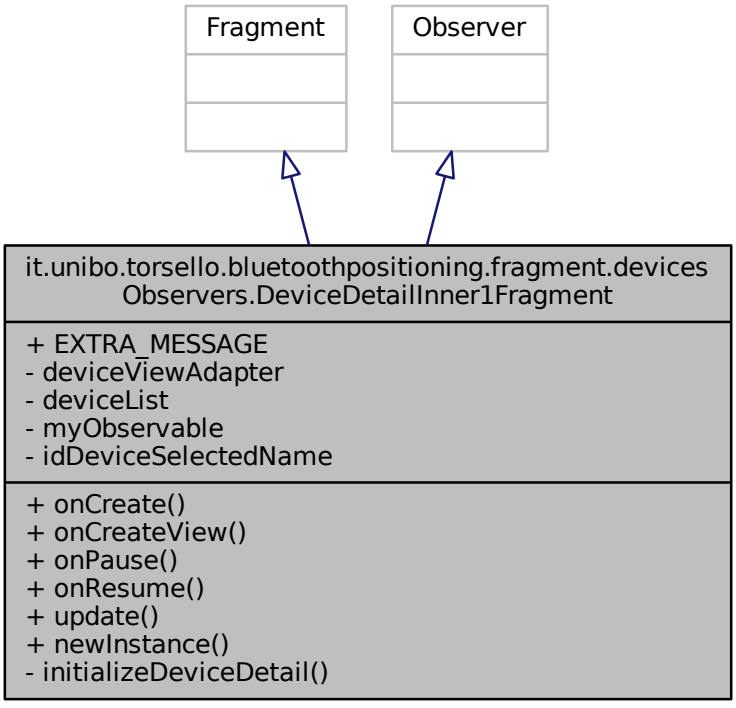
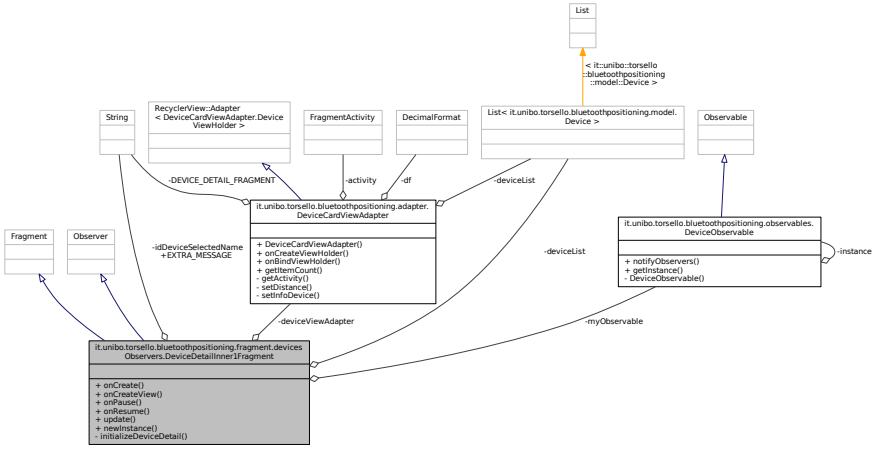


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceDetail↩
Inner1Fragment:



Membri pubblici

- void onCreate (@Nullable Bundle savedInstanceState)
- View onCreateView (LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
- void onPause ()
- void onResume ()
- void update (Observable o, Object arg)

Membri pubblici statici

- static DeviceDetailInner1Fragment newInstance (String message)

Attributi pubblici statici

- static final String EXTRA_MESSAGE = "EXTRA_MESSAGE"

Membri privati

- void initializeDeviceDetail (View root)

Attributi privati

- DeviceCardViewAdapter deviceViewAdapter
- List< Device > deviceList
- DeviceObservable myObservable
- String idDeviceSelectedName

6.14.1 Descrizione dettagliata

Created by Federico Torsello. federico.torsello@studio.unibo.it

6.14.2 Documentazione delle funzioni membro

6.14.2.1 initializeDeviceDetail()

```
void it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceDetailInner1↩
Fragment.initializeDeviceDetail (
            View root )  [private]
```

```
77                                                  {
78         // add RecyclerView
79         RecyclerView recyclerView = (RecyclerView) root.findViewById(R.id.recycler_view);
80         recyclerView.setLayoutManager(new LinearLayoutManager(getContext()));
81         recyclerView.setAdapter(deviceViewAdapter);
82     }
```

6.14.2.2 newInstance()

```
static DeviceDetailInner1Fragment it.unibo.torsello.bluetoothpositioning.fragment.devices↩
Observers.DeviceDetailInner1Fragment.newInstance (
            String message )  [static]
```

```
36                                                                          {
37         DeviceDetailInner1Fragment fragment = new DeviceDetailInner1Fragment();
38         Bundle args = new Bundle();
39         args.putString(EXTRA_MESSAGE, message);
40         fragment.setArguments(args);
41         return fragment;
42     }
```

6.14.2.3 onCreate()

```
void it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceDetailInner1↩
Fragment.onCreate (
            @Nullable Bundle savedInstanceState )
```

```
45                                                          {
46         super.onCreate(savedInstanceState);
47
48         myObservable = DeviceObservable.getInstance();
49
50         idDeviceSelectedName = getArguments().getString(
      EXTRA_MESSAGE);
51         deviceList = new ArrayList<>();
52         deviceViewAdapter = new DeviceCardViewAdapter(getActivity(),
      deviceList);
53
54     }
```

6.14.2.4 onCreateView()

```
View it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceDetailInner1↩
Fragment.onCreateView (
            LayoutInflater inflater,
            ViewGroup container,
            Bundle savedInstanceState )
```

```
57                                                                                  {
58         View root = inflater.inflate(R.layout.fragment_device_detail_inner_1, container, false);
59
60         initializeDeviceDetail(root);
61
62         return root;
63     }
```

6.14.2.5 onPause()

```
void it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceDetailInner1↩
Fragment.onPause ( )
```

```
66                          {
67         myObservable.deleteObserver(this);
68         super.onPause();
69     }
```

6.14.2.6 onResume()

```
void it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceDetailInner1↩
Fragment.onResume ( )
```

```
72                           {
73         super.onResume();
74         myObservable.addObserver(this);
75     }
```

6.14.2.7 update()

```
void it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceDetailInner1↩
Fragment.update (
              Observable o,
              Object arg )
```

```
85                                                {
86
87         if (arg instanceof List) {
88
89             if (!deviceList.isEmpty()) {
90                 deviceList.clear();
91             }
92
93             List<Device> devices = (List<Device>) arg;
94
95             for (Device deviceSelected : devices) {
96                 if (deviceSelected.getFriendlyName().equals(idDeviceSelectedName) ||
97                         deviceSelected.getAddress().equals(idDeviceSelectedName)) {
98                     deviceList.add(deviceSelected);
99                 }
100             }
101
102             deviceViewAdapter.notifyDataSetChanged();
103         }
104     }
```

6.14.3 Documentazione dei membri dato

6.14.3.1 deviceList

```
List<Device> it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceDetail↩
Inner1Fragment.deviceList  [private]
```

6.14.3.2 deviceViewAdapter

```
DeviceCardViewAdapter it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.↩
DeviceDetailInner1Fragment.deviceViewAdapter  [private]
```

6.14.3.3 EXTRA_MESSAGE

```
final String it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceDetail↩
Inner1Fragment.EXTRA_MESSAGE = "EXTRA_MESSAGE"  [static]
```

6.14.3.4 idDeviceSelectedName

```
String it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceDetailInner1↩
Fragment.idDeviceSelectedName  [private]
```

6.14.3.5 myObservable

```
DeviceObservable it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.Device↩
DetailInner1Fragment.myObservable  [private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- DeviceDetailInner1Fragment.java

6.15 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragmet

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragmet

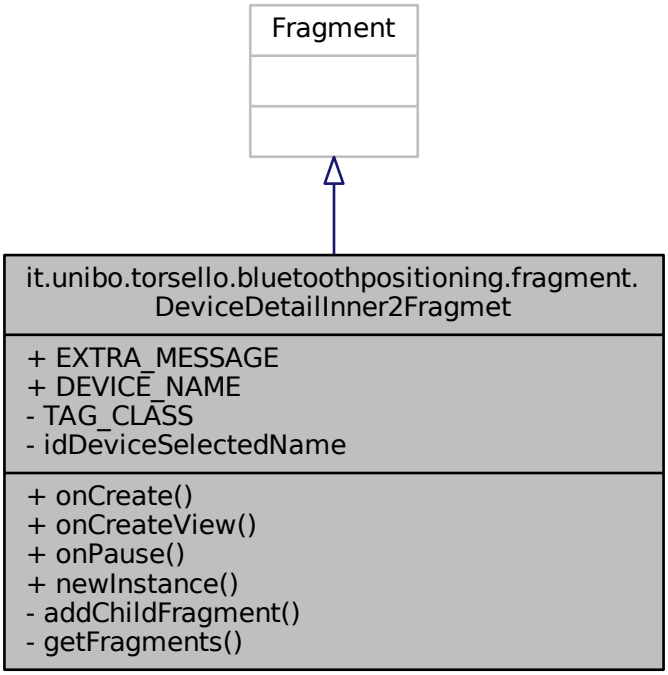
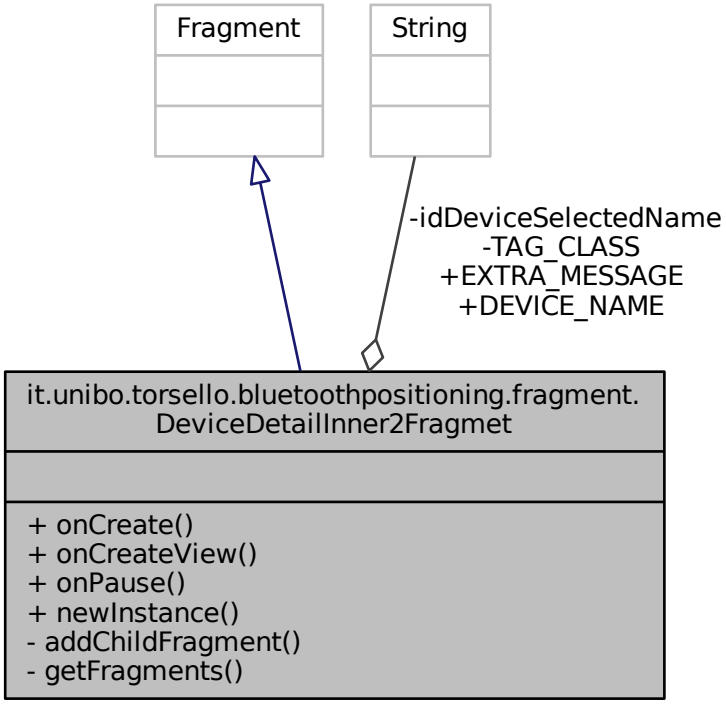


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragmet:



Membri pubblici

- void onCreate (Bundle savedInstanceState)
- View onCreateView (LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
- void onPause ()

Membri pubblici statici

- static DeviceDetailInner2Fragmet newInstance (String message, String deviceName)

Attributi pubblici statici

- static final String EXTRA_MESSAGE = "EXTRA_MESSAGE"
- static final String DEVICE_NAME = "DEVICE_NAME"

Membri privati

- void addChildFragment (View root)
- ArrayList< Fragment > getFragments ()

Attributi privati

- final String TAG_CLASS = getClass().getSimpleName()
- String idDeviceSelectedName

6.15.1 Descrizione dettagliata

Created by federico on 03/10/16.

6.15.2 Documentazione delle funzioni membro

6.15.2.1 addChildFragment()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragmet.addChild↩
Fragment (
            View root )  [private]
```

```
59                                          {
60
61         ViewPager mViewPager = (ViewPager) root.findViewById(R.id.view_pager);
62         StatePagerAdapter myPageAdapter = new StatePagerAdapter(getChildFragmentManager(),
       getFragments());
63         mViewPager.setAdapter(myPageAdapter);
64
65         TabLayout tabLayout = (TabLayout) root.findViewById(R.id.sliding_tabs);
66         tabLayout.setupWithViewPager(mViewPager);
67     }
```

6.15.2.2 getFragments()

```
ArrayList<Fragment> it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragmet.↩
getFragments ( )  [private]
```

```
69                                              {
70         ArrayList<Fragment> fragments = new ArrayList<>();
71
72         // inner fragment 0
73         ArrayList<String> params1 = new ArrayList<>();
74         params1.add(getString(R.string.chart_arduino));
75         params1.add(getString(R.string.chart_raw_distance));
76         params1.add(getString(R.string.chart_altbeacon));
77         params1.add(getString(R.string.chart_kalman_filter));
78
79         fragments.add(DeviceChartFragment.newInstance("chart1",
       idDeviceSelectedName, params1));
80
81         // inner fragment 1
82         ArrayList<String> params2 = new ArrayList<>();
83         params2.add(getString(R.string.chart_arduino));
84         params2.add(getString(R.string.chart_raw_distance));
85         params2.add(getString(R.string.chart_kalman_filter));
86
87         fragments.add(DeviceChartFragment.newInstance("chart2",
       idDeviceSelectedName, params2));
88
89         // inner fragment 2
90         ArrayList<String> params3 = new ArrayList<>();
91         params3.add(getString(R.string.chart_arduino));
92         params3.add(getString(R.string.chart_altbeacon));
93         params3.add(getString(R.string.chart_kalman_filter));
94
95         fragments.add(DeviceChartFragment.newInstance("chart3",
       idDeviceSelectedName, params3));
96
97         return fragments;
98     }
```

6.15.2.3 newInstance()

```
static DeviceDetailInner2Fragmet it.unibo.torsello.bluetoothpositioning.fragment.Device↩
DetailInner2Fragmet.newInstance (
            String message,
            String deviceName )  [static]
```

```
29                                                                                  {
30         DeviceDetailInner2Fragmet fragment = new DeviceDetailInner2Fragmet();
31         Bundle args = new Bundle();
32         args.putString(EXTRA_MESSAGE, message);
33         args.putString(DEVICE_NAME, deviceName);
34         fragment.setArguments(args);
35         return fragment;
36     }
```

6.15.2.4 onCreate()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragmet.onCreate (
            Bundle savedInstanceState )
```

```
39                                                {
40         super.onCreate(savedInstanceState);
41
42         idDeviceSelectedName = getArguments().getString(
      DEVICE_NAME);
43     }
```

6.15.2.5 onCreateView()

```
View it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragmet.onCreateView (
            LayoutInflater inflater,
            ViewGroup container,
            Bundle savedInstanceState )
```

```
46                                                                                  {
47         View root = inflater.inflate(R.layout.fragment_device_detail_inner_2, container, false);
48
49         addChildFragment(root);
50
51         return root;
52     }
```

6.15.2.6 onPause()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragmet.onPause ( )
```

```
55                            {
56         super.onPause();
57     }
```

6.15.3 Documentazione dei membri dato

6.15.3.1 DEVICE_NAME

```
final String it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragmet.DEVIC↩
E_NAME = "DEVICE_NAME"  [static]
```

6.15.3.2 EXTRA_MESSAGE

```
final String it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragmet.EXTRA↩
_MESSAGE = "EXTRA_MESSAGE"  [static]
```

6.15.3.3 idDeviceSelectedName

```
String it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragmet.idDevice↩
SelectedName  [private]
```

6.15.3.4 TAG_CLASS

```
final String it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragmet.TAG_C↩
LASS = getClass().getSimpleName()  [private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- DeviceDetailInner2Fragmet.java

6.16 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceListFragment

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceListFragment

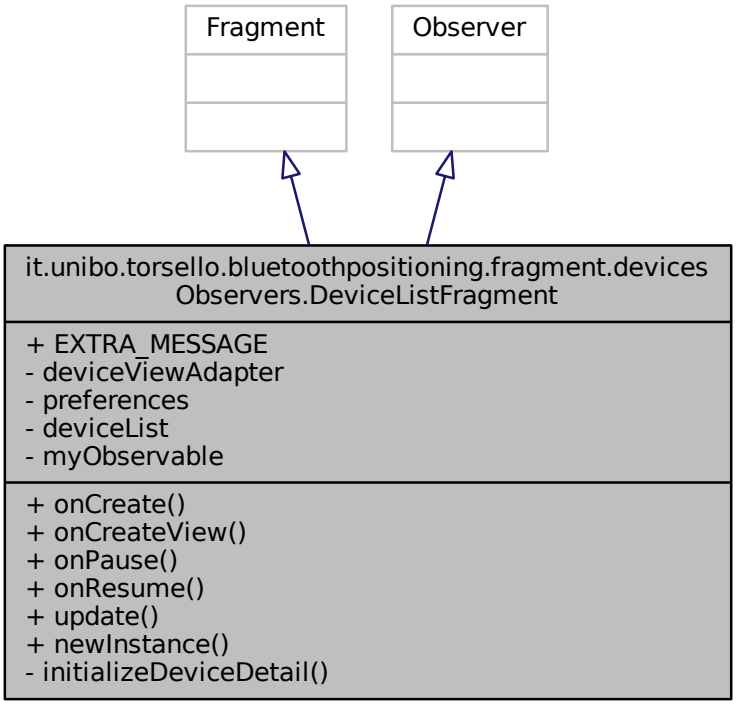
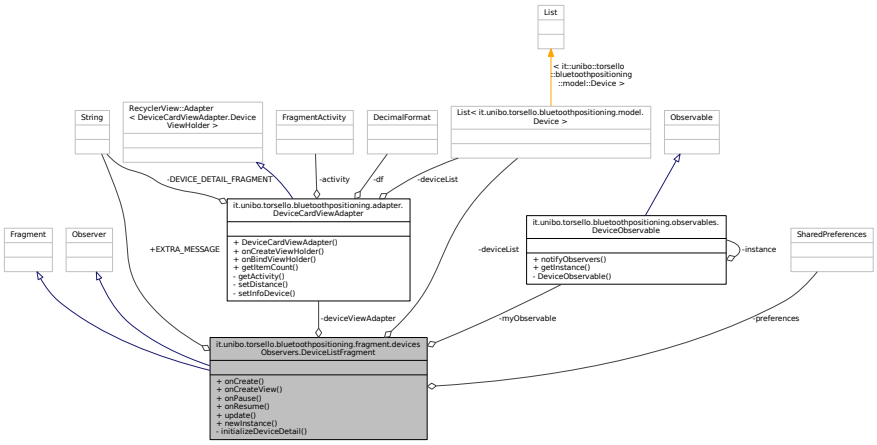


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceList↩
Fragment:



Membri pubblici

- void onCreate (@Nullable Bundle savedInstanceState)
- View onCreateView (LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
- void onPause ()
- void onResume ()
- void update (Observable o, Object arg)

Membri pubblici statici

- static DeviceListFragment newInstance ()

Attributi pubblici statici

- static final String EXTRA_MESSAGE = "EXTRA_MESSAGE"

Membri privati

- void initializeDeviceDetail (View root)

Attributi privati

- DeviceCardViewAdapter deviceViewAdapter
- SharedPreferences preferences
- List< Device > deviceList
- DeviceObservable myObservable

6.16.1 Descrizione dettagliata

Created by Federico Torsello. federico.torsello@studio.unibo.it

6.16.2 Documentazione delle funzioni membro

6.16.2.1 initializeDeviceDetail()

```
void it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceListFragment.↩
initializeDeviceDetail (
            View root )  [private]
```

```
84                                                  {
85         // add RecyclerView
86         RecyclerView recyclerView = (RecyclerView) root.findViewById(R.id.recycler_view);
87         recyclerView.setLayoutManager(new LinearLayoutManager(getContext()));
88         deviceViewAdapter = new DeviceCardViewAdapter(getActivity(),
     deviceList);
89         recyclerView.setAdapter(deviceViewAdapter);
90     }
```

6.16.2.2 newInstance()

```
static DeviceListFragment it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.↩
DeviceListFragment.newInstance ( )  [static]
```

