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 constant
- package distanceEstimation
- package extra
- package filters
- package fragment
- package model
- package observables
- package task
- 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.constant

Composti

- class DeviceConstants
- class KFilterConstants
- class SettingConstants

5.8 Package it.unibo.torsello.bluetoothpositioning.distanceEstimation

Composti

- class Estimation

5.9 Package it.unibo.torsello.bluetoothpositioning.extra

Composti

- class FABBehavior

5.10 Package it.unibo.torsello.bluetoothpositioning.filters

Package

- package kalmanFilter
- package kalmanFilter2

Composti

- class MyArmaRssiFilter

5.11 Package it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter

Composti

- class KFilter
- class KFilterBuilder

5.12 Package it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter2

Composti

- class KFilter2

5.13 Package it.unibo.torsello.bluetoothpositioning.fragment

Composti

- class CameraFragment
- class DeviceChartFragment
- class DeviceDetailFragment
- class DeviceDetailInner0Fragment
- class DeviceDetailInner1Fragment
- class DeviceDetailInner2Fragment
- class DeviceDetailReportFragment
- class DeviceDetailResumeFragment
- class DeviceListFragment
- class SettingsFragment
- class UsbMeasurementFragment

5.14 Package it.unibo.torsello.bluetoothpositioning.model

Composti

- class Device

5.15 Package it.unibo.torsello.bluetoothpositioning.observables

Composti

- class DeviceObservable
- class UsbMeasurementObservable

5.16 Package it.unibo.torsello.bluetoothpositioning.task

Composti

- class SaveImageTask

5.17 Package it.unibo.torsello.bluetoothpositioning.util

Composti

- class CameraPreviewUtil
- class ChartUtil
- class ReportUtils
- class UsbUtil

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

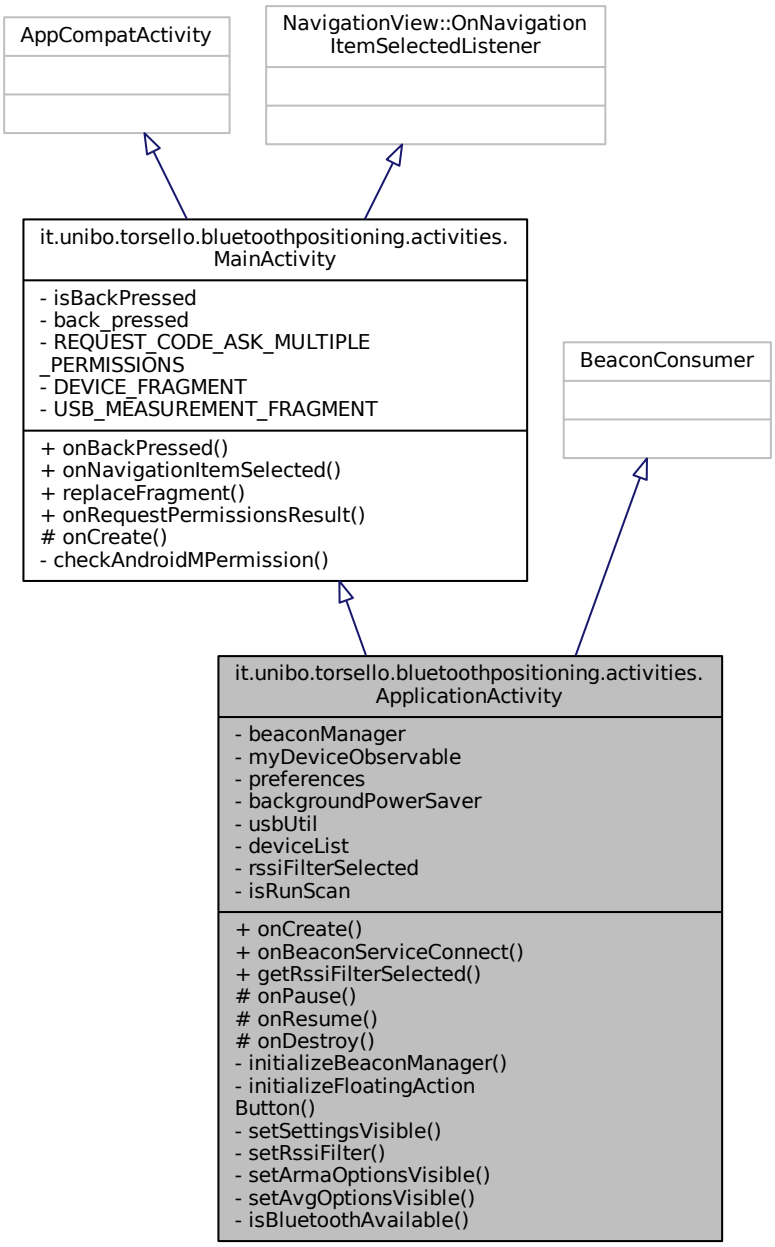
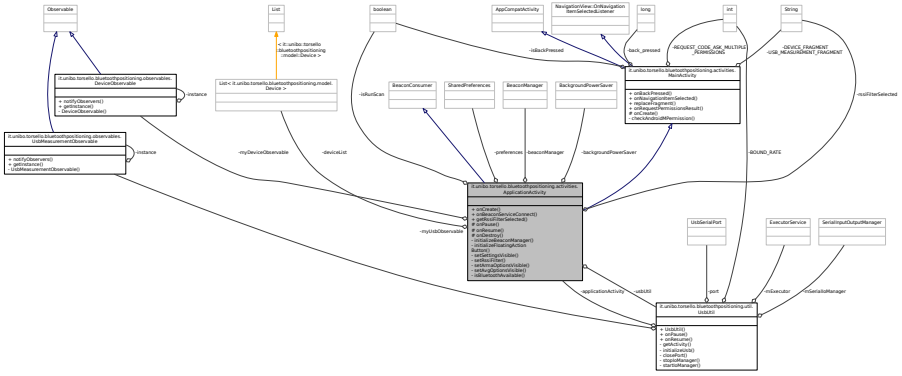


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity:



Membri pubblici

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

Membri protetti

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

Membri privati

- void initializeBeaconManager ()
- void initializeFloatingActionButton ()
- void setSettingsVisible (final boolean scanStarted)
- void setRssiFilter ()
- void setArmaOptionsVisible (final boolean visible)
- void setAvgOptionsVisible (final boolean visible)
- boolean isBluetoothAvailable ()

Attributi privati

- BeaconManager beaconManager
- DeviceObservable myDeviceObservable
- SharedPreferences preferences
- BackgroundPowerSaver backgroundPowerSaver
- UsbUtil usbUtil
- List< Device > deviceList
- String rssiFilterSelected
- boolean isRunScan = false

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 getRssiFilterSelected()

```
String it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity.getRssiFilter↩
Selected ( )
```

```
287                                         {
288         return rssiFilterSelected;
289     }
```

6.1.2.2 initializeBeaconManager()

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

```
72                                          {
73         beaconManager = BeaconManager.getInstanceForApplication(getApplicationContext());
74         beaconManager.bind(this);
75
76         // Save battery whenever the application is not visible.
77         // This reduces bluetooth power usage by about 60%
78         backgroundPowerSaver = new BackgroundPowerSaver(getApplicationContext());
79
80 //        Log.i("AltBeacon filter used:", BeaconManager.getRssiFilterImplClass().getSimpleName());
81
82         // for finding different type of beacon,
83         beaconManager.getBeaconParsers().clear();
84
85         // Alt beacon
86         beaconManager.getBeaconParsers().add(new BeaconParser()
87                 .setBeaconLayout(BeaconParser.ALTBEACON_LAYOUT));
88         // Detect the main identifier (UID) frame:
89         beaconManager.getBeaconParsers().add(new BeaconParser()
90                 .setBeaconLayout(BeaconParser.EDDYSTONE_UID_LAYOUT));
91         // Detect the telemetry (TLM) frame:
92         beaconManager.getBeaconParsers().add(new BeaconParser()
93                 .setBeaconLayout(BeaconParser.EDDYSTONE_TLM_LAYOUT));
94         // Detect the URL frame:
95         beaconManager.getBeaconParsers().add(new BeaconParser()
96                 .setBeaconLayout(BeaconParser.EDDYSTONE_URL_LAYOUT));
97         // Standard Apple iBeacon
98         beaconManager.getBeaconParsers().add(new BeaconParser()
99                 .setBeaconLayout(DeviceConstants.APPLE_BEACON_LAYOUT));
100         // Estimote Nearable
101         beaconManager.getBeaconParsers().add(new BeaconParser()
102                 .setBeaconLayout(DeviceConstants.ESTIMOTE_NEARABLE_LAYOUT));
103
104         beaconManager.setForegroundScanPeriod(500L);
105         beaconManager.setForegroundBetweenScanPeriod(0L);
106
107 //        beaconManager.setMaxTrackingAge(2000);
108     }
```

6.1.2.3 initializeFloatingActionButton()

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

```
110                                                 {
111         final FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
112         Snackbar.make(fab, R.string.snackBar_start_scanning, Snackbar.LENGTH_INDEFINITE).show();
113         fab.setOnClickListener(new View.OnClickListener() {
114             @Override
115             public void onClick(View view) {
116 
117                 if (isBluetoothAvailable()) {
118 
119                     isRunScan = !isRunScan;
120                     Region region = new Region("RegionId", null, null, null);
121 
122                     setSettingsVisible(isRunScan);
123 
124                     if (isRunScan) {
125                         fab.setImageResource(R.drawable.ic_bluetooth_searching_white_24dp);
126                         try {
127                             beaconManager.startRangingBeaconsInRegion(region);
128                         } catch (RemoteException e) {
129                             e.printStackTrace();
130                         }
131                         Snackbar.make(view, R.string.snackBar_scanning_enabled,
132                                 Snackbar.LENGTH_SHORT).show();
133                     } else {
134                         fab.setImageResource(R.drawable.ic_bluetooth_white_24dp);
135                         try {
136                             beaconManager.stopRangingBeaconsInRegion(region);
137                         } catch (RemoteException e) {
138                             e.printStackTrace();
139                         }
140                         Snackbar.make(view, R.string.snackBar_scanning_disabled,
141                                 Snackbar.LENGTH_INDEFINITE).show();
142                     }
143                 }
144             }
145         });
146     }
```

6.1.2.4 isBluetoothAvailable()

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

```
355                                             {
356         try {
357             final FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
358             if (!beaconManager.checkAvailability()) {
359 
360                 new AlertDialog.Builder(this)
361                         .setTitle(R.string.dialog_bluetooth_title)
362                         .setMessage(R.string.dialog_bluetooth_text)
363                         .setPositiveButton(android.R.string.ok, null)
364                         .setOnDismissListener(new DialogInterface.OnDismissListener() {
365                             @Override
366                             public void onDismiss(DialogInterface dialog) {
367                                 BluetoothAdapter.getDefaultAdapter().enable();
368                                 fab.setImageResource(R.drawable.ic_bluetooth_white_24dp);
369                                 Snackbar.make(fab, R.string.snackBar_start_scanning,
370                                         Snackbar.LENGTH_INDEFINITE).show();
371                             }
372                         }).show();
373                 fab.setImageResource(R.drawable.ic_bluetooth_disabled_black_24dp);
374                 Snackbar.make(fab, R.string.snackBar_scanning_disabled,
375                         Snackbar.LENGTH_INDEFINITE).show();
376                 return false;
377             }
378         } catch (RuntimeException e) {
379             e.getStackTrace();
380         }
381         return true;
382     }
```

6.1.2.5 onBeaconServiceConnect()

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

```
164                                         {
165
166         try {
167             beaconManager.updateScanPeriods();
168         } catch (RemoteException e) {
169             e.printStackTrace();
170         }
171
172         beaconManager.addRangeNotifier(new RangeNotifier() {
173             @Override
174             public void didRangeBeaconsInRegion(final Collection<Beacon> beacons, Region region) {
175
176                 setRssiFilter();
177
178                 for (Beacon b : beacons) {
179
180                     // take from the list the device
181                     Device device = DeviceConstants.DEVICE_MAP.get(b.getBluetoothAddress());
182
183                     if (device != null) { // useful only if DEVICE_MAP is empty
184                         double processNoise = preferences.getFloat(SettingConstants.
    KALMAN_NOISE_VALUE, 0);
185                         device.setBeacon(b);
186                         device.updateDistance(processNoise);
187
188                         if (!deviceList.contains(device)) {
189                             deviceList.add(device);
190                         }
191                     }
192                 }
193
194                 new Thread(new Runnable() {
195                     @Override
196                     public void run() {
197                         runOnUiThread(new Runnable() {
198
199                             @Override
200                             public void run() {
201                                 myDeviceObservable.
    notifyObservers(deviceList);
202                             }
203                         });
204                     }
205                 }).start();
206             }
207         });
208     }
```

6.1.2.6 onCreate()

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

```
55                                                  {
56         super.onCreate(savedInstanceState);
57
58         deviceList = new ArrayList<>();
59
60         myDeviceObservable = DeviceObservable.getInstance();
61
62         preferences = getSharedPreferences(SettingConstants.SETTINGS_PREFERENCES, 0);
63
64         usbUtil = new UsbUtil(this);
65
66         initializeBeaconManager();
67
68         initializeFloatingActionButton();
69
70     }
```

6.1.2.7 onDestroy()

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

```
346                                 {
347         if (beaconManager.isBound(this)) {
348             beaconManager.unbind(this);
349             backgroundPowerSaver.onActivityDestroyed(this);
350         }
351 
352         super.onDestroy();
353     }
```

6.1.2.8 onPause()

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

```
319                                 {
320         if (beaconManager.isBound(this)) {
321             beaconManager.setBackgroundMode(true);
322             backgroundPowerSaver.onActivityPaused(this);
323         }
324 
325         usbUtil.onPause();
326 
327         super.onPause();
328     }
```

6.1.2.9 onResume()

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

```
331                                 {
332         super.onResume();
333 
334         if (beaconManager.isBound(this)) {
335             beaconManager.setBackgroundMode(false);
336             backgroundPowerSaver.onActivityResumed(this);
337         }
338 
339         isBluetoothAvailable();
340 
341         usbUtil.onResume();
342 
343     }
```

6.1.2.10 setArmaOptionsVisible()

void it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity.setArmaOptions↩
Visible (
final boolean *visible*) [private]

```
291                                                         {
292         runOnUiThread(new Runnable() {
293             @Override
294             public void run() {
295                 if (visible) {
296                     findViewById(R.id.radioGroupArmaOptions).setVisibility(View.VISIBLE);
297                 } else {
298                     findViewById(R.id.radioGroupArmaOptions).setVisibility(View.GONE);
299                 }
300             }
301         });
302     }
```

6.1.2.11 setAvgOptionsVisible()

```
void it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity.setAvgOptions↩
Visible (
            final boolean visible )  [private]
```

```
304                                                                 {
305         runOnUiThread(new Runnable() {
306             @Override
307             public void run() {
308                 if (visible) {
309                     findViewById(R.id.radioGroupAverageOptions).setVisibility(View.VISIBLE);
310                 } else {
311                     findViewById(R.id.radioGroupAverageOptions).setVisibility(View.GONE);
312                 }
313             }
314         });
315     }
```

6.1.2.12 setRssiFilter()

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

```
210                                 {
211
212         StringBuilder selectOption = new StringBuilder();
213
214         int sorting = preferences.getInt(SettingConstants.FILTER_RSSI, 0);
215         switch (sorting) {
216             case 0:
217             case R.id.radioButton_no_rssi_filtering:
218                 setArmaOptionsVisible(false);
219                 setAvgOptionsVisible(false);
220
221                 selectOption.append("No RSSI filtering");
222
223                 MyArmaRssiFilter.enableArmaFilter(false);
224                 BeaconManager.setRssiFilterImplClass(MyArmaRssiFilter.class);
225                 break;
226
227             case R.id.radioButton_arma_rssi_filter:
228                 setArmaOptionsVisible(true);
229                 setAvgOptionsVisible(false);
230
231                 selectOption.append("ARMA RSSI filter");
232
233                 int armaOption = preferences.getInt(SettingConstants.ARMA_OPTION, 0);
234                 switch (armaOption) {
235                     case 0:
236                     case R.id.radioButton_arma_op1:
237                         MyArmaRssiFilter.setArmaSpeed(0.1D);
238                         break;
239                     case R.id.radioButton_arma_op2:
240                         MyArmaRssiFilter.setArmaSpeed(0.25D);
241                         break;
242                     case R.id.radioButton_arma_op3:
243                         MyArmaRssiFilter.setArmaSpeed(0.5D);
244                         break;
245                 }
246
247                 selectOption.append(" - Speed ").append(MyArmaRssiFilter.getArmaSpeed());
248
249                 MyArmaRssiFilter.enableArmaFilter(true);
250                 BeaconManager.setRssiFilterImplClass(MyArmaRssiFilter.class);
251                 break;
252
253             case R.id.radioButton_average_rssi_filter:
254
255                 setArmaOptionsVisible(false);
256                 setAvgOptionsVisible(true);
257
258                 selectOption.append("AVG RSSI filter");
259
260                 int avgOption = preferences.getInt(SettingConstants.AVG_OPTION, 0);
261
262                 long sampleExpirationMilliseconds = 0;
```

```
263
264                 switch (avgOption) {
265                     case 0:
266                     case R.id.radioButton_avg_op1:
267                         sampleExpirationMilliseconds = 20000L;
268                         break;
269                     case R.id.radioButton_avg_op2:
270                         sampleExpirationMilliseconds = 10000L;
271                         break;
272                     case R.id.radioButton_avg_op3:
273                         sampleExpirationMilliseconds = 5000L;
274                         break;
275                 }
276
277                 selectOption.append(" - ").append(sampleExpirationMilliseconds).append(" ms");
278
279                 RunningAverageRssiFilter.setSampleExpirationMilliseconds(sampleExpirationMilliseconds);
280                 BeaconManager.setRssiFilterImplClass(RunningAverageRssiFilter.class);
281                 break;
282         }
283
284         rssiFilterSelected = String.valueOf(selectOption);
285     }
```

6.1.2.13 setSettingsVisible()

```
void it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity.setSettingsVisible
(
            final boolean scanStarted )  [private]
```

```
148                                                         {
149         runOnUiThread(new Runnable() {
150             @Override
151             public void run() {
152                 if (!scanStarted) {
153                     findViewById(R.id.text_setting_disabled).setVisibility(View.VISIBLE);
154                     findViewById(R.id.nested_setting).setVisibility(View.GONE);
155                 } else {
156                     findViewById(R.id.text_setting_disabled).setVisibility(View.GONE);
157                     findViewById(R.id.nested_setting).setVisibility(View.VISIBLE);
158                 }
159             }
160         });
161     }
```

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 deviceList

```
List<Device> it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity.deviceList
[private]
```

6.1.3.4 isRunScan

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

6.1.3.5 myDeviceObservable

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

6.1.3.6 preferences

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

6.1.3.7 rssiFilterSelected

```
String it.unibo.torsello.bluetoothpositioning.activities.ApplicationActivity.rssiFilter↩
Selected  [private]
```

6.1.3.8 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

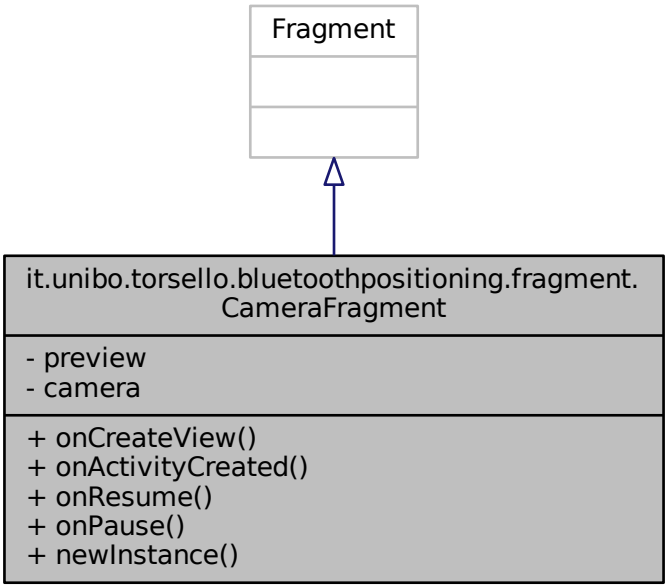
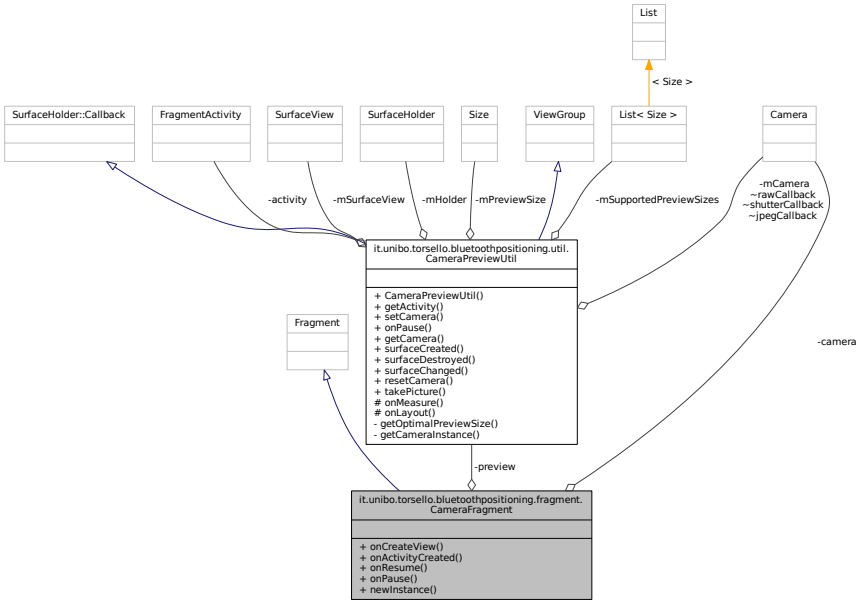


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.CameraFragment:



Membri pubblici

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

Membri pubblici statici

- static CameraFragment newInstance ()

Attributi privati

- CameraPreviewUtil preview
- Camera camera

6.2.1 Descrizione dettagliata

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

6.2.2 Documentazione delle funzioni membro

6.2.2.1 newInstance()

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

```
26                                          {
27         return new CameraFragment();
28     }
```

6.2.2.2 onActivityCreated()

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

```
71                                                                          {
72          super.onActivityCreated(savedInstanceState);
73
74          getActivity().findViewById(R.id.fab_camera).setOnClickListener(new View.OnClickListener() {
75              @Override
76              public void onClick(View v) {
77                  preview.takePicture();
78              }
79          });
80
81      }
```

6.2.2.3 onCreateView()

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

```
31                                                                                                  {
32
33          View root = inflater.inflate(R.layout.fragment_camera, container, false);
34          preview = new CameraPreviewUtil(getContext(), (SurfaceView) root.findViewById(R.id.
      surfaceView));
35          ((FrameLayout) root.findViewById(R.id.layout)).addView(preview);
36          preview.setKeepScreenOn(true);
37          preview.setOnClickListener(new OnClickListener() {
38
39              @Override
40              public void onClick(View arg0) {
41  //                preview.takePicture();
42                  camera.autoFocus(new Camera.AutoFocusCallback() {
43                      @Override
44                      public void onAutoFocus(boolean success, Camera arg1) {
45  //                          if (success) {
46  //                              preview.takePicture();
47  //                          }
48                      }
49                  });
50              }
51          });
52          preview.setOnLongClickListener(new View.OnLongClickListener() {
53              @Override
54              public boolean onLongClick(View arg0) {
55
56                  camera.autoFocus(new Camera.AutoFocusCallback() {
57                      @Override
58                      public void onAutoFocus(boolean success, Camera arg1) {
59  //                          if (success) {
60  //                              preview.takePicture();
61  //                          }
62                      }
63                  });
64                  return true;
65              }
66          });
67          return root;
68      }
```

6.2.2.4 onPause()

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

```
91                          {
92          preview.onPause();
93          super.onPause();
94      }
```

6.2.2.5 onResume()

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

```
84                          {
85         super.onResume();
86         preview.setCamera(getActivity());
87         camera = preview.getCamera();
88     }
```

6.2.3 Documentazione dei membri dato

6.2.3.1 camera

```
Camera it.unibo.torsello.bluetoothpositioning.fragment.CameraFragment.camera  [private]
```

6.2.3.2 preview

```
CameraPreviewUtil it.unibo.torsello.bluetoothpositioning.fragment.CameraFragment.preview  [private]
```