```
56                                                  {
57         DeviceListFragment fragment = new DeviceListFragment();
58         Bundle args = new Bundle();
59         args.putString(EXTRA_MESSAGE, "Scan Device");
60         fragment.setArguments(args);
61         return fragment;
62     }
```

6.16.2.3 onCreate()

```
void it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceListFragment.on↩
Create (
            @Nullable Bundle savedInstanceState )
```

```
65                                                          {
66         super.onCreate(savedInstanceState);
67
68         myObservable = DeviceObservable.getInstance();
69
70         deviceList = new ArrayList<>();
71         preferences = getActivity().getSharedPreferences(SettingConstants.SETTINGS_PREFERENCES,
     0);
72     }
```

6.16.2.4 onCreateView()

```
View it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceListFragment.on↩
CreateView (
            LayoutInflater inflater,
            ViewGroup container,
            Bundle savedInstanceState )
```

```
75                                                                                          {
76
77         View root = inflater.inflate(R.layout.fragment_device_list, container, false);
78
79         initializeDeviceDetail(root);
80
81         return root;
82     }
```

6.16.2.5 onPause()

```
void it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceListFragment.on↩
Pause ( )
```

```
93                          {
94         myObservable.deleteObserver(this);
95         super.onPause();
96     }
```

6.16.2.6 onResume()

```
void it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceListFragment.on↩
Resume ( )
```

```
99                           {
100         super.onResume();
101         myObservable.addObserver(this);
102     }
```

6.16.2.7 update()

```
void it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceListFragment.↩
update (
              Observable o,
              Object arg )
```

```
105                                                 {
106
107         if (arg instanceof List) {
108
109             if (!deviceList.isEmpty()) {
110                 deviceList.clear();
111             }
112
113             List<Device> devices = (List<Device>) arg;
114
115             // optional sorting
116             Collections.sort(devices, new Comparator<Device>() {
117                 public int compare(Device b1, Device b2) {
118                     int sorting = preferences.getInt(SettingConstants.DISTANCE_SORTING, 0);
119                     switch (sorting) {
120                         case 0:
121                         case R.id.radioButton_default_sorting:
122                             return Double.compare(b1.getIndex(), b2.getIndex());
123                         case R.id.radioButton_color_sorting:
124                             return Double.compare(b1.getColor(), b2.getColor());
125                         case R.id.radioButton_distance_sorting:
126                             return Double.compare(b1.getKalmanFilterDistance(), b2.getKalmanFilterDistance(
      ));
127                     } // default sorting (a good basic ordering for the other options)
128                     return Double.compare(b1.getIndex(), b2.getIndex());
129                 }
130             });
131
132             deviceList.addAll(devices);
133             deviceViewAdapter.notifyDataSetChanged();
134         }
135     }
```

6.16.3 Documentazione dei membri dato

6.16.3.1 deviceList

```
List<Device> it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceList↩
Fragment.deviceList  [private]
```

6.16.3.2 deviceViewAdapter

```
DeviceCardViewAdapter it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.↩
DeviceListFragment.deviceViewAdapter  [private]
```

6.16.3.3 EXTRA_MESSAGE

```
final String it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceList↩
Fragment.EXTRA_MESSAGE = "EXTRA_MESSAGE"  [static]
```

6.16.3.4 myObservable

```
DeviceObservable it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceList↩
Fragment.myObservable  [private]
```

6.16.3.5 preferences

```
SharedPreferences it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.Device↩
ListFragment.preferences  [private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- DeviceListFragment.java

6.17 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.observables.DeviceObservable

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.observables.DeviceObservable

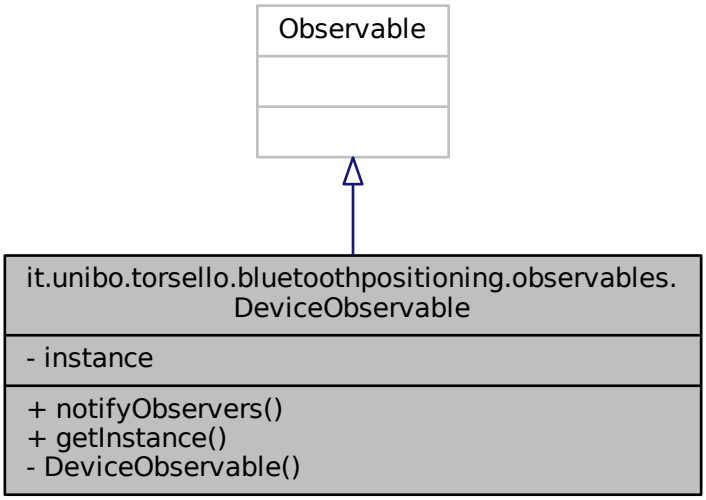
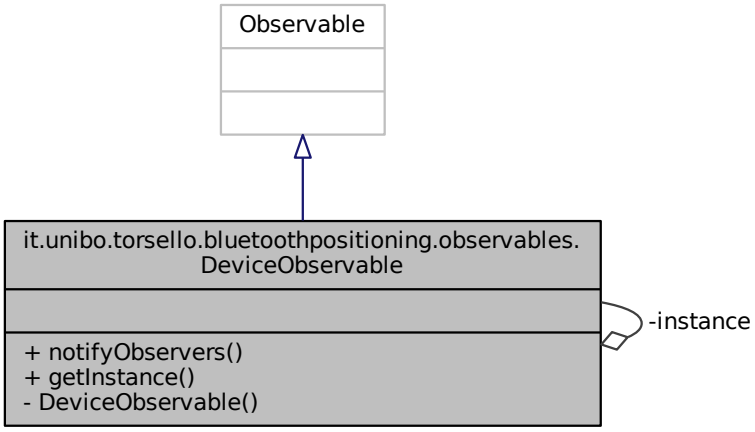


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.observables.DeviceObservable:



Membri pubblici

- void notifyObservers (Object data)

Membri pubblici statici

- static DeviceObservable getInstance ()

Membri privati

- DeviceObservable ()

Attributi privati statici

- static DeviceObservable instance = new DeviceObservable()

6.17.1 Descrizione dettagliata

Created by Federico Torsello. federico.torsello@studio.unibo.it

6.17.2 Documentazione dei costruttori e dei distruttori

6.17.2.1 DeviceObservable()

```
it.unibo.torsello.bluetoothpositioning.observables.DeviceObservable.DeviceObservable ( )  [private]
```

```
17                                  {
18         super();
19     }
```

6.17.3 Documentazione delle funzioni membro

6.17.3.1 getInstance()

```
static DeviceObservable it.unibo.torsello.bluetoothpositioning.observables.DeviceObservable.↩
getInstance ( )  [static]
```

```
13                                              {
14         return instance;
15     }
```

6.17.3.2 notifyObservers()

```
void it.unibo.torsello.bluetoothpositioning.observables.DeviceObservable.notifyObservers (
              Object data )
```

```
22                                              {
23         setChanged();
24         super.notifyObservers(data);
25     }
```

6.17.4 Documentazione dei membri dato

6.17.4.1 instance

```
DeviceObservable it.unibo.torsello.bluetoothpositioning.observables.DeviceObservable.instance
= new DeviceObservable()  [static], [private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- DeviceObservable.java

6.18 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.↩ DeviceViewHolder

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceView↩ Holder

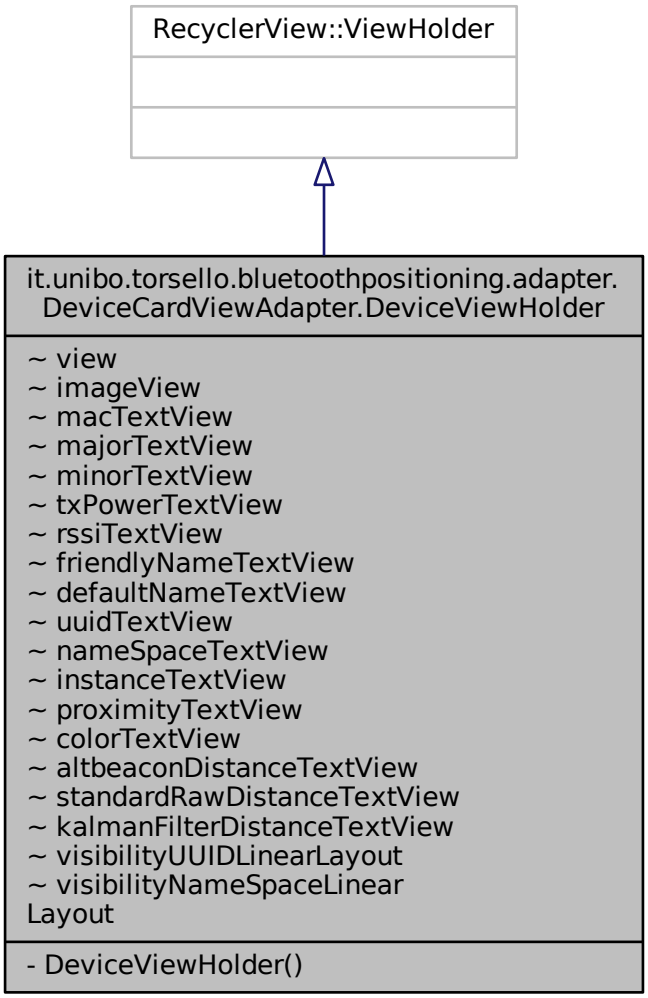
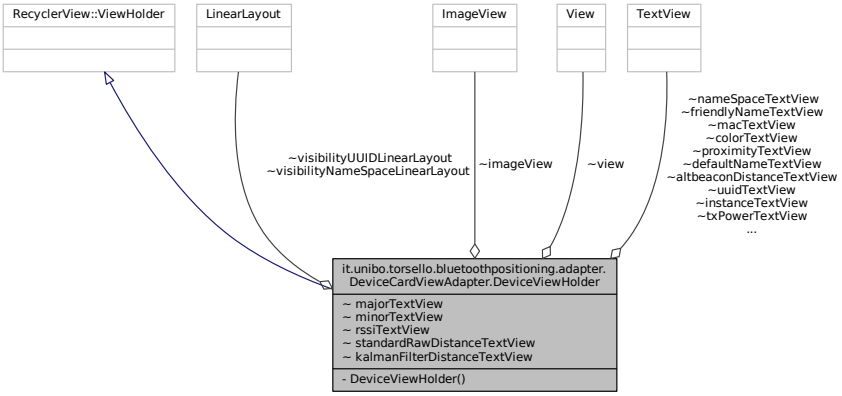


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.Device↩

ViewHolder:



Attributi con visibilità di package

- View view
- ImageView imageView
- TextView macTextView
- TextView majorTextView
- TextView minorTextView
- TextView txPowerTextView
- TextView rssiTextView
- TextView friendlyNameTextView
- TextView defaultNameTextView
- TextView uuidTextView
- TextView nameSpaceTextView
- TextView instanceTextView
- TextView proximityTextView
- TextView colorTextView
- TextView altbeaconDistanceTextView
- TextView standardRawDistanceTextView
- TextView kalmanFilterDistanceTextView
- LinearLayout visibilityUUIDLinearLayout
- LinearLayout visibilityNameSpaceLinearLayout

Membri privati

- DeviceViewHolder (View view)

6.18.1 Documentazione dei costruttori e dei distruttori

6.18.1.1 DeviceViewHolder()

```
it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceViewHolder.Device↩
ViewHolder (
            View view ) [private]
```

```
244                                         {
245
246             super(view);
247             this.view = view;
248             imageView = (ImageView) view.findViewById(R.id.imageBeacon);
249             defaultNameTextView = (TextView) view.findViewById(R.id.
    value_default_name);
250             friendlyNameTextView = (TextView) view.findViewById(R.id.
    value_friendly_name);
251             macTextView = (TextView) view.findViewById(R.id.value_mac_address);
252             majorTextView = (TextView) view.findViewById(R.id.value_major);
253             minorTextView = (TextView) view.findViewById(R.id.value_minor);
254             txPowerTextView = (TextView) view.findViewById(R.id.value_power);
255             rssiTextView = (TextView) view.findViewById(R.id.value_rssi);
256             uuidTextView = (TextView) view.findViewById(R.id.value_uuid);
257             nameSpaceTextView = (TextView) view.findViewById(R.id.value_name_space);
258             proximityTextView = (TextView) view.findViewById(R.id.value_proximity);
259             instanceTextView = (TextView) view.findViewById(R.id.value_instance);
260             colorTextView = (TextView) view.findViewById(R.id.value_color);
261
262             altbeaconDistanceTextView = (TextView) view.findViewById(R.id.
    value_altbeacon_distance);
263             kalmanFilterDistanceTextView = (TextView)
    view.findViewById(R.id.value_kalman_filter_distance);
264             standardRawDistanceTextView = (TextView)
    view.findViewById(R.id.value_standard_raw_distance);
265
266             visibilityUUIDLinearLayout = (LinearLayout)
    view.findViewById(R.id.visibility_uuid_minor_major_nmb);
267             visibilityNameSpaceLinearLayout = (LinearLayout)
    view.findViewById(R.id.visibilityNameSpace_Instance);
268         }
```

6.18.2 Documentazione dei membri dato

6.18.2.1 altbeaconDistanceTextView

```
TextView it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceView↩
Holder.altbeaconDistanceTextView  [package]
```

6.18.2.2 colorTextView

```
TextView it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceView↩
Holder.colorTextView  [package]
```

6.18.2.3 defaultNameTextView

```
TextView it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceView↩
Holder.defaultNameTextView  [package]
```

6.18.2.4 friendlyNameTextView

```
TextView it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceView↩
Holder.friendlyNameTextView  [package]
```

6.18.2.5 imageView

```
ImageView it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceView↩
Holder.imageView  [package]
```

6.18.2.6 instanceTextView

```
TextView it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceView↩
Holder.instanceTextView  [package]
```

6.18.2.7 kalmanFilterDistanceTextView

```
TextView it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceView↩
Holder.kalmanFilterDistanceTextView [package]
```

6.18.2.8 macTextView

```
TextView it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceView↩
Holder.macTextView [package]
```

6.18.2.9 majorTextView

```
TextView it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceView↩
Holder.majorTextView [package]
```

6.18.2.10 minorTextView

```
TextView it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceView↩
Holder.minorTextView [package]
```

6.18.2.11 nameSpaceTextView

```
TextView it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceView↩
Holder.nameSpaceTextView [package]
```

6.18.2.12 proximityTextView

```
TextView it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceView↩
Holder.proximityTextView [package]
```

6.18.2.13 rssiTextView

```
TextView it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceView↩
Holder.rssiTextView [package]
```

6.18.2.14 standardRawDistanceTextView

```
TextView it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceView↩
Holder.standardRawDistanceTextView [package]
```

6.18.2.15 txPowerTextView

```
TextView it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceView↩
Holder.txPowerTextView [package]
```

6.18.2.16 uuidTextView

```
TextView it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceView↩
Holder.uuidTextView [package]
```

6.18.2.17 view

```
View it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceViewHolder.↩
view  [package]
```

6.18.2.18 visibilityNameSpaceLinearLayout

```
LinearLayout it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceView↩
Holder.visibilityNameSpaceLinearLayout  [package]
```

6.18.2.19 visibilityUUIDLinearLayout

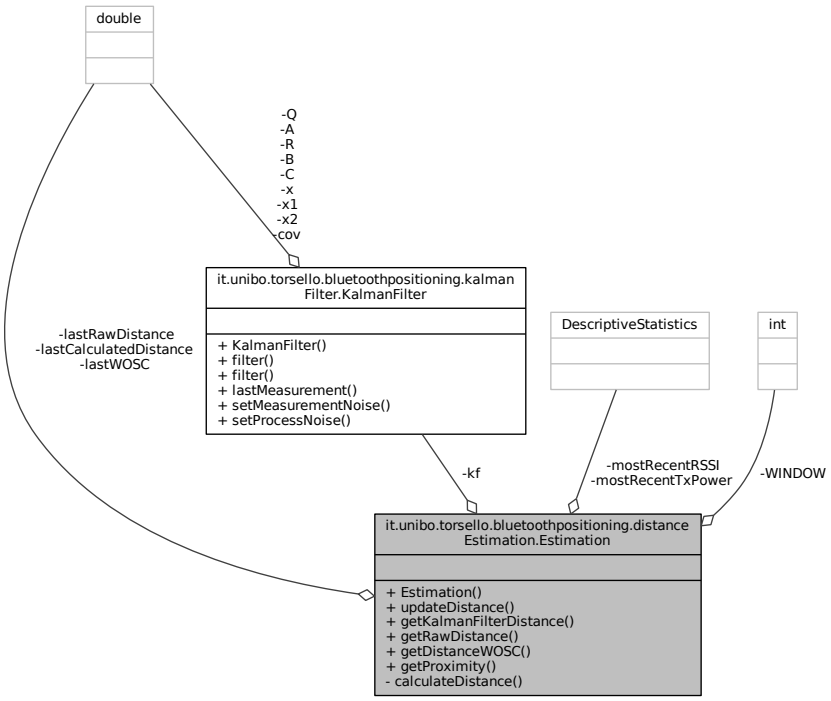
```
LinearLayout it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceView↩
Holder.visibilityUUIDLinearLayout  [package]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- DeviceCardViewAdapter.java

6.19 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation

Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation:



Membri pubblici

- Estimation ()
- void updateDistance (Beacon b, double processNoise)
- double getKalmanFilterDistance ()
- double getRawDistance ()
- double getDistanceWOSC ()
- String getProximity ()

Membri privati

- double calculateDistance (double txPower, double rssi)

Attributi privati

- DescriptiveStatistics mostRecentRSSI
- DescriptiveStatistics mostRecentTxPower
- KalmanFilter kf
- double lastCalculatedDistance
- double lastRawDistance
- double lastWOSC

Attributi privati statici

- static final int WINDOW = 20

6.19.1 Descrizione dettagliata

A helper class

Autore

Jonathan Vidmar

Versione

1.0

6.19.2 Documentazione dei costruttori e dei distruttori

6.19.2.1 Estimation()

`it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.Estimation ( )`

```
25                          {
26
27          // limit on the number of values that can be stored in the dataset
28          mostRecentRSSI = new DescriptiveStatistics();
29          mostRecentRSSI.setWindowSize(WINDOW);
30          mostRecentTxPower = new DescriptiveStatistics();
31          mostRecentTxPower.setWindowSize(WINDOW);
32
33          lastCalculatedDistance = 0;
34          lastRawDistance = 0;
35          lastWOSC = 0;
36
37          kf = new KFBuilder()
38                  // filter for RSSI
39                  .R(10) // Initial process noise
40                  .Q(60.0) // Initial measurement noise
41                  .build();
42      }
```

6.19.3 Documentazione delle funzioni membro

6.19.3.1 calculateDistance()

```
double it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.calculateDistance
(
              double txPower,
              double rssi )  [private]
```

```
66                                                                      {
67
68          if (rssi == 0.0D) {
69              return -1.0D; // if we cannot determine accuracy, return -1.
70          }
71
72          double ratio = (rssi * 1.0D) / txPower;
73          if (ratio < 1.0D) {
74              return Math.pow(ratio, 10.0D);
75          }
76
77 //         return (0.89976D) * Math.pow(ratio, 7.7095D) + 0.125D;
78          return (0.89976d * Math.pow(ratio, 7.7095D)) + 0.111D;
79
80      /*
81       * RSSI = TxPower - 10 * n * lg(d)
82       * n = 2 (in free space)
83       * d = 10 ^ ((TxPower - RSSI) / (10 * n))
84       */
85          // return Math.pow(10D, (txPower - rssi) / (10 * 2));
86      }
```

6.19.3.2 getDistanceWOSC()

```
double it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.getDistanceWOSC (
)
```

```
96                                      {
97          return lastWOSC;
98      }
```

6.19.3.3 getKalmanFilterDistance()

```
double it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.getKalmanFilter↩
Distance ( )
```

```
88                                              {
89          return lastCalculatedDistance;
90      }
```