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.CameraPreviewUtil

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

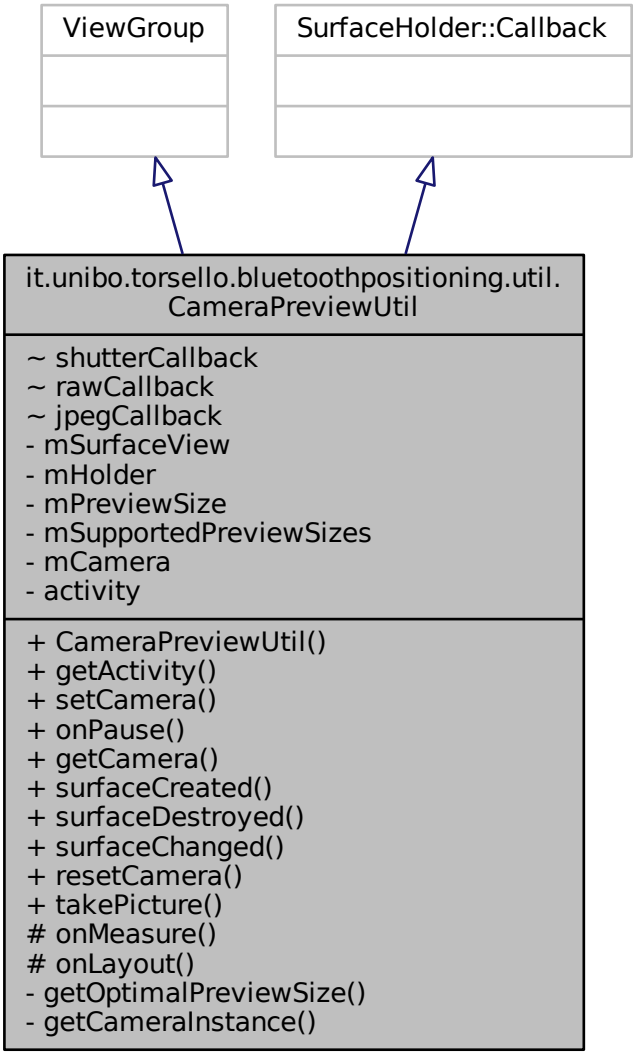
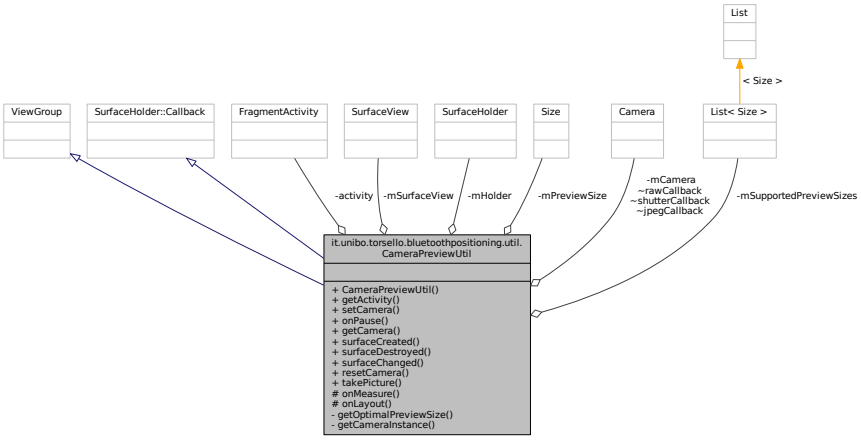


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.util.CameraPreviewUtil:



Membri pubblici

- CameraPreviewUtil (Context context, SurfaceView sv)
- FragmentActivity getActivity ()
- 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)

Attributi con visibilità di package

- 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

- SurfaceView mSurfaceView
- SurfaceHolder mHolder
- Size mPreviewSize
- List< Size > mSupportedPreviewSizes
- Camera mCamera
- FragmentActivity activity

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 CameraPreviewUtil()

```
it.unibo.torsello.bluetoothpositioning.util.CameraPreviewUtil.CameraPreviewUtil (
            Context context,
            SurfaceView sv )
```

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

6.3.3 Documentazione delle funzioni membro

6.3.3.1 getActivity()

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

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

6.3.3.2 getCamera()

```
Camera it.unibo.torsello.bluetoothpositioning.util.CameraPreviewUtil.getCamera ( )
```

```
81                              {
82         return mCamera;
83     }
```

6.3.3.3 getCameraInstance()

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

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

```
88                                              {
89
90          Camera c = null;
91
92          try {
93              int numCams = Camera.getNumberOfCameras();
94              if (numCams > 0) {
95                  c = Camera.open(0); // attempt to get a CameraUtil instance
96              }
97          } catch (RuntimeException e) {
98              // CameraUtil is not available (in use or does not exist)
99              e.getStackTrace();
100          }
101
102          return c; // returns null if camera is unavailable
103      }
```

6.3.3.4 getOptimalPreviewSize()

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

```
169                                                                         {
170         final double ASPECT_TOLERANCE = 0.1;
171         double targetRatio = (double) w / h;
172         if (sizes == null) return null;
173
174         Size optimalSize = null;
175         double minDiff = Double.MAX_VALUE;
176
177 //        int targetHeight = h;
178
179         // Try to find an size match aspect ratio and size
180         for (Size size : sizes) {
181             double ratio = (double) size.width / size.height;
182             if (Math.abs(ratio - targetRatio) > ASPECT_TOLERANCE) continue;
183             if (Math.abs(size.height - h) < minDiff) {
184                 optimalSize = size;
185                 minDiff = Math.abs(size.height - h);
186             }
187         }
188
189         // Cannot find the one match the aspect ratio, ignore the requirement
190         if (optimalSize == null) {
191             minDiff = Double.MAX_VALUE;
192             for (Size size : sizes) {
193                 if (Math.abs(size.height - h) < minDiff) {
194                     optimalSize = size;
195                     minDiff = Math.abs(size.height - h);
196                 }
197             }
198         }
199         return optimalSize;
200     }
```

6.3.3.5 onLayout()

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

```
120                                                                     {
121         if (changed && getChildCount() > 0) {
122             final View child = getChildAt(0);
123 
124             final int width = r - l;
125             final int height = b - t;
126 
127             int previewWidth = width;
128             int previewHeight = height;
129             if (mPreviewSize != null) {
130                 previewWidth = mPreviewSize.width;
131                 previewHeight = mPreviewSize.height;
132             }
133 
134             // Center the child SurfaceView within the parent.
135             if (width * previewHeight > height * previewWidth) {
136                 final int scaledChildWidth = previewWidth * height / previewHeight;
137                 child.layout((width - scaledChildWidth) / 2, 0,
138                         (width + scaledChildWidth) / 2, height);
139             } else {
140                 final int scaledChildHeight = previewHeight * width / previewWidth;
141                 child.layout(0, (height - scaledChildHeight) / 2,
142                         width, (height + scaledChildHeight) / 2);
143             }
144         }
145     }
```

6.3.3.6 onMeasure()

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

```
106                                                                       {
107         // We purposely disregard child measurements because act as a
108         // wrapper to a SurfaceView that centers the camera preview instead
109         // of stretching it.
110         final int width = resolveSize(getSuggestedMinimumWidth(), widthMeasureSpec);
111         final int height = resolveSize(getSuggestedMinimumHeight(), heightMeasureSpec);
112         setMeasuredDimension(width, height);
113 
114         if (mSupportedPreviewSizes != null) {
115             mPreviewSize = getOptimalPreviewSize(
    mSupportedPreviewSizes, width, height);
116         }
117     }
```

6.3.3.7 onPause()

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

```
73                          {
74         if (mCamera != null) {
75             mCamera.stopPreview();
76             mCamera.release();
77             mCamera = null;
78         }
79     }
```

6.3.3.8 resetCamera()

```
void it.unibo.torsello.bluetoothpositioning.util.CameraPreviewUtil.resetCamera ( )
```

```
215                                 {
216         new Thread(new Runnable() {
217             @Override
218             public void run() {
219                 mCamera.startPreview();
220             }
221         }).start();
222 
223 
224     }
```

6.3.3.9 setCamera()

```
void it.unibo.torsello.bluetoothpositioning.util.CameraPreviewUtil.setCamera (
              FragmentActivity fragmentActivity )
```

```
45                                                                  {
46 
47         this.activity = fragmentActivity;
48         try {
49             mCamera = getCameraInstance();
50         } catch (RuntimeException ex) {
51             Toast.makeText(fragmentActivity, "camera_not_found", Toast.LENGTH_LONG).show();
52         }
53 
54         if (mCamera != null) {
55 
56             mSupportedPreviewSizes = mCamera.getParameters().
    getSupportedPreviewSizes();
57             requestLayout();
58 
59             // get Camera parameters
60             Camera.Parameters params = mCamera.getParameters();
61 
62             List<String> focusModes = params.getSupportedFocusModes();
63             if (focusModes.contains(Camera.Parameters.FOCUS_MODE_AUTO)) {
64                 // set the focus mode
65                 params.setFocusMode(Camera.Parameters.FOCUS_MODE_AUTO);
66                 // set Camera parameters
67                 mCamera.setParameters(params);
68             }
69         }
70 
71     }
```

6.3.3.10 surfaceChanged()

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

```
203                                                                             {
204         if (mCamera != null) {
205             Camera.Parameters parameters = mCamera.getParameters();
206             parameters.setPreviewSize(mPreviewSize.width,
    mPreviewSize.height);
207             requestLayout();
208 
209             mCamera.setParameters(parameters);
210 
211             resetCamera();
212         }
213     }
```

6.3.3.11 surfaceCreated()

```
void it.unibo.torsello.bluetoothpositioning.util.CameraPreviewUtil.surfaceCreated (
              SurfaceHolder holder )
```

```
148                                                 {
149         // The Surface has been created, acquire the camera and tell it where
150         // to draw.
151         try {
152             if (mCamera != null) {
153                 mCamera.setDisplayOrientation(90);
154                 mCamera.setPreviewDisplay(holder);
155             }
156         } catch (IOException exception) {
157 //             Log.e(TAG, "IOException caused by setPreviewDisplay()", exception);
158         }
159     }
```

6.3.3.12 surfaceDestroyed()

```
void it.unibo.torsello.bluetoothpositioning.util.CameraPreviewUtil.surfaceDestroyed (
              SurfaceHolder holder )
```

```
162                                                   {
163         // Surface will be destroyed when we return, so stop the preview.
164         if (mCamera != null) {
165             mCamera.stopPreview();
166         }
167     }
```

6.3.3.13 takePicture()

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

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

6.3.4 Documentazione dei membri dato

6.3.4.1 activity

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

6.3.4.2 jpegCallback

```
Camera.PictureCallback it.unibo.torsello.bluetoothpositioning.util.CameraPreviewUtil.jpeg↩
Callback  [package]
```

Valore iniziale:

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

            resetCamera();
        }
    }
```

6.3.4.3 mCamera

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

6.3.4.4 mHolder

```
SurfaceHolder it.unibo.torsello.bluetoothpositioning.util.CameraPreviewUtil.mHolder  [private]
```

6.3.4.5 mPreviewSize

```
Size it.unibo.torsello.bluetoothpositioning.util.CameraPreviewUtil.mPreviewSize  [private]
```

6.3.4.6 mSupportedPreviewSizes

```
List<Size> it.unibo.torsello.bluetoothpositioning.util.CameraPreviewUtil.mSupportedPreview↩
Sizes  [private]
```

6.3.4.7 mSurfaceView

```
SurfaceView it.unibo.torsello.bluetoothpositioning.util.CameraPreviewUtil.mSurfaceView  [private]
```

6.3.4.8 rawCallback

```
Camera.PictureCallback it.unibo.torsello.bluetoothpositioning.util.CameraPreviewUtil.raw↩
Callback  [package]
```

Valore iniziale:

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

        }
    }
```

6.3.4.9 shutterCallback

```
Camera.ShutterCallback it.unibo.torsello.bluetoothpositioning.util.CameraPreviewUtil.shutter↩
Callback  [package]
```

Valore iniziale:

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

        }
    }
```

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

- CameraPreviewUtil.java

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

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

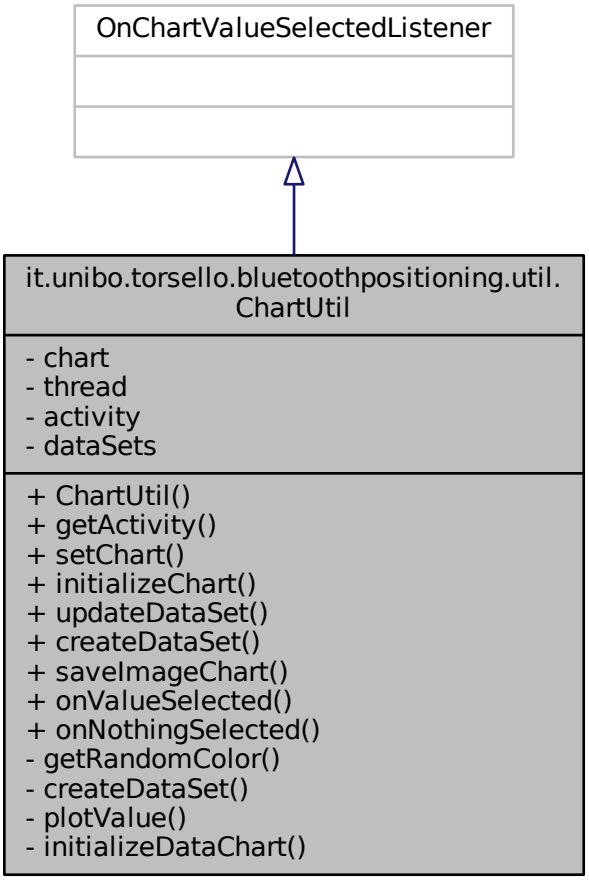
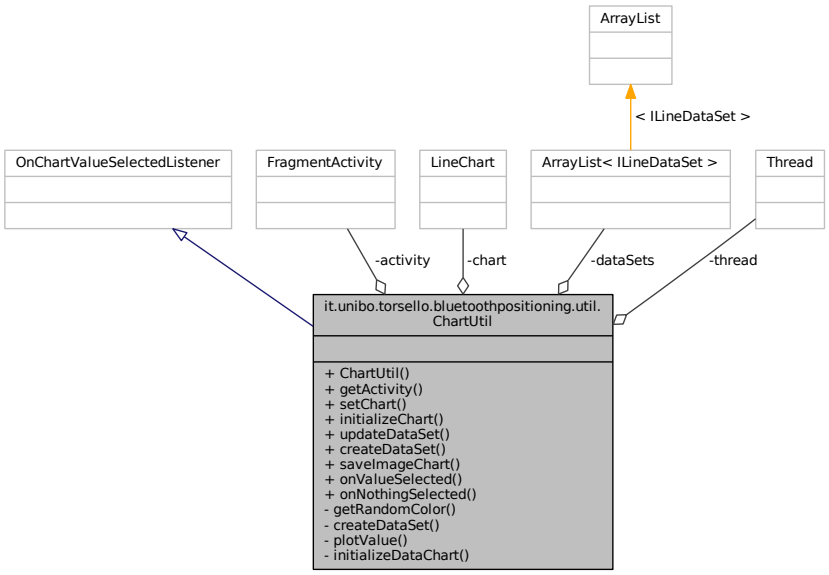


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.util.ChartUtil:



Membri pubblici

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

Membri privati

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

Attributi privati

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

6.4.1 Descrizione dettagliata

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

6.4.2 Documentazione dei costruttori e dei distruttori

6.4.2.1 ChartUtil()

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

```
42                                                  {
43         this.activity = fragmentActivity;
44     }
```

6.4.3 Documentazione delle funzioni membro

6.4.3.1 createDataSet() [1/2]

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

```
134                                                 {
135         // create a dataset and give it a type
136
137         for (String s : args) {
138             if (s != null) {
139                 if (s.equals(getActivity().getString(R.string.chart_arduino))) {
140                     dataSets.add(createDataSet(s, Color.RED));
141                 } else if (s.equals(getActivity().getString(R.string.chart_raw_distance))) {
142                     dataSets.add(createDataSet(s, Color.GREEN));
143                 } else if (s.equals(getActivity().getString(R.string.chart_altbeacon))) {
144                     dataSets.add(createDataSet(s, Color.BLUE));
145                 } else if (s.equals(getActivity().getString(R.string.chart_kalman_filter))) {
146                     dataSets.add(createDataSet(s, Color.CYAN));
147                 } else {
148                     dataSets.add(createDataSet(s,
      getRandomColor()));
149                 }
150             }
151         }
152
153     }
```

6.4.3.2 createDataSet() [2/2]

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

```
160                                                                 {
161         LineDataSet set = new LineDataSet(null, nameDataSet);
162         set.setColor(color);
163         return set;
164     }
```

6.4.3.3 getActivity()

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

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

6.4.3.4 getRandomColor()

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

```
155                                 {
156         Random rnd = new Random();
157         return Color.argb(255, rnd.nextInt(255), rnd.nextInt(255), rnd.nextInt(255));
158     }
```

6.4.3.5 initializeChart()

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

```
54                                  {
55         dataSets = new ArrayList<>();
56
57         chart.setOnChartValueSelectedListener(this);
58
59         // no description text
60         chart.setDescription("");
61         chart.setNoDataTextDescription("You need to provide data for the chart.");
62
63         chart.setDrawGridBackground(true);
64
65         // if disabled, scaling can be done on x- and y-axis separately
66         chart.setPinchZoom(true);
67
68         // set an alternative background color
69         chart.setBackgroundColor(Color.LTGRAY);
70
71         Typeface mTfLight = Typeface.createFromAsset(getActivity().getAssets(), "
    OpenSans-Light.ttf");
72         Typeface mTfBold = Typeface.createFromAsset(getActivity().getAssets(), "
    OpenSans-Bold.ttf");
73
74         // get the legend (only possible after setting data)
75         Legend l = chart.getLegend();
76 //         l.setPosition(Legend.LegendPosition.RIGHT_OF_CHART);
77 //         l.setOrientation(Legend.LegendOrientation.VERTICAL);
78         l.setXEntrySpace(7f);
79         l.setYEntrySpace(7f);
80
81         XAxis xl = chart.getXAxis();
82         xl.setTypeface(mTfLight);
83         xl.setGridColor(Color.LTGRAY);
84         xl.setTextColor(Color.WHITE);
85
86         YAxis leftAxis = chart.getAxisLeft();
87         leftAxis.setTypeface(mTfLight);
88         leftAxis.setTextColor(Color.WHITE);
89
90         YAxis rightAxis = chart.getAxisRight();
91         rightAxis.setTypeface(mTfBold);
92
93     }
```

6.4.3.6 initializeDataChart()

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

```
185                                                                     {
186
187         // create a data object with the datasets
188         LineData lineData = new LineData(dataSets);
189         lineData.setValueTextColor(Color.RED);
190         lineData.setValueTextSize(9f);
191         lineData.setValueFormatter(new DefaultValueFormatter(2));
192
193         // set data
194         chart.setData(lineData);
195     }
```

6.4.3.7 onNothingSelected()

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

```
213                                 {
214 //        Log.i("Nothing selected", "Nothing selected.");
215     }
```

6.4.3.8 onValueSelected()

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

```
207                                                     {
208         Snackbar.make(getActivity().findViewById(R.id.fab), e.copy() + " selected"
209                 , Snackbar.LENGTH_SHORT).show();
210     }
```

6.4.3.9 plotValue()

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

```
166                                                                 {
167
168         ILineDataSet set = data.getDataSetByIndex(index);
169
170         set.addEntry(new Entry(set.getEntryCount(), value.floatValue()));
171
172         data.notifyDataChanged();
173
174         // let the chart know it's data has changed
175         chart.notifyDataSetChanged();
176
177         // limit the number of visible entries
178         chart.setVisibleXRangeMaximum(10);
179
180         // move to the latest entry
181         chart.moveViewToX(data.getEntryCount());
182
183     }
```

6.4.3.10 saveImageChart()

```
void it.unibo.torsello.bluetoothpositioning.util.ChartUtil.saveImageChart (
              String chartName,
              String formattedDate )
```

```
197                                                                     {
198         String nameImage = chartName + " " + System.currentTimeMillis();
199         chart.saveToGallery(nameImage, chartName
200                 + " - " + formattedDate, null, Bitmap.CompressFormat.JPEG, 100);
201
202         Snackbar.make(getActivity().findViewById(R.id.fab), chartName + " stored"
203                 , Snackbar.LENGTH_SHORT).show();
204     }
```

6.4.3.11 setChart()

```
void it.unibo.torsello.bluetoothpositioning.util.ChartUtil.setChart (
            LineChart chart )
```

```
50                                          {
51         this.chart = chart;
52     }
```

6.4.3.12 updateDataSet()

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

```
95                                                                          {
96         if (thread != null)
97             thread.interrupt();
98
99         thread = new Thread(new Runnable() {
100            @Override
101            public void run() {
102                if (getActivity() != null) {
103
104                    getActivity().runOnUiThread(new Runnable() {
105                        @Override
106                        public void run() {
107
108                            LineData data = chart.getData();
109                            if (data == null) {
110                                if (dataSets != null) {
111                                    initializeDataChart(
    dataSets);
112                                } else {
113                                    throw new Error("Error: dataSet is null!!!");
114                                }
115                            } else {
116                                if (data.getDataSetCount() > 0) {
117
118 //                                   Log.i("miao", data.getDataSetCount()  +" ");
119                                    for (int i = 0; i < doubleArrayList.size(); i++) {
120                                        plotValue(data, i, doubleArrayList.get(i));
121                                    }
122
123                                }
124                            }
125                        }
126                    });
127                }
128            }
129        });
130
131        thread.start();
132    }
```

6.4.4 Documentazione dei membri dato

6.4.4.1 activity

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

6.4.4.2 chart

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

6.4.4.3 dataSets

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

6.4.4.4 thread

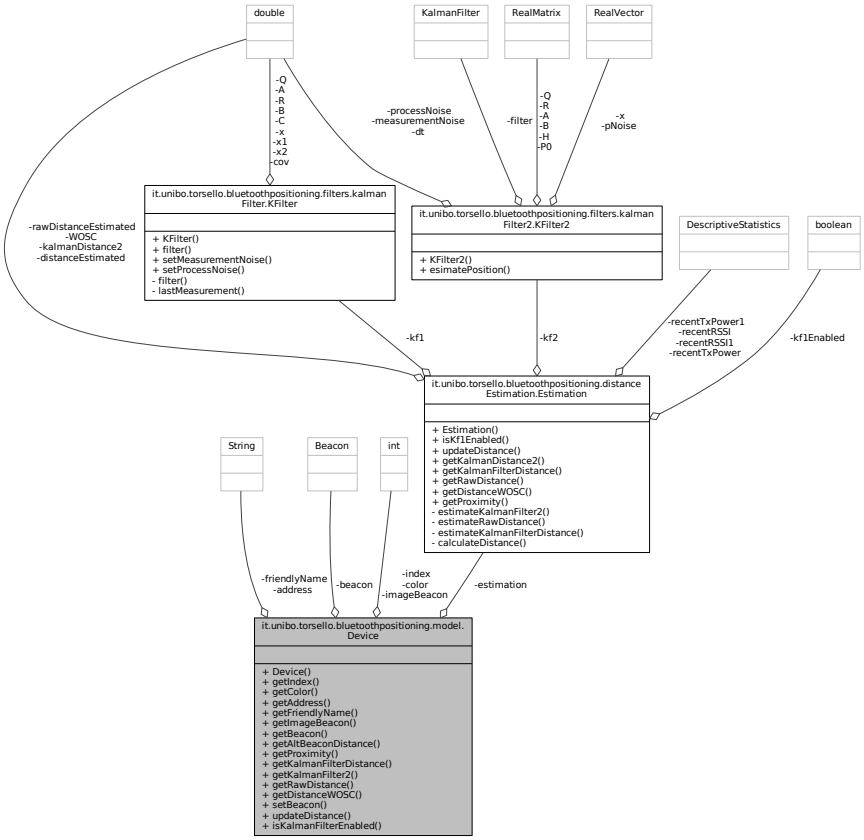
```
Thread it.unibo.torsello.bluetoothpositioning.util.ChartUtil.thread  [private]
```

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

- ChartUtil.java

6.5 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 getKalmanFilter2 ()
- double getRawDistance ()
- double getDistanceWOSC ()
- void setBeacon (Beacon beacon)
- void updateDistance (double processNoise)
- boolean isKalmanFilterEnabled ()

Attributi privati

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

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 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      }
```

6.5.3 Documentazione delle funzioni membro

6.5.3.1 getAddress()

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

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

6.5.3.2 getAltBeaconDistance()

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

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

6.5.3.3 getBeacon()

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

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

6.5.3.4 getColor()

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

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

6.5.3.5 getDistanceWOSC()

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

```
79                                  {
80         return estimation.getDistanceWOSC();
81     }
```

6.5.3.6 getFriendlyName()

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

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

6.5.3.7 getImageBeacon()

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

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

6.5.3.8 getIndex()

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

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

6.5.3.9 getKalmanFilter2()

```
double it.unibo.torsello.bluetoothpositioning.model.Device.getKalmanFilter2 ( )
```

```
71                                  {
72         return estimation.getKalmanDistance2();
73     }
```

6.5.3.10 getKalmanFilterDistance()

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

```
63                                          {
64         if (!estimation.isKf1Enabled()) {
65             return 0;
66         }
67
68         return estimation.getKalmanFilterDistance();
69     }
```

6.5.3.11 getProximity()

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

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

6.5.3.12 getRawDistance()

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

```
75                                  {
76         return estimation.getRawDistance();
77     }
```

6.5.3.13 isKalmanFilterEnabled()

```
boolean it.unibo.torsello.bluetoothpositioning.model.Device.isKalmanFilterEnabled ( )
```

```
93                                          {
94         return estimation.isKf1Enabled();
95     }
```

6.5.3.14 setBeacon()

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

```
83                                      {
84         this.beacon = beacon;
85     }
```

6.5.3.15 updateDistance()

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

```
87                                                  {
88         if (beacon != null) {
89             estimation.updateDistance(beacon, processNoise);
90         }
91     }
```

6.5.4 Documentazione dei membri dato

6.5.4.1 address

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

6.5.4.2 beacon

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

6.5.4.3 color

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

6.5.4.4 estimation

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

6.5.4.5 friendlyName

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

6.5.4.6 imageBeacon

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

6.5.4.7 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

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

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

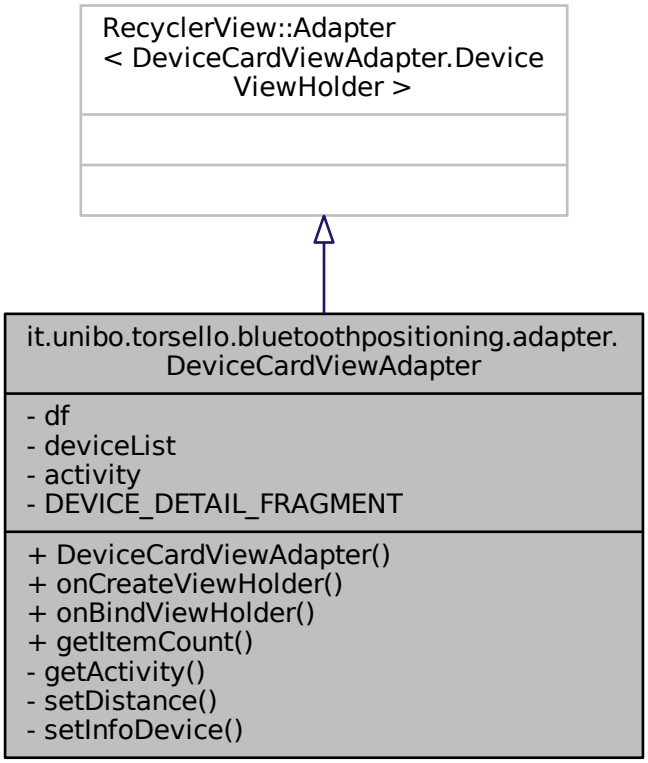
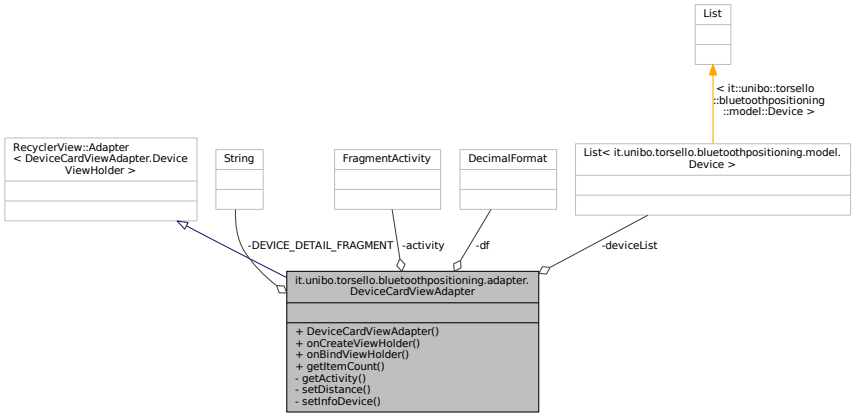