6.19.3.4 getProximity()

```
String it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.getProximity ( )
```

```
100                                     {
101         double proximity = lastCalculatedDistance;
102         String accuracy;
103
104         if (proximity <= 0) {
105             accuracy = "unknown";
106         } else if (proximity <= 0.5) {
107             accuracy = "immediate";
108         } else if (proximity <= 4.0) {
109             accuracy = "near";
110         } else {
111             accuracy = "far";
112         }
113         return accuracy;
114     }
```

6.19.3.5 getRawDistance()

```
double it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.getRawDistance ( )
```

```
92                                  {
93         return lastRawDistance;
94     }
```

6.19.3.6 updateDistance()

```
void it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.updateDistance (
              Beacon b,
              double processNoise )
```

```
44                                                              {
45         double lastFilteredReading = -1;
46 
47         mostRecentRSSI.addValue(b.getRssi());
48         mostRecentTxPower.addValue(b.getTxPower());
49 
50         // Update measurement noise continually
51         double mNoise = Math.sqrt((100 * 9 / Math.log(10)) *
52                 Math.log(1 + Math.pow(mostRecentRSSI.getMean() /
     mostRecentRSSI.getStandardDeviation(), 2)));
53 
54         if (!Double.isInfinite(mNoise) && !Double.isNaN(mNoise)) {
55             kf.setMeasurementNoise(mNoise);
56         }
57 
58         kf.setProcessNoise(processNoise);
59         lastFilteredReading = kf.filter(mostRecentRSSI.getPercentile(50));
60         lastCalculatedDistance = calculateDistance(
     mostRecentTxPower.getPercentile(50), lastFilteredReading);
61         lastRawDistance = calculateDistance(b.getTxPower(), b.getRssi());
62         lastWOSC = calculateDistance(b.getTxPower(), lastFilteredReading);
63     }
```

6.19.4 Documentazione dei membri dato

6.19.4.1 kf

```
KalmanFilter it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.kf  [private]
```

6.19.4.2 lastCalculatedDistance

```
double it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.lastCalculated↩
Distance  [private]
```

6.19.4.3 lastRawDistance

```
double it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.lastRawDistance
[private]
```

6.19.4.4 lastWOSC

```
double it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.lastWOSC  [private]
```

6.19.4.5 mostRecentRSSI

```
DescriptiveStatistics it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.↩
mostRecentRSSI  [private]
```

6.19.4.6 mostRecentTxPower

```
DescriptiveStatistics it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.↩
mostRecentTxPower  [private]
```

6.19.4.7 WINDOW

```
final int it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.WINDOW = 20
[static], [private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- Estimation.java

6.20 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.extra.FABBehavior

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.extra.FABBehavior

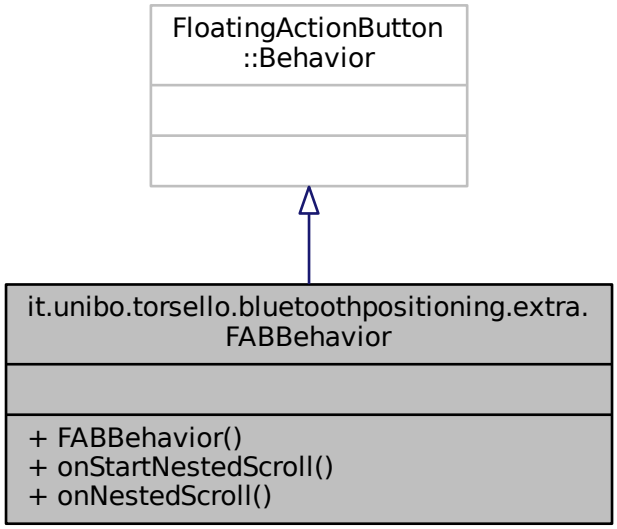
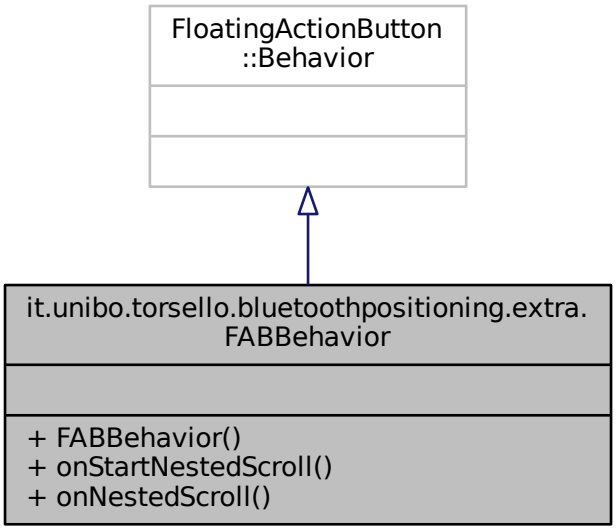


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.extra.FABBehavior:



Membri pubblici

- FABBehavior (Context context, AttributeSet attrs)
- boolean onStartNestedScroll (CoordinatorLayout coordinatorLayout, final FloatingActionButton child, View directTargetChild, View target, int nestedScrollAxes)
- void onNestedScroll (CoordinatorLayout coordinatorLayout, final FloatingActionButton child, View target, int dxConsumed, int dyConsumed, int dxUnconsumed, int dyUnconsumed)

6.20.1 Descrizione dettagliata

Created by Federico Torsello. federico.torsello@studio.unibo.it

6.20.2 Documentazione dei costruttori e dei distruttori

6.20.2.1 FABBehavior()

```
it.unibo.torsello.bluetoothpositioning.extra.FABBehavior.FABBehavior (
            Context context,
            AttributeSet attrs )
```

```
17                                                  {
18         super();
19     }
```

6.20.3 Documentazione delle funzioni membro

6.20.3.1 onNestedScroll()

```
void it.unibo.torsello.bluetoothpositioning.extra.FABBehavior.onNestedScroll (
            CoordinatorLayout coordinatorLayout,
            final FloatingActionButton child,
            View target,
            int dxConsumed,
            int dyConsumed,
            int dxUnconsumed,
            int dyUnconsumed )
```

```
32
        {
33      super.onNestedScroll(coordinatorLayout, child, target, dxConsumed, dyConsumed, dxUnconsumed,
34              dyUnconsumed);
35
36      if ((dyConsumed > 0 || dyUnconsumed == 0) && child.getVisibility() == View.VISIBLE) {
37          child.hide();
38          new Handler().postDelayed(new Runnable() {
39              @Override
40              public void run() {
41                  child.show();
42              }
43          }, 1000);
44      }
45  }
```

6.20.3.2 onStartNestedScroll()

```
boolean it.unibo.torsello.bluetoothpositioning.extra.FABBehavior.onStartNestedScroll (
            CoordinatorLayout coordinatorLayout,
            final FloatingActionButton child,
            View directTargetChild,
            View target,
            int nestedScrollAxes )
```

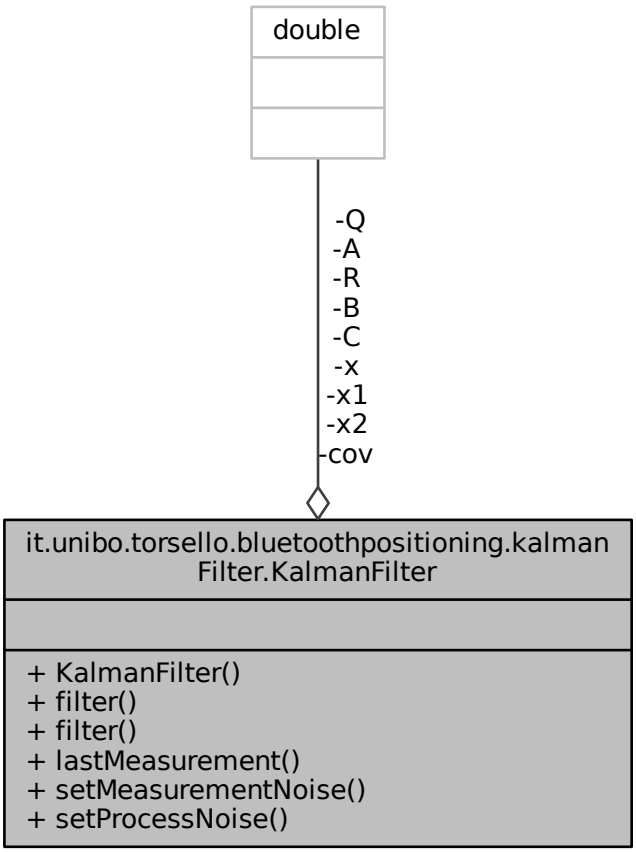
```
23                                                                                                  {
24
25      return nestedScrollAxes == ViewCompat.SCROLL_AXIS_VERTICAL ||
26              super.onStartNestedScroll(coordinatorLayout, child, directTargetChild, target,
27                      nestedScrollAxes);
28  }
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- FABBehavior.java

6.21 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.kalmanFilter.KalmanFilter

Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.kalmanFilter.KalmanFilter:



Membri pubblici

- KalmanFilter (double R, double Q, double A, double B, double C)
- double filter (double z)
- double filter (double z, double u)
- double lastMeasurement ()
- void setMeasurementNoise (double noise)
- void setProcessNoise (double noise)

Attributi privati

- double R
- double Q
- double A
- double B
- double C

- double cov
- double x
- double x1
- double x2

6.21.1 Descrizione dettagliata

Originally written in JS by Wouter Bulten 2015 Rewritten to Java by Jonathan Vidmar 2016 Copyright 2015 Wouter Bulten GNU LESSER GENERAL PUBLIC LICENSE v3

6.21.2 Documentazione dei costruttori e dei distruttori

6.21.2.1 KalmanFilter()

```
it.unibo.torsello.bluetoothpositioning.kalmanFilter.KalmanFilter.KalmanFilter (
            double R,
            double Q,
            double A,
            double B,
            double C )
```

Create 1-dimensional kalman filter

Parametri

R	Process noise
Q	Measurement noise
A	State vector
B	Control vector
C	Measurement vector

```
30                                                                  {
31
32          this.R = R;
33          this.Q = Q;
34          this.A = A;
35          this.B = B;
36          this.C = C;
37
38          cov = Double.NaN;
39          x = Double.NaN;
40      }
```

6.21.3 Documentazione delle funzioni membro

6.21.3.1 filter() [1/2]

```
double it.unibo.torsello.bluetoothpositioning.kalmanFilter.KalmanFilter.filter (
            double z )
```

```
43                                  {
44          return filter(z, 0);
45      }
```

6.21.3.2 filter() [2/2]

```
double it.unibo.torsello.bluetoothpositioning.kalmanFilter.KalmanFilter.filter (
            double z,
            double u )
```

Filter a new value

Parametri

z	Measurement
u	Control

Restituisce

x

```
54                                          {
55
56          if (Double.isNaN(x)) {
57              x = (1 / C) * z;
58              x1 = x;
59              x2 = x1;
60              cov = (1 / C) * Q * (1 / C);
61          } else {
62
63              // Calculate previous update step
64              B = (x - x1) / 2;
65
66              // Compute prediction
67              double predX = (A * x) + (B * u);
68              double predCov = ((A * cov) * A) + R;
69
70              // Kalman gain
71              double K = predCov * C * (1 / ((C * predCov * C) + Q));
72
73              // Correction
74              x1 = x;
75              x = predX + K * (z - (C * predX));
76              cov = predCov - (K * C * predCov);
77          }
78
79          return x;
80      }
```

6.21.3.3 lastMeasurement()

```
double it.unibo.torsello.bluetoothpositioning.kalmanFilter.KalmanFilter.lastMeasurement ( )
```

Return the last filtered measurement

Restituisce

x Estimated signal without noise

```
87                                      {
88          return x;
89      }
```

6.21.3.4 setMeasurementNoise()

```
void it.unibo.torsello.bluetoothpositioning.kalmanFilter.KalmanFilter.setMeasurementNoise (
            double noise )
```

Set measurement noise Q

Parametri

noise	Measurement noise

```
96                                                  {
97          Q = noise;
98      }
```

6.21.3.5 setProcessNoise()

```
void it.unibo.torsello.bluetoothpositioning.kalmanFilter.KalmanFilter.setProcessNoise (
              double noise )
```

Set the process noise R

Parametri

noise	Process noise

```
105                                                 {
106         R = noise;
107     }
```

6.21.4 Documentazione dei membri dato

6.21.4.1 A

```
double it.unibo.torsello.bluetoothpositioning.kalmanFilter.KalmanFilter.A  [private]
```

6.21.4.2 B

```
double it.unibo.torsello.bluetoothpositioning.kalmanFilter.KalmanFilter.B  [private]
```

6.21.4.3 C

```
double it.unibo.torsello.bluetoothpositioning.kalmanFilter.KalmanFilter.C  [private]
```

6.21.4.4 cov

```
double it.unibo.torsello.bluetoothpositioning.kalmanFilter.KalmanFilter.cov  [private]
```

6.21.4.5 Q

```
double it.unibo.torsello.bluetoothpositioning.kalmanFilter.KalmanFilter.Q  [private]
```

6.21.4.6 R

```
double it.unibo.torsello.bluetoothpositioning.kalmanFilter.KalmanFilter.R  [private]
```

6.21.4.7 x

```
double it.unibo.torsello.bluetoothpositioning.kalmanFilter.KalmanFilter.x  [private]
```

6.21.4.8 x1

```
double it.unibo.torsello.bluetoothpositioning.kalmanFilter.KalmanFilter.x1  [private]
```

6.21.4.9 x2

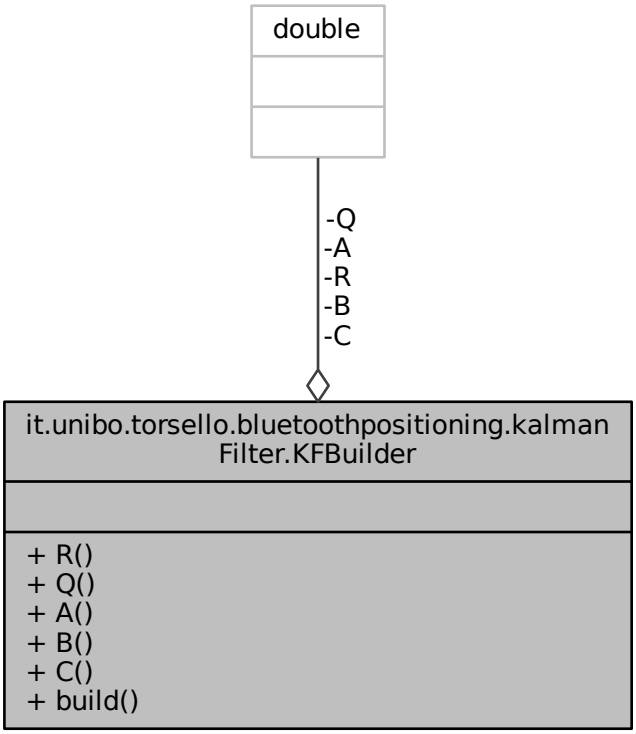
```
double it.unibo.torsello.bluetoothpositioning.kalmanFilter.KalmanFilter.x2  [private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- KalmanFilter.java

6.22 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.kalmanFilter.KFBuilder

Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.kalmanFilter.KFBuilder:



Membri pubblici

- KFBuilder R (double R)
- KFBuilder Q (double Q)
- KFBuilder A (double A)
- KFBuilder B (double B)
- KFBuilder C (double C)
- KalmanFilter build ()

Attributi privati

- double R = 1
- double Q = 1
- double A = 1
- double B = 0
- double C = 1

6.22.1 Descrizione dettagliata

Simple builder class for 1-dimensional Kalman filter with predefined

6.22.2 Documentazione delle funzioni membro

6.22.2.1 A()

```
KFBuilder it.unibo.torsello.bluetoothpositioning.kalmanFilter.KFBuilder.A (
            double A )
```

```
23                                  {
24          this.A = A;
25          return this;
26      }
```

6.22.2.2 B()

```
KFBuilder it.unibo.torsello.bluetoothpositioning.kalmanFilter.KFBuilder.B (
            double B )
```

```
28                                  {
29          this.B = B;
30          return this;
31      }
```

6.22.2.3 build()

```
KalmanFilter it.unibo.torsello.bluetoothpositioning.kalmanFilter.KFBuilder.build ( )
```

```
38                                  {
39          return new KalmanFilter(R, Q, A, B, C);
40      }
```

6.22.2.4 C()

```
KFBuilder it.unibo.torsello.bluetoothpositioning.kalmanFilter.KFBuilder.C (
            double C )
```

```
33                                  {
34         this.C = C;
35         return this;
36     }
```

6.22.2.5 Q()

```
KFBuilder it.unibo.torsello.bluetoothpositioning.kalmanFilter.KFBuilder.Q (
            double Q )
```

```
18                                  {
19         this.Q = Q;
20         return this;
21     }
```

6.22.2.6 R()

```
KFBuilder it.unibo.torsello.bluetoothpositioning.kalmanFilter.KFBuilder.R (
            double R )
```

```
13                                  {
14         this.R = R;
15         return this;
16     }
```

6.22.3 Documentazione dei membri dato

6.22.3.1 A

```
double it.unibo.torsello.bluetoothpositioning.kalmanFilter.KFBuilder.A = 1  [private]
```

6.22.3.2 B

```
double it.unibo.torsello.bluetoothpositioning.kalmanFilter.KFBuilder.B = 0  [private]
```

6.22.3.3 C

```
double it.unibo.torsello.bluetoothpositioning.kalmanFilter.KFBuilder.C = 1  [private]
```

6.22.3.4 Q

```
double it.unibo.torsello.bluetoothpositioning.kalmanFilter.KFBuilder.Q = 1  [private]
```

6.22.3.5 R

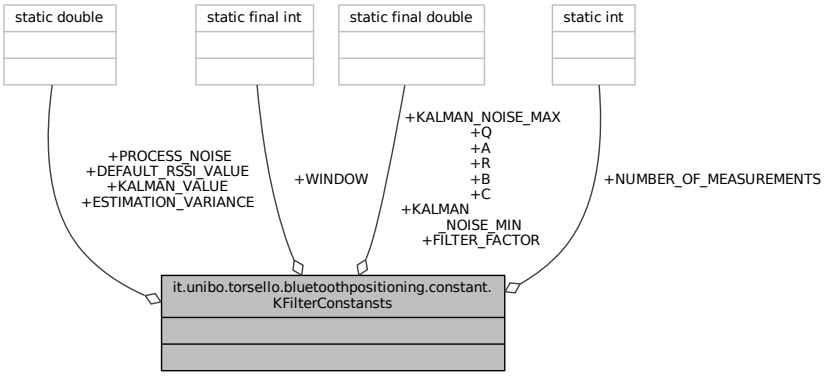
```
double it.unibo.torsello.bluetoothpositioning.kalmanFilter.KFBuilder.R = 1  [private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- KFBuilder.java

6.23 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.constant.KFilterConstansts

Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.constant.KFilterConstansts:



Attributi pubblici statici

- static final double KALMAN_NOISE_MIN = 0.00D
- static final double KALMAN_NOISE_MAX = 5.0D
- static final double FILTER_FACTOR = 0.1D
- static final int WINDOW = 20
- static final double R = 10
- static final double Q = 60.0
- static final double A = 1
- static final double B = 0
- static final double C = 1
- static double KALMAN_VALUE = 2D
- static double DEFAULT_RSSI_VALUE = -70D
- static int NUMBER_OF_MEASUREMENTS = 1
- static double ESTIMATION_VARIANCE = 0.01D
- static double PROCESS_NOISE = 10

6.23.1 Descrizione dettagliata

Created by Federico Torsello. federico.torsello@studio.unibo.it

6.23.2 Documentazione dei membri dato

6.23.2.1 A

```
final double it.unibo.torsello.bluetoothpositioning.constant.KFilterConstansts.A = 1  [static]
```

6.23.2.2 B

```
final double it.unibo.torsello.bluetoothpositioning.constant.KFilterConstansts.B = 0  [static]
```

6.23.2.3 C

```
final double it.unibo.torsello.bluetoothpositioning.constant.KFilterConstansts.C = 1  [static]
```

6.23.2.4 DEFAULT_RSSI_VALUE

```
double it.unibo.torsello.bluetoothpositioning.constant.KFilterConstansts.DEFAULT_RSSI_VALUE =
-70D  [static]
```

6.23.2.5 ESTIMATION_VARIANCE

```
double it.unibo.torsello.bluetoothpositioning.constant.KFilterConstansts.ESTIMATION_VARIANCE =
0.01D  [static]
```

6.23.2.6 FILTER_FACTOR

```
final double it.unibo.torsello.bluetoothpositioning.constant.KFilterConstansts.FILTER_FACTOR =
0.1D  [static]
```

6.23.2.7 KALMAN_NOISE_MAX

```
final double it.unibo.torsello.bluetoothpositioning.constant.KFilterConstansts.KALMAN_NOISE_↩
MAX = 5.0D  [static]
```

6.23.2.8 KALMAN_NOISE_MIN

```
final double it.unibo.torsello.bluetoothpositioning.constant.KFilterConstansts.KALMAN_NOISE_↩
MIN = 0.00D  [static]
```

6.23.2.9 KALMAN_VALUE

```
double it.unibo.torsello.bluetoothpositioning.constant.KFilterConstansts.KALMAN_VALUE = 2D
[static]
```

6.23.2.10 NUMBER_OF_MEASUREMENTS

```
int it.unibo.torsello.bluetoothpositioning.constant.KFilterConstansts.NUMBER_OF_MEASUREMENTS =
1  [static]
```

6.23.2.11 PROCESS_NOISE

```
double it.unibo.torsello.bluetoothpositioning.constant.KFilterConstansts.PROCESS_NOISE = 10
[static]
```

6.23.2.12 Q

```
final double it.unibo.torsello.bluetoothpositioning.constant.KFilterConstansts.Q = 60.0  [static]
```

6.23.2.13 R

```
final double it.unibo.torsello.bluetoothpositioning.constant.KFilterConstansts.R = 10  [static]
```

6.23.2.14 WINDOW

```
final int it.unibo.torsello.bluetoothpositioning.constant.KFilterConstansts.WINDOW = 20  [static]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- KFilterConstansts.java

6.24 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.activities.MainActivity

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.activities.MainActivity

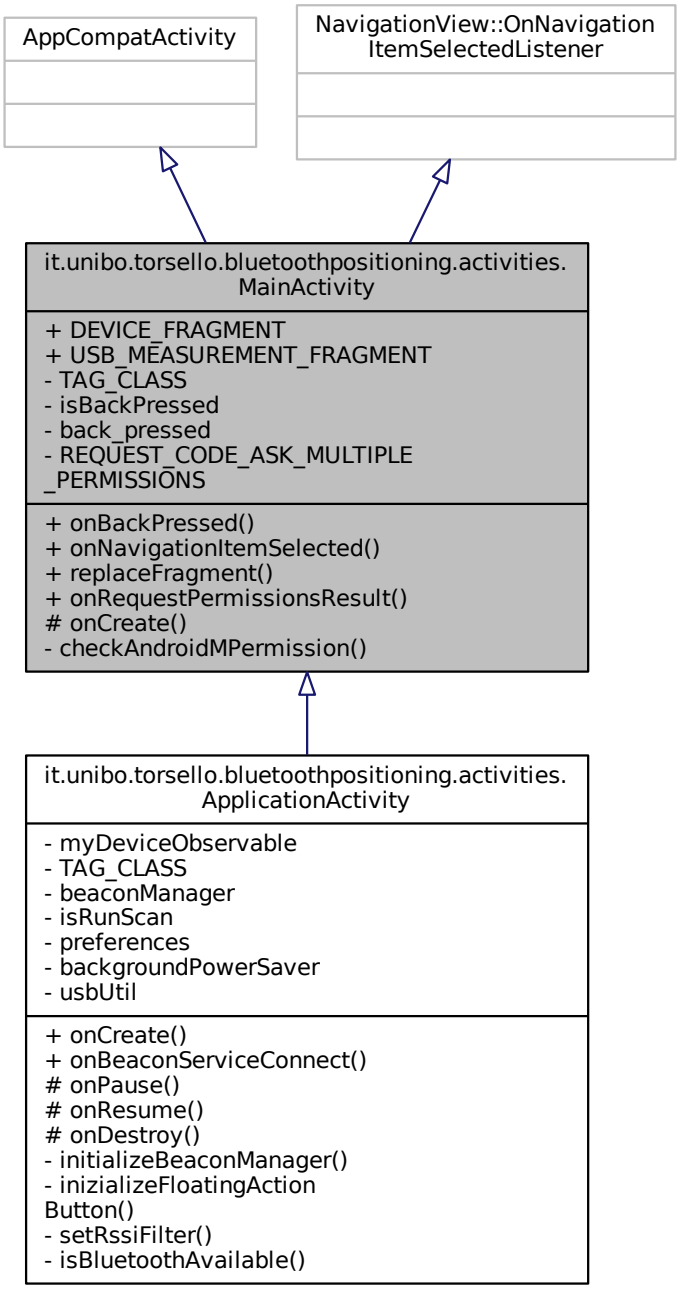
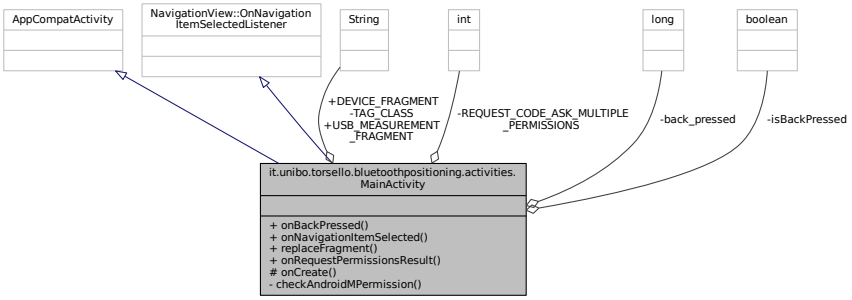


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.activities.MainActivity:



Membri pubblici

- void onBackPressed ()
- boolean onNavigationItemSelected (MenuItem item)
- void replaceFragment (String fragTag)
- void onRequestPermissionsResult (int requestCode, @NonNull String permissions[ ], @NonNull int[ ] grant↩ Results)

Attributi pubblici statici

- static final String DEVICE_FRAGMENT = "device"
- static final String USB_MEASUREMENT_FRAGMENT = "usb measurement"

Membri protetti

- void onCreate (Bundle savedInstanceState)

Membri privati

- void checkAndroidMPermission ()

Attributi privati

- final String TAG_CLASS = getClass().getSimpleName()
- boolean isBackPressed = false
- long back_pressed
- final int REQUEST_CODE_ASK_MULTIPLE_PERMISSIONS = 124

6.24.1 Descrizione dettagliata

Created by Federico Torsello. federico.torsello@studio.unibo.it

6.24.2 Documentazione delle funzioni membro

6.24.2.1 checkAndroidMPermission()

void it.unibo.torsello.bluetoothpositioning.activities.MainActivity.checkAndroidMPermission ( ) [private]

```
173                                         {
174
175         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
176             final List<String> permissions = new ArrayList<>();
177
178             if (checkSelfPermission(Manifest.permission.ACCESS_FINE_LOCATION)
179                     != PackageManager.PERMISSION_GRANTED) {
180                 permissions.add(Manifest.permission.ACCESS_FINE_LOCATION);
181             }
182
183             if (checkSelfPermission(Manifest.permission.ACCESS_COARSE_LOCATION)
184                     != PackageManager.PERMISSION_GRANTED) {
185                 permissions.add(Manifest.permission.ACCESS_COARSE_LOCATION);
186             }
187
188             if (!permissions.isEmpty()) {
189                 new AlertDialog.Builder(this)
190                         .setTitle(R.string.dialog_location_access_title)
191                         .setMessage(R.string.dialog_bluetooth_text)
192                         .setPositiveButton(android.R.string.ok, null)
193                         .setOnDismissListener(new DialogInterface.OnDismissListener() {
194                             @TargetApi(23)
195                             @Override
196                             public void onDismiss(DialogInterface dialog) {
197                                 requestPermissions(permissions.toArray(new String[permissions.size()]),
198                                         REQUEST_CODE_ASK_MULTIPLE_PERMISSIONS
    );
199                             }
200
201                         }).show();
202             }
203         }
204     }
```

6.24.2.2 onBackPressed()

void it.unibo.torsello.bluetoothpositioning.activities.MainActivity.onBackPressed ( )

```
70                                  {
71
72         DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
73         if (drawer.isDrawerOpen(GravityCompat.START)) {
74             drawer.closeDrawer(GravityCompat.START);
75         } else if (drawer.isDrawerOpen(GravityCompat.END)) {
76             drawer.closeDrawer(GravityCompat.END);
77         } else {
78
79             replaceFragment(DEVICE_FRAGMENT);
80
81             final long DOUBLE_PRESS_INTERVAL = 1500L;
82             if (!isBackPressed || back_pressed + DOUBLE_PRESS_INTERVAL <= System.
    currentTimeMillis()) {
83                 isBackPressed = true;
84                 FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
85                 assert fab != null;
86                 Snackbar.make(fab, R.string.snackBar_exit, Snackbar.LENGTH_SHORT).show();
87             } else {
88                 super.finish();
89             }
90             back_pressed = System.currentTimeMillis();
91         }
92     }
```

6.24.2.3 onCreate()

```
void it.unibo.torsello.bluetoothpositioning.activities.MainActivity.onCreate (
            Bundle savedInstanceState )  [protected]
```

```
47                                                  {
48         super.onCreate(savedInstanceState);
49         setContentView(R.layout.activity_main);
50
51         Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
52         setSupportActionBar(toolbar);
53
54         DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
55         ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
56                 this, drawer, toolbar, R.string.navigation_drawer_open, R.string.navigation_drawer_close);
57         drawer.addDrawerListener(toggle);
58         toggle.syncState();
59
60         ((NavigationView) findViewById(R.id.nav_view)).setNavigationItemSelectedListener(this);
61
62         ((NavigationView) findViewById(R.id.nav_view2)).setNavigationItemSelectedListener(this);
63
64         replaceFragment(DEVICE_FRAGMENT);
65
66         checkAndroidMPermission();
67     }
```

6.24.2.4 onNavigationItemSelected()

```
boolean it.unibo.torsello.bluetoothpositioning.activities.MainActivity.onNavigationItem↩
Selected (
            MenuItem item )
```

```
96                                                          {
97
98         DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
99
100         // Handle navigation view item clicks here.
101         switch (item.getItemId()) {
102             case R.id.nav_home:
103                 replaceFragment(DEVICE_FRAGMENT);
104                 break;
105             case R.id.nav_settings:
106                 drawer.openDrawer(GravityCompat.END);
107                 break;
108             case R.id.nav_measurement:
109                 replaceFragment(USB_MEASUREMENT_FRAGMENT);
110                 break;
111 //            case R.id.nav_share:
112 //                fragment = CamTestFragment.newInstance();
113 //                break;
114 //            case R.id.nav_send:
115 //                fragment = ViewPagerFragment.newInstance(getFragments());
116 //                break;
117         }
118
119         if (drawer.isDrawerOpen(GravityCompat.START)) {
120             drawer.closeDrawer(GravityCompat.START);
121         }
122
123         return true;
124     }
```

6.24.2.5 onRequestPermissionsResult()

```
void it.unibo.torsello.bluetoothpositioning.activities.MainActivity.onRequestPermissionsResult
(
            int requestCode,
            @NonNull String permissions[],
            @NonNull int [] grantResults )
```

```
146                                                                         {
147         switch (requestCode) {
148             case REQUEST_CODE_ASK_MULTIPLE_PERMISSIONS:
149                 for (int i = 0; i < permissions.length; i++) {
150                     if (grantResults[i] == PackageManager.PERMISSION_GRANTED) {
151 //                      Log.d(TAG_CLASS, "Permission Granted: " + permissions[i]);
152                     } else if (grantResults[i] == PackageManager.PERMISSION_DENIED) {
153 //                      Log.d(TAG_CLASS, "Permission Denied: " + permissions[i]);
154                         new AlertDialog.Builder(this)
155                                 .setTitle(R.string.dialog_permissions_location_access_title)
156                                 .setMessage(R.string.dialog_permissions_location_access_text)
157                                 .setPositiveButton(android.R.string.ok, null)
158                                 .setOnDismissListener(new DialogInterface.OnDismissListener() {
159 
160                                     @Override
161                                     public void onDismiss(DialogInterface dialog) {
162                                     }
163 
164                                 }).show();
165                     }
166                 }
167                 break;
168             default:
169                 super.onRequestPermissionsResult(requestCode, permissions, grantResults);
170         }
171     }
```

6.24.2.6 replaceFragment()

```
void it.unibo.torsello.bluetoothpositioning.activities.MainActivity.replaceFragment (
            String fragTag )
```

```
126                                                 {
127         Fragment currentFragment = getSupportFragmentManager().findFragmentByTag(fragTag);
128         switch (fragTag) {
129             case DEVICE_FRAGMENT:
130                 currentFragment = DeviceListFragment.newInstance();
131                 break;
132             case USB_MEASUREMENT_FRAGMENT:
133                 currentFragment = UsbMeasurementFragment.newInstance();
134                 break;
135         }
136 
137         if (currentFragment != null) {
138             getSupportFragmentManager().beginTransaction()
139                     .replace(R.id.contentMainLayout, currentFragment, fragTag)
140                     .commit();
141         }
142     }
```

6.24.3 Documentazione dei membri dato

6.24.3.1 back_pressed

```
long it.unibo.torsello.bluetoothpositioning.activities.MainActivity.back_pressed  [private]
```

6.24.3.2 DEVICE_FRAGMENT

```
final String it.unibo.torsello.bluetoothpositioning.activities.MainActivity.DEVICE_FRAGMENT =
"device"  [static]
```

6.24.3.3 isBackPressed

```
boolean it.unibo.torsello.bluetoothpositioning.activities.MainActivity.isBackPressed = false
[private]
```

6.24.3.4 REQUEST_CODE_ASK_MULTIPLE_PERMISSIONS

```
final int it.unibo.torsello.bluetoothpositioning.activities.MainActivity.REQUEST_CODE_ASK_MU↩
LTIPLE_PERMISSIONS = 124  [private]
```

6.24.3.5 TAG_CLASS

```
final String it.unibo.torsello.bluetoothpositioning.activities.MainActivity.TAG_CLASS = get↩
Class().getSimpleName()  [private]
```

6.24.3.6 USB_MEASUREMENT_FRAGMENT

```
final String it.unibo.torsello.bluetoothpositioning.activities.MainActivity.USB_MEASUREMENT_↩
FRAGMENT = "usb measurement"  [static]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- MainActivity.java

6.25 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.configuration.MyArmaRssiFilter

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.configuration.MyArmaRssiFilter

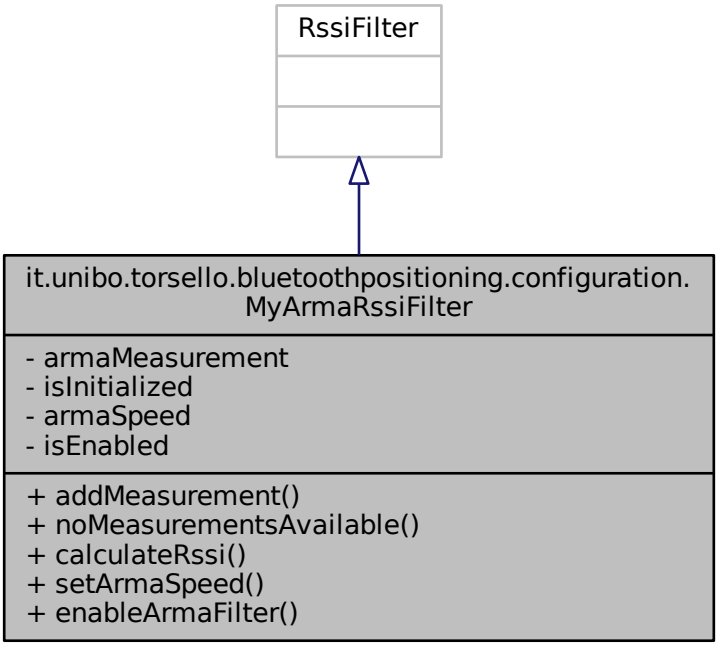
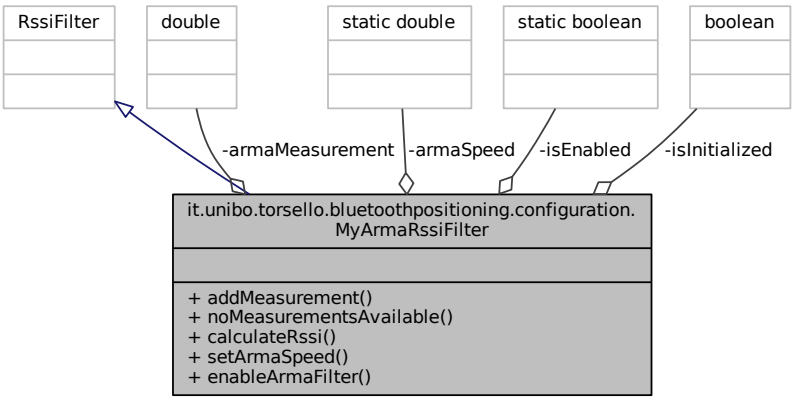


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.configuration.MyArmaRssiFilter:



Membri pubblici

- void addMeasurement (Integer rssi)
- boolean noMeasurementsAvailable ()
- double calculateRssi ()

Membri pubblici statici

- static void setArmaSpeed (double arma_speed)
- static void enableArmaFilter (boolean set)

Attributi privati

- double armaMeasurement
- boolean isInitialized = false

Attributi privati statici

- static double armaSpeed = 0.08D
- static boolean isEnabled = true

6.25.1 Documentazione delle funzioni membro

6.25.1.1 addMeasurement()

```
void it.unibo.torsello.bluetoothpositioning.configuration.MyArmaRssiFilter.addMeasurement (
            Integer rssi )
```

```
25                                          {
26
27         if (isEnabled) {
28             if (!isInitialized) {
29                 armaMeasurement = rssi;
30                 isInitialized = true;
31             }
32
33             armaMeasurement = (armaMeasurement -
       armaSpeed * (armaMeasurement - rssi));
34         } else {
35             armaMeasurement = rssi;
36         }
37
38     }
```

6.25.1.2 calculateRssi()

```
double it.unibo.torsello.bluetoothpositioning.configuration.MyArmaRssiFilter.calculateRssi ( )
```

```
46                                  {
47         return armaMeasurement;
48     }
```

6.25.1.3 enableArmaFilter()

```
static void it.unibo.torsello.bluetoothpositioning.configuration.MyArmaRssiFilter.enableArma↩
Filter (
              boolean set )  [static]
```

```
20                                                  {
21         isEnabled = set;
22     }
```

6.25.1.4 noMeasurementsAvailable()

```
boolean it.unibo.torsello.bluetoothpositioning.configuration.MyArmaRssiFilter.noMeasurements↩
Available ( )
```

```
41                                          {
42         return false;
43     }
```

6.25.1.5 setArmaSpeed()

```
static void it.unibo.torsello.bluetoothpositioning.configuration.MyArmaRssiFilter.setArmaSpeed
(
              double arma_speed )  [static]
```

```
16                                                    {
17         armaSpeed = arma_speed;
18     }
```

6.25.2 Documentazione dei membri dato

6.25.2.1 armaMeasurement

```
double it.unibo.torsello.bluetoothpositioning.configuration.MyArmaRssiFilter.armaMeasurement
[private]
```

6.25.2.2 armaSpeed

```
double it.unibo.torsello.bluetoothpositioning.configuration.MyArmaRssiFilter.armaSpeed = 0.08D
[static], [private]
```

6.25.2.3 isEnabled

```
boolean it.unibo.torsello.bluetoothpositioning.configuration.MyArmaRssiFilter.isEnabled = true
[static], [private]
```