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter:



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"

6.6.1 Descrizione dettagliata

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

6.6.2 Documentazione dei costruttori e dei distruttori

6.6.2.1 DeviceCardViewAdapter()

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

```
39                                                                                  {
40
41         this.deviceList = new ArrayList<>();
42         this.deviceList = deviceList;
43         this.activity = fragmentActivity;
44
45         df = new DecimalFormat("0.00", DecimalFormatSymbols.getInstance());
46     }
```

6.6.3 Documentazione delle funzioni membro

6.6.3.1 getActivity()

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

```
48                                          {
49         return activity;
50     }
```

6.6.3.2 getItemCount()

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

```
226                                 {
227         return deviceList.size();
228     }
```

6.6.3.3 onBindViewHolder()

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

```
61                                                                                  {
62
63          final Beacon beacon = deviceList.get(position).getBeacon();
64          final Device device = deviceList.get(position);
65
66          setInfoDevice(holder, beacon);
67
68          setDistance(holder, device);
69
70          final Integer imageBeacon = device.getImageBeacon();
71          if (imageBeacon != null) {
72              holder.imageView.setImageResource(imageBeacon);
73          } else {
74              holder.imageView.setImageResource(R.drawable.beacon_unknown);
75          }
76
77          holder.rssiTextView.setText(String.format("%sdb", beacon.getRssi()));
78
79          holder.txPowerTextView.setText(String.format("%sdb", beacon.getTxPower()));
80
81          final String friendlyName = device.getFriendlyName();
82          if (friendlyName != null) {
83              holder.friendlyNameTextView.setText(friendlyName);
84          } else {
85              holder.friendlyNameTextView.setText(android.R.string.unknownName);
86          }
87
88          final String bluetoothName = beacon.getBluetoothName();
89          if (bluetoothName != null) {
90              holder.defaultNameTextView.setText(bluetoothName);
91          } else {
92              holder.defaultNameTextView.setText(android.R.string.unknownName);
93          }
94
95          final String macAddress = beacon.getBluetoothAddress();
96          if (macAddress != null) {
97              holder.macTextView.setText(macAddress);
98          } else {
99              holder.macTextView.setText(android.R.string.unknownName);
100         }
101
102         final String proximity = device.getProximity();
103         if (proximity != null) {
104             holder.proximityTextView.setText(proximity);
105         } else {
106             holder.proximityTextView.setText(android.R.string.unknownName);
107         }
108
109         final Integer color = device.getColor();
110         if (color != null) {
111             holder.colorTextView.setText(color);
112         } else {
113             holder.colorTextView.setText(android.R.string.unknownName);
114         }
115
116         holder.view.setOnClickListener(new View.OnClickListener() {
117             @Override
118             public void onClick(View v) {
119                 final String deviceDetailName;
120                 if (device.getFriendlyName() != null) {
121                     deviceDetailName = device.getFriendlyName();
122                 } else {
123                     deviceDetailName = device.getAddress();
```

```
124                 }
125
126                 new Thread(new Runnable() {
127                     @Override
128                     public void run() {
129                         getActivity().runOnUiThread(new Runnable() {
130                             @Override
131                             public void run() {
132
133                                 Fragment currentFrag = getActivity().getSupportFragmentManager()
134                                         .findFragmentByTag(
    DEVICE_DETAIL_FRAGMENT);
135
136                                 if (currentFrag == null) {
137                                     getActivity().getSupportFragmentManager().beginTransaction()
138                                             .replace(R.id.contentMainLayout,
139                                                     DeviceDetailFragment.newInstance(deviceDetailName),
140                                                     DEVICE_DETAIL_FRAGMENT)
141                                             .commit();
142                                 }
143                             }
144                         });
145                     }
146                 }).start();
147             }
148         });
149     }
```

6.6.3.4 onCreateViewHolder()

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

```
53                                                                                  {
54
55         View root = LayoutInflater.from(parent.getContext())
56                 .inflate(R.layout.list_items, parent, false);
57         return new DeviceViewHolder(root);
58     }
```

6.6.3.5 setDistance()

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

```
151                                                                 {
152         holder.altbeaconDistanceTextView.setText(String.format("%sm", df.format(device.
    getAltBeaconDistance())));
153
154         holder.standardRawDistanceTextView.setText(String.format("%sm", df.format(device.getRawDistance()
    )));
155
156         if (!device.isKalmanFilterEnabled()) {
157             holder.kalmanFilterDistanceTextView.setText(getActivity().getText(R.string.
    kalman_filter_disabled));
158         } else {
159             holder.kalmanFilterDistanceTextView.setText(String.format("%sm", df.format(device.
    getKalmanFilterDistance())));
160         }
161
162         holder.kf2TextView.setText(String.format("%sm", df.format(device.getKalmanFilter2())));
163     }
```

6.6.3.6 setInfoDevice()

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

```
165                                                                     {
166         if (beacon.getServiceUuid() == 0xfeaa) {
167
168             holder.visibilityUUIDLinearLayout.setVisibility(View.GONE);
169             holder.visibilityNameSpaceLinearLayout.setVisibility(View.VISIBLE);
170
171             if (beacon.getBeaconTypeCode() == 0x00) {
172                 // Eddystone-UID
173                 if (beacon.getId1() != null) {
174                     holder.nameSpaceTextView.setText(beacon.getId1().toString());
175                 } else {
176                     holder.nameSpaceTextView.setText(android.R.string.unknownName);
177                 }
178
179                 if (beacon.getId2() != null) {
180                     holder.instanceTextView.setText(beacon.getId2().toString());
181                 } else {
182                     holder.instanceTextView.setText(android.R.string.unknownName);
183                 }
184
185             } else if (beacon.getBeaconTypeCode() == 0x10) {
186                 // Eddystone-URL
187                 // String url = UrlBeaconUrlCompressor.uncompress(beacon.getId1().toByteArray());
188             } else if (beacon.getBeaconTypeCode() == 0x20) {
189                 if (!beacon.getExtraDataFields().isEmpty()) {
190                     // Eddystone-TLM
191                 }
192             }
193         } else if (beacon.getServiceUuid() == 0xbeac) {
194             // AltBeacon
195         } else if (beacon.getBeaconTypeCode() == 0x0215) { //533 in dec)
196
197             holder.visibilityUUIDLinearLayout.setVisibility(View.VISIBLE);
198             holder.visibilityNameSpaceLinearLayout.setVisibility(View.GONE);
199
200             // AppleIBeacon
201             if (beacon.getId1() != null) {
202                 holder.uuidTextView.setText(beacon.getId1().toString());
203             } else {
204                 holder.uuidTextView.setText(android.R.string.unknownName);
205             }
206
207             if (beacon.getId2() != null) {
208                 holder.majorTextView.setText(beacon.getId2().toString());
209             } else {
210                 holder.majorTextView.setText(android.R.string.unknownName);
211             }
212
213             if (beacon.getId3() != null) {
214                 holder.minorTextView.setText(beacon.getId3().toString());
215             } else {
216                 holder.minorTextView.setText(android.R.string.unknownName);
217             }
218
219         } else if (beacon.getBeaconTypeCode() == 0x0101) {
220             // EstimoteNearable
221         }
222
223     }
```

6.6.4 Documentazione dei membri dato

6.6.4.1 activity

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

6.6.4.2 DEVICE_DETAIL_FRAGMENT

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

6.6.4.3 deviceList

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

6.6.4.4 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

6.7 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.fragment.DeviceChartFragment

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

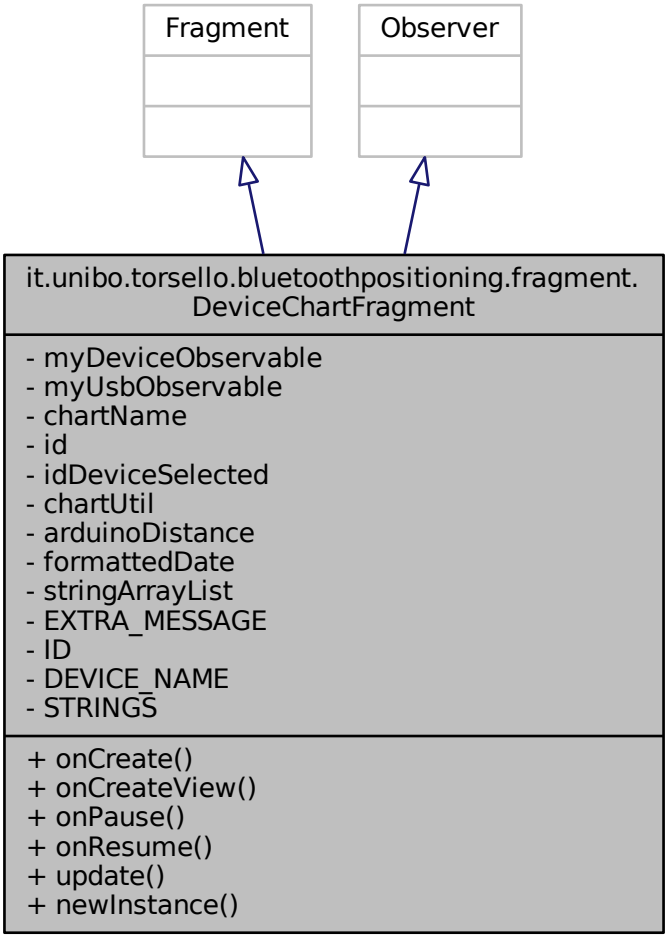
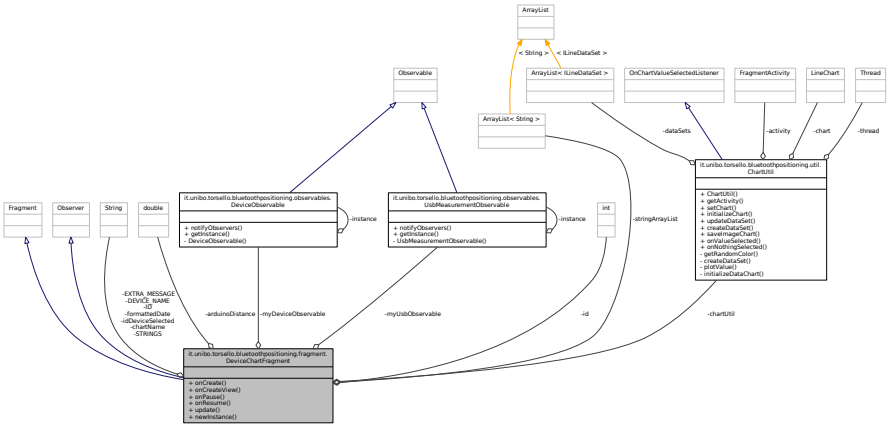


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.DeviceChartFragment:



Membri pubblici

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

Membri pubblici statici

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

Attributi privati

- DeviceObservable myDeviceObservable
- UsbMeasurementObservable myUsbObservable
- String chartName
- int id
- String idDeviceSelected
- ChartUtil chartUtil
- double arduinoDistance = 0D
- String formattedDate
- ArrayList< String > stringArrayList

Attributi privati statici

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

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

```
52                                                                          {
53         DeviceChartFragment fragment = new DeviceChartFragment();
54         Bundle args = new Bundle();
55         args.putString(EXTRA_MESSAGE, message);
56         args.putInt(ID, id);
57         args.putString(DEVICE_NAME, deviceName);
58         args.putStringArrayList(STRINGS, strings);
59         fragment.setArguments(args);
60         return fragment;
61     }
```

6.7.2.2 onCreate()

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

```
64                                                  {
65         super.onCreate(savedInstanceState);
66
67         id = getArguments().getInt(ID);
68         chartName = getArguments().getString(EXTRA_MESSAGE);
69         idDeviceSelected = getArguments().getString(DEVICE_NAME);
70         stringArrayList = getArguments().getStringArrayList(
     STRINGS);
71
72         myDeviceObservable = DeviceObservable.getInstance();
73         myUsbObservable = UsbMeasurementObservable.getInstance();
74
75         chartUtil = new ChartUtil(getActivity());
76
77         SimpleDateFormat df = new SimpleDateFormat("dd-MMM-yyyy", Locale.getDefault());
78         formattedDate = df.format(Calendar.getInstance().getTime());
79     }
```

6.7.2.3 onCreateView()

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

```
82                                                                                                                  {
83
84          View root = inflater.inflate(R.layout.fragment_device_detail_chart, container, false);
85
86          Button button = (Button) root.findViewById(R.id.button_save_image);
87          button.setOnClickListener(new View.OnClickListener() {
88              @Override
89              public void onClick(View v) {
90                  chartUtil.saveImageChart(chartName,
      formattedDate);
91              }
92          });
93
94          LineChart lineChart = (LineChart) root.findViewById(R.id.chart);
95
96          // add the charts
97          chartUtil.setChart(lineChart);
98
99          return root;
100     }
```

6.7.2.4 onPause()

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

```
103                          {
104         myDeviceObservable.deleteObserver(this);
105         myUsbObservable.deleteObserver(this);
106         super.onPause();
107     }
```

6.7.2.5 onResume()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceChartFragment.onResume ( )
```

```
110                           {
111         super.onResume();
112         myDeviceObservable.addObserver(this);
113         myUsbObservable.addObserver(this);
114         chartUtil.initializeChart();
115     }
```

6.7.2.6 update()

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

```
118                                                          {
119
120         if (observable instanceof UsbMeasurementObservable) {
121             if (arg instanceof Double) {
122                 arduinoDistance = (Double) arg;
123             }
124         }
125
126         if (observable instanceof DeviceObservable) {
127             if (arg instanceof List) {
128
129                 List<Device> devices = (List<Device>) arg;
130
131                 for (Device deviceSelected : devices) {
132                     if (deviceSelected.getFriendlyName().equals(idDeviceSelected) ||
133                             deviceSelected.getAddress().equals(idDeviceSelected)) {
134
135                         if (chartUtil != null) {
136
137                             chartUtil.createDataSet(
      stringArrayList);
```

```
138
139                             ArrayList<Double> dataSetForUpdates = new ArrayList<>();
140
141                             for (String s : stringArrayList) {
142                                 if (s.equals(getString(R.string.chart_arduino))) {
143                                     dataSetForUpdates.add(arduinoDistance);
144                                 }
145
146                                 if (s.equals(getString(R.string.chart_raw_distance))) {
147                                     dataSetForUpdates.add(deviceSelected.getRawDistance());
148                                 }
149
150                                 if (s.equals(getString(R.string.chart_altbeacon))) {
151                                     dataSetForUpdates.add(deviceSelected.getAltBeaconDistance());
152
153                                 }
154
155                                 if (s.equals(getString(R.string.chart_kalman_filter))) {
156                                     dataSetForUpdates.add(deviceSelected.getKalmanFilterDistance());
157                                 }
158                             }
159
160                             chartUtil.updateDataSet(dataSetForUpdates);
161                         }
162                     }
163                 }
164             }
165         }
166     }
```

6.7.3 Documentazione dei membri dato

6.7.3.1 arduinoDistance

double it.unibo.torsello.bluetoothpositioning.fragment.DeviceChartFragment.arduinoDistance = 0D [private]

6.7.3.2 chartName

String it.unibo.torsello.bluetoothpositioning.fragment.DeviceChartFragment.chartName [private]

6.7.3.3 chartUtil

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

6.7.3.4 DEVICE_NAME

final String it.unibo.torsello.bluetoothpositioning.fragment.DeviceChartFragment.DEVICE_NAME = "DEVICE_NAME" [static], [private]

6.7.3.5 EXTRA_MESSAGE

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

6.7.3.6 formattedDate

String it.unibo.torsello.bluetoothpositioning.fragment.DeviceChartFragment.formattedDate [private]

6.7.3.7 ID

final String it.unibo.torsello.bluetoothpositioning.fragment.DeviceChartFragment.ID = "ID" [static], [private]

6.7.3.8 id

```
int it.unibo.torsello.bluetoothpositioning.fragment.DeviceChartFragment.id  [private]
```

6.7.3.9 idDeviceSelected

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

6.7.3.10 myDeviceObservable

```
DeviceObservable it.unibo.torsello.bluetoothpositioning.fragment.DeviceChartFragment.my↩
DeviceObservable  [private]
```

6.7.3.11 myUsbObservable

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

6.7.3.12 stringArrayList

```
ArrayList<String> it.unibo.torsello.bluetoothpositioning.fragment.DeviceChartFragment.string↩
ArrayList  [private]
```

6.7.3.13 STRINGS

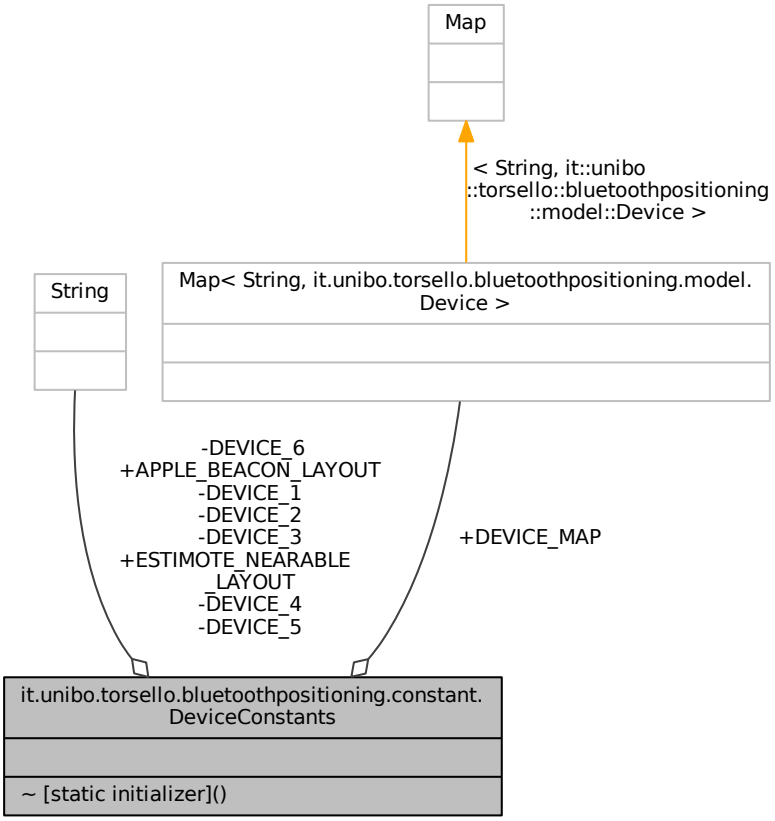
```
final String it.unibo.torsello.bluetoothpositioning.fragment.DeviceChartFragment.STRINGS =
"STRINGS"  [static], [private]
```

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

- DeviceChartFragment.java

6.8 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"

6.8.1 Descrizione dettagliata

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

6.8.2 Documentazione delle funzioni membro

6.8.2.1 [static initializer]()

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

6.8.3 Documentazione dei membri dato

6.8.3.1 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]
```

6.8.3.2 DEVICE_1

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

6.8.3.3 DEVICE_2

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

6.8.3.4 DEVICE_3

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

6.8.3.5 DEVICE_4

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

6.8.3.6 DEVICE_5

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

6.8.3.7 DEVICE_6

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

6.8.3.8 DEVICE_MAP

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

6.8.3.9 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

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

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

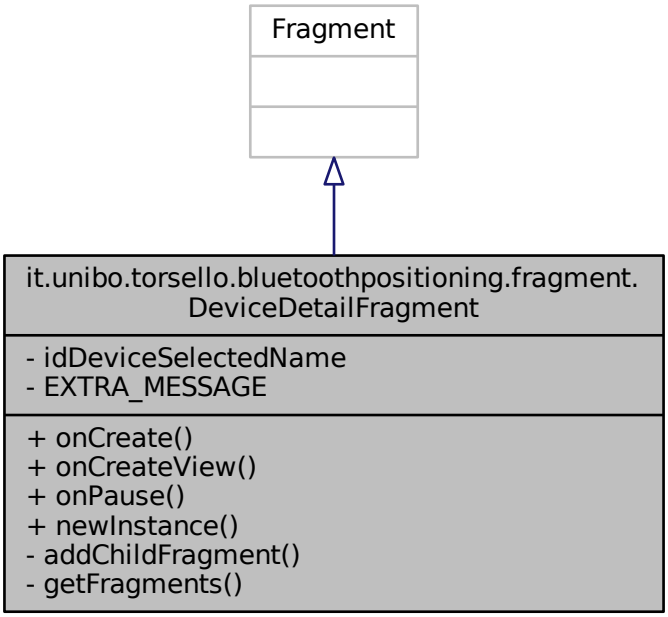
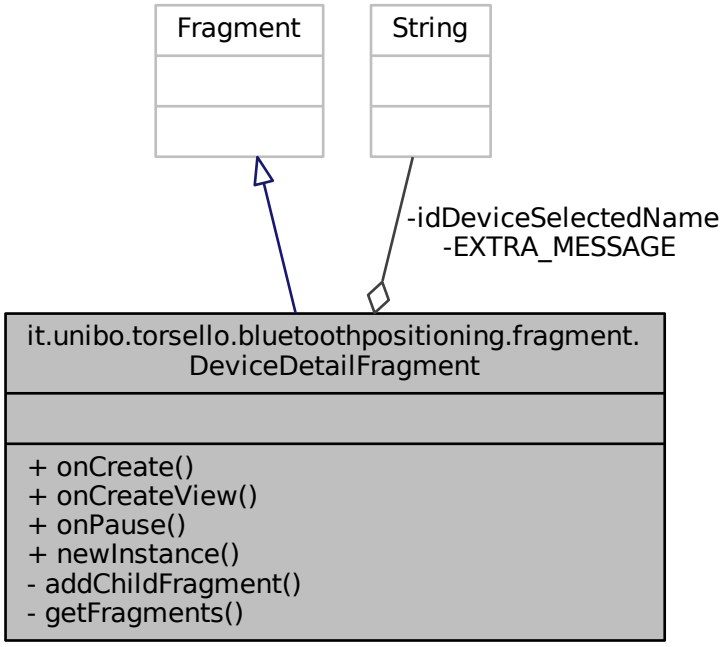


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailFragment:



Membri pubblici

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

Membri pubblici statici

- static DeviceDetailFragment newInstance (String message)

Membri privati

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

Attributi privati

- String idDeviceSelectedName

Attributi privati statici

- static final String EXTRA_MESSAGE = "EXTRA_MESSAGE"

6.9.1 Descrizione dettagliata

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

6.9.2 Documentazione delle funzioni membro

6.9.2.1 addChildFragment()

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

```
62                                          {
63
64          ViewPager mViewPager = (ViewPager) root.findViewById(R.id.view_pager);
65          StatePagerAdapter myPageAdapter = new StatePagerAdapter(getChildFragmentManager(),
    getFragments());
66          mViewPager.setAdapter(myPageAdapter);
67          mViewPager.setOffscreenPageLimit(getFragments().size());
68          mViewPager.setCurrentItem(1);
69
70          TabLayout tabLayout = (TabLayout) root.findViewById(R.id.sliding_tabs);
71          tabLayout.setupWithViewPager(mViewPager);
72      }
```

6.9.2.2 getFragments()

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

```
74                                              {
75
76          ArrayList<Fragment> fragments = new ArrayList<>();
77
78          // fragment 0
79          fragments.add(DeviceDetailInner0Fragment.newInstance(
    idDeviceSelectedName));
80
81          // fragment 1
82          fragments.add(DeviceDetailInner1Fragment.newInstance(
    idDeviceSelectedName));
83
84          // fragment 2
85          fragments.add(DeviceDetailInner2Fragment.newInstance(
    idDeviceSelectedName));
86
87
88          return fragments;
89      }
```

6.9.2.3 newInstance()

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

```
28                                                                  {
29          DeviceDetailFragment fragment = new DeviceDetailFragment();
30          Bundle args = new Bundle();
31          args.putString(EXTRA_MESSAGE, message);
32          fragment.setArguments(args);
33          return fragment;
34      }
```

6.9.2.4 onCreate()

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

```
37                                                  {
38         super.onCreate(savedInstanceState);
39
40         idDeviceSelectedName = getArguments().getString(
      EXTRA_MESSAGE);
41     }
```

6.9.2.5 onCreateView()

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

```
44                                                                                                  {
45         final View root = inflater.inflate(R.layout.fragment_device_detail, container, false);
46
47         getActivity().findViewById(R.id.toolbar).setVisibility(View.GONE);
48
49 //         ((AppBarLayout) root.findViewById(R.id.appbar_detail)).setExpanded(false);
50 //         ((CollapsingToolbarLayout)
      root.findViewById(R.id.collapsing_toolbar)).setTitle(idDeviceSelectedName);
51
52         addChildFragment(root);
53         return root;
54     }
```

6.9.2.6 onPause()

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

```
57                            {
58         getActivity().findViewById(R.id.toolbar).setVisibility(View.VISIBLE);
59         super.onPause();
60     }
```