6.25.2.4 isInitialized

```
boolean it.unibo.torsello.bluetoothpositioning.configuration.MyArmaRssiFilter.isInitialized =
false  [private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- MyArmaRssiFilter.java

6.26 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview

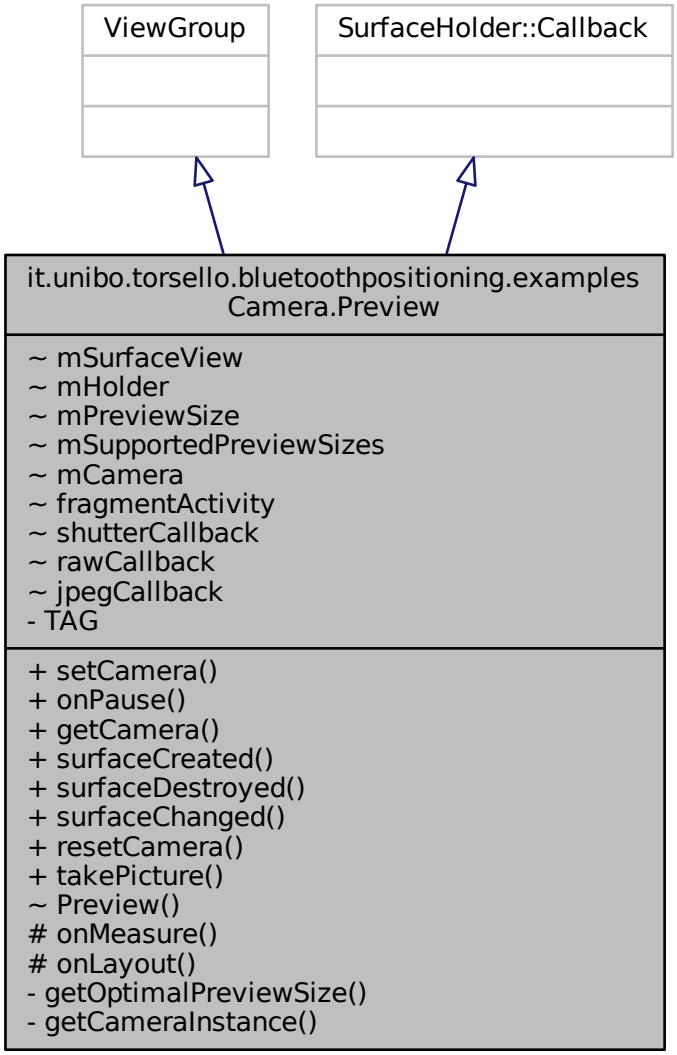
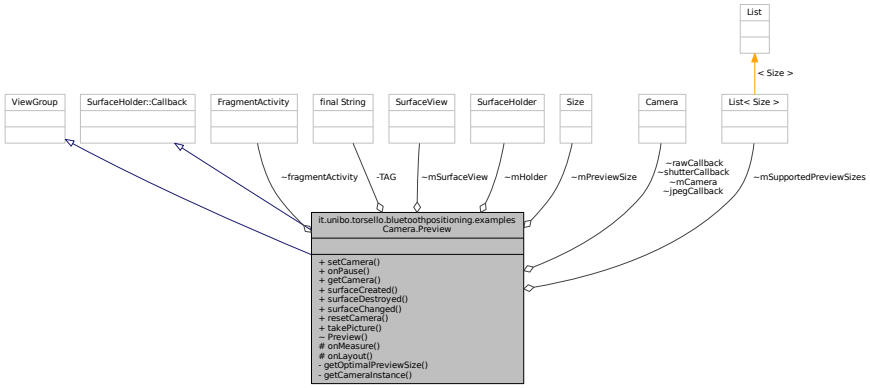


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview:



Membri pubblici

- void setCamera (FragmentActivity fragmentActivity)
- void onPause ()
- Camera getCamera ()
- void surfaceCreated (SurfaceHolder holder)
- void surfaceDestroyed (SurfaceHolder holder)
- void surfaceChanged (SurfaceHolder holder, int format, int w, int h)
- void resetCamera ()
- void takePicture ()

Membri protetti

- void onMeasure (int widthMeasureSpec, int heightMeasureSpec)
- void onLayout (boolean changed, int l, int t, int r, int b)

Funzioni con visibilità di package

- Preview (Context context, SurfaceView sv)

Attributi con visibilità di package

- SurfaceView mSurfaceView
- SurfaceHolder mHolder
- Size mPreviewSize
- List< Size > mSupportedPreviewSizes
- Camera mCamera
- FragmentActivity fragmentActivity
- Camera.ShutterCallback shutterCallback
- Camera.PictureCallback rawCallback
- Camera.PictureCallback jpegCallback

Membri privati

- Size getOptimalPreviewSize (List< Size > sizes, int w, int h)

Membri privati statici

- static Camera getCameraInstance ()

Attributi privati

- final String TAG = "Preview"

6.26.1 Documentazione dei costruttori e dei distruttori

6.26.1.1 Preview()

```
it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview.Preview (
              Context context,
              SurfaceView sv )  [package]
```

```
32                                              {
33         super(context);
34
35
36         mSurfaceView = sv;
37
38         mHolder = mSurfaceView.getHolder();
39         mHolder.addCallback(this);
40         mHolder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);
41     }
```

6.26.2 Documentazione delle funzioni membro

6.26.2.1 getCamera()

```
Camera it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview.getCamera ( )
```

```
79                                  {
80         return mCamera;
81     }
```

6.26.2.2 getCameraInstance()

```
static Camera it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview.getCameraInstance
( )  [static], [private]
```

A safe way to get an instance of the CameraUtil object.

```
86                                                  {
87
88         Camera c = null;
89
90         try {
91             int numCams = Camera.getNumberOfCameras();
92             if (numCams > 0) {
93                 c = Camera.open(0); // attempt to get a CameraUtil instance
94             }
95         } catch (RuntimeException e) {
96             // CameraUtil is not available (in use or does not exist)
97             e.getStackTrace();
98         }
99
100         return c; // returns null if camera is unavailable
101     }
```

6.26.2.3 getOptimalPreviewSize()

```
Size it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview.getOptimalPreviewSize (
            List< Size > sizes,
            int w,
            int h )  [private]
```

```
167                                                                     {
168         final double ASPECT_TOLERANCE = 0.1;
169         double targetRatio = (double) w / h;
170         if (sizes == null) return null;
171
172         Size optimalSize = null;
173         double minDiff = Double.MAX_VALUE;
174
175 //        int targetHeight = h;
176
177         // Try to find an size match aspect ratio and size
178         for (Size size : sizes) {
179             double ratio = (double) size.width / size.height;
180             if (Math.abs(ratio - targetRatio) > ASPECT_TOLERANCE) continue;
181             if (Math.abs(size.height - h) < minDiff) {
182                 optimalSize = size;
183                 minDiff = Math.abs(size.height - h);
184             }
185         }
186
187         // Cannot find the one match the aspect ratio, ignore the requirement
188         if (optimalSize == null) {
189             minDiff = Double.MAX_VALUE;
190             for (Size size : sizes) {
191                 if (Math.abs(size.height - h) < minDiff) {
192                     optimalSize = size;
193                     minDiff = Math.abs(size.height - h);
194                 }
195             }
196         }
197         return optimalSize;
198     }
```

6.26.2.4 onLayout()

```
void it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview.onLayout (
            boolean changed,
            int l,
            int t,
            int r,
            int b )  [protected]
```

```
118                                                                       {
119         if (changed && getChildCount() > 0) {
120             final View child = getChildAt(0);
121
122             final int width = r - l;
123             final int height = b - t;
124
125             int previewWidth = width;
126             int previewHeight = height;
127             if (mPreviewSize != null) {
128                 previewWidth = mPreviewSize.width;
129                 previewHeight = mPreviewSize.height;
130             }
131
132             // Center the child SurfaceView within the parent.
133             if (width * previewHeight > height * previewWidth) {
134                 final int scaledChildWidth = previewWidth * height / previewHeight;
135                 child.layout((width - scaledChildWidth) / 2, 0,
136                         (width + scaledChildWidth) / 2, height);
137             } else {
138                 final int scaledChildHeight = previewHeight * width / previewWidth;
139                 child.layout(0, (height - scaledChildHeight) / 2,
140                         width, (height + scaledChildHeight) / 2);
141             }
142         }
143     }
```

6.26.2.5 onMeasure()

```
void it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview.onMeasure (
              int widthMeasureSpec,
              int heightMeasureSpec )  [protected]
```

```
104                                                                          {
105         // We purposely disregard child measurements because act as a
106         // wrapper to a SurfaceView that centers the camera preview instead
107         // of stretching it.
108         final int width = resolveSize(getSuggestedMinimumWidth(), widthMeasureSpec);
109         final int height = resolveSize(getSuggestedMinimumHeight(), heightMeasureSpec);
110         setMeasuredDimension(width, height);
111
112         if (mSupportedPreviewSizes != null) {
113             mPreviewSize = getOptimalPreviewSize(
      mSupportedPreviewSizes, width, height);
114         }
115     }
```

6.26.2.6 onPause()

```
void it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview.onPause ( )
```

```
71                           {
72         if (mCamera != null) {
73             mCamera.stopPreview();
74             mCamera.release();
75             mCamera = null;
76         }
77     }
```

6.26.2.7 resetCamera()

```
void it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview.resetCamera ( )
```

```
213                              {
214         new Thread(new Runnable() {
215             @Override
216             public void run() {
217                 mCamera.startPreview();
218             }
219         }).start();
220
221
222     }
```

6.26.2.8 setCamera()

```
void it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview.setCamera (
              FragmentActivity fragmentActivity )
```

```
43                                                              {
44
45         this.fragmentActivity = fragmentActivity;
46         try {
47             mCamera = getCameraInstance();
48         } catch (RuntimeException ex) {
49             Toast.makeText(fragmentActivity, "camera_not_found", Toast.LENGTH_LONG).show();
50         }
51
52         if (mCamera != null) {
53
54             mSupportedPreviewSizes = mCamera.getParameters().
      getSupportedPreviewSizes();
```

```
55              requestLayout();
56
57              // get Camera parameters
58              Camera.Parameters params = mCamera.getParameters();
59
60              List<String> focusModes = params.getSupportedFocusModes();
61              if (focusModes.contains(Camera.Parameters.FOCUS_MODE_AUTO)) {
62                  // set the focus mode
63                  params.setFocusMode(Camera.Parameters.FOCUS_MODE_AUTO);
64                  // set Camera parameters
65                  mCamera.setParameters(params);
66              }
67          }
68
69      }
```

6.26.2.9 surfaceChanged()

```
void it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview.surfaceChanged (
              SurfaceHolder holder,
              int format,
              int w,
              int h )
```

```
201                                                                                 {
202          if (mCamera != null) {
203              Camera.Parameters parameters = mCamera.getParameters();
204              parameters.setPreviewSize(mPreviewSize.width,
     mPreviewSize.height);
205              requestLayout();
206
207              mCamera.setParameters(parameters);
208
209              resetCamera();
210          }
211      }
```

6.26.2.10 surfaceCreated()

```
void it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview.surfaceCreated (
              SurfaceHolder holder )
```

```
146                                                 {
147          // The Surface has been created, acquire the camera and tell it where
148          // to draw.
149          try {
150              if (mCamera != null) {
151                  mCamera.setPreviewDisplay(holder);
152              }
153          } catch (IOException exception) {
154              Log.e(TAG, "IOException caused by setPreviewDisplay()", exception);
155          }
156      }
```

6.26.2.11 surfaceDestroyed()

```
void it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview.surfaceDestroyed (
              SurfaceHolder holder )
```

```
159                                                   {
160          // Surface will be destroyed when we return, so stop the preview.
161          if (mCamera != null) {
162              mCamera.stopPreview();
163          }
164      }
```

6.26.2.12 takePicture()

```
void it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview.takePicture ( )
```

```
246                                 {
247
248         mCamera.takePicture(shutterCallback, rawCallback,
      jpegCallback);
249     }
```

6.26.3 Documentazione dei membri dato

6.26.3.1 fragmentActivity

```
FragmentActivity it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview.fragment↩
Activity  [package]
```

6.26.3.2 jpegCallback

```
Camera.PictureCallback it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview.jpeg↩
Callback  [package]
```

Valore iniziale:

```
= new Camera.PictureCallback() {
        public void onPictureTaken(byte[] data, final Camera camera) {
            new SaveImageTask(fragmentActivity).execute(data);

            resetCamera();

            Log.d(TAG, "onPictureTaken - jpeg");
        }
    }
```

6.26.3.3 mCamera

```
Camera it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview.mCamera  [package]
```

6.26.3.4 mHolder

```
SurfaceHolder it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview.mHolder  [package]
```

6.26.3.5 mPreviewSize

```
Size it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview.mPreviewSize  [package]
```

6.26.3.6 mSupportedPreviewSizes

```
List<Size> it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview.mSupportedPreview↩
Sizes  [package]
```

6.26.3.7 mSurfaceView

```
SurfaceView it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview.mSurfaceView  [package]
```

6.26.3.8 rawCallback

```
Camera.PictureCallback it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview.raw↩
Callback  [package]
```

Valore iniziale:

```
= new Camera.PictureCallback() {
        public void onPictureTaken(byte[] data, Camera camera) {

        }
    }
```

6.26.3.9 shutterCallback

```
Camera.ShutterCallback it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview.shutter↩
Callback  [package]
```

Valore iniziale:

```
= new Camera.ShutterCallback() {
        public void onShutter() {

        }
    }
```

6.26.3.10 TAG

```
final String it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview.TAG = "Preview"
[private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- Preview.java

6.27 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.examplesCamera.SaveImageTask

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.examplesCamera.SaveImageTask

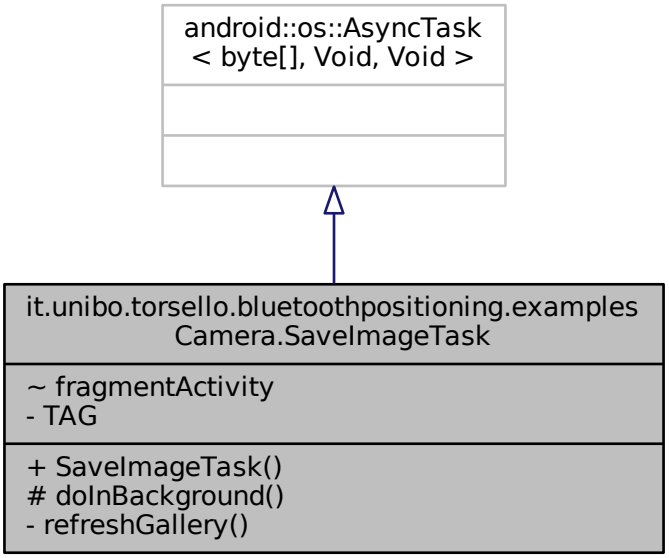
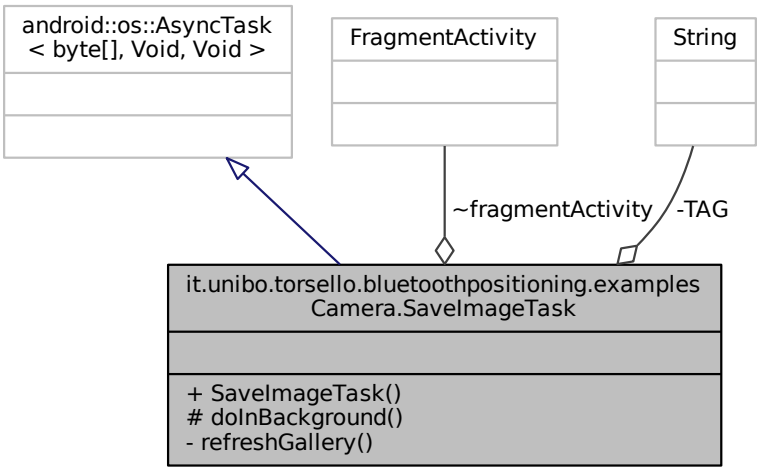


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.examplesCamera.SaveImageTask:



Membri pubblici

- SaveImageTask (FragmentActivity fragmentActivity)

Membri protetti

- Void doInBackground (byte[ ]... data)

Attributi con visibilità di package

- FragmentActivity fragmentActivity

Membri privati

- void refreshGallery (File file)

Attributi privati statici

- static final String TAG = "SaveImageTask"

6.27.1 Descrizione dettagliata

Created by federico on 21/09/16.

6.27.2 Documentazione dei costruttori e dei distruttori

6.27.2.1 SaveImageTask()

```
it.unibo.torsello.bluetoothpositioning.examplesCamera.SaveImageTask.SaveImageTask (
            FragmentActivity fragmentActivity )
```

```
23                                                          {
24         this.fragmentActivity = fragmentActivity;
25     }
```

6.27.3 Documentazione delle funzioni membro

6.27.3.1 doInBackground()

```
Void it.unibo.torsello.bluetoothpositioning.examplesCamera.SaveImageTask.doInBackground (
            byte...  [] data )  [protected]
```

```
28                                                  {
29
30         // Write to SD Card
31         try {
32             File sdCard = Environment.getExternalStorageDirectory();
33             File dir = new File(sdCard.getAbsolutePath() + "/camtest");
34 //                  dir.mkdirs();
35
36             String fileName = String.format(Locale.getDefault(), "%d.jpg", System.currentTimeMillis());
37             File outFile = new File(dir, fileName);
38
39             FileOutputStream outStream = new FileOutputStream(outFile);
40             outStream.write(data[0]);
41             outStream.flush();
42             outStream.close();
43
44             Log.d(TAG, "onPictureTaken - wrote bytes: " + data.length + " to " + outFile.getAbsolutePath
     ());
45
46             refreshGallery(outFile);
47         } catch (IOException e) {
48             e.printStackTrace();
49         }
50         return null;
51     }
```

6.27.3.2 refreshGallery()

```
void it.unibo.torsello.bluetoothpositioning.examplesCamera.SaveImageTask.refreshGallery (
            File file )  [private]
```

```
53                                          {
54         Intent mediaScanIntent = new Intent(Intent.ACTION_MEDIA_SCANNER_SCAN_FILE);
55         mediaScanIntent.setData(Uri.fromFile(file));
56         fragmentActivity.sendBroadcast(mediaScanIntent);
57     }
```

6.27.4 Documentazione dei membri dato

6.27.4.1 fragmentActivity

```
FragmentActivity it.unibo.torsello.bluetoothpositioning.examplesCamera.SaveImageTask.fragment↩
Activity  [package]
```

6.27.4.2 TAG

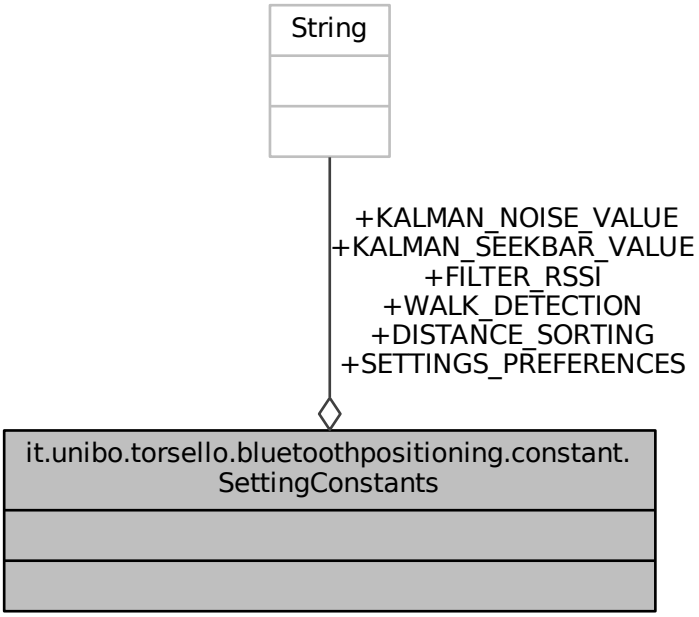
```
final String it.unibo.torsello.bluetoothpositioning.examplesCamera.SaveImageTask.TAG = "Save↩
ImageTask"  [static], [private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- SaveImageTask.java

6.28 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.constant.SettingConstants

Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.constant.SettingConstants:



Attributi pubblici statici

- static final String SETTINGS_PREFERENCES = "settings_preferences"
- static final String FILTER_RSSI = "filter_rssi"
- static final String WALK_DETECTION = "walk_detection"
- static final String KALMAN_SEEKBAR_VALUE = "kalman_filter_seek_value"
- static final String KALMAN_NOISE_VALUE = "kalman_filter_noise_value"
- static final String DISTANCE_SORTING = "distance_sorting_selected"

6.28.1 Descrizione dettagliata

Created by Federico Torsello. federico.torsello@studio.unibo.it A class containing constants for the SharedPreference objects.

6.28.2 Documentazione dei membri dato

6.28.2.1 DISTANCE_SORTING

```
final String it.unibo.torsello.bluetoothpositioning.constant.SettingConstants.DISTANCE_SORTING
= "distance_sorting_selected"  [static]
```

6.28.2.2 FILTER_RSSI

```
final String it.unibo.torsello.bluetoothpositioning.constant.SettingConstants.FILTER_RSSI =
"filter_rssi"  [static]
```

6.28.2.3 KALMAN_NOISE_VALUE

```
final String it.unibo.torsello.bluetoothpositioning.constant.SettingConstants.KALMAN_NOISE_V↩
ALUE = "kalman_filter_noise_value"  [static]
```

6.28.2.4 KALMAN_SEEKBAR_VALUE

```
final String it.unibo.torsello.bluetoothpositioning.constant.SettingConstants.KALMAN_SEEKBAR↩
_VALUE = "kalman_filter_seek_value"  [static]
```

6.28.2.5 SETTINGS_PREFERENCES

```
final String it.unibo.torsello.bluetoothpositioning.constant.SettingConstants.SETTINGS_PREFE↩
RENCES = "settings_preferences"  [static]
```

6.28.2.6 WALK_DETECTION

```
final String it.unibo.torsello.bluetoothpositioning.constant.SettingConstants.WALK_DETECTION =
"walk_detection"  [static]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- SettingConstants.java

6.29 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.fragment.SettingsFragment

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.fragment.SettingsFragment

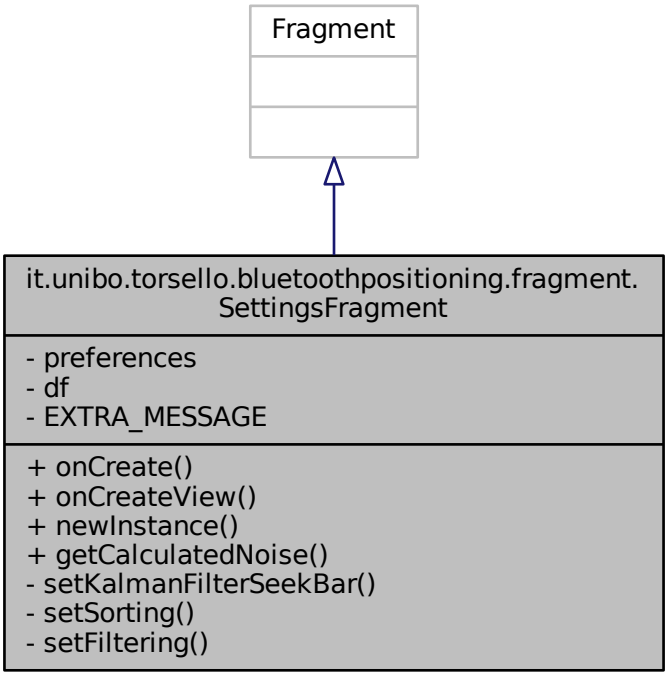
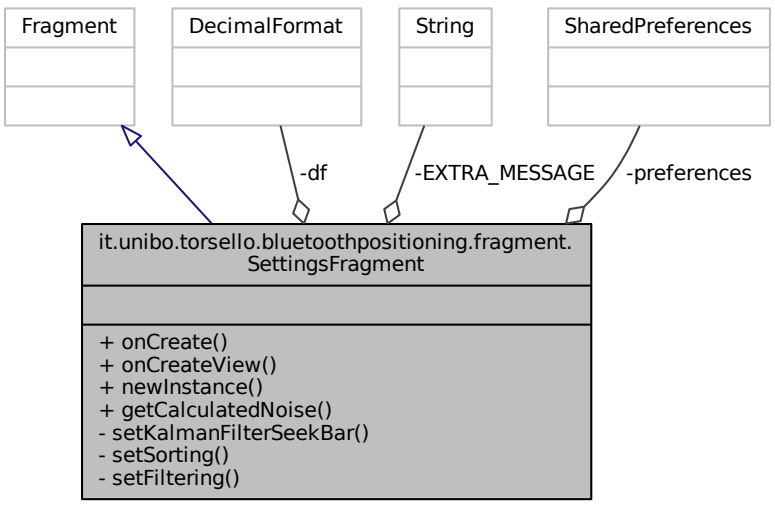


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.SettingsFragment:



Membri pubblici

- void onCreate (Bundle savedInstanceState)
- View onCreateView (LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)

Membri pubblici statici

- static SettingsFragment newInstance ()
- static float getCalculatedNoise (int p)

Membri privati

- void setKalmanFilterSeekBar (View root)
- void setSorting (View root)
- void setFiltering (View root)

Attributi privati

- SharedPreferences preferences
- DecimalFormat df

Attributi privati statici

- static final String EXTRA_MESSAGE = "EXTRA_MESSAGE"

6.29.1 Descrizione dettagliata

Created by Federico Torsello. federico.torsello@studio.unibo.it

6.29.2 Documentazione delle funzioni membro

6.29.2.1 getCalculatedNoise()

```
static float it.unibo.torsello.bluetoothpositioning.fragment.SettingsFragment.getCalculated↩
Noise (
            int p )  [static]
```

```
91                                              {
92         double percent = (p / 10D);
93         double noise = KFilterConstansts.KALMAN_NOISE_MIN +
94                 (KFilterConstansts.KALMAN_NOISE_MAX - KFilterConstansts.KALMAN_NOISE_MIN) * percent;
95
96         return (float) noise;
97
98     }
```

6.29.2.2 newInstance()

```
static SettingsFragment it.unibo.torsello.bluetoothpositioning.fragment.SettingsFragment.new↩
Instance ( )  [static]
```

```
30                                                {
31         SettingsFragment fragment = new SettingsFragment();
32         Bundle args = new Bundle();
33         args.putString(EXTRA_MESSAGE, "Settings");
34         fragment.setArguments(args);
35         return fragment;
36     }
```

6.29.2.3 onCreate()

```
void it.unibo.torsello.bluetoothpositioning.fragment.SettingsFragment.onCreate (
              Bundle savedInstanceState )
```

```
39                                                    {
40         super.onCreate(savedInstanceState);
41
42         preferences = getActivity().getSharedPreferences(SettingConstants.SETTINGS_PREFERENCES,
      0);
43         df = new DecimalFormat("0.0#");
44     }
```

6.29.2.4 onCreateView()

```
View it.unibo.torsello.bluetoothpositioning.fragment.SettingsFragment.onCreateView (
              LayoutInflater inflater,
              ViewGroup container,
              Bundle savedInstanceState )
```

```
47                                                                                                {
48         View root = inflater.inflate(R.layout.fragment_settings, container, false);
49
50         setKalmanFilterSeekBar(root);
51
52         setSorting(root);
53
54         setFiltering(root);
55
56         return root;
57     }
```

6.29.2.5 setFiltering()

```
void it.unibo.torsello.bluetoothpositioning.fragment.SettingsFragment.setFiltering (
              View root )  [private]
```

```
119                                       {
120         RadioGroup rg = (RadioGroup) root.findViewById(R.id.radioGroupFilter);
121         int checkedRadioButton;
122         if (rg.getCheckedRadioButtonId() != 0) {
123             checkedRadioButton = preferences.getInt(SettingConstants.FILTER_RSSI, rg.
      getCheckedRadioButtonId());
124         } else {
125             checkedRadioButton = 0;
126         }
127         rg.check(checkedRadioButton);
128         rg.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
129             @Override
130             public void onCheckedChanged(RadioGroup group, int checkedId) {
131                 SharedPreferences.Editor editor = preferences.edit();
132                 editor.putInt(SettingConstants.FILTER_RSSI, checkedId);
133                 editor.apply();
134             }
135         });
136     }
```

6.29.2.6 setKalmanFilterSeekBar()

```
void it.unibo.torsello.bluetoothpositioning.fragment.SettingsFragment.setKalmanFilterSeekBar (
            View root )  [private]
```

```
60                                                  {
61         SeekBar kalmanSeek = (SeekBar) root.findViewById(R.id.kalmanSeek);
62         int seekValue = preferences.getInt(SettingConstants.KALMAN_SEEKBAR_VALUE, 4);
63         kalmanSeek.setMax(10);
64         kalmanSeek.setProgress(kalmanSeek.getMax() / 2);
65
66         final TextView kalmanFilterValue = (TextView) root.findViewById(R.id.kalmanValue);
67         kalmanFilterValue.setText(df.format(getCalculatedNoise(seekValue)));
68
69         kalmanSeek.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
70             @Override
71             public void onProgressChanged(SeekBar seekBar, int seekValue, boolean fromUser) {
72                 kalmanFilterValue.setText(df.format(getCalculatedNoise(seekValue)));
73             }
74
75             @Override
76             public void onStartTrackingTouch(SeekBar seekBar) {
77             }
78
79             @Override
80             public void onStopTrackingTouch(SeekBar seekBar) {
81                 SharedPreferences.Editor editor = preferences.edit();
82                 int progress = seekBar.getProgress();
83                 editor.putInt(SettingConstants.KALMAN_SEEKBAR_VALUE, progress);
84                 editor.putFloat(SettingConstants.KALMAN_NOISE_VALUE,
    getCalculatedNoise(progress));
85                 editor.apply();
86                 kalmanFilterValue.setText(df.format(getCalculatedNoise(progress)));
87             }
88         });
89     }
```

6.29.2.7 setSorting()

```
void it.unibo.torsello.bluetoothpositioning.fragment.SettingsFragment.setSorting (
            View root )  [private]
```

```
100                                     {
101         RadioGroup rg = (RadioGroup) root.findViewById(R.id.radioGroupSortingMode);
102         int checkedRadioButton;
103         if (rg.getCheckedRadioButtonId() != 0) {
104             checkedRadioButton = preferences.getInt(SettingConstants.DISTANCE_SORTING, rg.
    getCheckedRadioButtonId());
105         } else {
106             checkedRadioButton = 0;
107         }
108         rg.check(checkedRadioButton);
109         rg.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
110             @Override
111             public void onCheckedChanged(RadioGroup group, int checkedId) {
112                 SharedPreferences.Editor editor = preferences.edit();
113                 editor.putInt(SettingConstants.DISTANCE_SORTING, checkedId);
114                 editor.apply();
115             }
116         });
117     }
```

6.29.3 Documentazione dei membri dato

6.29.3.1 df

```
DecimalFormat it.unibo.torsello.bluetoothpositioning.fragment.SettingsFragment.df  [private]
```

6.29.3.2 EXTRA_MESSAGE

```
final String it.unibo.torsello.bluetoothpositioning.fragment.SettingsFragment.EXTRA_MESSAGE =
"EXTRA_MESSAGE"  [static], [private]
```

6.29.3.3 preferences

```
SharedPreferences it.unibo.torsello.bluetoothpositioning.fragment.SettingsFragment.preferences
[private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- SettingsFragment.java

6.30 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.adapter.StatePagerAdapter

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.adapter.StatePagerAdapter

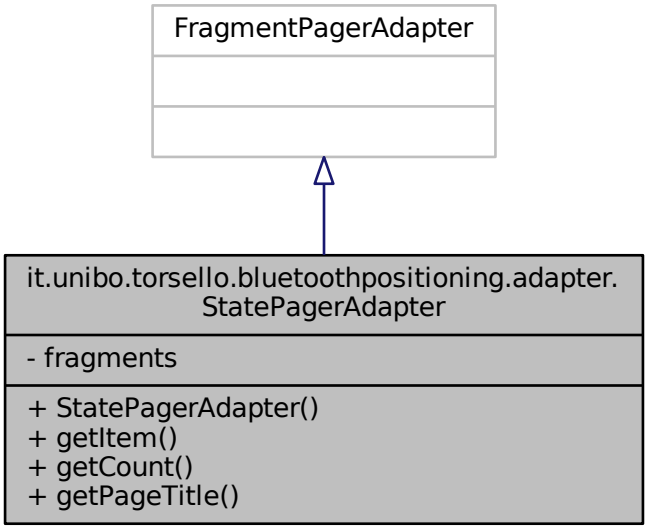
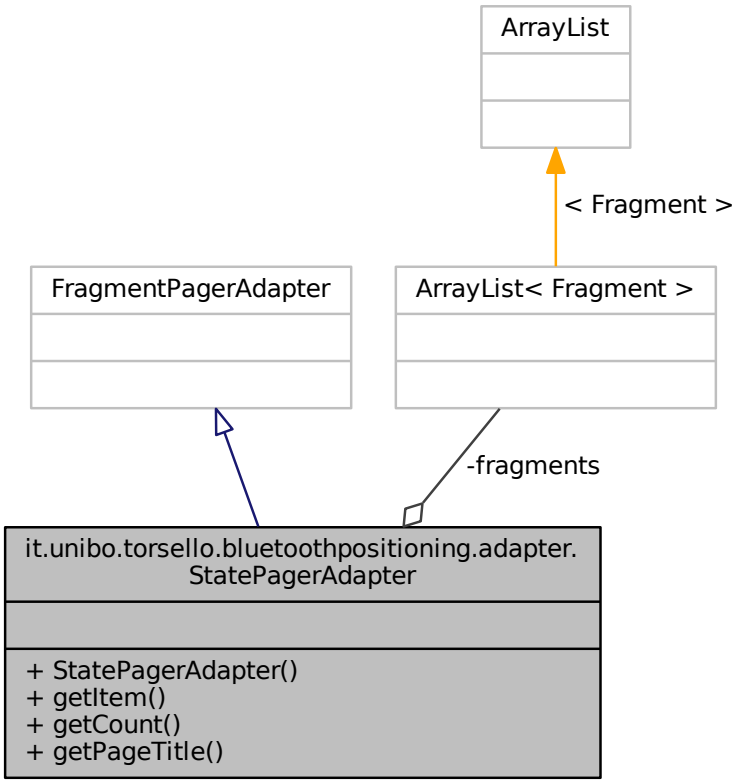


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.adapter.StatePagerAdapter:



Membri pubblici

- StatePagerAdapter (FragmentManager fm, ArrayList< Fragment > fragments)
- Fragment getItem (int position)
- int getCount ()
- CharSequence getPageTitle (int position)

Attributi privati

- ArrayList< Fragment > fragments

6.30.1 Descrizione dettagliata

Created by Federico Torsello. federico.torsello@studio.unibo.it

6.30.2 Documentazione dei costruttori e dei distruttori

6.30.2.1 StatePagerAdapter()

```
it.unibo.torsello.bluetoothpositioning.adapter.StatePagerAdapter.StatePagerAdapter (
            FragmentManager fm,
            ArrayList< Fragment > fragments )
```

```
19                                                                          {
20         super(fm);
21         this.fragments = fragments;
22     }
```

6.30.3 Documentazione delle funzioni membro

6.30.3.1 getCount()

```
int it.unibo.torsello.bluetoothpositioning.adapter.StatePagerAdapter.getCount ( )
```

```
30                           {
31         return this.fragments.size();
32     }
```

6.30.3.2 getItem()

```
Fragment it.unibo.torsello.bluetoothpositioning.adapter.StatePagerAdapter.getItem (
            int position )
```

```
25                                       {
26         return this.fragments.get(position);
27     }
```

6.30.3.3 getPageTitle()

```
CharSequence it.unibo.torsello.bluetoothpositioning.adapter.StatePagerAdapter.getPageTitle (
            int position )
```

```
35                                                {
36         return fragments.get(position).getArguments().getString("EXTRA_MESSAGE");
37     }
```

6.30.4 Documentazione dei membri dato

6.30.4.1 fragments

```
ArrayList<Fragment> it.unibo.torsello.bluetoothpositioning.adapter.StatePagerAdapter.fragments
[private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- StatePagerAdapter.java

6.31 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.fragment.usbObservers.Usb↩ MeasurementFragment

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.fragment.usbObservers.UsbMeasurement↩ Fragment

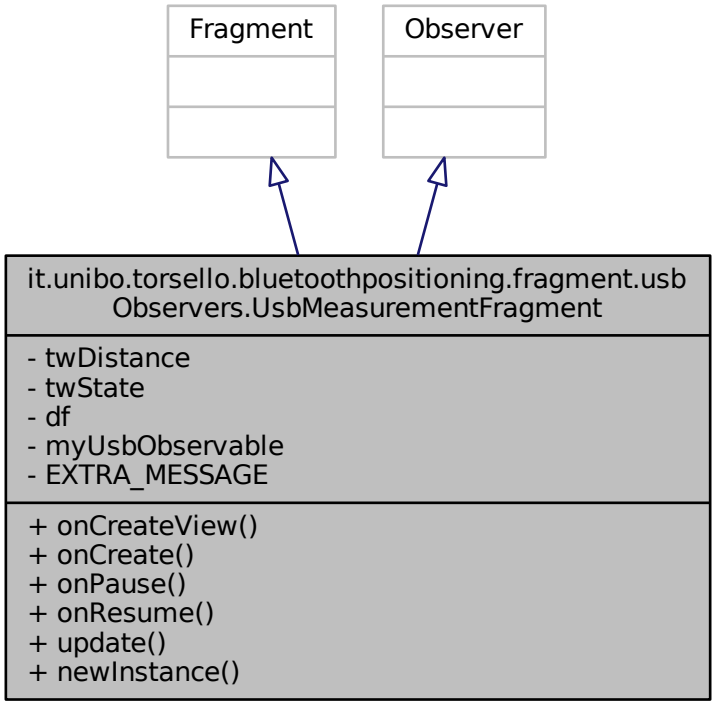
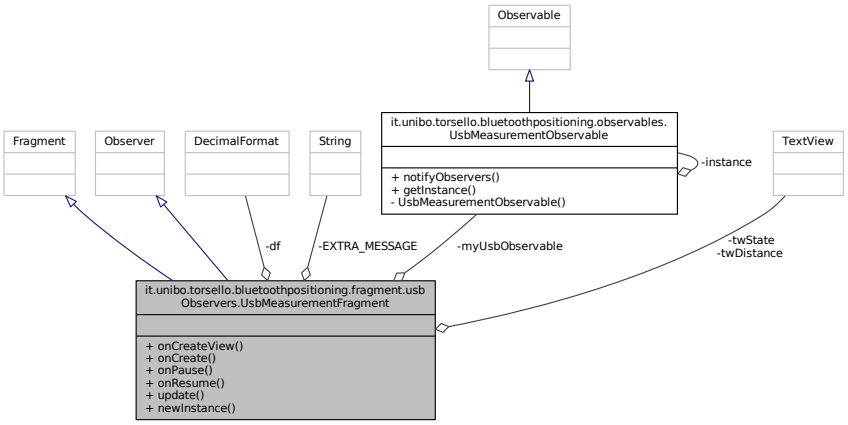


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.usbObservers.UsbMeasurement↩ Fragment:



Membri pubblici

- View onCreateView (LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
- void onCreate (@Nullable Bundle savedInstanceState)
- void onPause ()
- void onResume ()
- void update (Observable o, final Object arg)

Membri pubblici statici

- static UsbMeasurementFragment newInstance ()

Attributi privati

- TextView twDistance
- TextView twState
- DecimalFormat df
- UsbMeasurementObservable myUsbObservable