6.9.3 Documentazione dei membri dato

6.9.3.1 EXTRA_MESSAGE

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

6.9.3.2 idDeviceSelectedName

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

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

- DeviceDetailFragment.java

6.10 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment

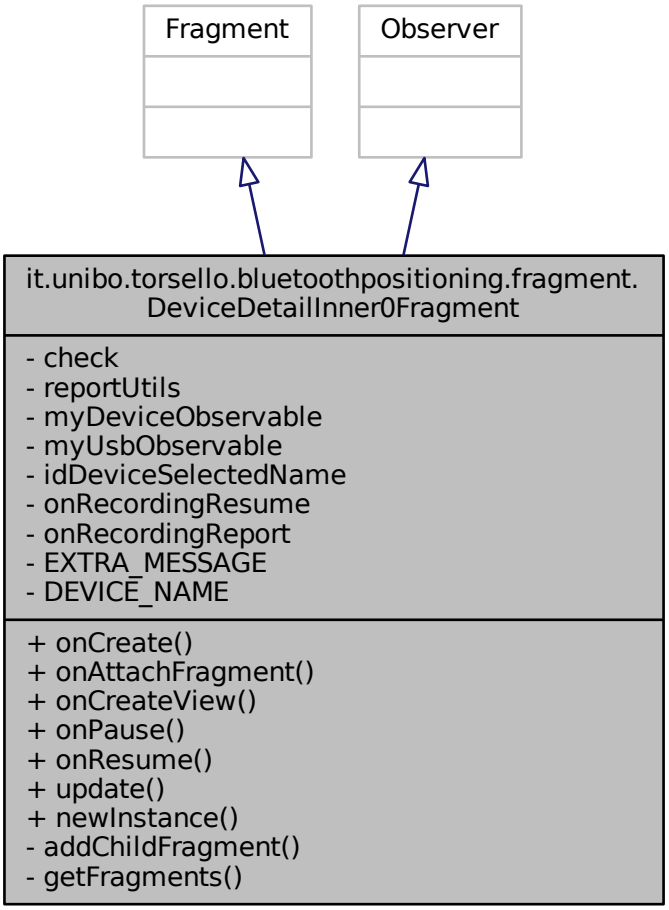
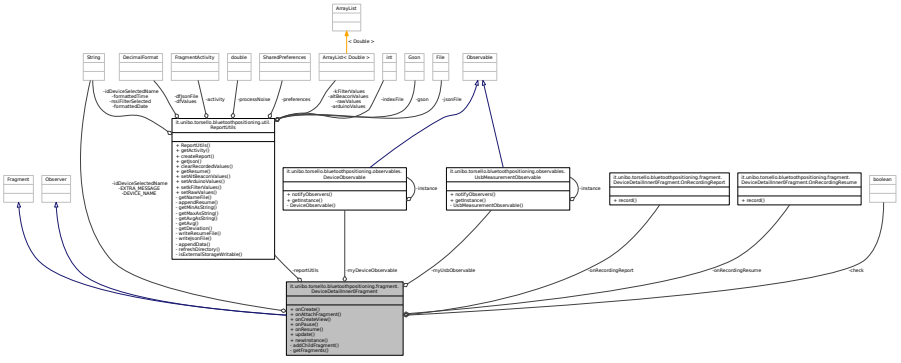


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment:



Composti

- interface OnRecordingReport
- interface OnRecordingResume

Membri pubblici

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

Membri pubblici statici

- static DeviceDetailInner0Fragment newInstance (String deviceName)

Membri privati

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

Attributi privati

- boolean check
- ReportUtils reportUtils
- DeviceObservable myDeviceObservable
- UsbMeasurementObservable myUsbObservable
- String idDeviceSelectedName
- OnRecordingResume onRecordingResume
- OnRecordingReport onRecordingReport

Attributi privati statici

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

6.10.1 Descrizione dettagliata

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

6.10.2 Documentazione delle funzioni membro

6.10.2.1 addChildFragment()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.addChild↩
Fragment (
            View root )  [private]
```

```
143                                             {
144
145         ViewPager mViewPager = (ViewPager) root.findViewById(R.id.view_pager_inner_0);
146
147         // avoid casual fragment's destruction
148         mViewPager.setOffscreenPageLimit(getFragments().size());
149
150         StatePagerAdapter myPageAdapter = new StatePagerAdapter(getChildFragmentManager(),
      getFragments());
151         mViewPager.setAdapter(myPageAdapter);
152
153         TabLayout tabLayout = (TabLayout) root.findViewById(R.id.sliding_tabs);
154         tabLayout.setupWithViewPager(mViewPager);
155     }
```

6.10.2.2 getFragments()

```
ArrayList<Fragment> it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.↩
getFragments ( )  [private]
```

```
157                                               {
158         ArrayList<Fragment> fragments = new ArrayList<>();
159
160         // nested fragment 0
161         fragments.add(DeviceDetailResumeFragment.newInstance());
162
163         // nested fragment 1
164         fragments.add(DeviceDetailReportFragment.newInstance());
165
166         return fragments;
167     }
```

6.10.2.3 newInstance()

```
static DeviceDetailInner0Fragment it.unibo.torsello.bluetoothpositioning.fragment.Device↩
DetailInner0Fragment.newInstance (
            String deviceName )  [static]
```

```
58                                                                                  {
59         DeviceDetailInner0Fragment fragment = new DeviceDetailInner0Fragment();
60         Bundle args = new Bundle();
61         args.putString(EXTRA_MESSAGE, "testing");
62         args.putString(DEVICE_NAME, deviceName);
63         fragment.setArguments(args);
64         return fragment;
65     }
```

6.10.2.4 onAttachFragment()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.onAttach↩
Fragment (
            Fragment childFragment )
```

```
80                                                      {
81         super.onAttachFragment(childFragment);
82         if (childFragment instanceof DeviceDetailReportFragment) {
83             onRecordingReport = (OnRecordingReport) childFragment;
84         }
85
86         if (childFragment instanceof DeviceDetailResumeFragment) {
87             onRecordingResume = (OnRecordingResume) childFragment;
88         }
89     }
```

6.10.2.5 onCreate()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.onCreate (
            Bundle savedInstanceState )
```

```
68                                                  {
69         super.onCreate(savedInstanceState);
70
71         idDeviceSelectedName = getArguments().getString(
     DEVICE_NAME);
72
73         myDeviceObservable = DeviceObservable.getInstance();
74         myUsbObservable = UsbMeasurementObservable.getInstance();
75
76         reportUtils = new ReportUtils(getActivity(),
     idDeviceSelectedName);
77      }
```

6.10.2.6 onCreateView()

```
View it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.onCreateView (
            LayoutInflater inflater,
            ViewGroup container,
            Bundle savedInstanceState )
```

```
92                                                                                                  {
93         final View root = inflater.inflate(R.layout.fragment_device_detail_inner_0, container, false);
94
95         final ToggleButton toggle = (ToggleButton) root.findViewById(R.id.toggleButton);
96         toggle.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
97             public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
98                 check = isChecked;
99                 if (isChecked) {
100
101                     reportUtils.clearRecordedValues();
102
103                     toggle.setClickable(false);
104
105                     new Handler().postDelayed(new Runnable() {
106                         @Override
107                         public void run() {
108                             toggle.setClickable(true);
109                             toggle.setChecked(false);
110
111                             onRecordingReport.record(
     reportUtils.getJson());
112                             onRecordingResume.record(
     reportUtils.getResume());
113                         }
114                     }, 5000);
115
116                     Snackbar.make(getActivity().findViewById(R.id.fab),
117                             "Start recording", Snackbar.LENGTH_SHORT).show();
118                 } else {
119                     reportUtils.createReport();
120                 }
121             }
122         });
123
124         addChildFragment(root);
125
126         return root;
127     }
```

6.10.2.7 onPause()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.onPause ( )
```

```
130                         {
131         myDeviceObservable.deleteObserver(this);
132         myUsbObservable.deleteObserver(this);
133         super.onPause();
134     }
```

6.10.2.8 onResume()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.onResume ( )
```

```
137                             {
138         super.onResume();
139         myDeviceObservable.addObserver(this);
140         myUsbObservable.addObserver(this);
141     }
```

6.10.2.9 update()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.update (
              Observable observable,
              Object arg )
```

```
170                                                         {
171         if (observable instanceof UsbMeasurementObservable) {
172             if (arg instanceof Double) {
173                 Double arduinoDistance = (Double) arg;
174                 if (check) {
175                     reportUtils.setArduinoValues(arduinoDistance);
176                 }
177             }
178         }
179 
180         if (observable instanceof DeviceObservable) {
181             if (arg instanceof List) {
182 
183                 List<Device> devices = (List<Device>) arg;
184 
185                 for (Device deviceSelected : devices) {
186                     if (deviceSelected.getFriendlyName().equals(
    idDeviceSelectedName) ||
187                             deviceSelected.getAddress().equals(
    idDeviceSelectedName)) {
188 
189                         if (check) {
190                             reportUtils.setAltBeaconValues(deviceSelected.
    getAltBeaconDistance());
191                             reportUtils.setkFilterValues(deviceSelected.
    getKalmanFilterDistance());
192                             reportUtils.setRawValues(deviceSelected.getRawDistance()
    );
193                         }
194                     }
195                 }
196             }
197         }
198 
199     }
```

6.10.3 Documentazione dei membri dato

6.10.3.1 check

```
boolean it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.check  [private]
```

6.10.3.2 DEVICE_NAME

```
final String it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.DEVI↩
CE_NAME = "DEVICE_NAME"  [static], [private]
```

6.10.3.3 EXTRA_MESSAGE

```
final String it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.EXTR↩
A_MESSAGE = "EXTRA_MESSAGE"  [static], [private]
```

6.10.3.4 idDeviceSelectedName

```
String it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.idDevice↩
SelectedName  [private]
```

6.10.3.5 myDeviceObservable

```
DeviceObservable it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.↩
myDeviceObservable  [private]
```

6.10.3.6 myUsbObservable

```
UsbMeasurementObservable it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0↩
Fragment.myUsbObservable  [private]
```

6.10.3.7 onRecordingReport

```
OnRecordingReport it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.↩
onRecordingReport  [private]
```

6.10.3.8 onRecordingResume

```
OnRecordingResume it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.↩
onRecordingResume  [private]
```

6.10.3.9 reportUtils

```
ReportUtils it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.↩
reportUtils  [private]
```

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

- DeviceDetailInner0Fragment.java

6.11 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner1Fragment

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

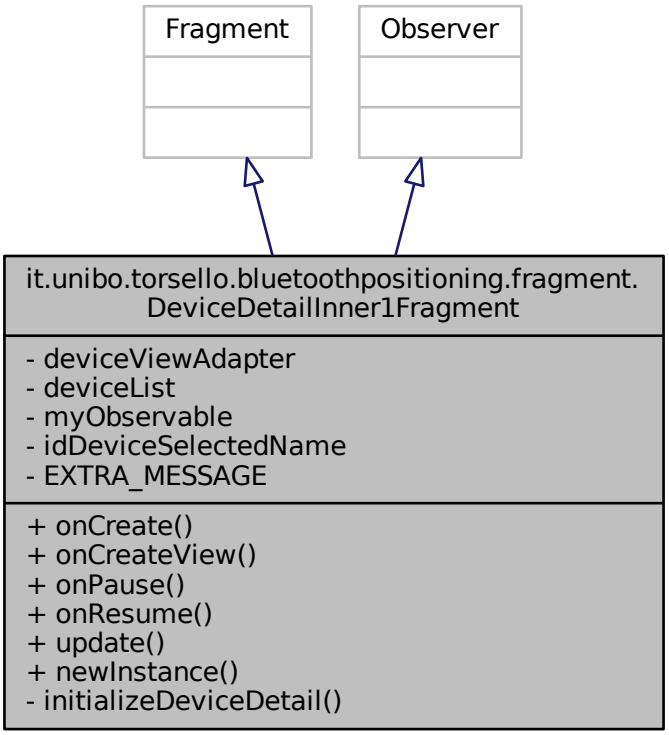
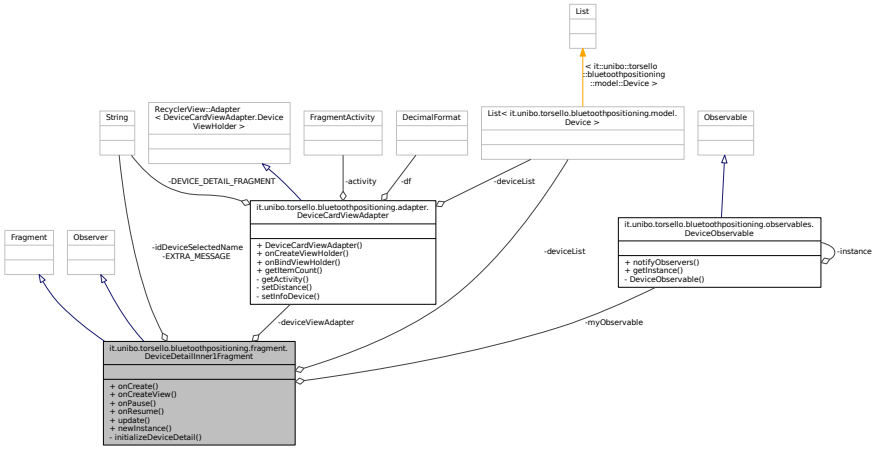


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner1Fragment:



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)

Membri privati

- void initializeDeviceDetail (View root)

Attributi privati

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

Attributi privati statici

- static final String EXTRA_MESSAGE = "EXTRA_MESSAGE"

6.11.1 Descrizione dettagliata

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

6.11.2 Documentazione delle funzioni membro

6.11.2.1 initializeDeviceDetail()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner1Fragment.initialize↩
DeviceDetail (
            View root )  [private]
```

```
75                                                  {
76         // add RecyclerView
77         RecyclerView recyclerView = (RecyclerView) root.findViewById(R.id.recycler_view_detail);
78         recyclerView.setLayoutManager(new LinearLayoutManager(getContext()));
79         recyclerView.setAdapter(deviceViewAdapter);
80     }
```

6.11.2.2 newInstance()

```
static DeviceDetailInner1Fragment it.unibo.torsello.bluetoothpositioning.fragment.Device↩
DetailInner1Fragment.newInstance (
            String message )  [static]
```

```
35                                                                          {
36         DeviceDetailInner1Fragment fragment = new DeviceDetailInner1Fragment();
37         Bundle args = new Bundle();
38         args.putString(EXTRA_MESSAGE, message);
39         fragment.setArguments(args);
40         return fragment;
41     }
```

6.11.2.3 onCreate()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner1Fragment.onCreate (
            @Nullable Bundle savedInstanceState )
```

```
44                                                          {
45         super.onCreate(savedInstanceState);
46
47         myObservable = DeviceObservable.getInstance();
48
49         idDeviceSelectedName = getArguments().getString(
      EXTRA_MESSAGE);
50         deviceList = new ArrayList<>();
51         deviceViewAdapter = new DeviceCardViewAdapter(getActivity(),
      deviceList);
52      }
```

6.11.2.4 onCreateView()

```
View it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner1Fragment.onCreateView (
            LayoutInflater inflater,
            ViewGroup container,
            Bundle savedInstanceState )
```

```
55                                                                                  {
56         View root = inflater.inflate(R.layout.fragment_device_detail_inner_1, container, false);
57
58         initializeDeviceDetail(root);
59
60         return root;
61     }
```

6.11.2.5 onPause()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner1Fragment.onPause ( )
```

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

6.11.2.6 onResume()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner1Fragment.onResume ( )
```

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

6.11.2.7 update()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner1Fragment.update (
              Observable o,
              Object arg )
```

```
83                                                {
84
85         if (arg instanceof List) {
86
87             if (!deviceList.isEmpty()) {
88                 deviceList.clear();
89             }
90
91             List<Device> devices = (List<Device>) arg;
92
93             for (Device deviceSelected : devices) {
94                 if (deviceSelected.getFriendlyName().equals(idDeviceSelectedName) ||
95                         deviceSelected.getAddress().equals(idDeviceSelectedName)) {
96                     deviceList.add(deviceSelected);
97                 }
98             }
99
100             deviceViewAdapter.notifyDataSetChanged();
101         }
102     }
```

6.11.3 Documentazione dei membri dato

6.11.3.1 deviceList

```
List<Device> it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner1Fragment.↩
deviceList  [private]
```

6.11.3.2 deviceViewAdapter

```
DeviceCardViewAdapter it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner1↩
Fragment.deviceViewAdapter  [private]
```

6.11.3.3 EXTRA_MESSAGE

```
final String it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner1Fragment.EXTR↩
A_MESSAGE = "EXTRA_MESSAGE"  [static], [private]
```

6.11.3.4 idDeviceSelectedName

```
String it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner1Fragment.idDevice↩
SelectedName  [private]
```

6.11.3.5 myObservable

```
DeviceObservable it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner1Fragment.↩
myObservable [private]
```

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

- DeviceDetailInner1Fragment.java

6.12 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragment

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragment

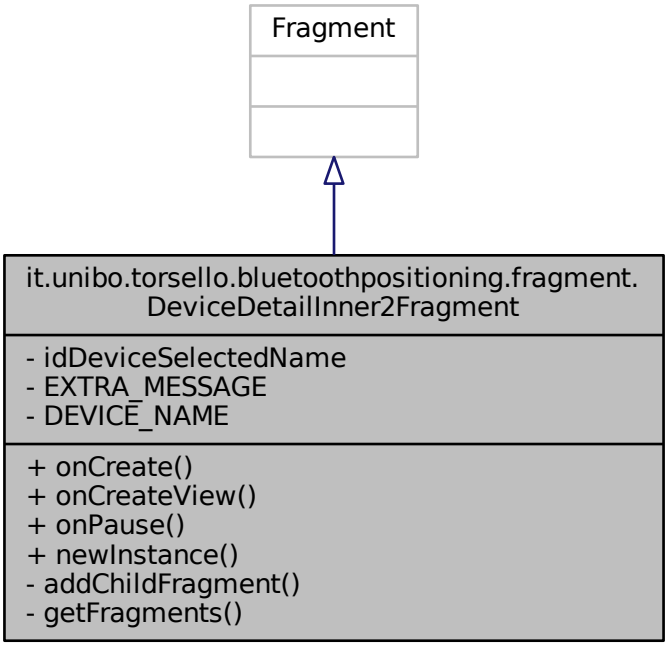
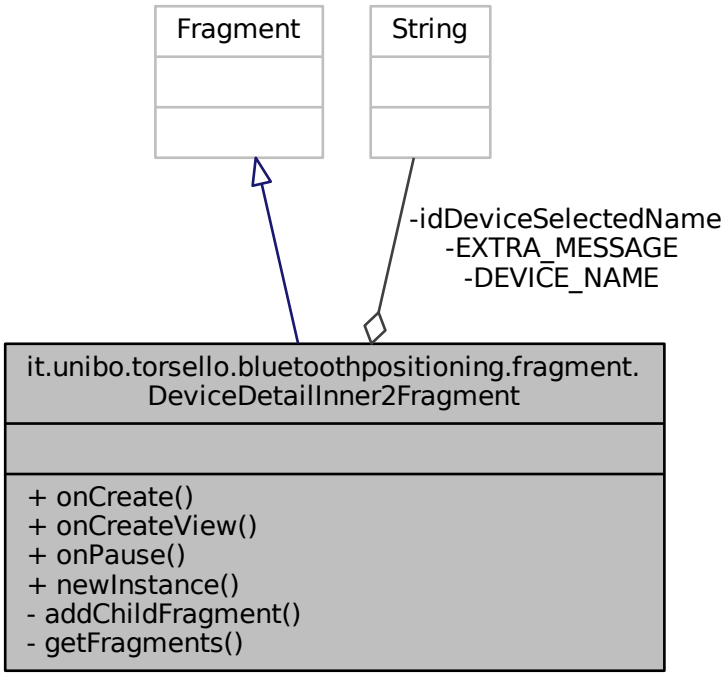


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragment:



Membri pubblici

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

Membri pubblici statici

- static DeviceDetailInner2Fragment newInstance (String deviceName)

Membri privati

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

Attributi privati

- String idDeviceSelectedName

Attributi privati statici

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

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 addChildFragment()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragment.addChild↩
Fragment (
            View root )  [private]
```

```
57                                          {
58
59          ViewPager mViewPager = (ViewPager) root.findViewById(R.id.view_pager_inner_2);
60
61          // avoid casual fragment's destruction
62          mViewPager.setOffscreenPageLimit(getFragments().size());
63
64          StatePagerAdapter myPageAdapter = new StatePagerAdapter(getChildFragmentManager(),
      getFragments());
65          mViewPager.setAdapter(myPageAdapter);
66
67          TabLayout tabLayout = (TabLayout) root.findViewById(R.id.sliding_tabs);
68          tabLayout.setupWithViewPager(mViewPager);
69      }
```

6.12.2.2 getFragments()

```
ArrayList<Fragment> it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragment.↩
getFragments ( )  [private]
```

```
71                                              {
72          ArrayList<Fragment> fragments = new ArrayList<>();
73
74          // inner fragment 0
75          ArrayList<String> params1 = new ArrayList<>();
76          params1.add(getString(R.string.chart_arduino));
77          params1.add(getString(R.string.chart_raw_distance));
78          params1.add(getString(R.string.chart_altbeacon));
79          params1.add(getString(R.string.chart_kalman_filter));
80
81          fragments.add(DeviceChartFragment.newInstance("chart1", 1,
      idDeviceSelectedName, params1));
82
83          // inner fragment 1
84          ArrayList<String> params2 = new ArrayList<>();
85          params2.add(getString(R.string.chart_arduino));
86          params2.add(getString(R.string.chart_raw_distance));
87          params2.add(getString(R.string.chart_kalman_filter));
88
89          fragments.add(DeviceChartFragment.newInstance("chart2", 2,
      idDeviceSelectedName, params2));
90
91          // inner fragment 2
92          ArrayList<String> params3 = new ArrayList<>();
93          params3.add(getString(R.string.chart_arduino));
94          params3.add(getString(R.string.chart_altbeacon));
95          params3.add(getString(R.string.chart_kalman_filter));
96
97          fragments.add(DeviceChartFragment.newInstance("chart3", 3,
      idDeviceSelectedName, params3));
98
99          return fragments;
100     }
```

6.12.2.3 newInstance()

```
static DeviceDetailInner2Fragment it.unibo.torsello.bluetoothpositioning.fragment.Device↩
DetailInner2Fragment.newInstance (
            String deviceName )  [static]
```

```
27                                                                          {
28         DeviceDetailInner2Fragment fragment = new DeviceDetailInner2Fragment();
29         Bundle args = new Bundle();
30         args.putString(EXTRA_MESSAGE, "real time charts");
31         args.putString(DEVICE_NAME, deviceName);
32         fragment.setArguments(args);
33         return fragment;
34     }
```

6.12.2.4 onCreate()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragment.onCreate (
            Bundle savedInstanceState )
```

```
37                                                {
38         super.onCreate(savedInstanceState);
39
40         idDeviceSelectedName = getArguments().getString(
      DEVICE_NAME);
41     }
```

6.12.2.5 onCreateView()

```
View it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragment.onCreateView (
            LayoutInflater inflater,
            ViewGroup container,
            Bundle savedInstanceState )
```

```
44                                                                                {
45         View root = inflater.inflate(R.layout.fragment_device_detail_inner_2, container, false);
46
47         addChildFragment(root);
48
49         return root;
50     }
```

6.12.2.6 onPause()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragment.onPause ( )
```

```
53                           {
54         super.onPause();
55     }
```

6.12.3 Documentazione dei membri dato

6.12.3.1 DEVICE_NAME

```
final String it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragment.DEVI↩
CE_NAME = "DEVICE_NAME"  [static], [private]
```

6.12.3.2 EXTRA_MESSAGE

```
final String it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragment.EXTR↩
A_MESSAGE = "EXTRA_MESSAGE"  [static], [private]
```

6.12.3.3 idDeviceSelectedName

```
String it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragment.idDevice↩
SelectedName  [private]
```

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

- DeviceDetailInner2Fragment.java

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

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

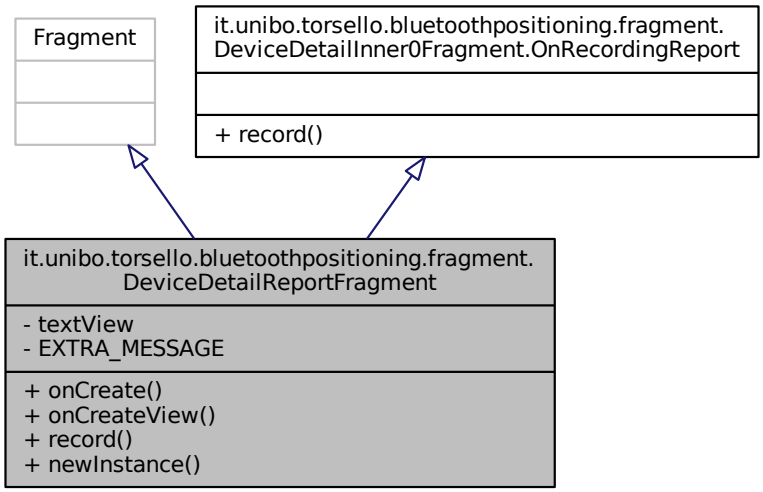
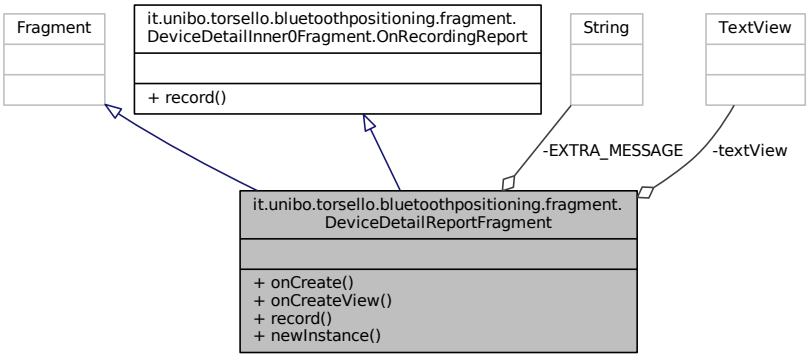


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailReportFragment:



Membri pubblici

- void onCreate (@Nullable Bundle savedInstanceState)
- View onCreateView (LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
- void record (String newRecord)

Membri pubblici statici

- static DeviceDetailReportFragment newInstance ()

Attributi privati

- TextView textView

Attributi privati statici

- static final String EXTRA_MESSAGE = "EXTRA_MESSAGE"

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 newInstance()

```
static DeviceDetailReportFragment it.unibo.torsello.bluetoothpositioning.fragment.Device↩
DetailReportFragment.newInstance ( )  [static]
```

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

6.13.2.2 onCreate()

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

```
36                                                              {
37         super.onCreate(savedInstanceState);
38
39     }
```

6.13.2.3 onCreateView()

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

```
43                                                  {
44         // Inflate the layout for this fragment
45         final View root = inflater.inflate(R.layout.fragment_device_detail_report, container, false);
46
47         textView = (TextView) root.findViewById(R.id.textView_report);
48
49         return root;
50     }
```

6.13.2.4 record()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailReportFragment.record (
            String newRecord )
```

Implementa it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.OnRecordingReport.

```
53                                          {
54         textView.setText(newRecord);
55     }
```

6.13.3 Documentazione dei membri dato

6.13.3.1 EXTRA_MESSAGE

```
final String it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailReportFragment.EXTR↩
A_MESSAGE = "EXTRA_MESSAGE"  [static], [private]
```

6.13.3.2 textView

```
TextView it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailReportFragment.textView
[private]
```

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

- DeviceDetailReportFragment.java

6.14 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailResume↩ Fragment

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

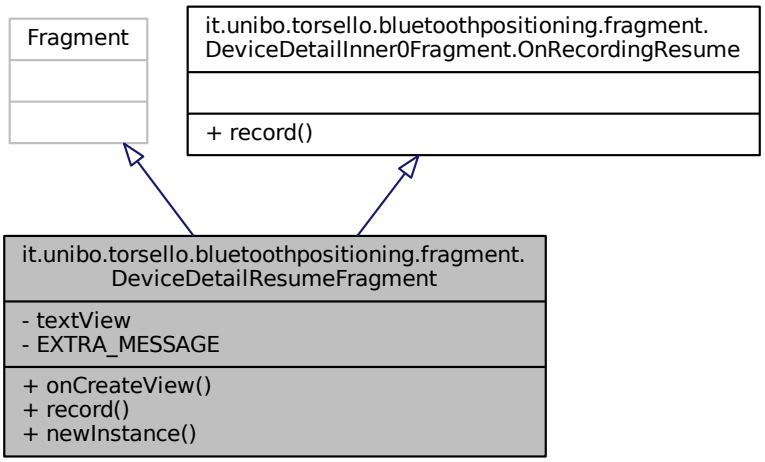
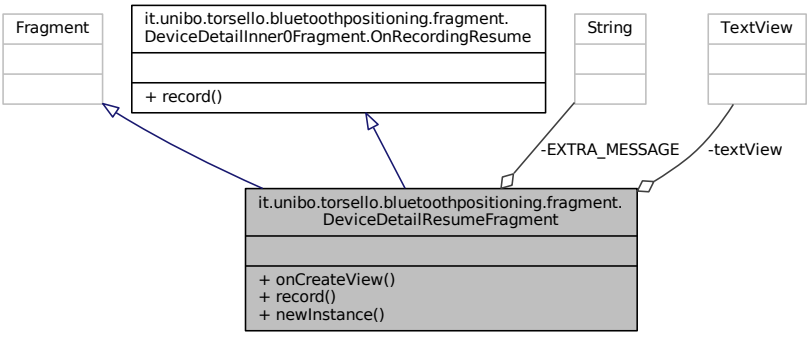


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailResumeFragment:



Membri pubblici

- View onCreateView (LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
- void record (String newRecord)

Membri pubblici statici

- static DeviceDetailResumeFragment newInstance ()

Attributi privati

- TextView textView

Attributi privati statici

- static final String EXTRA_MESSAGE = "EXTRA_MESSAGE"

6.14.1 Descrizione dettagliata

Created by federico on 12/10/16.

6.14.2 Documentazione delle funzioni membro

6.14.2.1 newInstance()

```
static DeviceDetailResumeFragment it.unibo.torsello.bluetoothpositioning.fragment.Device↩
DetailResumeFragment.newInstance ( )  [static]
```

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

6.14.2.2 onCreateView()

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

```
37                                                  {
38         // Inflate the layout for this fragment
39         final View root = inflater.inflate(R.layout.fragment_device_detail_resume, container, false);
40
41         textView = (TextView) root.findViewById(R.id.textView_resume);
42
43         return root;
44     }
```

6.14.2.3 record()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailResumeFragment.record (
              String newRecord )
```

Implementa it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.OnRecordingResume.

```
47                                          {
48         textView.setText(newRecord);
49     }
```

6.14.3 Documentazione dei membri dato

6.14.3.1 EXTRA_MESSAGE

```
final String it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailResumeFragment.EXTR↩
A_MESSAGE = "EXTRA_MESSAGE"  [static], [private]
```

6.14.3.2 textView

```
TextView it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailResumeFragment.textView
[private]
```

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

- DeviceDetailResumeFragment.java

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

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

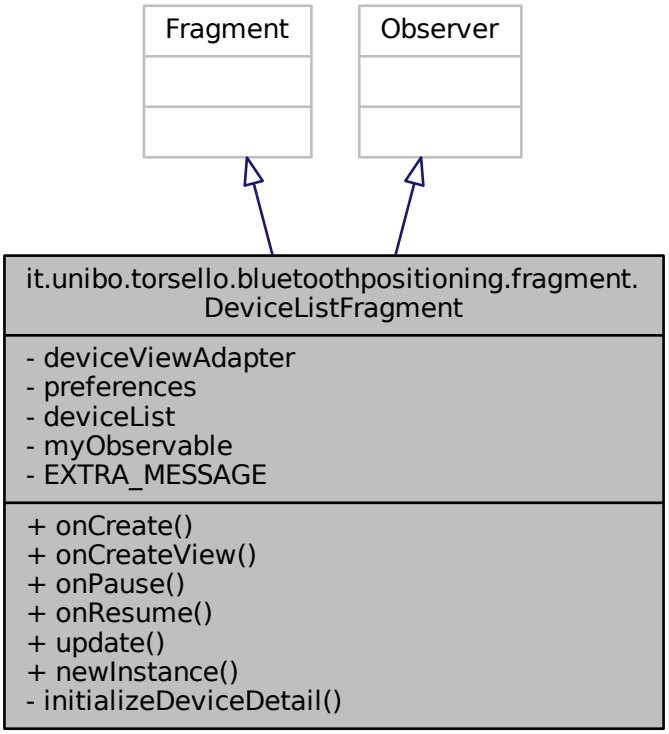
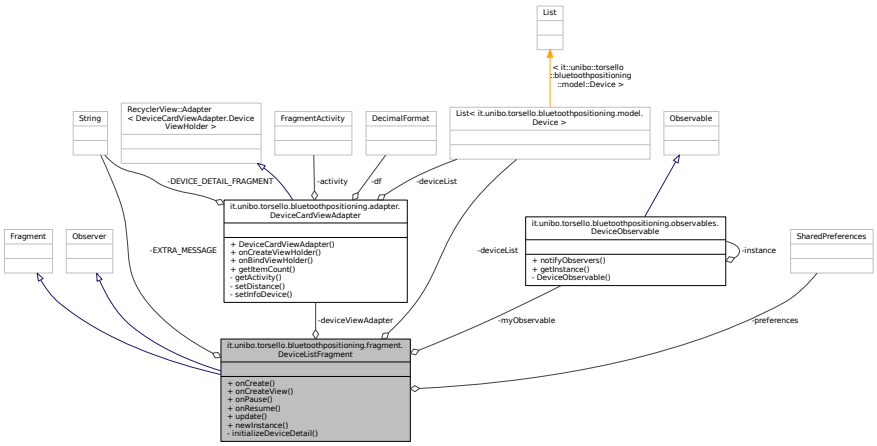


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.DeviceListFragment:



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 ()

Membri privati

- void initializeDeviceDetail (View root)

Attributi privati

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

Attributi privati statici

- static final String EXTRA_MESSAGE = "EXTRA_MESSAGE"

6.15.1 Descrizione dettagliata

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

6.15.2 Documentazione delle funzioni membro

6.15.2.1 initializeDeviceDetail()

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

```
83                                                  {
84         // add RecyclerView
85         RecyclerView recyclerView = (RecyclerView) root.findViewById(R.id.recycler_view);
86         recyclerView.setLayoutManager(new LinearLayoutManager(getContext()));
87         deviceViewAdapter = new DeviceCardViewAdapter(getActivity(),
     deviceList);
88         recyclerView.setAdapter(deviceViewAdapter);
89     }
```

6.15.2.2 newInstance()

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

```
55                                                  {
56         DeviceListFragment fragment = new DeviceListFragment();
57         Bundle args = new Bundle();
58         args.putString(EXTRA_MESSAGE, "Scan Device");
59         fragment.setArguments(args);
60         return fragment;
61     }
```

6.15.2.3 onCreate()

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

```
64                                                          {
65         super.onCreate(savedInstanceState);
66
67         myObservable = DeviceObservable.getInstance();
68
69         deviceList = new ArrayList<>();
70         preferences = getActivity().getSharedPreferences(SettingConstants.SETTINGS_PREFERENCES,
     0);
71     }
```

6.15.2.4 onCreateView()

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

```
74                                                                                          {
75
76         View root = inflater.inflate(R.layout.fragment_device_list, container, false);
77
78         initializeDeviceDetail(root);
79
80         return root;
81     }
```

6.15.2.5 onPause()

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

```
92                          {
93         myObservable.deleteObserver(this);
94         super.onPause();
95     }
```

6.15.2.6 onResume()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceListFragment.onResume ( )
```

```
98                           {
99         super.onResume();
100         myObservable.addObserver(this);
101     }
```

6.15.2.7 update()

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

```
104                                                 {
105
106         if (arg instanceof List) {
107
108             if (!deviceList.isEmpty()) {
109                 deviceList.clear();
110             }
111
112             List<Device> devices = (List<Device>) arg;
113
114             // optional sorting
115             Collections.sort(devices, new Comparator<Device>() {
116                 public int compare(Device b1, Device b2) {
117                     int sorting = preferences.getInt(SettingConstants.DISTANCE_SORTING, 0);
118                     switch (sorting) {
119                         case 0:
120                         case R.id.radioButton_default_sorting:
121                             return Double.compare(b1.getIndex(), b2.getIndex());
122                         case R.id.radioButton_color_sorting:
123                             return Double.compare(b1.getColor(), b2.getColor());
124                         case R.id.radioButton_distance_sorting:
125                             return Double.compare(b1.getKalmanFilterDistance(), b2.getKalmanFilterDistance(
      ));
126                     } // default sorting (a good basic ordering for the other options)
127                     return Double.compare(b1.getIndex(), b2.getIndex());
128                 }
129             });
130
131             deviceList.addAll(devices);
132             deviceViewAdapter.notifyDataSetChanged();
133         }
134     }
```

6.15.3 Documentazione dei membri dato

6.15.3.1 deviceList

```
List<Device> it.unibo.torsello.bluetoothpositioning.fragment.DeviceListFragment.deviceList
[private]
```

6.15.3.2 deviceViewAdapter

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

6.15.3.3 EXTRA_MESSAGE

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

6.15.3.4 myObservable

```
DeviceObservable it.unibo.torsello.bluetoothpositioning.fragment.DeviceListFragment.myObservable
[private]
```

6.15.3.5 preferences

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

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

- DeviceListFragment.java

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

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

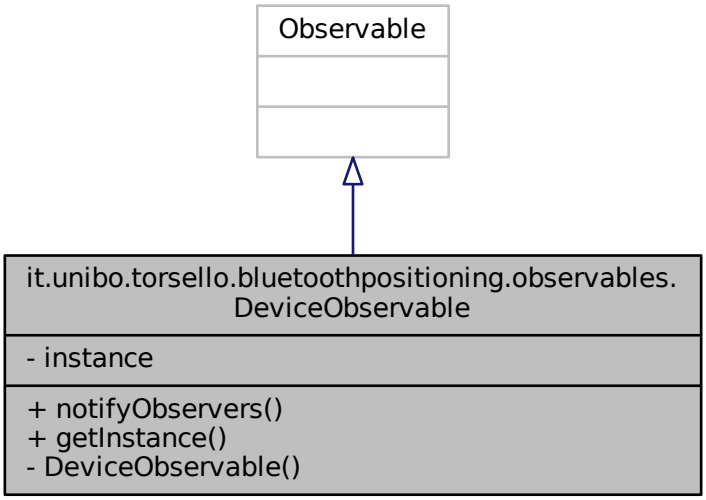
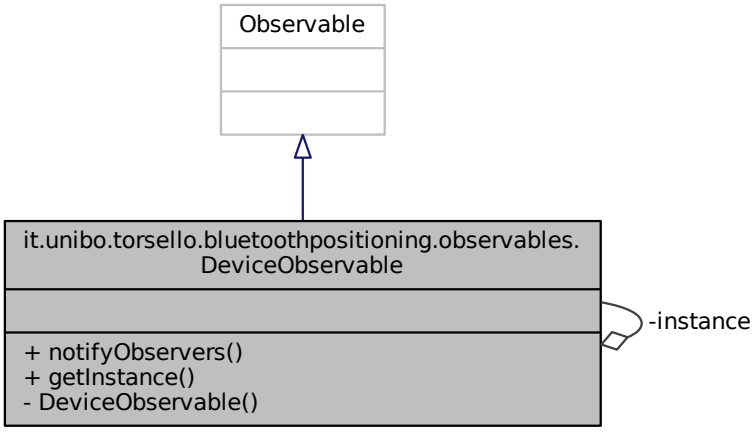


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.observables.DeviceObservable:



Membri pubblici

- void notifyObservers (Object data)

Membri pubblici statici

- static DeviceObservable getInstance ()

Membri privati

- DeviceObservable ()

Attributi privati statici

- static DeviceObservable instance = new DeviceObservable()

6.16.1 Descrizione dettagliata

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

6.16.2 Documentazione dei costruttori e dei distruttori

6.16.2.1 DeviceObservable()

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

```
13                                  {
14          super();
15      }
```

6.16.3 Documentazione delle funzioni membro

6.16.3.1 getInstance()

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

```
17                                              {
18         return instance;
19     }
```

6.16.3.2 notifyObservers()

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

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

6.16.4 Documentazione dei membri dato

6.16.4.1 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

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

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

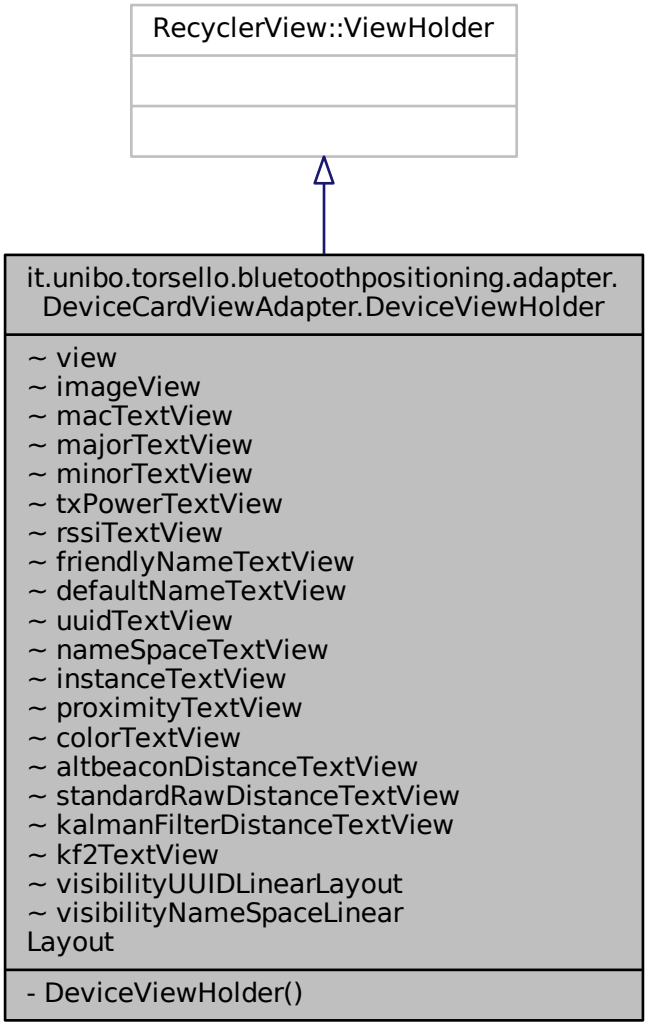
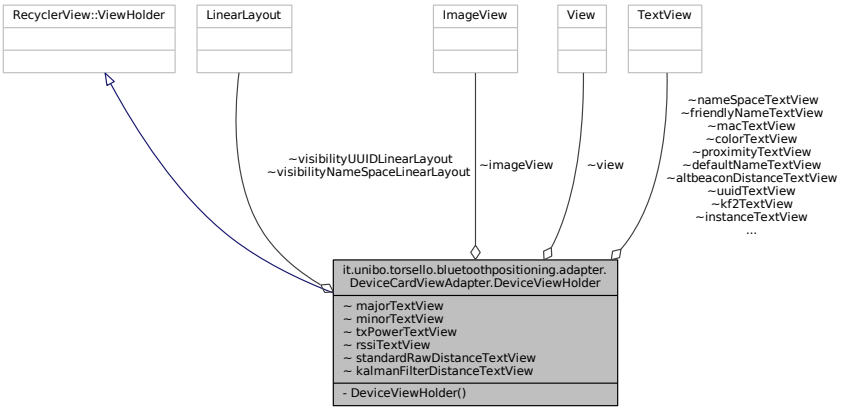


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.Device↩

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
- TextView kf2TextView
- LinearLayout visibilityUUIDLinearLayout
- LinearLayout visibilityNameSpaceLinearLayout

Membri privati

- DeviceViewHolder (View view)

6.17.1 Documentazione dei costruttori e dei distruttori

6.17.1.1 DeviceViewHolder()

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

```
255                                          {
256
257             super(view);
258             this.view = view;
259             imageView = (ImageView) view.findViewById(R.id.imageBeacon);
260             defaultNameTextView = (TextView) view.findViewById(R.id.
      value_default_name);
261             friendlyNameTextView = (TextView) view.findViewById(R.id.
      value_friendly_name);
262             macTextView = (TextView) view.findViewById(R.id.value_mac_address);
263             majorTextView = (TextView) view.findViewById(R.id.value_major);
264             minorTextView = (TextView) view.findViewById(R.id.value_minor);
265             txPowerTextView = (TextView) view.findViewById(R.id.value_power);
266             rssiTextView = (TextView) view.findViewById(R.id.value_rssi);
267             uuidTextView = (TextView) view.findViewById(R.id.value_uuid);
268             nameSpaceTextView = (TextView) view.findViewById(R.id.value_name_space);
269             proximityTextView = (TextView) view.findViewById(R.id.value_proximity);
270             instanceTextView = (TextView) view.findViewById(R.id.value_instance);
271             colorTextView = (TextView) view.findViewById(R.id.value_color);
272
273             altbeaconDistanceTextView = (TextView) view.findViewById(R.id.
      value_altbeacon_distance);
274             kalmanFilterDistanceTextView = (TextView)
      view.findViewById(R.id.value_kalman_filter_distance);
275             standardRawDistanceTextView = (TextView)
      view.findViewById(R.id.value_standard_raw_distance);
276
277             kf2TextView = (TextView) view.findViewById(R.id.value_kalman_filter_2);
278
279             visibilityUUIDLinearLayout = (LinearLayout)
      view.findViewById(R.id.visibility_uuid_minor_major_nmb);
280             visibilityNameSpaceLinearLayout = (LinearLayout)
      view.findViewById(R.id.visibilityNameSpace_Instance);
281         }
```

6.17.2 Documentazione dei membri dato

6.17.2.1 altbeaconDistanceTextView

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

6.17.2.2 colorTextView

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

6.17.2.3 defaultNameTextView

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

6.17.2.4 friendlyNameTextView

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

6.17.2.5 imageView

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

6.17.2.6 instanceTextView

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

6.17.2.7 kalmanFilterDistanceTextView

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

6.17.2.8 kf2TextView

```
TextView it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceView↩
Holder.kf2TextView  [package]
```

6.17.2.9 macTextView

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

6.17.2.10 majorTextView

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

6.17.2.11 minorTextView

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

6.17.2.12 nameSpaceTextView

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

6.17.2.13 proximityTextView

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

6.17.2.14 rssiTextView

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

6.17.2.15 standardRawDistanceTextView

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

6.17.2.16 txPowerTextView

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

6.17.2.17 uuidTextView

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

6.17.2.18 view

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

6.17.2.19 visibilityNameSpaceLinearLayout

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

6.17.2.20 visibilityUUIDLinearLayout

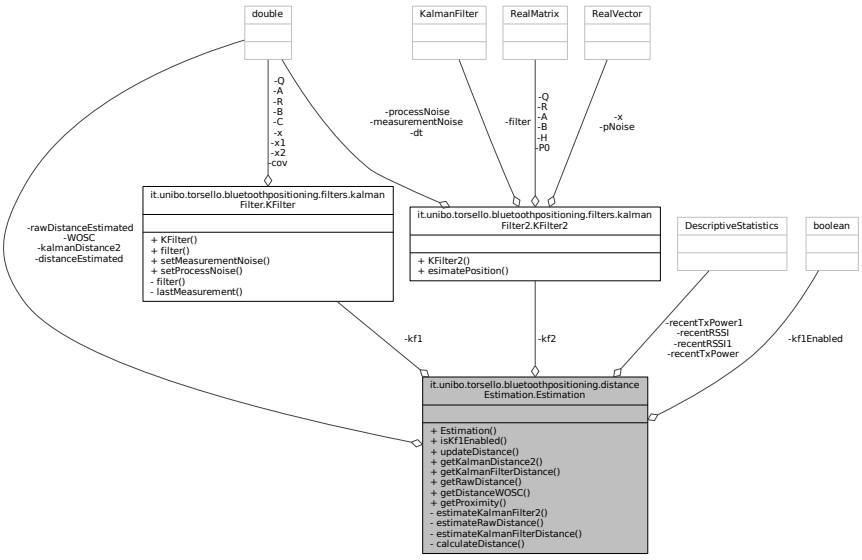
```
LinearLayout it.unibo.torsello.bluetoothpositioning.adapter.DeviceCardViewAdapter.DeviceView↩
Holder.visibilityUUIDLinearLayout  [package]
```

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

- DeviceCardViewAdapter.java

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

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



Membri pubblici

- Estimation ()
- boolean isKf1Enabled ()
- void updateDistance (Beacon b, double processNoise)
- double getKalmanDistance2 ()
- double getKalmanFilterDistance ()
- double getRawDistance ()
- double getDistanceWOSC ()
- String getProximity ()

Membri privati

- void estimateKalmanFilter2 (Beacon b, double myProcess)
- void estimateRawDistance (Beacon b)
- void estimateKalmanFilterDistance (Beacon b, double processNoise)
- double calculateDistance (double txPower, double rssi)

Attributi privati

- DescriptiveStatistics recentRSSI
- DescriptiveStatistics recentTxPower
- KFilter kf1
- double distanceEstimated
- double rawDistanceEstimated
- double WOSC
- boolean kf1Enabled
- KFilter2 kf2
- double kalmanDistance2
- DescriptiveStatistics recentRSSI1
- DescriptiveStatistics recentTxPower1

6.18.1 Descrizione dettagliata

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

6.18.2 Documentazione dei costruttori e dei distruttori

6.18.2.1 Estimation()

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

```
38                          {
39
40          rawDistanceEstimated = 0;
41
42          // limit on the number of values that can be stored in the dataset
43          recentRSSI = new DescriptiveStatistics();
44          recentRSSI.setWindowSize(KFilterConstants.WINDOW);
45          recentTxPower = new DescriptiveStatistics();
46          recentTxPower.setWindowSize(KFilterConstants.WINDOW);
47
48          distanceEstimated = 0;
49          WOSC = 0;
50
51          kf1 = new KFilterBuilder()
52                  // filter for RSSI
53                  .R(KFilterConstants.INITIAL_PROCESS_NOISE) // Initial process noise
54                  .Q(KFilterConstants.INITIAL_MEASUREMENT_NOISE) // Initial measurement noise
55                  .build();
56
57          recentRSSI1 = new DescriptiveStatistics();
58          recentRSSI1.setWindowSize(KFilterConstants.WINDOW);
59          recentTxPower1 = new DescriptiveStatistics();
60          recentTxPower1.setWindowSize(KFilterConstants.WINDOW);
61
62          kf2 = new KFilter2();
63
64      }
```

6.18.3 Documentazione delle funzioni membro

6.18.3.1 calculateDistance()

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

```
126                                                                 {
127
128         if (rssi == 0.0D) {
129             return -1.0D; // if we cannot determine accuracy, return -1.
130         }
131
132         double ratio = (rssi * 1.0D) / txPower;
133         if (ratio < 1.0D) {
134             return Math.pow(ratio, 10.0D);
135         }
136
137 //        return (0.89976D * Math.pow(ratio, 7.7095D)) + 0.125D;
138         return (0.89976D * Math.pow(ratio, 7.7095D)) + 0.111D;
139     }
```

6.18.3.2 estimateKalmanFilter2()

```
void it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.estimateKalman↩
Filter2 (
              Beacon b,
              double myProcess )  [private]
```

```
80                                                                  {
81
82          double rssiFiltered = kf2.esimatePosition(b.getRssi(), myProcess);
83 //         rssiFiltered = Math.round(rssiFiltered * 10.0) / 10.0;
84          kalmanDistance2 = calculateDistance(b.getTxPower(), rssiFiltered);
85      }
```

6.18.3.3 estimateKalmanFilterDistance()

```
void it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.estimateKalman↩
FilterDistance (
            Beacon b,
            double processNoise )  [private]
```