Attributi privati statici

- static final String EXTRA_MESSAGE = "EXTRA_MESSAGE"

6.31.1 Descrizione dettagliata

Created by Federico Torsello. federico.torsello@studio.unibo.it

6.31.2 Documentazione delle funzioni membro

6.31.2.1 newInstance()

```
static UsbMeasurementFragment it.unibo.torsello.bluetoothpositioning.fragment.usbObservers.↩
UsbMeasurementFragment.newInstance ( )  [static]
```

```
35                                                  {
36         UsbMeasurementFragment fragment = new UsbMeasurementFragment();
37         Bundle args = new Bundle();
38         args.putString(EXTRA_MESSAGE, "Measurement");
39         fragment.setArguments(args);
40         return fragment;
41     }
```

6.31.2.2 onCreate()

```
void it.unibo.torsello.bluetoothpositioning.fragment.usbObservers.UsbMeasurementFragment.on↩
Create (
              @Nullable Bundle savedInstanceState )
```

```
55                                                          {
56         super.onCreate(savedInstanceState);
57
58         myUsbObservable = UsbMeasurementObservable.getInstance();
59         df = new DecimalFormat("0.00", DecimalFormatSymbols.getInstance());
60
61     }
```

6.31.2.3 onCreateView()

```
View it.unibo.torsello.bluetoothpositioning.fragment.usbObservers.UsbMeasurementFragment.on↩
CreateView (
            LayoutInflater inflater,
            ViewGroup container,
            Bundle savedInstanceState )
```

```
45                                                          {
46         View root = inflater.inflate(R.layout.fragment_usb_measurement, container, false);
47
48         twDistance = (TextView) root.findViewById(R.id.tw_distance_value);
49         twState = (TextView) root.findViewById(R.id.tw_state_value);
50
51         return root;
52     }
```

6.31.2.4 onPause()

```
void it.unibo.torsello.bluetoothpositioning.fragment.usbObservers.UsbMeasurementFragment.on↩
Pause ( )
```

```
64                          {
65         myUsbObservable.deleteObserver(this);
66         super.onPause();
67     }
```

6.31.2.5 onResume()

```
void it.unibo.torsello.bluetoothpositioning.fragment.usbObservers.UsbMeasurementFragment.on↩
Resume ( )
```

```
70                           {
71         super.onResume();
72         myUsbObservable.addObserver(this);
73     }
```

6.31.2.6 update()

```
void it.unibo.torsello.bluetoothpositioning.fragment.usbObservers.UsbMeasurementFragment.↩
update (
            Observable o,
            final Object arg )
```

```
76                                                      {
77
78         getActivity().runOnUiThread(new Runnable() {
79             @Override
80             public void run() {
81                 if (arg instanceof Double) {
82                     Double arduinoDistance = (Double) arg;
83                     twDistance.setText(String.format("%s m", df.format(arduinoDistance)));
84                 } else if (arg instanceof String) {
85                     String message = (String) arg;
86                     twState.setText(message);
87                 } else if (arg instanceof Boolean) {
88                     boolean state = (Boolean) arg;
89                     if (state) {
90                         twState.setTextColor(Color.GREEN);
91                     } else {
92                         twState.setTextColor(Color.RED);
93                     }
94                 }
95             }
96         });
97     }
```

6.31.3 Documentazione dei membri dato

6.31.3.1 df

```
DecimalFormat it.unibo.torsello.bluetoothpositioning.fragment.usbObservers.UsbMeasurement↩
Fragment.df  [private]
```

6.31.3.2 EXTRA_MESSAGE

```
final String it.unibo.torsello.bluetoothpositioning.fragment.usbObservers.UsbMeasurement↩
Fragment.EXTRA_MESSAGE = "EXTRA_MESSAGE"  [static], [private]
```

6.31.3.3 myUsbObservable

```
UsbMeasurementObservable it.unibo.torsello.bluetoothpositioning.fragment.usbObservers.Usb↩
MeasurementFragment.myUsbObservable  [private]
```

6.31.3.4 twDistance

```
TextView it.unibo.torsello.bluetoothpositioning.fragment.usbObservers.UsbMeasurementFragment.↩
twDistance  [private]
```

6.31.3.5 twState

```
TextView it.unibo.torsello.bluetoothpositioning.fragment.usbObservers.UsbMeasurementFragment.↩
twState  [private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- UsbMeasurementFragment.java

6.32 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.observables.UsbMeasurement↩ Observable

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.observables.UsbMeasurementObservable

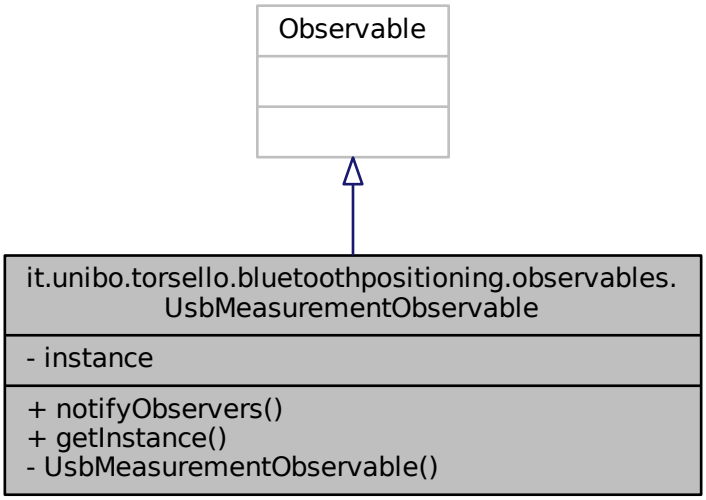
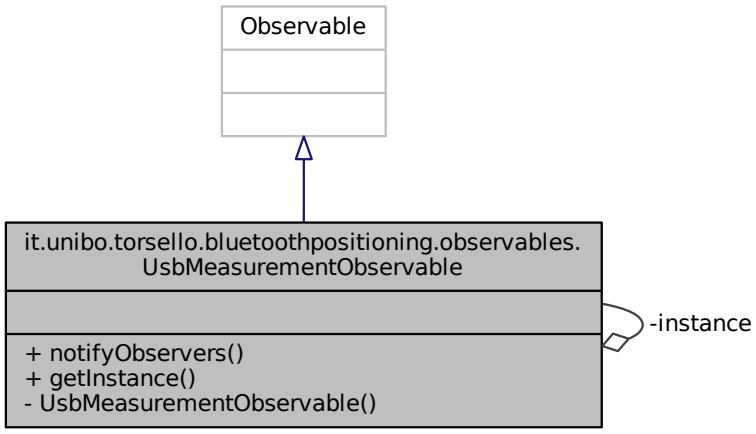


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.observables.UsbMeasurementObservable:



Membri pubblici

- void notifyObservers (Object data)

Membri pubblici statici

- static UsbMeasurementObservable getInstance ()

Membri privati

- UsbMeasurementObservable ()

Attributi privati statici

- static UsbMeasurementObservable instance = new UsbMeasurementObservable()

6.32.1 Descrizione dettagliata

Created by Federico Torsello. federico.torsello@studio.unibo.it

6.32.2 Documentazione dei costruttori e dei distruttori

6.32.2.1 UsbMeasurementObservable()

```
it.unibo.torsello.bluetoothpositioning.observables.UsbMeasurementObservable.UsbMeasurement↩
Observable ( )  [private]
```

```
17                                      {
18         super();
19     }
```

6.32.3 Documentazione delle funzioni membro

6.32.3.1 getInstance()

```
static UsbMeasurementObservable it.unibo.torsello.bluetoothpositioning.observables.UsbMeasurement↩
Observable.getInstance ( )  [static]
```

```
13                                                  {
14         return instance;
15     }
```

6.32.3.2 notifyObservers()

```
void it.unibo.torsello.bluetoothpositioning.observables.UsbMeasurementObservable.notify↩
Observers (
              Object data )
```

```
22                                          {
23         setChanged();
24         super.notifyObservers(data);
25     }
```

6.32.4 Documentazione dei membri dato

6.32.4.1 instance

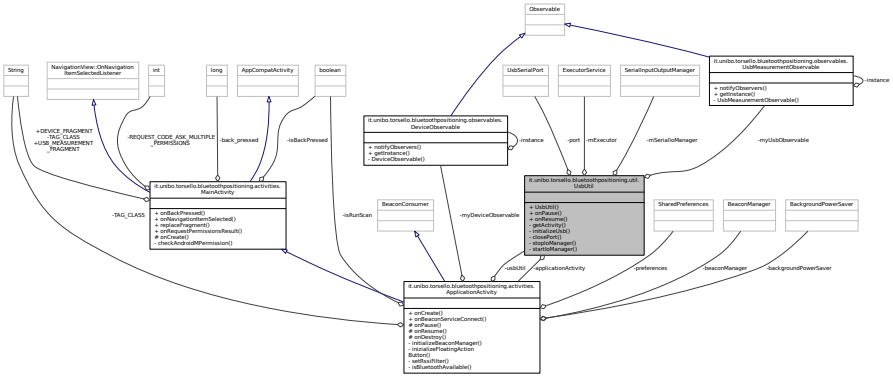
```
UsbMeasurementObservable it.unibo.torsello.bluetoothpositioning.observables.UsbMeasurement↩
Observable.instance = new UsbMeasurementObservable()  [static], [private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- UsbMeasurementObservable.java

6.33 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.util.UsbUtil

Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.util.UsbUtil:



Membri pubblici

- UsbUtil (ApplicationActivity applicationActivity)
- void onPause ()
- void onResume ()

Membri privati

- ApplicationActivity getActivity ()
- void initializeUsb ()
- void closePort ()
- void stopIoManager ()
- void startIoManager ()

Attributi privati

- UsbMeasurementObservable myUsbObservable
- final ExecutorService mExecutor = Executors.newSingleThreadExecutor()
- UsbSerialPort port
- SerialInputOutputManager mSerialIoManager
- ApplicationActivity applicationActivity

6.33.1 Descrizione dettagliata

Created by federico on 02/10/16.

6.33.2 Documentazione dei costruttori e dei distruttori

6.33.2.1 UsbUtil()

```
it.unibo.torsello.bluetoothpositioning.util.UsbUtil.UsbUtil (
              ApplicationActivity applicationActivity )
```

```
37                                                          {
38         this.applicationActivity = applicationActivity;
39         myUsbObservable = UsbMeasurementObservable.getInstance();
40     }
```

6.33.3 Documentazione delle funzioni membro

6.33.3.1 closePort()

```
void it.unibo.torsello.bluetoothpositioning.util.UsbUtil.closePort ( )  [private]
```

```
110                                 {
111         if (port != null) {
112             try {
113                 port.close();
114             } catch (IOException e) {
115                 // Ignore.
116             }
117             port = null;
118         }
119     }
```

6.33.3.2 getActivity()

```
ApplicationActivity it.unibo.torsello.bluetoothpositioning.util.UsbUtil.getActivity ( )  [private]
```

```
42                                          {
43         return applicationActivity;
44     }
```

6.33.3.3 initializeUsb()

```
void it.unibo.torsello.bluetoothpositioning.util.UsbUtil.initializeUsb ( )  [private]
```

```
56                                  {
57
58         // Find all available drivers from attached devices.
59         UsbManager usbManager = (UsbManager) getActivity().getSystemService(Context.USB_SERVICE)
     ;
60         List<UsbSerialDriver> availableDrivers = UsbSerialProber.getDefaultProber().findAllDrivers(
     usbManager);
61         if (!availableDrivers.isEmpty()) {
62
63             // Open a connection to the first available driver.
64             UsbSerialDriver driver = availableDrivers.get(0);
65
66             if (usbManager.hasPermission(driver.getDevice())) {
67                 if (usbManager.openDevice(driver.getDevice()) != null) {
68                     // Read some data! Most have just one port (port 0).
69                     port = driver.getPorts().get(0);
70                 }
71             } else {
72                 Intent startIntent = new Intent(getActivity(), getClass());
73                 PendingIntent pendingIntent =
74                         PendingIntent.getService(getActivity(), 0, startIntent, PendingIntent.
     FLAG_CANCEL_CURRENT);
75                 usbManager.requestPermission(driver.getDevice(), pendingIntent);
76             }
77
78             if (port != null) {
79
80                 UsbDeviceConnection connection = usbManager.openDevice(port.getDriver().getDevice());
81
82                 if (connection != null) {
83                     try {
84                         port.open(connection);
85                         port.setParameters(115200, 8, UsbSerialPort.STOPBITS_1, UsbSerialPort.
     PARITY_NONE);
86
87 //                    String details = "CD  - Carrier Detect" + port.getCD() + '\n' +
88 //                            "CTS - Clear To Send" + port.getCTS() + '\n' +
89 //                            "DSR - Data Set Ready" + port.getDSR() + '\n' +
90 //                            "DTR - Data Terminal Ready" + port.getDTR() + '\n' +
91 //                            "DSR - Data Set Ready" + port.getDSR() + '\n' +
92 //                            "RI  - Ring Indicator" + port.getRI() + '\n' +
93 //                            "RTS - Request To Send" + port.getRTS();
94
95                     } catch (IOException e) {
96                         myUsbObservable.notifyObservers(
     getActivity().getString(R.string.error_opening_device)
97                                 + " " + e.getMessage());
98                         myUsbObservable.notifyObservers(false);
99                         closePort();
100                         return;
101                     }
102
103                     stopIoManager();
104                     startIoManager();
105                 }
106             }
107         }
108     }
```

6.33.3.4 onPause()

```
void it.unibo.torsello.bluetoothpositioning.util.UsbUtil.onPause ( )
```

```
46                          {
47         stopIoManager();
48         closePort();
49     }
```

6.33.3.5 onResume()

```
void it.unibo.torsello.bluetoothpositioning.util.UsbUtil.onResume ( )
```

```
51                          {
52         initializeUsb();
53     }
```

6.33.3.6 startIoManager()

```
void it.unibo.torsello.bluetoothpositioning.util.UsbUtil.startIoManager ( )  [private]
```

```
128                                 {
129         if (port != null) {
130
131             SerialInputOutputManager.Listener mListener =
132                     new SerialInputOutputManager.Listener() {
133
134                         @Override
135                         public void onRunError(Exception e) {
136                             myUsbObservable.notifyObservers(false);
137                             myUsbObservable.notifyObservers(
      getActivity().getString(R.string.usb_device_not_connected));
138                             myUsbObservable.notifyObservers(0D);
139                         }
140
141                         @Override
142                         public void onNewData(final byte[] data) {
143                             try {
144                                 myUsbObservable.notifyObservers(true);
145                                 myUsbObservable.notifyObservers(
      getActivity().getString(R.string.usb_device_connected));
146                                 myUsbObservable.notifyObservers(Double.
      valueOf(new String(data).trim()) / 100);
147                             } catch (NumberFormatException nfe) {
148                             }
149                         }
150                     };
151
152             mSerialIoManager = new SerialInputOutputManager(port, mListener);
153             mExecutor.submit(mSerialIoManager);
154         }
155     }
```

6.33.3.7 stopIoManager()

```
void it.unibo.torsello.bluetoothpositioning.util.UsbUtil.stopIoManager ( )  [private]
```

```
121                                 {
122         if (mSerialIoManager != null) {
123             mSerialIoManager.stop();
124             mSerialIoManager = null;
125         }
126     }
```

6.33.4 Documentazione dei membri dato

6.33.4.1 applicationActivity

```
ApplicationActivity it.unibo.torsello.bluetoothpositioning.util.UsbUtil.applicationActivity
[private]
```

6.33.4.2 mExecutor

```
final ExecutorService it.unibo.torsello.bluetoothpositioning.util.UsbUtil.mExecutor = Executors.↩
newSingleThreadExecutor()  [private]
```

6.33.4.3 mSerialIoManager

```
SerialInputOutputManager it.unibo.torsello.bluetoothpositioning.util.UsbUtil.mSerialIoManager
[private]
```

6.33.4.4 myUsbObservable

```
UsbMeasurementObservable it.unibo.torsello.bluetoothpositioning.util.UsbUtil.myUsbObservable
[private]
```

6.33.4.5 port

```
UsbSerialPort it.unibo.torsello.bluetoothpositioning.util.UsbUtil.port  [private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- UsbUtil.java

6.34 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil

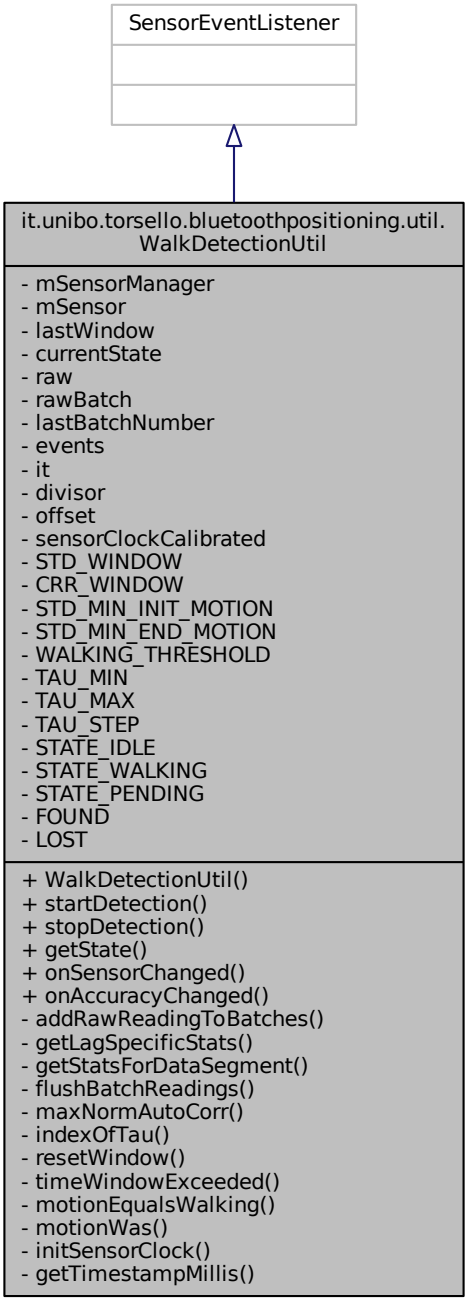
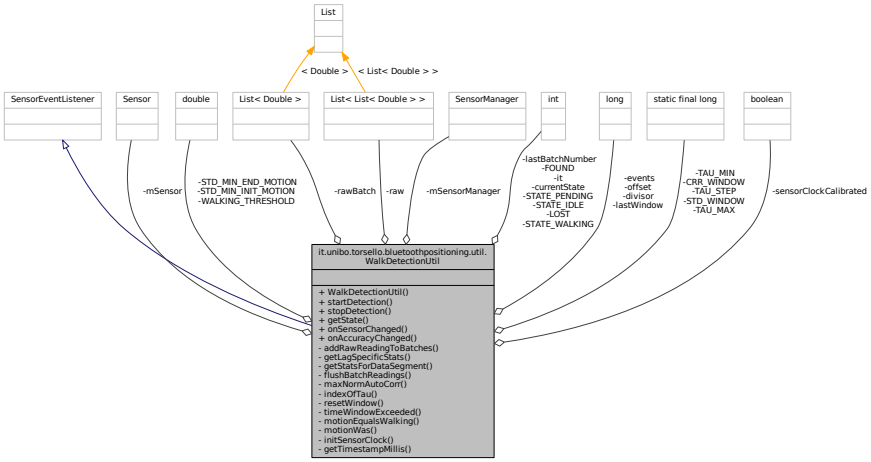


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil:



Membri pubblici

- WalkDetectionUtil (Application app)
- void startDetection ()
- void stopDetection ()
- int getState ()
- void onSensorChanged (SensorEvent event)
- void onAccuracyChanged (Sensor sensor, int accuracy)

Membri privati

- void addRawReadingToBatches (SensorEvent event)
- DescriptiveStatistics getLagSpecificStats (long maxLag)
- DescriptiveStatistics getStatsForDataSegment (List< Double > array)
- void flushBatchReadings ()
- double maxNormAutoCorr ()
- int indexOfTau (int tau)
- void resetWindow ()
- boolean timeWindowExceeded (long windowType)
- boolean motionEqualsWalking ()
- boolean motionWas (int event)
- void initSensorClock (long sys)
- long getTimestampMillis (long timestamp)

Attributi privati

- SensorManager mSensorManager
- Sensor mSensor
- long lastWindow = -1
- int currentState = STATE_IDLE
- List< List< Double > > raw
- List< Double > rawBatch
- int lastBatchNumber = 0
- long events [ ] = new long[2]
- int it = 0
- long divisor
- long offset
- boolean sensorClockCalibrated = false

Attributi privati statici

- static final long STD_WINDOW = 800
- static final long CRR_WINDOW = 2000
- static final double STD_MIN_INIT_MOTION = 0.6
- static final double STD_MIN_END_MOTION = 0.1
- static final double WALKING_THRESHOLD = 0.5
- static final long TAU_MIN = 600
- static final long TAU_MAX = 1500
- static final long TAU_STEP = 100
- static final int STATE_IDLE = 0
- static final int STATE_WALKING = 1
- static final int STATE_PENDING = 2
- static final int FOUND = 3
- static final int LOST = 4

6.34.1 Descrizione dettagliata

Created by Federico Torsello. federico.torsello@studio.unibo.it

6.34.2 Documentazione dei costruttori e dei distruttori

6.34.2.1 WalkDetectionUtil()