```
96                                                                          {
97
98          if (!(processNoise > 0)) {
99              kf1Enabled = false;
100              distanceEstimated = 0;
101          } else {
102              kf1Enabled = true;
103
104              recentRSSI.addValue(b.getRssi());
105              recentTxPower.addValue(b.getTxPower());
106
107              // Update measurement noise continually
108              double mNoise = Math.sqrt((100 * 9 / Math.log(10)) *
109                      Math.log(1 + Math.pow(recentRSSI.getMean() /
      recentRSSI.getStandardDeviation(), 2)));
110
111              if (!Double.isInfinite(mNoise) && !Double.isNaN(mNoise)) {
112                  Log.i("cuai1", "asd " + mNoise);
113                  kf1.setMeasurementNoise(mNoise);
114              }
115
116              // update measurement, z parameter
117              double lastFilteredReading = kf1.filter(recentRSSI.getPercentile(50));
118
119              WOSC = calculateDistance(b.getTxPower(), lastFilteredReading);
120
121              distanceEstimated = calculateDistance(
      recentTxPower.getPercentile(50), lastFilteredReading);
122          }
123      }
```

6.18.3.4 estimateRawDistance()

```
void it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.estimateRawDistance
(
            Beacon b )  [private]
```

```
91                                              {
92          // raw distance
93          rawDistanceEstimated = calculateDistance(b.getTxPower(), b.
      getRssi());
94      }
```

6.18.3.5 getDistanceWOSC()

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

```
149                                     {
150         return WOSC;
151     }
```

6.18.3.6 getKalmanDistance2()

```
double it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.getKalmanDistance2
( )
```

```
87                                      {
88          return kalmanDistance2;
89      }
```

6.18.3.7 getKalmanFilterDistance()

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

```
141                                             {
142         return distanceEstimated;
143     }
```

6.18.3.8 getProximity()

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

```
153                                     {
154         double proximity = distanceEstimated;
155         String accuracy;
156 
157         if (proximity <= 0) {
158             accuracy = "unknown";
159         } else if (proximity <= 0.5) {
160             accuracy = "immediate";
161         } else if (proximity <= 4.0) {
162             accuracy = "near";
163         } else {
164             accuracy = "far";
165         }
166         return accuracy;
167     }
```

6.18.3.9 getRawDistance()

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

```
145                                       {
146         return rawDistanceEstimated;
147     }
```

6.18.3.10 isKf1Enabled()

```
boolean it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.isKf1Enabled ( )
```

```
66                                      {
67         return kf1Enabled;
68     }
```

6.18.3.11 updateDistance()

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

```
70                                                              {
71 
72         estimateRawDistance(b);
73 
74         estimateKalmanFilterDistance(b, processNoise);
75 
76         estimateKalmanFilter2(b, processNoise);
77 
78     }
```

6.18.4 Documentazione dei membri dato

6.18.4.1 distanceEstimated

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

6.18.4.2 kalmanDistance2

```
double it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.kalmanDistance2
[private]
```

6.18.4.3 kf1

```
KFilter it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.kf1  [private]
```

6.18.4.4 kf1Enabled

```
boolean it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.kf1Enabled  [private]
```

6.18.4.5 kf2

```
KFilter2 it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.kf2  [private]
```

6.18.4.6 rawDistanceEstimated

```
double it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.rawDistance↩
Estimated  [private]
```

6.18.4.7 recentRSSI

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

6.18.4.8 recentRSSI1

```
DescriptiveStatistics it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.↩
recentRSSI1  [private]
```

6.18.4.9 recentTxPower

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

6.18.4.10 recentTxPower1

```
DescriptiveStatistics it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation.↩
recentTxPower1  [private]
```

6.18.4.11 WOSC

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

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

- Estimation.java

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

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

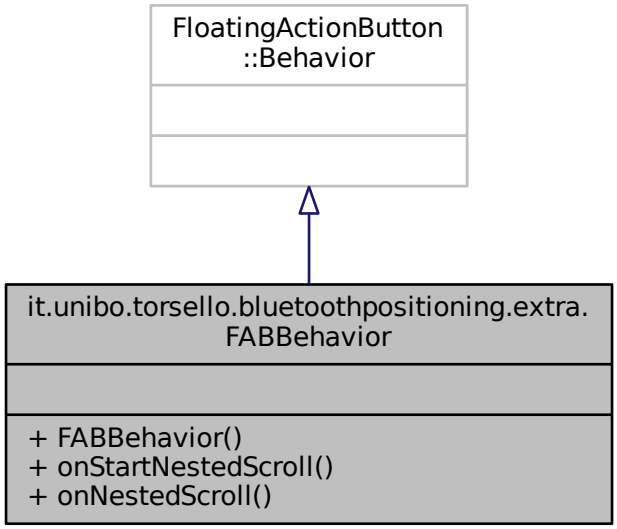
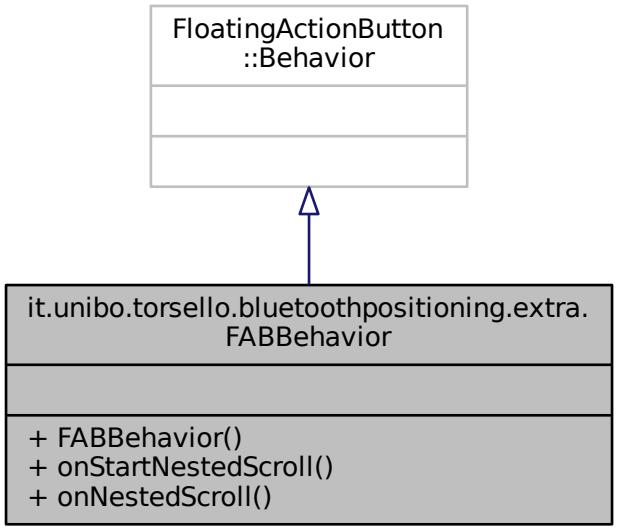


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.extra.FABBehavior:



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)

6.19.1 Descrizione dettagliata

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

6.19.2 Documentazione dei costruttori e dei distruttori

6.19.2.1 FABBehavior()

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

```
18                                                      {
19         super();
20     }
```

6.19.3 Documentazione delle funzioni membro

6.19.3.1 onNestedScroll()

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

```
34
          {
35        super.onNestedScroll(coordinatorLayout, child, target, dxConsumed, dyConsumed, dxUnconsumed,
36                dyUnconsumed);
37
38        if ((dyConsumed > 0 || dyUnconsumed == 0) && child.getVisibility() == View.VISIBLE) {
39            child.hide();
40            new Handler().postDelayed(new Runnable() {
41                @Override
42                public void run() {
43                    child.show();
44                }
45            }, 1000);
46        }
47    }
```

6.19.3.2 onStartNestedScroll()

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

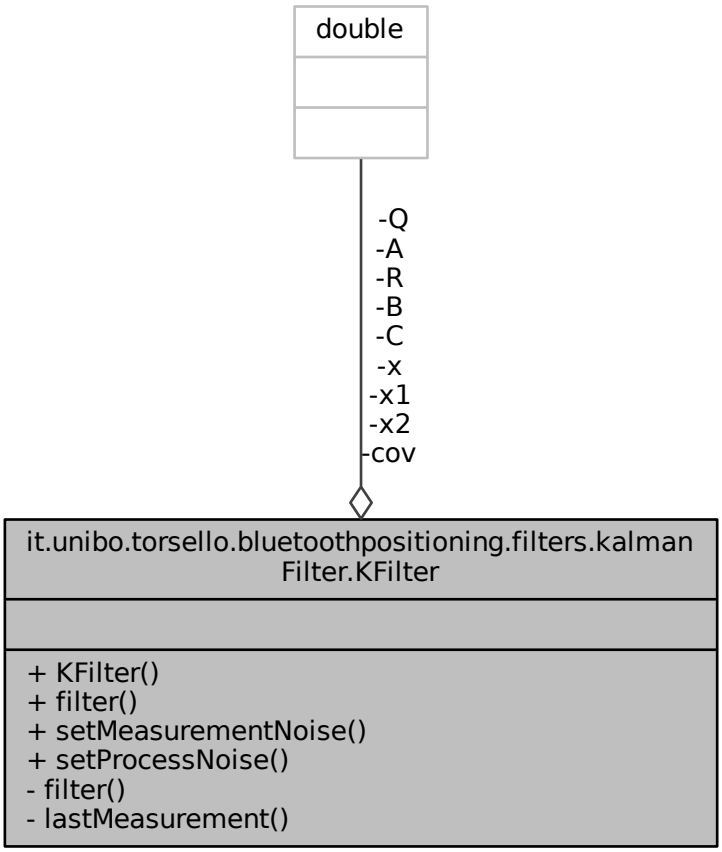
```
24                                                                                                  {
25
26 //        return nestedScrollAxes == ViewCompat.SCROLL_AXIS_VERTICAL ||
27 //                super.onStartNestedScroll(coordinatorLayout, child, directTargetChild, target,
28 //                        nestedScrollAxes);
29        return true;
30    }
```

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

- FABBehavior.java

6.20 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter.KFilter

Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter.KFilter:



Membri pubblici

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

Membri privati

- double filter (double z, double u)
- double lastMeasurement ()

Attributi privati

- double R
- double Q
- double A
- double B
- double C
- double cov
- double x
- double x1
- double x2

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 KFilter()

```
it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter.KFilter.KFilter (
            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

```
28                                                                  {
29
30          this.R = R;
31          this.Q = Q;
32          this.A = A;
33          this.B = B;
34          this.C = C;
35
36          cov = Double.NaN;
37          x = Double.NaN;
38      }
```

6.20.3 Documentazione delle funzioni membro

6.20.3.1 filter() [1/2]

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

```
41                                  {
42          return filter(z, 0);
43      }
```

6.20.3.2 filter() [2/2]

```
double it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter.KFilter.filter (
              double z,
              double u )  [private]
```

Filter a new value

Parametri

z	Measurement
u	Control

Restituisce

x

```
52                                              {
53
54          if (Double.isNaN(x)) {
55              x = (1 / C) * z;
56              x1 = x;
57              x2 = x1;
58              cov = (1 / C) * Q * (1 / C);
59          } else {
60
61              // Calculate previous update step
62              B = (x - x1) / 2;
63
64              // Compute prediction
65              double predX = (A * x) + (B * u);
66              double predCov = ((A * cov) * A) + R;
67
68              // KFilter2 gain
69              double K = predCov * C * (1 / ((C * predCov * C) + Q));
70
71              // Correction
72              x1 = x;
73              x = predX + K * (z - (C * predX));
74              cov = predCov - (K * C * predCov);
75          }
76
77          return x;
78      }
```

6.20.3.3 lastMeasurement()

```
double it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter.KFilter.lastMeasurement ( )
[private]
```

Return the last filtered measurement

Restituisce

x Estimated signal without noise

```
85                                      {
86          return x;
87      }
```

6.20.3.4 setMeasurementNoise()

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

Set measurement noise INITIAL_MEASUREMENT_NOISE

Parametri

noise	Measurement noise

```
94                                                  {
95          Q = noise;
96      }
```

6.20.3.5 setProcessNoise()

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

Set the process noise INITIAL_PROCESS_NOISE

Parametri

noise	Process noise

```
103                                                 {
104         R = noise;
105     }
```

6.20.4 Documentazione dei membri dato

6.20.4.1 A

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

6.20.4.2 B

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

6.20.4.3 C

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

6.20.4.4 cov

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

6.20.4.5 Q

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

6.20.4.6 R

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

6.20.4.7 x

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

6.20.4.8 x1

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

6.20.4.9 x2

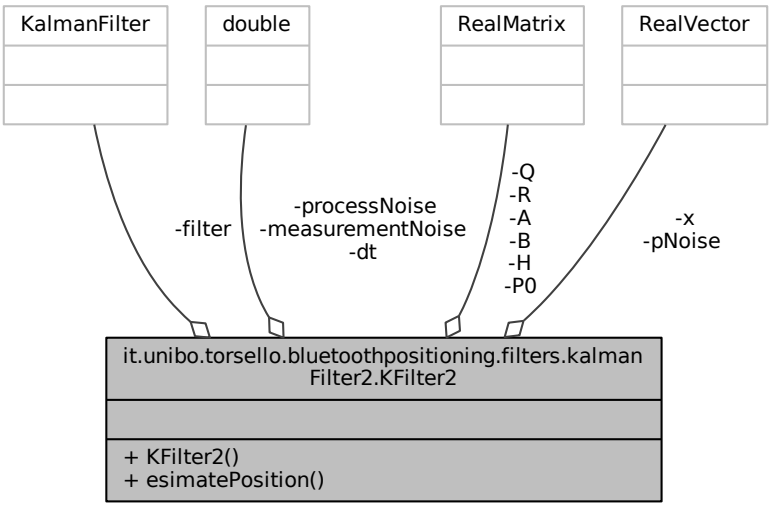
```
double it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter.KFilter.x2  [private]
```

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

- KFilter.java

6.21 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter2.KFilter2

Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter2.KFilter2:



Membri pubblici

- KFilter2 ()
- double esimatePosition (double pos, double myProcess)

Attributi privati

- RealMatrix A
- RealMatrix B
- RealMatrix H
- RealMatrix Q
- RealMatrix R
- RealVector x
- RealMatrix P0
- final double dt = 0.1d
- final double measurementNoise = 0.15d
- RealVector pNoise
- double processNoise = 0.00001d
- KalmanFilter filter

6.21.1 Descrizione dettagliata

Created by federico on 13/10/16.

6.21.2 Documentazione dei costruttori e dei distruttori

6.21.2.1 KFilter2()

`it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter2.KFilter2.KFilter2 ( )`

```
48                      {
49
50          A = new Array2DRowRealMatrix(new double[]{1d});
51
52          B = null;
53
54          H = new Array2DRowRealMatrix(new double[]{1d});
55
56          Q = new Array2DRowRealMatrix(new double[]{processNoise});
57
58          P0 = new Array2DRowRealMatrix(new double[][]{{65d}});
59
60          R = new Array2DRowRealMatrix(new double[]{
61                  Math.pow(measurementNoise, 2d)
62          });
63
64          ProcessModel pm = new DefaultProcessModel(A, B, Q, x, P0);
65          MeasurementModel mm = new DefaultMeasurementModel(H, R);
66          filter = new KalmanFilter(pm, mm);
67
68          pNoise = new ArrayRealVector(1);
69      }
```

6.21.3 Documentazione delle funzioni membro

6.21.3.1 esimatePosition()

```
double it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter2.KFilter2.esimatePosition (
            double pos,
            double myProcess )
```

```
71                                                                  {
72
73 //        pNoise.setEntry(0, myProcess);
74
75         // x = [ 0 0 0 0] state consists of position and velocity[pX, pY, vX, vY]
76         x = new ArrayRealVector(new double[]{pos});
77
78         // predict the state estimate one time-step ahead
79         filter.predict();
80
81         // x = A * x + B * u (state prediction)
82 //        x = A.operate(x).add(pNoise);
83         x = A.operate(x);
84
85         // z = H * x  (measurement prediction)
86         RealVector z = H.operate(x);
87
88         // correct the state estimate with the latest measurement
89         filter.correct(z);
90
91         //get the corrected state - the position
92
93         return filter.getStateEstimation()[0];
94     }
```

6.21.4 Documentazione dei membri dato

6.21.4.1 A

RealMatrix it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter2.KFilter2.A [private]

6.21.4.2 B

RealMatrix it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter2.KFilter2.B [private]

6.21.4.3 dt

final double it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter2.KFilter2.dt = 0.1d [private]

6.21.4.4 filter

KalmanFilter it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter2.KFilter2.filter [private]

6.21.4.5 H

RealMatrix it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter2.KFilter2.H [private]

6.21.4.6 measurementNoise

final double it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter2.KFilter2.measurement↩Noise = 0.15d [private]

6.21.4.7 P0

RealMatrix it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter2.KFilter2.P0 [private]

6.21.4.8 pNoise

```
RealVector it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter2.KFilter2.pNoise [private]
```

6.21.4.9 processNoise

```
double it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter2.KFilter2.processNoise =
0.00001d [private]
```

6.21.4.10 Q

```
RealMatrix it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter2.KFilter2.Q [private]
```

6.21.4.11 R

```
RealMatrix it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter2.KFilter2.R [private]
```

6.21.4.12 x

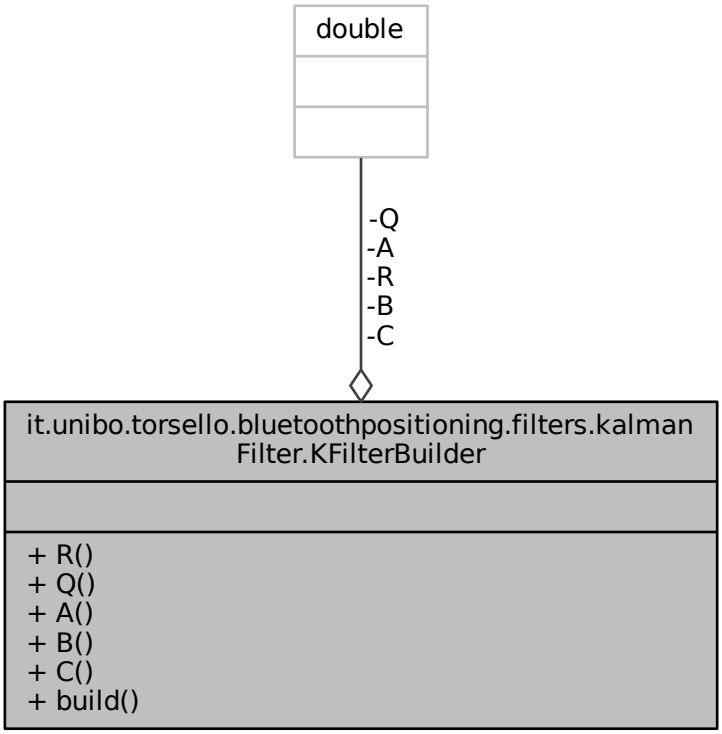
```
RealVector it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter2.KFilter2.x [private]
```

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

- KFilter2.java

6.22 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter.KFilterBuilder

Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter.KFilterBuilder:



Membri pubblici

- KFilterBuilder R (double R)
- KFilterBuilder Q (double Q)
- KFilterBuilder A (double A)
- KFilterBuilder B (double B)
- KFilterBuilder C (double C)
- KFilter build ()

Attributi privati

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

6.22.1 Descrizione dettagliata

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

Simple builder class for 1-dimensional KFilter2 filter with predefined

6.22.2 Documentazione delle funzioni membro

6.22.2.1 A()

```
KFilterBuilder it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter.KFilterBuilder.A (
              double A )
```

```
27                                      {
28         this.A = A;
29         return this;
30     }
```

6.22.2.2 B()

```
KFilterBuilder it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter.KFilterBuilder.B (
              double B )
```

```
32                                      {
33         this.B = B;
34         return this;
35     }
```

6.22.2.3 build()

```
KFilter it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter.KFilterBuilder.build ( )
```

```
42                              {
43         return new KFilter(R, Q, A, B, C);
44     }
```

6.22.2.4 C()

```
KFilterBuilder it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter.KFilterBuilder.C (
            double C )
```

```
37                                      {
38         this.C = C;
39         return this;
40     }
```

6.22.2.5 Q()

```
KFilterBuilder it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter.KFilterBuilder.Q (
            double Q )
```

```
22                                      {
23         this.Q = Q;
24         return this;
25     }
```

6.22.2.6 R()

```
KFilterBuilder it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter.KFilterBuilder.R (
            double R )
```

```
17                                      {
18         this.R = R;
19         return this;
20     }
```

6.22.3 Documentazione dei membri dato

6.22.3.1 A

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

6.22.3.2 B

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

6.22.3.3 C

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

6.22.3.4 Q

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

6.22.3.5 R

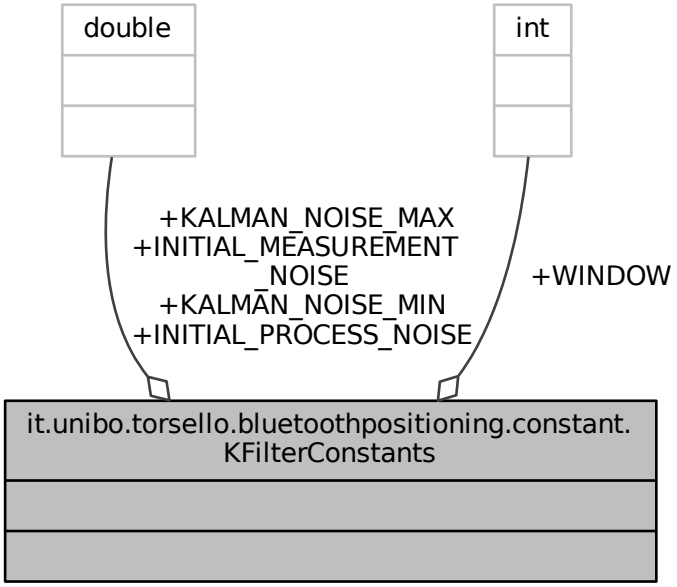
```
double it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter.KFilterBuilder.R = 1  [private]
```

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

- KFilterBuilder.java

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

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



Attributi pubblici statici

- static final double KALMAN_NOISE_MIN = 0D
- static final double KALMAN_NOISE_MAX = 5D
- static final double INITIAL_PROCESS_NOISE = 5D
- static final double INITIAL_MEASUREMENT_NOISE = 5D
- static final int WINDOW = 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 INITIAL_MEASUREMENT_NOISE

```
final double it.unibo.torsello.bluetoothpositioning.constant.KFilterConstants.INITIAL_MEASUR↩
EMENT_NOISE = 5D  [static]
```

6.23.2.2 INITIAL_PROCESS_NOISE

```
final double it.unibo.torsello.bluetoothpositioning.constant.KFilterConstants.INITIAL_PROCES↩
S_NOISE = 5D  [static]
```

6.23.2.3 KALMAN_NOISE_MAX

```
final double it.unibo.torsello.bluetoothpositioning.constant.KFilterConstants.KALMAN_NOISE_MAX
= 5D  [static]
```

6.23.2.4 KALMAN_NOISE_MIN

```
final double it.unibo.torsello.bluetoothpositioning.constant.KFilterConstants.KALMAN_NOISE_MIN
= 0D  [static]
```

6.23.2.5 WINDOW

```
final int it.unibo.torsello.bluetoothpositioning.constant.KFilterConstants.WINDOW = 10  [static]
```

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

- KFilterConstants.java

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

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

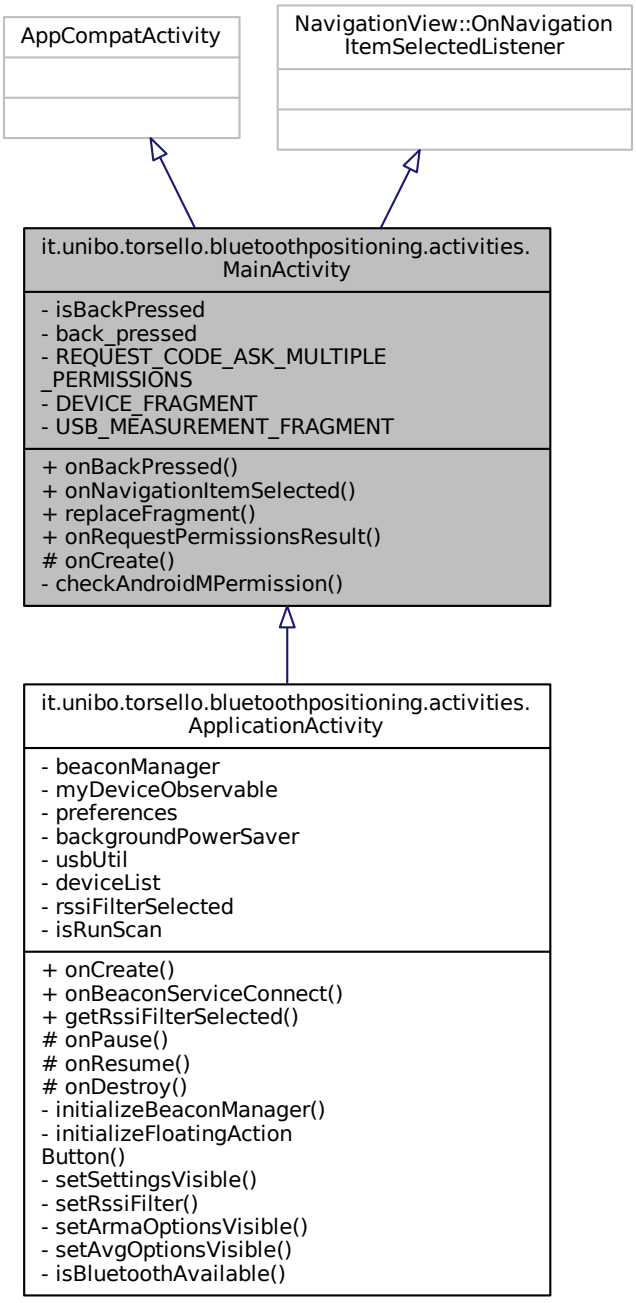
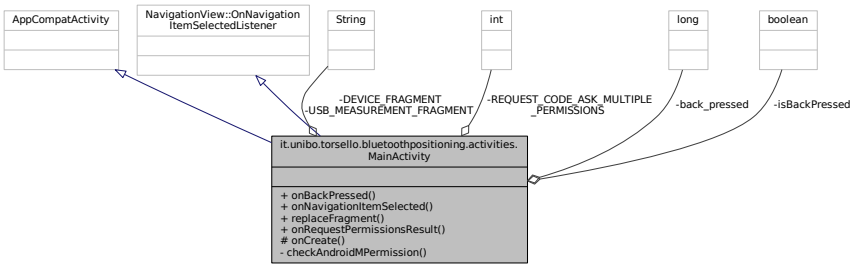


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.activities.MainActivity:



Membri pubblici

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

Membri protetti

- void onCreate (Bundle savedInstanceState)

Membri privati

- void checkAndroidMPermission ()

Attributi privati

- boolean isBackPressed = false
- long back_pressed
- final int REQUEST_CODE_ASK_MULTIPLE_PERMISSIONS = 124

Attributi privati statici

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

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]

```
167                                         {
168
169         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
170             final List<String> permissions = new ArrayList<>();
171
172             if (checkSelfPermission(Manifest.permission.ACCESS_FINE_LOCATION)
173                     != PackageManager.PERMISSION_GRANTED) {
174                 permissions.add(Manifest.permission.ACCESS_FINE_LOCATION);
175             }
176
177             if (checkSelfPermission(Manifest.permission.ACCESS_COARSE_LOCATION)
178                     != PackageManager.PERMISSION_GRANTED) {
179                 permissions.add(Manifest.permission.ACCESS_COARSE_LOCATION);
180             }
181
182             if (!permissions.isEmpty()) {
183                 new AlertDialog.Builder(this)
184                         .setTitle(R.string.dialog_location_access_title)
185                         .setMessage(R.string.dialog_bluetooth_text)
186                         .setPositiveButton(android.R.string.ok, null)
187                         .setOnDismissListener(new DialogInterface.OnDismissListener() {
188                             @TargetApi(23)
189                             @Override
190                             public void onDismiss(DialogInterface dialog) {
191                                 requestPermissions(permissions.toArray(new String[permissions.size()]),
192                                         REQUEST_CODE_ASK_MULTIPLE_PERMISSIONS
    );
193                             }
194
195                         }).show();
196             }
197         }
198     }
```

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]
```

```
45                                                  {
46         super.onCreate(savedInstanceState);
47         setContentView(R.layout.activity_main);
48
49         Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
50         setSupportActionBar(toolbar);
51
52         DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
53         ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
54                 this, drawer, toolbar, R.string.navigation_drawer_open, R.string.navigation_drawer_close);
55         drawer.addDrawerListener(toggle);
56         toggle.syncState();
57
58         ((NavigationView) findViewById(R.id.nav_view)).setNavigationItemSelectedListener(this);
59
60         ((NavigationView) findViewById(R.id.nav_view2)).setNavigationItemSelectedListener(this);
61
62         replaceFragment(DEVICE_FRAGMENT);
63
64         checkAndroidMPermission();
65
66
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         }
112
113         if (drawer.isDrawerOpen(GravityCompat.START)) {
114             drawer.closeDrawer(GravityCompat.START);
115         }
116
117         return true;
118     }
```

6.24.2.5 onRequestPermissionsResult()

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

```
140                                                                         {
141         switch (requestCode) {
142             case REQUEST_CODE_ASK_MULTIPLE_PERMISSIONS:
143                 for (int i = 0; i < permissions.length; i++) {
144                     if (grantResults[i] == PackageManager.PERMISSION_GRANTED) {
145 //                      Log.d(TAG_CLASS, "Permission Granted: " + permissions[i]);
146                     } else if (grantResults[i] == PackageManager.PERMISSION_DENIED) {
147 //                      Log.d(TAG_CLASS, "Permission Denied: " + permissions[i]);
148                         new AlertDialog.Builder(this)
149                                 .setTitle(R.string.dialog_permissions_location_access_title)
150                                 .setMessage(R.string.dialog_permissions_location_access_text)
151                                 .setPositiveButton(android.R.string.ok, null)
152                                 .setOnDismissListener(new DialogInterface.OnDismissListener() {
153
154                                     @Override
155                                     public void onDismiss(DialogInterface dialog) {
156                                     }
157
158                                 }).show();
159                     }
160                 }
161                 break;
162             default:
163                 super.onRequestPermissionsResult(requestCode, permissions, grantResults);
164         }
165     }
```

6.24.2.6 replaceFragment()

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

```
120                                             {
121         Fragment currentFragment = getSupportFragmentManager().findFragmentByTag(fragTag);
122         switch (fragTag) {
123             case DEVICE_FRAGMENT:
124                 currentFragment = DeviceListFragment.newInstance();
125                 break;
126             case USB_MEASUREMENT_FRAGMENT:
127                 currentFragment = UsbMeasurementFragment.newInstance();
128                 break;
129         }
130
131         if (currentFragment != null) {
132             getSupportFragmentManager().beginTransaction()
133                     .replace(R.id.contentMainLayout, currentFragment, fragTag)
134                     .commit();
135         }
136     }
```

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], [private]
```

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 USB_MEASUREMENT_FRAGMENT

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

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.filters.MyArmaRssiFilter

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

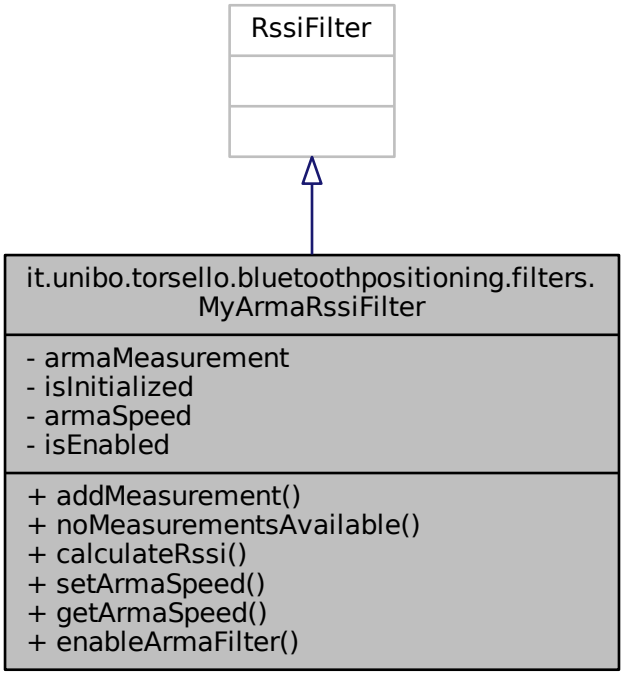
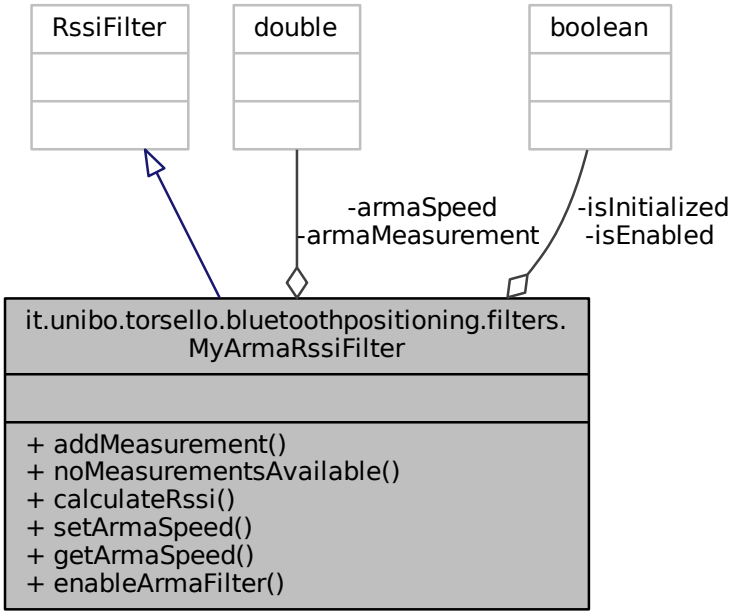


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.filters.MyArmaRssiFilter:



Membri pubblici

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

Membri pubblici statici

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

Attributi privati

- double armaMeasurement
- boolean isInitialized = false

Attributi privati statici

- static double armaSpeed = 0.1D
- static boolean isEnabled = true

6.25.1 Descrizione dettagliata

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

6.25.2 Documentazione delle funzioni membro

6.25.2.1 addMeasurement()

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

```
30                                          {
31
32         if (isEnabled) {
33             if (!isInitialized) {
34                 armaMeasurement = rssi;
35                 isInitialized = true;
36             }
37
38             armaMeasurement = (armaMeasurement -
      armaSpeed * (armaMeasurement - rssi));
39         } else {
40             armaMeasurement = rssi;
41         }
42
43     }
```

6.25.2.2 calculateRssi()

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

```
51                                  {
52         return armaMeasurement;
53     }
```

6.25.2.3 enableArmaFilter()

```
static void it.unibo.torsello.bluetoothpositioning.filters.MyArmaRssiFilter.enableArmaFilter (
            boolean set )  [static]
```

```
25                                                  {
26         isEnabled = set;
27     }
```

6.25.2.4 getArmaSpeed()

```
static double it.unibo.torsello.bluetoothpositioning.filters.MyArmaRssiFilter.getArmaSpeed ( )
[static]
```

```
21                                      {
22         return armaSpeed;
23     }
```

6.25.2.5 noMeasurementsAvailable()

```
boolean it.unibo.torsello.bluetoothpositioning.filters.MyArmaRssiFilter.noMeasurementsAvailable
( )
```

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

6.25.2.6 setArmaSpeed()

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

```
17                                                  {
18         armaSpeed = arma_speed;
19     }
```

6.25.3 Documentazione dei membri dato

6.25.3.1 armaMeasurement

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

6.25.3.2 armaSpeed

```
double it.unibo.torsello.bluetoothpositioning.filters.MyArmaRssiFilter.armaSpeed = 0.1D  [static],
[private]
```

6.25.3.3 isEnabled

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

6.25.3.4 isInitialized

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

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

- MyArmaRssiFilter.java

6.26 Riferimenti per l'interfaccia it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0↩ Fragment.OnRecordingReport

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.On↩ RecordingReport

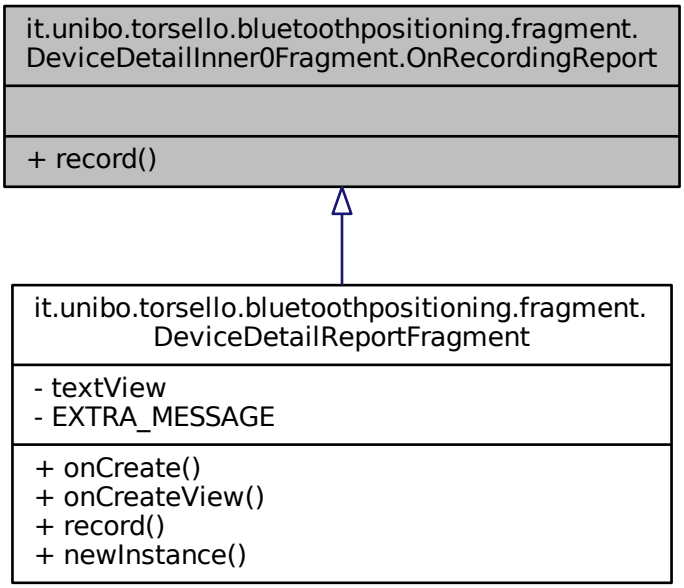
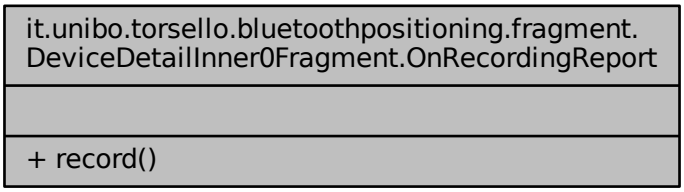


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.On↩ RecordingReport:



Membri pubblici

- void record (String newRecord)

6.26.1 Documentazione delle funzioni membro

6.26.1.1 record()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.OnRecording↩
Report.record (
            String newRecord )
```

Implementato in it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailReportFragment.

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

- DeviceDetailInner0Fragment.java

6.27 Riferimenti per l'interfaccia it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0↩ Fragment.OnRecordingResume

Diagramma delle classi per it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.On↩ RecordingResume

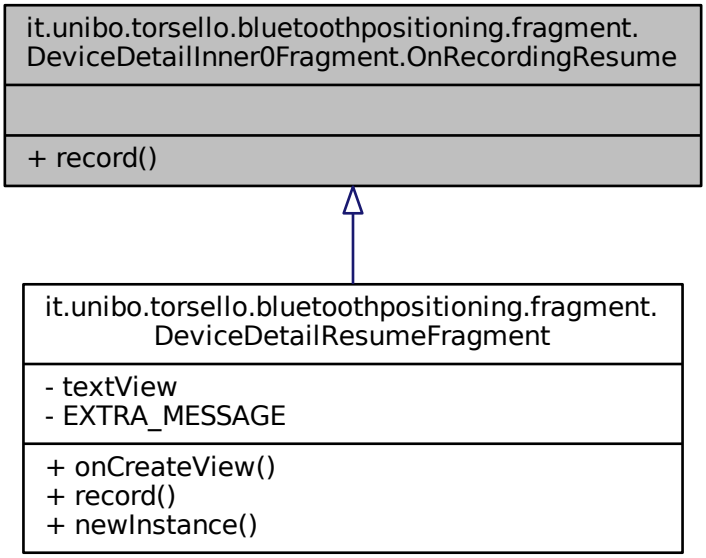
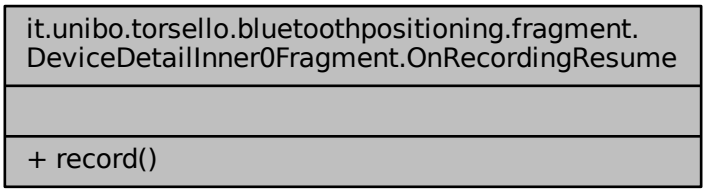


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.On↩
RecordingResume:



Membri pubblici

- void record (String newRecord)

6.27.1 Documentazione delle funzioni membro

6.27.1.1 record()

```
void it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.OnRecording↩
Resume.record (
            String newRecord )
```

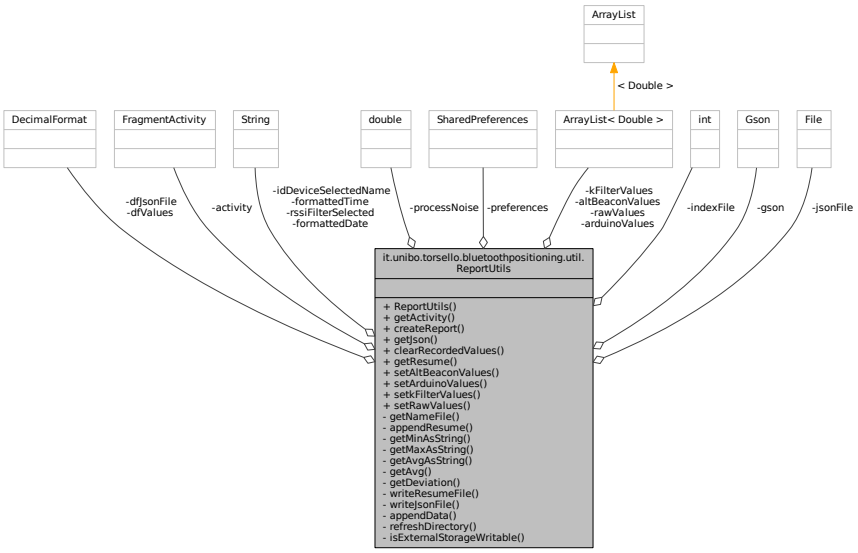
Implementato in it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailResumeFragment.

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

- DeviceDetailInner0Fragment.java

6.28 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.util.ReportUtils

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



Membri pubblici

- ReportUtils (FragmentActivity fragmentActivity, String deviceName)
- FragmentActivity getActivity ()
- void createReport ()
- String getJson ()
- void clearRecordedValues ()
- String getResume ()
- void setAltBeaconValues (Double altBeaconValues)
- void setArduinoValues (Double arduinoValues)
- void setkFilterValues (Double kFilterValues)
- void setRawValues (Double rawValues)

Membri privati

- String getNameFile ()
- String appendResume (String arg, ArrayList< Double > value)
- String getMinAsString (ArrayList< Double > value)
- String getMaxAsString (ArrayList< Double > value)
- String getAvgAsString (ArrayList< Double > value)
- Double getAvg (ArrayList< Double > value)
- String getDeviation (Double value)
- void writeResumeFile ()
- void writeJsonFile ()
- JsonWriter appendData (JsonWriter jw, ArrayList< Double > value) throws IOException
- void refreshDirectory (File file)
- boolean isExternalStorageWritable ()

Attributi privati

- String idDeviceSelectedName
- FragmentActivity activity
- String rssiFilterSelected
- ArrayList< Double > arduinoValues
- ArrayList< Double > rawValues
- ArrayList< Double > altBeaconValues
- ArrayList< Double > kFilterValues
- String formattedDate
- String formattedTime
- DecimalFormat dfValues
- DecimalFormat dfJsonFile
- Gson gson
- File jsonFile
- int indexFile = 1
- SharedPreferences preferences
- double processNoise

6.28.1 Descrizione dettagliata

Created by federico on 11/10/16.

6.28.2 Documentazione dei costruttori e dei distruttori

6.28.2.1 ReportUtils()

```
it.unibo.torsello.bluetoothpositioning.util.ReportUtils.ReportUtils (
            FragmentActivity fragmentActivity,
            String deviceName )
```

```
65                                                                          {
66          this.activity = fragmentActivity;
67          this.idDeviceSelectedName = deviceName;
68
69          arduinoValues = new ArrayList<>();
70          rawValues = new ArrayList<>();
71          altBeaconValues = new ArrayList<>();
72          kFilterValues = new ArrayList<>();
73
74          formattedDate = new SimpleDateFormat("dd-MMM-yyyy", Locale.getDefault())
75                  .format(Calendar.getInstance().getTime());
76
77          formattedTime = new SimpleDateFormat("HH:mm", Locale.getDefault())
78                  .format(Calendar.getInstance().getTime());
79
80          dfValues = new DecimalFormat("0.00", DecimalFormatSymbols.getInstance());
81
82          dfJsonFile = new DecimalFormat("00", DecimalFormatSymbols.getInstance());
83
84
85          gson = new GsonBuilder().setPrettyPrinting().create();
86
87          preferences = getActivity().
88                  getSharedPreferences(SettingConstants.SETTINGS_PREFERENCES, 0);
89      }
```

6.28.3 Documentazione delle funzioni membro

6.28.3.1 appendData()

```
JsonWriter it.unibo.torsello.bluetoothpositioning.util.ReportUtils.appendData (
            JsonWriter jw,
            ArrayList< Double > value ) throws IOException  [private]
```

```
398                                                                                 {
399         jw.name("min").value(getMinAsString(value));
400         jw.name("max").value(getMaxAsString(value));
401         jw.name("avg").value(getAvgAsString(value));
402         jw.name("min dev").value(getDeviation(Collections.min(value)));
403         jw.name("max dev").value(getDeviation(Collections.max(value)));
404         jw.name("avg dev").value(getDeviation(getAvg(value)));
405         return jw;
406     }
```

6.28.3.2 appendResume()

```
String it.unibo.torsello.bluetoothpositioning.util.ReportUtils.appendResume (
            String arg,
            ArrayList< Double > value )  [private]
```

```
228                                                             {
229
230         StringBuilder sb = new StringBuilder();
231
232         sb.append(arg).append("\n");
233         sb.append("Min:\t").append(getMinAsString(value))
234                 .append("\t\tdeviation:\t").append(getDeviation(Collections.min(value))).append
      ("\n");
235         sb.append("Max:\t").append(getMaxAsString(value))
236                 .append("\t\tdeviation:\t").append(getDeviation(Collections.max(value))).append
      ("\n");
237         sb.append("Avg:\t").append(getAvgAsString(value))
238                 .append("\t\tdeviation:\t").append(getDeviation(
     getAvg(value))).append("\n");
239         sb.append("\n");
240         return String.valueOf(sb);
241     }
```

6.28.3.3 clearRecordedValues()

```
void it.unibo.torsello.bluetoothpositioning.util.ReportUtils.clearRecordedValues ( )
```

```
185                                     {
186         arduinoValues.clear();
187         rawValues.clear();
188         altBeaconValues.clear();
189         kFilterValues.clear();
190     }
```

6.28.3.4 createReport()

```
void it.unibo.torsello.bluetoothpositioning.util.ReportUtils.createReport ( )
```

```
95                                {
96
97          double processNoise = preferences.getFloat(SettingConstants.
    KALMAN_NOISE_VALUE, 0);
98
99          String tempFilter = "";
100          if (rssiFilterSelected != null) {
101              tempFilter = rssiFilterSelected;
102          }
103
104          rssiFilterSelected = ((ApplicationActivity)
    getActivity()).getRssiFilterSelected();
105
106          if (isExternalStorageWritable()) {
107              File root = Environment.getExternalStorageDirectory();
108
109              File dir = new File(root.getAbsolutePath()
110                      + File.separator
111                      + getActivity().getString(R.string.app_name)
112                      + " Report");
113              dir.mkdirs();
114
115              File subDir1 = new File(dir.getAbsolutePath() + File.separator
116                      + formattedDate);
117              subDir1.mkdirs();
118
119              File subDir2 = new File(subDir1.getAbsolutePath() + File.separator
120                      + "KF process noise " + processNoise);
121              subDir2.mkdirs();
122
123              File subDir3 = new File(subDir2.getAbsolutePath() + File.separator
124                      + rssiFilterSelected);
125              subDir3.mkdirs();
126
127              if (!rssiFilterSelected.equals(tempFilter)) {
128                  indexFile = 1;
129              }
130
131              jsonFile = new File(subDir3, getNameFile());
132
133              if (jsonFile.exists()) {
134                  if (!rssiFilterSelected.equals(tempFilter)) {
135                      indexFile = 1;
136                  } else {
137                      String name = jsonFile.getName();
138                      String substring = name.substring(name.length() - 7, name.length() - 5);
139                      indexFile = Integer.parseInt(substring);
140                      indexFile++;
141
142                      jsonFile = new File(subDir3, getNameFile());
143                  }
144              } else {
145                  indexFile = 1;
146              }
147
148              try {
149                  jsonFile.createNewFile();
150              } catch (IOException e) {
151                  e.printStackTrace();
152              }
153
154              writeJsonFile();
155
156              writeResumeFile();
157          }
158      }
```

6.28.3.5 getActivity()

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

```
91                                    {
92          return activity;
93      }
```

6.28.3.6 getAvg()

```
Double it.unibo.torsello.bluetoothpositioning.util.ReportUtils.getAvg (
            ArrayList< Double > value )  [private]
```

```
255                                                 {
256         Double sum = 0D;
257         if (!value.isEmpty()) {
258             for (Double mark : value) {
259                 sum += mark;
260             }
261             return sum / value.size();
262         }
263         return sum;
264     }
```

6.28.3.7 getAvgAsString()

```
String it.unibo.torsello.bluetoothpositioning.util.ReportUtils.getAvgAsString (
            ArrayList< Double > value )  [private]
```

```
251                                                         {
252         return dfValues.format(getAvg(value)) + "m";
253     }
```

6.28.3.8 getDeviation()

```
String it.unibo.torsello.bluetoothpositioning.util.ReportUtils.getDeviation (
            Double value )  [private]
```

```
266                                             {
267
268         double deviation = (value - indexFile);
269         if (deviation >= 0) {
270             return "+" + dfValues.format(deviation) + "m";
271         } else {
272             return dfValues.format(deviation) + "m";
273         }
274     }
```

6.28.3.9 getJson()

```
String it.unibo.torsello.bluetoothpositioning.util.ReportUtils.getJson ( )
```

```
164                             {
165
166         String temp = "";
167
168         try {
169             FileInputStream fin = new FileInputStream(jsonFile);
170
171             int chars;
172
173             while ((chars = fin.read()) != -1) {
174                 temp = temp + Character.toString((char) chars);
175             }
176
177             //string temp contains all the data of the file.
178             fin.close();
179         } catch (IOException e) {
180             e.printStackTrace();
181         }
182         return temp;
183     }
```

6.28.3.10 getMaxAsString()

```
String it.unibo.torsello.bluetoothpositioning.util.ReportUtils.getMaxAsString (
            ArrayList< Double > value )  [private]
```

```
247                                                     {
248         return dfValues.format(Collections.max(value)) + "m";
249     }
```

6.28.3.11 getMinAsString()

```
String it.unibo.torsello.bluetoothpositioning.util.ReportUtils.getMinAsString (
            ArrayList< Double > value )  [private]
```

```
243                                                     {
244         return dfValues.format(Collections.min(value)) + "m";
245     }
```

6.28.3.12 getNameFile()

```
String it.unibo.torsello.bluetoothpositioning.util.ReportUtils.getNameFile ( )  [private]
```

```
160                                 {
161         return String.format(idDeviceSelectedName + " - %s.json",
      dfJsonFile.format(indexFile));
162     }
```

6.28.3.13 getResume()

```
String it.unibo.torsello.bluetoothpositioning.util.ReportUtils.getResume ( )
```

```
192                                 {
193
194         boolean isKalmanFilterEnabled = preferences
195                 .getBoolean(SettingConstants.KALMAN_FILTER_ENABLED, false);
196
197         StringBuilder sb = new StringBuilder();
198
199         sb.append("Date: ").append(formattedDate).append("\n");
200
201         sb.append("Time: ").append(formattedTime).append("\n");
202
203         sb.append("Filter: ").append(rssiFilterSelected).append("\n");
204
205         sb.append("Reference: ").append(indexFile).append("m").append("\n\n");
206
207         sb.append(appendResume("Raw", rawValues));
208
209         sb.append(appendResume("AltBeacon", altBeaconValues));
210
211         if (isKalmanFilterEnabled) {
212             sb.append(appendResume("KFilter Filter", kFilterValues));
213             sb.append("ProcessNoise: ").append(processNoise).append("\n\n");
214         } else {
215             sb.append("KFilter Filter").append("\n");
216             sb.append("no data - filter disabled").append("\n\n");
217         }
218         if (!arduinoValues.isEmpty()) {
219             sb.append(appendResume("Arduino", arduinoValues));
220         } else {
221             sb.append("Arduino").append("\n");
222             sb.append("no data - arduino not connected");
223         }
224
225         return String.valueOf(sb);
226     }
```

6.28.3.14 isExternalStorageWritable()

```
boolean it.unibo.torsello.bluetoothpositioning.util.ReportUtils.isExternalStorageWritable ( )
[private]
```

```
423                                                 {
424         String state = Environment.getExternalStorageState();
425         if (Environment.MEDIA_MOUNTED.equals(state)) {
426             return true;
427         }
428         Log.d("sure", "not writeable");
429         return false;
430     }
```

6.28.3.15 refreshDirectory()