```
it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.WalkDetectionUtil (
            Application app )
```

```
61                                              {
62         mSensorManager = (SensorManager) app.getSystemService(Context.SENSOR_SERVICE);
63         mSensor = mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
64         raw = new ArrayList<>();
65         rawBatch = new ArrayList<>();
66     }
```

6.34.3 Documentazione delle funzioni membro

6.34.3.1 addRawReadingToBatches()

```
void it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.addRawReadingToBatches (
            SensorEvent event )  [private]
```

```
152                                                         {
153         int delta = FastMath.toIntExact((getTimestampMillis(event.timestamp) -
      lastWindow) / 100);
154         double measurement = sqrt(pow(event.values[0], 2) + pow(event.values[1], 2) + pow(event.values[2],
      2));
155         if (delta != lastBatchNumber) {
156             flushBatchReadings();
157             lastBatchNumber = delta;
158         }
159         rawBatch.add(measurement);
160     }
```

6.34.3.2 flushBatchReadings()

```
void it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.flushBatchReadings ( )
[private]
```

```
181                                     {
182         ArrayList<Double> array = new ArrayList<>(rawBatch.size());
183         array.addAll(rawBatch);
184         raw.add(array);
185         rawBatch.clear();
186     }
```

6.34.3.3 getLagSpecificStats()

```
DescriptiveStatistics it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.getLag↩
SpecificStats (
             long maxLag )  [private]
```

```
162                                                                 {
163         DescriptiveStatistics stats = new DescriptiveStatistics();
164         int maxBatch = FastMath.toIntExact(maxLag) / 100 + 1;
165         maxBatch = maxBatch > raw.size() ? raw.size() : maxBatch;
166         for (int i = 0; i < maxBatch; i++)
167             for (double d : raw.get(i)) {
168                 stats.addValue(d);
169             }
170         return stats;
171     }
```

6.34.3.4 getState()

```
int it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.getState ( )
```

```
78                          {
79         return currentState == STATE_PENDING ?
      STATE_IDLE : currentState;
80      }
```

6.34.3.5 getStatsForDataSegment()

```
DescriptiveStatistics it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.getStats↩
ForDataSegment (
             List< Double > array )  [private]
```

```
173                                                                 {
174         DescriptiveStatistics stats = new DescriptiveStatistics();
175         for (double d : array) {
176             stats.addValue(d);
177         }
178         return stats;
179     }
```

6.34.3.6 getTimestampMillis()

```
long it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.getTimestampMillis (
             long timestamp )  [private]
```

```
280                                             {
281         return timestamp / divisor + offset;
282     }
```

6.34.3.7 indexOfTau()

```
int it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.indexOfTau (
            int tau )  [private]
```

```
226                                     {
227         int index = 0;
228         for (int k = 0; k < tau; k++) {
229             index += raw.get(k).size();
230         }
231         return index;
232     }
```

6.34.3.8 initSensorClock()

```
void it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.initSensorClock (
            long sys )  [private]
```

```
264                                         {
265
266         sensorClockCalibrated = true;
267         long timestampDelta = events[1] - events[0];
268
269         if (timestampDelta > 500) { // in reality ~1 vs ~1,000,000
270             // timestamps are in nanoseconds
271             divisor = 1000000;
272         } else {
273             // timestamps are in milliseconds
274             divisor = 1;
275         }
276
277         offset = sys - events[0] / divisor;
278     }
```

6.34.3.9 maxNormAutoCorr()

```
double it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.maxNormAutoCorr ( )  [private]
```

Computes the maximum normalized auto correlation of raw with respect to time lags TAU_MIN and TAU_MAX

```
191                                     {
192         List<Double> rawConcat = new ArrayList<>();
193         double std, mean, stdTau, meanTau, sum;
194         double max = Double.MIN_VALUE;
195         //int tauOpt = 0;
196         for (List<Double> l :
197                 raw) {
198             rawConcat.addAll(l);
199         }
200         //Log.i("walk", "Raw size: " + rawConcat.size());
201         for (int tau = (FastMath.toIntExact(TAU_MIN) / 100) - 1; tau < FastMath.toIntExact(
    TAU_MAX) / 100; tau += TAU_STEP / 100) {
202             int tauIndex = indexOfTau(tau);
203             for (int m = 0; m < rawConcat.size() - (tauIndex * 2) - 1; m++) {
204                 DescriptiveStatistics stats = getStatsForDataSegment(rawConcat.
    subList(m, m + tauIndex));
205                 //Log.i("walk", "stats size: " + stats.getValues().length);
206                 std = stats.getStandardDeviation();
207                 mean = stats.getMean();
208                 DescriptiveStatistics statsTau = getStatsForDataSegment(rawConcat.
    subList(m + tauIndex, m + tauIndex * 2));
209                 //Log.i("walk", "statsTau size: " + statsTau.getValues().length);
210                 stdTau = statsTau.getStandardDeviation();
211                 meanTau = statsTau.getMean();
212                 sum = 0;
213                 int k;
214                 for (k = 0; k < tauIndex - 1; k++) {
215                     sum += (rawConcat.get(m + k) - mean) * (rawConcat.get(m + k + tauIndex) - meanTau);
216                 }
217                 double temp = sum / (std * stdTau * tauIndex);
218                 //Log.i("walk", "Tau: " + tau + " - X(" + m + "," + tauIndex + ") = " + temp);
219                 max = max(max, temp);
220                 //if (temp == max) tauOpt = tau;
221             }
222         }
223         return max;
224     }
```

6.34.3.10 motionEqualsWalking()

```
boolean it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.motionEqualsWalking ( )
[private]
```

```
244                                         {
245         double temp = maxNormAutoCorr();
246         //Log.i("walk", "corr: " + temp);
247         //Log.i("walk", "std" + getLagSpecificStats(CRR_WINDOW).getStandardDeviation());
248         return temp >= WALKING_THRESHOLD;
249     }
```

6.34.3.11 motionWas()

```
boolean it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.motionWas (
              int event )  [private]
```

```
251                                         {
252         boolean eventResult = false;
253         switch (event) {
254             case FOUND:
255                 eventResult = getLagSpecificStats(STD_WINDOW).
      getStandardDeviation() > STD_MIN_INIT_MOTION;
256                 break;
257             case LOST:
258                 eventResult = getLagSpecificStats(STD_WINDOW).
      getStandardDeviation() < STD_MIN_END_MOTION;
259                 break;
260         }
261         return eventResult;
262     }
```

6.34.3.12 onAccuracyChanged()

```
void it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.onAccuracyChanged (
              Sensor sensor,
              int accuracy )
```

Called when the accuracy of the registered sensor has changed.

Parametri

sensor	
accuracy	The new accuracy of this sensor, one of `SensorManager.SENSOR_STATUS_*`

```
148                                                     {
149         // Unneeded
150     }
```

6.34.3.13 onSensorChanged()

```
void it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.onSensorChanged (
              SensorEvent event )
```

Called when sensor values have changed.

Parametri

event	the SensorEvent.

```
88                                                  {
89          if (mSensor.getType() == Sensor.TYPE_ACCELEROMETER) {
90              if (sensorClockCalibrated) {
91
92                  if (lastWindow == -1) {
93                      lastWindow = getTimestampMillis(event.timestamp);
94                  }
95
96                  addRawReadingToBatches(event);
97
98                  switch (currentState) {
99                      case STATE_IDLE:
100                         if (timeWindowExceeded(STD_WINDOW)) {
101                             if (motionWas(FOUND)) {
102                                 currentState = STATE_PENDING;
103                                 Log.i("walk", "PENDING");
104                             } else {
105                                 resetWindow();
106                             }
107                         }
108                         break;
109                     case STATE_PENDING:
110                         if (timeWindowExceeded(CRR_WINDOW)) {
111                             if (motionEqualsWalking()) {
112                                 currentState = STATE_WALKING;
113                                 Log.i("walk", "WALKING");
114                             }
115                             resetWindow();
116                         }
117                         break;
118                     case STATE_WALKING:
119                         if (timeWindowExceeded(STD_WINDOW)) {
120                             if (motionWas(LOST)) {
121                                 currentState = STATE_IDLE;
122                                 Log.i("walk", "IDLE");
123                             }
124                             resetWindow();
125                         }
126                         break;
127                 }
128             } else {
129                 if (it == 0) {
130                     events[0] = event.timestamp;
131                     it++;
132                 } else {
133                     events[1] = event.timestamp;
134                     initSensorClock(System.currentTimeMillis());
135                 }
136             }
137         }
138     }
```

6.34.3.14 resetWindow()

```
void it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.resetWindow ( )  [private]
```

```
234                                 {
235         raw.clear();
236         lastWindow = -1;
237     }
```

6.34.3.15 startDetection()

```
void it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.startDetection ( )
```

```
68                                  {
69          mSensorManager.registerListener(this, mSensor, 1000);
70      }
```

6.34.3.16 stopDetection()

```
void it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.stopDetection ( )
```

```
72                                {
73          currentState = STATE_PENDING;
74          mSensorManager.unregisterListener(this);
75      }
```

6.34.3.17 timeWindowExceeded()

```
boolean it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.timeWindowExceeded (
              long windowType )  [private]
```

```
239                                                     {
240          flushBatchReadings();
241          return System.currentTimeMillis() - lastWindow >= windowType;
242      }
```

6.34.4 Documentazione dei membri dato

6.34.4.1 CRR_WINDOW

```
final long it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.CRR_WINDOW = 2000
[static], [private]
```

6.34.4.2 currentState

```
int it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.currentState = STATE_IDLE
[private]
```

6.34.4.3 divisor

```
long it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.divisor  [private]
```

6.34.4.4 events

```
long it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.events[] = new long[2]
[private]
```

6.34.4.5 FOUND

```
final int it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.FOUND = 3  [static],
[private]
```

6.34.4.6 it

```
int it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.it = 0  [private]
```

6.34.4.7 lastBatchNumber

```
int it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.lastBatchNumber = 0  [private]
```

6.34.4.8 lastWindow

```
long it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.lastWindow = -1  [private]
```

6.34.4.9 LOST

```
final int it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.LOST = 4  [static],
[private]
```

6.34.4.10 mSensor

```
Sensor it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.mSensor  [private]
```

6.34.4.11 mSensorManager

```
SensorManager it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.mSensorManager
[private]
```

6.34.4.12 offset

```
long it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.offset  [private]
```

6.34.4.13 raw

```
List<List<Double> > it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.raw  [private]
```

6.34.4.14 rawBatch

```
List<Double> it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.rawBatch  [private]
```

6.34.4.15 sensorClockCalibrated

```
boolean it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.sensorClockCalibrated =
false  [private]
```

6.34.4.16 STATE_IDLE

```
final int it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.STATE_IDLE = 0  [static],
[private]
```

6.34.4.17 STATE_PENDING

```
final int it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.STATE_PENDING = 2
[static], [private]
```

6.34.4.18 STATE_WALKING

```
final int it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.STATE_WALKING = 1
[static], [private]
```

6.34.4.19 STD_MIN_END_MOTION

```
final double it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.STD_MIN_END_MOTION
= 0.1  [static], [private]
```

6.34.4.20 STD_MIN_INIT_MOTION

```
final double it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.STD_MIN_INIT_MOTION
= 0.6  [static], [private]
```

6.34.4.21 STD_WINDOW

```
final long it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.STD_WINDOW = 800
[static], [private]
```

6.34.4.22 TAU_MAX

```
final long it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.TAU_MAX = 1500  [static],
[private]
```

6.34.4.23 TAU_MIN

```
final long it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.TAU_MIN = 600  [static],
[private]
```

6.34.4.24 TAU_STEP

```
final long it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.TAU_STEP = 100  [static],
[private]
```

6.34.4.25 WALKING_THRESHOLD

```
final double it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil.WALKING_THRESHOLD =
0.5  [static], [private]
```

La documentazione per questa classe è stata generata a partire dal seguente file:

- WalkDetectionUtil.java

7 Documentazione dei file

7.1 Riferimenti per il file ApplicationActivity.java

Composti

- class it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity

Package

- package it.unibo.torsello.bluetoothpositioning.activities

7.2 Riferimenti per il file CameraFragment.java

Composti

- class it.unibo.torsello.bluetoothpositioning.fragment.CameraFragment

Package

- package it.unibo.torsello.bluetoothpositioning.fragment

7.3 Riferimenti per il file CameraUtil.java

Composti

- class it.unibo.torsello.bluetoothpositioning.util.CameraUtil

Package

- package it.unibo.torsello.bluetoothpositioning.util

7.4 Riferimenti per il file CamTestFragment.java

Composti

- class it.unibo.torsello.bluetoothpositioning.examplesCamera.CamTestFragment

Package

- package it.unibo.torsello.bluetoothpositioning.examplesCamera

7.5 Riferimenti per il file ChartUtil.java

Composti

- class it.unibo.torsello.bluetoothpositioning.util.ChartUtil

Package

- package it.unibo.torsello.bluetoothpositioning.util

7.6 Riferimenti per il file CompassFragment.java

Composti

- class it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassFragment

Package

- package it.unibo.torsello.bluetoothpositioning.fragment.oldFragment

7.7 Riferimenti per il file CompassMagnoFragment.java

Composti

- class it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CompassMagnoFragment

Package

- package it.unibo.torsello.bluetoothpositioning.fragment.oldFragment

7.8 Riferimenti per il file CountPassFragment.java

Composti

- class it.unibo.torsello.bluetoothpositioning.fragment.oldFragment.CountPassFragment

Package

- package it.unibo.torsello.bluetoothpositioning.fragment.oldFragment

7.9 Riferimenti per il file Device.java

Composti

- class it.unibo.torsello.bluetoothpositioning.model.Device

Package

- package it.unibo.torsello.bluetoothpositioning.model

7.10 Riferimenti per il file DeviceCardViewAdapter.java

Composti

- class it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter
- class it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceViewHolder

Package

- package it.unibo.torsello.bluetoothpositioning.adapter

7.11 Riferimenti per il file DeviceChartFragment.java

Composti

- class it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceChartFragment

Package

- package it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers

7.12 Riferimenti per il file DeviceConstants.java

Composti

- class it.unibo.torsello.bluetoothpositioning.constant.DeviceConstants

Package

- package it.unibo.torsello.bluetoothpositioning.constant

7.13 Riferimenti per il file DeviceDetailFragment.java

Composti

- class it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailFragment

Package

- package it.unibo.torsello.bluetoothpositioning.fragment

7.14 Riferimenti per il file DeviceDetailInner1Fragment.java

Composti

- class it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceDetailInner1Fragment

Package

- package it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers

7.15 Riferimenti per il file DeviceDetailInner2Fragmet.java

Composti

- class it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragmet

Package

- package it.unibo.torsello.bluetoothpositioning.fragment

7.16 Riferimenti per il file DeviceListFragment.java

Composti

- class it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers.DeviceListFragment

Package

- package it.unibo.torsello.bluetoothpositioning.fragment.devicesObservers

7.17 Riferimenti per il file DeviceObservable.java

Composti

- class it.unibo.torsello.bluetoothpositioning.observables.DeviceObservable

Package

- package it.unibo.torsello.bluetoothpositioning.observables

7.18 Riferimenti per il file Estimation.java

Composti

- class it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation

Package

- package it.unibo.torsello.bluetoothpositioning.distanceEstimation

7.19 Riferimenti per il file FABBehavior.java

Composti

- class it.unibo.torsello.bluetoothpositioning.extra.FABBehavior

Package

- package it.unibo.torsello.bluetoothpositioning.extra

7.20 Riferimenti per il file KalmanFilter.java

Composti

- class it.unibo.torsello.bluetoothpositioning.kalmanFilter.KalmanFilter

Package

- package it.unibo.torsello.bluetoothpositioning.kalmanFilter

7.21 Riferimenti per il file KFBuilder.java

Composti

- class it.unibo.torsello.bluetoothpositioning.kalmanFilter.KFBuilder

Package

- package it.unibo.torsello.bluetoothpositioning.kalmanFilter

7.22 Riferimenti per il file KFilterConstansts.java

Composti

- class it.unibo.torsello.bluetoothpositioning.constant.KFilterConstansts

Package

- package it.unibo.torsello.bluetoothpositioning.constant

7.23 Riferimenti per il file MainActivity.java

Composti

- class it.unibo.torsello.bluetoothpositioning.activities.MainActivity

Package

- package it.unibo.torsello.bluetoothpositioning.activities

7.24 Riferimenti per il file MyArmaRssiFilter.java

Composti

- class it.unibo.torsello.bluetoothpositioning.configuration.MyArmaRssiFilter

Package

- package it.unibo.torsello.bluetoothpositioning.configuration

7.25 Riferimenti per il file Preview.java

Composti

- class it.unibo.torsello.bluetoothpositioning.examplesCamera.Preview

Package

- package it.unibo.torsello.bluetoothpositioning.examplesCamera

7.26 Riferimenti per il file SaveImageTask.java

Composti

- class it.unibo.torsello.bluetoothpositioning.examplesCamera.SaveImageTask

Package

- package it.unibo.torsello.bluetoothpositioning.examplesCamera

7.27 Riferimenti per il file SettingConstants.java

Composti

- class it.unibo.torsello.bluetoothpositioning.constant.SettingConstants

Package

- package it.unibo.torsello.bluetoothpositioning.constant

7.28 Riferimenti per il file SettingsFragment.java

Composti

- class it.unibo.torsello.bluetoothpositioning.fragment.SettingsFragment

Package

- package it.unibo.torsello.bluetoothpositioning.fragment

7.29 Riferimenti per il file StatePagerAdapter.java

Composti

- class it.unibo.torsello.bluetoothpositioning.adapter.StatePagerAdapter

Package

- package it.unibo.torsello.bluetoothpositioning.adapter

7.30 Riferimenti per il file UsbMeasurementFragment.java

Composti

- class it.unibo.torsello.bluetoothpositioning.fragment.usbObservers.UsbMeasurementFragment

Package

- package it.unibo.torsello.bluetoothpositioning.fragment.usbObservers

7.31 Riferimenti per il file UsbMeasurementObservable.java

Composti

- class it.unibo.torsello.bluetoothpositioning.observables.UsbMeasurementObservable

Package

- package it.unibo.torsello.bluetoothpositioning.observables

7.32 Riferimenti per il file UsbUtil.java

Composti

- class it.unibo.torsello.bluetoothpositioning.util.UsbUtil

Package

- package it.unibo.torsello.bluetoothpositioning.util

7.33 Riferimenti per il file WalkDetectionUtil.java

Composti

- class it.unibo.torsello.bluetoothpositioning.util.WalkDetectionUtil

Package

- package it.unibo.torsello.bluetoothpositioning.util

Indice analitico