```
void it.unibo.torsello.bluetoothpositioning.util.ReportUtils.refreshDirectory (
             File file )  [private]
```

```
408                                                 {
409         if (file != null) {
410             Intent mediaScanIntent = new Intent(Intent.ACTION_MEDIA_SCANNER_SCAN_FILE);
411             mediaScanIntent.setData(Uri.fromFile(file));
412             getActivity().sendBroadcast(mediaScanIntent);
413 
414             Snackbar.make(getActivity().findViewById(R.id.fab),
415                     "Report saved", Snackbar.LENGTH_SHORT).show();
416         } else {
417             Snackbar.make(getActivity().findViewById(R.id.fab),
418                     "Report retrieval failed", Snackbar.LENGTH_SHORT).show();
419         }
420     }
```

6.28.3.16 setAltBeaconValues()

```
void it.unibo.torsello.bluetoothpositioning.util.ReportUtils.setAltBeaconValues (
             Double altBeaconValues )
```

```
432                                                         {
433         this.altBeaconValues.add(altBeaconValues);
434     }
```

6.28.3.17 setArduinoValues()

```
void it.unibo.torsello.bluetoothpositioning.util.ReportUtils.setArduinoValues (
             Double arduinoValues )
```

```
436                                                       {
437         this.arduinoValues.add(arduinoValues);
438     }
```

6.28.3.18 setkFilterValues()

```
void it.unibo.torsello.bluetoothpositioning.util.ReportUtils.setkFilterValues (
             Double kFilterValues )
```

```
440                                                       {
441         this.kFilterValues.add(kFilterValues);
442     }
```

6.28.3.19 setRawValues()

```
void it.unibo.torsello.bluetoothpositioning.util.ReportUtils.setRawValues (
            Double rawValues )
```

```
444                                                 {
445         this.rawValues.add(rawValues);
446     }
```

6.28.3.20 writeJsonFile()

```
void it.unibo.torsello.bluetoothpositioning.util.ReportUtils.writeJsonFile ( )  [private]
```

```
295                                 {
296
297         processNoise = preferences.getFloat(SettingConstants.KALMAN_NOISE_VALUE, 0);
298         boolean isKalmanFilterEnabled = preferences.getBoolean(SettingConstants.
    KALMAN_FILTER_ENABLED, true);
299
300         try {
301             Writer writer = new FileWriter(jsonFile);
302
303             JsonWriter jw = gson.newJsonWriter(writer);
304
305             jw.beginObject();
306             {
307                 jw.name("id").value("report " + dfJsonFile.format(
    indexFile));
308                 jw.name("date").value(formattedDate);
309                 jw.name("time").value(formattedTime);
310                 jw.name("filter").value(rssiFilterSelected);
311                 jw.name("reference").value(indexFile + "m");
312                 jw.name("distance_estimation");
313                 jw.beginObject();
314                 {
315                     jw.name("raw");
316                     jw.beginObject();
317                     {
318                         appendData(jw, rawValues);
319
320                         jw.name("raw_values");
321                         jw.beginArray();
322                         for (Double value : rawValues) {
323                             jw.value(value);
324                         }
325                         jw.endArray();
326                     }
327                     jw.endObject();
328
329                     jw.name("altbeacon");
330                     jw.beginObject();
331                     {
332                         appendData(jw, altBeaconValues);
333
334                         jw.name("altbeacon_values");
335                         jw.beginArray();
336                         for (Double value : altBeaconValues) {
337                             jw.value(value);
338                         }
339                         jw.endArray();
340                     }
341                     jw.endObject();
342
343
344                     if (isKalmanFilterEnabled) {
345
346                         jw.name("kFilter_processNoise").value(processNoise);
347
348                         jw.name("kFilter");
349                         jw.beginObject();
350                         {
351                             appendData(jw, kFilterValues);
352
353                             jw.name("kFilter_values");
354                             jw.beginArray();
355                             for (Double value : kFilterValues) {
356                                 jw.value(value);
357                             }
```

```
358                             jw.endArray();
359                         }
360                         jw.endObject();
361 
362                     } else {
363                         jw.name("kFilter").value(getActivity().getString(R.string.
    kalman_filter_disabled));
364                     }
365 
366                     if (!arduinoValues.isEmpty()) {
367                         jw.name("arduino");
368                         jw.beginObject();
369                         {
370                             appendData(jw, arduinoValues);
371 
372                             jw.name("arduino_values");
373                             jw.beginArray();
374                             for (Double value : arduinoValues) {
375                                 jw.value(value);
376                             }
377                             jw.endArray();
378                         }
379                         jw.endObject();
380 
381                     } else {
382                         jw.name("arduino").value("no data - arduino not connected");
383                     }
384                 }
385                 jw.endObject();
386             }
387             jw.endObject();
388 
389             writer.close();
390 
391             refreshDirectory(jsonFile);
392 
393         } catch (IOException e) {
394             e.printStackTrace();
395         }
396     }
```

6.28.3.21 writeResumeFile()

```
void it.unibo.torsello.bluetoothpositioning.util.ReportUtils.writeResumeFile ( )  [private]
```

```
276                                     {
277 
278         try {
279 
280             String fileTxtName = String.format(idDeviceSelectedName + " - %s.txt",
    dfJsonFile.format(indexFile));
281             File myFile = new File(jsonFile.getParent(), fileTxtName);
282             myFile.createNewFile();
283             FileOutputStream fOut = new FileOutputStream(myFile);
284             OutputStreamWriter myOutWriter = new OutputStreamWriter(fOut);
285             myOutWriter.append(getResume());
286             myOutWriter.close();
287             fOut.close();
288             refreshDirectory(myFile);
289         } catch (Exception e) {
290             e.printStackTrace();
291         }
292 
293     }
```

6.28.4 Documentazione dei membri dato

6.28.4.1 activity

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

6.28.4.2 altBeaconValues

```
ArrayList<Double> it.unibo.torsello.bluetoothpositioning.util.ReportUtils.altBeaconValues
[private]
```

6.28.4.3 arduinoValues

```
ArrayList<Double> it.unibo.torsello.bluetoothpositioning.util.ReportUtils.arduinoValues  [private]
```

6.28.4.4 dfJsonFile

```
DecimalFormat it.unibo.torsello.bluetoothpositioning.util.ReportUtils.dfJsonFile  [private]
```

6.28.4.5 dfValues

```
DecimalFormat it.unibo.torsello.bluetoothpositioning.util.ReportUtils.dfValues  [private]
```

6.28.4.6 formattedDate

```
String it.unibo.torsello.bluetoothpositioning.util.ReportUtils.formattedDate  [private]
```

6.28.4.7 formattedTime

```
String it.unibo.torsello.bluetoothpositioning.util.ReportUtils.formattedTime  [private]
```

6.28.4.8 gson

```
Gson it.unibo.torsello.bluetoothpositioning.util.ReportUtils.gson  [private]
```

6.28.4.9 idDeviceSelectedName

```
String it.unibo.torsello.bluetoothpositioning.util.ReportUtils.idDeviceSelectedName  [private]
```

6.28.4.10 indexFile

```
int it.unibo.torsello.bluetoothpositioning.util.ReportUtils.indexFile = 1  [private]
```

6.28.4.11 jsonFile

```
File it.unibo.torsello.bluetoothpositioning.util.ReportUtils.jsonFile  [private]
```

6.28.4.12 kFilterValues

```
ArrayList<Double> it.unibo.torsello.bluetoothpositioning.util.ReportUtils.kFilterValues  [private]
```

6.28.4.13 preferences

```
SharedPreferences it.unibo.torsello.bluetoothpositioning.util.ReportUtils.preferences  [private]
```

6.28.4.14 processNoise

```
double it.unibo.torsello.bluetoothpositioning.util.ReportUtils.processNoise  [private]
```

6.28.4.15 rawValues

```
ArrayList<Double> it.unibo.torsello.bluetoothpositioning.util.ReportUtils.rawValues  [private]
```

6.28.4.16 rssiFilterSelected

```
String it.unibo.torsello.bluetoothpositioning.util.ReportUtils.rssiFilterSelected  [private]
```

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

- ReportUtils.java

6.29 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.task.SaveImageTask

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

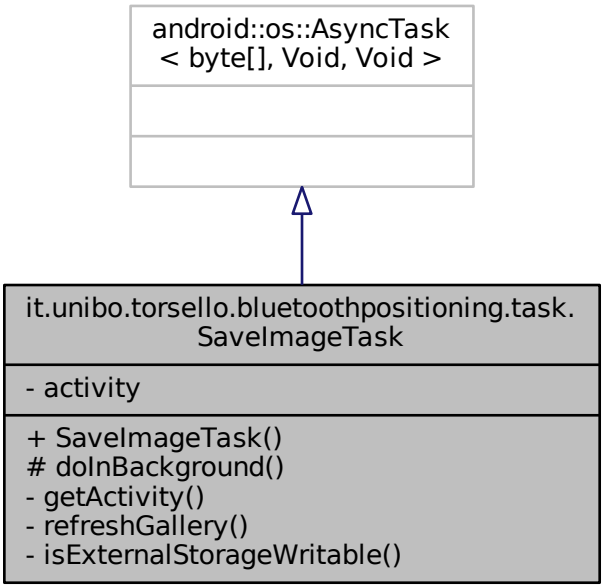
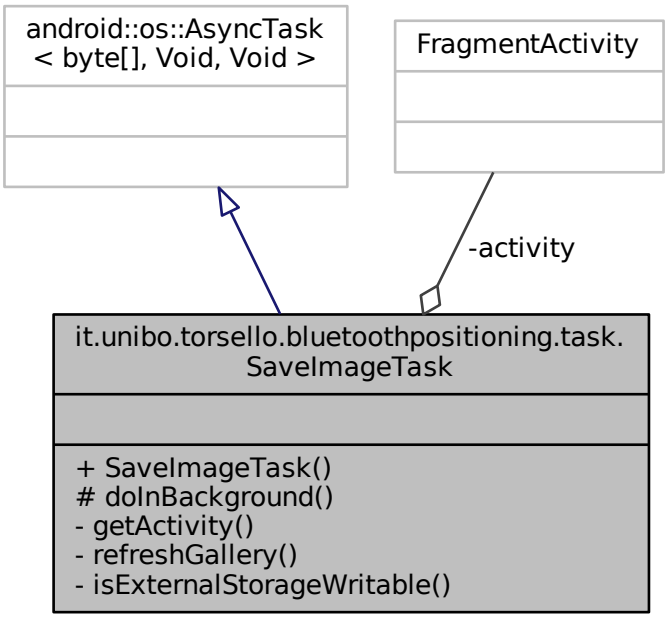


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.task.SaveImageTask:



Membri pubblici

- SaveImageTask (FragmentActivity fragmentActivity)

Membri protetti

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

Membri privati

- FragmentActivity getActivity ()
- void refreshGallery (File file)
- boolean isExternalStorageWritable ()

Attributi privati

- FragmentActivity activity

6.29.1 Descrizione dettagliata

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

6.29.2 Documentazione dei costruttori e dei distruttori

6.29.2.1 SaveImageTask()

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

```
26                                                          {
27         this.activity = fragmentActivity;
28     }
```

6.29.3 Documentazione delle funzioni membro

6.29.3.1 doInBackground()

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

```
35                                                  {
36
37         // Write to disk
38         if (isExternalStorageWritable()) {
39             File root = Environment.getExternalStorageDirectory();
40             File dir = new File(root.getAbsolutePath() + File.separator
41                     + getActivity().getString(R.string.app_name));
42             dir.mkdir();
43
44             String fileName = String.format(Locale.getDefault(), "%d.jpg", System.currentTimeMillis());
45             File outFile = new File(dir, fileName);
46             try {
47
48                 FileOutputStream outStream = new FileOutputStream(outFile);
49                 outStream.write(data[0]);
50                 outStream.flush();
51                 outStream.close();
52
53                 refreshGallery(outFile);
54             } catch (IOException e) {
55                 e.printStackTrace();
56             }
57         }
58         return null;
59     }
```

6.29.3.2 getActivity()

```
FragmentActivity it.unibo.torsello.bluetoothpositioning.task.SaveImageTask.getActivity ( )
[private]
```

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

6.29.3.3 isExternalStorageWritable()

```
boolean it.unibo.torsello.bluetoothpositioning.task.SaveImageTask.isExternalStorageWritable (
)  [private]
```

```
76                                                  {
77         String state = Environment.getExternalStorageState();
78         if (Environment.MEDIA_MOUNTED.equals(state)) {
79             return true;
80         }
81         Log.d("sure", "not writeable");
82         return false;
83     }
```

6.29.3.4 refreshGallery()

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

```
61                                          {
62         if (file != null) {
63             Intent mediaScanIntent = new Intent(Intent.ACTION_MEDIA_SCANNER_SCAN_FILE);
64             mediaScanIntent.setData(Uri.fromFile(file));
65             getActivity().sendBroadcast(mediaScanIntent);
66
67             Snackbar.make(getActivity().findViewById(R.id.fab),
68                     "Your picture has been saved", Snackbar.LENGTH_SHORT).show();
69         } else {
70             Snackbar.make(getActivity().findViewById(R.id.fab),
71                     "Image retrieval failed", Snackbar.LENGTH_SHORT).show();
72         }
73     }
```

6.29.4 Documentazione dei membri dato

6.29.4.1 activity

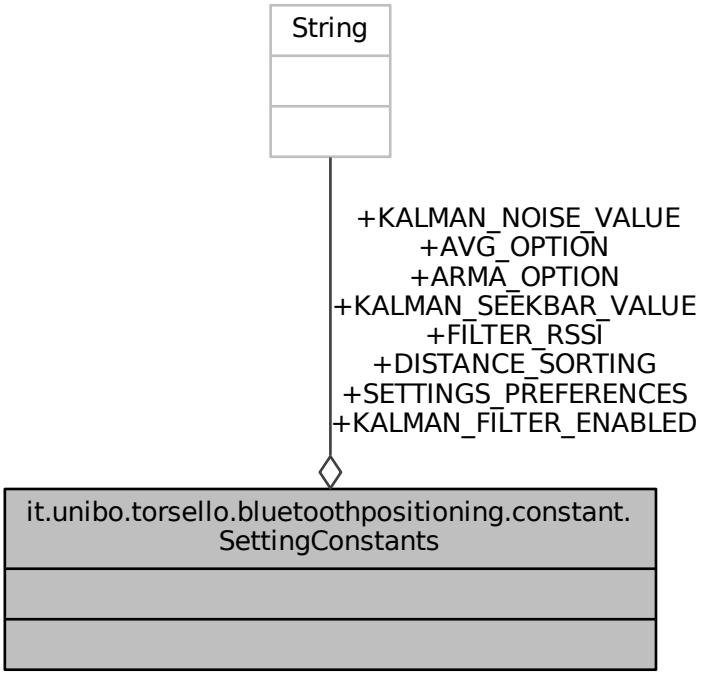
```
FragmentActivity it.unibo.torsello.bluetoothpositioning.task.SaveImageTask.activity  [private]
```

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

- SaveImageTask.java

6.30 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 ARMA_OPTION = "arma_option"
- static final String AVG_OPTION = "avg_option"
- static final String KALMAN_SEEKBAR_VALUE = "kalman_filter_seek_value"
- static final String KALMAN_NOISE_VALUE = "kalman_filter_noise_value"
- static final String KALMAN_FILTER_ENABLED = "kalma_filrer_enabled"
- static final String DISTANCE_SORTING = "distance_sorting_selected"

6.30.1 Descrizione dettagliata

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

A class containing constants for the SharedPreference objects.

6.30.2 Documentazione dei membri dato

6.30.2.1 ARMA_OPTION

```
final String it.unibo.torsello.bluetoothpositioning.constant.SettingConstants.ARMA_OPTION =
"arma_option"  [static]
```

6.30.2.2 AVG_OPTION

```
final String it.unibo.torsello.bluetoothpositioning.constant.SettingConstants.AVG_OPTION =
"avg_option"  [static]
```

6.30.2.3 DISTANCE_SORTING

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

6.30.2.4 FILTER_RSSI

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

6.30.2.5 KALMAN_FILTER_ENABLED

```
final String it.unibo.torsello.bluetoothpositioning.constant.SettingConstants.KALMAN_FILTER_↩
ENABLED = "kalma_filrer_enabled"  [static]
```

6.30.2.6 KALMAN_NOISE_VALUE

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

6.30.2.7 KALMAN_SEEKBAR_VALUE

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

6.30.2.8 SETTINGS_PREFERENCES

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

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

- SettingConstants.java

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

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

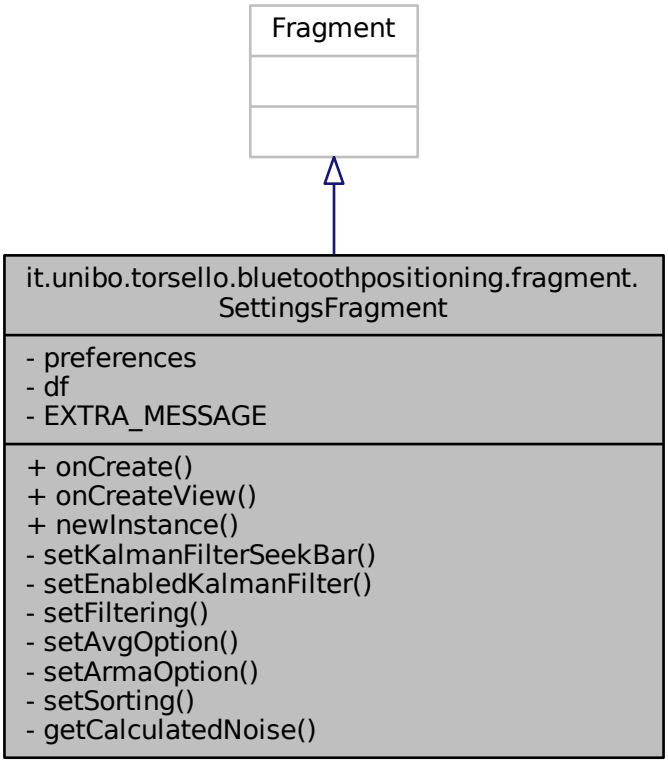
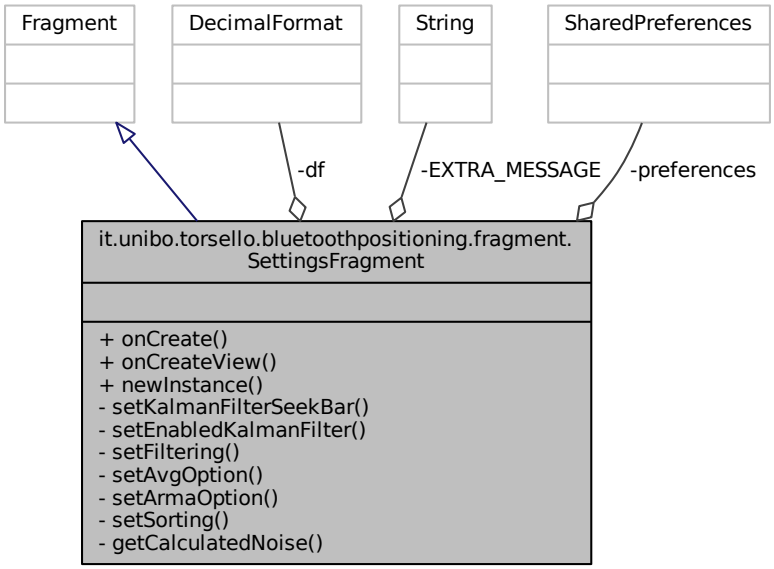


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.SettingsFragment:



Membri pubblici

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

Membri pubblici statici

- static SettingsFragment newInstance ()

Membri privati

- void setKalmanFilterSeekBar (View root)
- void setEnabledKalmanFilter (int progress)
- void setFiltering (View root)
- void setAvgOption (View root)
- void setArmaOption (View root)
- void setSorting (View root)

Membri privati statici

- static float getCalculatedNoise (int p)

Attributi privati

- SharedPreferences preferences
- DecimalFormat df

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 getCalculatedNoise()

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

```
112                                                 {
113         double percent = (p / 10D);
114         double noise = KFilterConstants.KALMAN_NOISE_MIN +
115                 (KFilterConstants.KALMAN_NOISE_MAX - KFilterConstants.KALMAN_NOISE_MIN) * percent;
116
117         return (float) noise;
118
119     }
```

6.31.2.2 newInstance()

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

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

6.31.2.3 onCreate()

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

```
38                                                  {
39         super.onCreate(savedInstanceState);
40
41         preferences = getActivity().getSharedPreferences(SettingConstants.SETTINGS_PREFERENCES,
     0);
42         df = new DecimalFormat("0.0#");
43     }
```

6.31.2.4 onCreateView()

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

```
46                                                                                                  {
47         View root = inflater.inflate(R.layout.fragment_settings, container, false);
48
49         setKalmanFilterSeekBar(root);
50
51         setFiltering(root);
52
53         setArmaOption(root);
54
55         setAvgOption(root);
56
57         setSorting(root);
58
59         return root;
60     }
```

6.31.2.5 setArmaOption()

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

```
159                                     {
160         RadioGroup rg = (RadioGroup) root.findViewById(R.id.radioGroupArmaOptions);
161         int checkedRadioButton;
162         if (rg.getCheckedRadioButtonId() != 0) {
163             checkedRadioButton = preferences.getInt(SettingConstants.ARMA_OPTION, rg.
    getCheckedRadioButtonId());
164         } else {
165             checkedRadioButton = 0;
166         }
167         rg.check(checkedRadioButton);
168         rg.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
169             @Override
170             public void onCheckedChanged(RadioGroup group, int checkedId) {
171                 SharedPreferences.Editor editor = preferences.edit();
172                 editor.putInt(SettingConstants.ARMA_OPTION, checkedId);
173                 editor.apply();
174             }
175         });
176     }
```

6.31.2.6 setAvgOption()

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

```
140                                     {
141         RadioGroup rg = (RadioGroup) root.findViewById(R.id.radioGroupAverageOptions);
142         int checkedRadioButton;
143         if (rg.getCheckedRadioButtonId() != 0) {
144             checkedRadioButton = preferences.getInt(SettingConstants.AVG_OPTION, rg.
    getCheckedRadioButtonId());
145         } else {
146             checkedRadioButton = 0;
147         }
148         rg.check(checkedRadioButton);
149         rg.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
150             @Override
151             public void onCheckedChanged(RadioGroup group, int checkedId) {
152                 SharedPreferences.Editor editor = preferences.edit();
153                 editor.putInt(SettingConstants.AVG_OPTION, checkedId);
154                 editor.apply();
155             }
156         });
157     }
```

6.31.2.7 setEnabledKalmanFilter()

```
void it.unibo.torsello.bluetoothpositioning.fragment.SettingsFragment.setEnabledKalmanFilter (
            int progress )  [private]
```

```
101                                                 {
102         SharedPreferences.Editor editor = preferences.edit();
103
104         if (progress > 0) {
105             editor.putBoolean(SettingConstants.KALMAN_FILTER_ENABLED, true);
106         } else {
107             editor.putBoolean(SettingConstants.KALMAN_FILTER_ENABLED, false);
108         }
109         editor.apply();
110     }
```

6.31.2.8 setFiltering()

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

```
121                                         {
122         RadioGroup rg = (RadioGroup) root.findViewById(R.id.radioGroupFilter);
123         int checkedRadioButton;
124         if (rg.getCheckedRadioButtonId() != 0) {
125             checkedRadioButton = preferences.getInt(SettingConstants.FILTER_RSSI, rg.
    getCheckedRadioButtonId());
126         } else {
127             checkedRadioButton = 0;
128         }
129         rg.check(checkedRadioButton);
130         rg.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
131             @Override
132             public void onCheckedChanged(RadioGroup group, int checkedId) {
133                 SharedPreferences.Editor editor = preferences.edit();
134                 editor.putInt(SettingConstants.FILTER_RSSI, checkedId);
135                 editor.apply();
136             }
137         });
138     }
```

6.31.2.9 setKalmanFilterSeekBar()

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

```
63                                                  {
64         SeekBar kalmanSeek = (SeekBar) root.findViewById(R.id.kalmanSeek);
65         int seekValue = preferences.getInt(SettingConstants.KALMAN_SEEKBAR_VALUE, 1);
66
67         setEnabledKalmanFilter(seekValue);
68
69         kalmanSeek.setProgress(seekValue);
70
71         final TextView kalmanFilterValue = (TextView) root.findViewById(R.id.kalmanValue);
72         kalmanFilterValue.setText(df.format(getCalculatedNoise(seekValue)));
73
74         kalmanSeek.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
75             @Override
76             public void onProgressChanged(SeekBar seekBar, int seekValue, boolean fromUser) {
77                 kalmanFilterValue.setText(df.format(getCalculatedNoise(seekValue)));
78             }
79
80             @Override
81             public void onStartTrackingTouch(SeekBar seekBar) {
82             }
83
84             @Override
85             public void onStopTrackingTouch(SeekBar seekBar) {
86                 SharedPreferences.Editor editor = preferences.edit();
87
```

```
88                 int progress = seekBar.getProgress();
89
90                 float storeProgress = getCalculatedNoise(progress);
91                 editor.putInt(SettingConstants.KALMAN_SEEKBAR_VALUE, progress);
92                 editor.putFloat(SettingConstants.KALMAN_NOISE_VALUE, storeProgress);
93                 editor.apply();
94
95                 setEnabledKalmanFilter(progress);
96                 kalmanFilterValue.setText(df.format(storeProgress));
97             }
98         });
99     }
```

6.31.2.10 setSorting()

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

```
178                                     {
179         RadioGroup rg = (RadioGroup) root.findViewById(R.id.radioGroupSortingMode);
180         int checkedRadioButton;
181         if (rg.getCheckedRadioButtonId() != 0) {
182             checkedRadioButton = preferences.getInt(SettingConstants.DISTANCE_SORTING, rg.
    getCheckedRadioButtonId());
183         } else {
184             checkedRadioButton = 0;
185         }
186         rg.check(checkedRadioButton);
187         rg.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
188             @Override
189             public void onCheckedChanged(RadioGroup group, int checkedId) {
190                 SharedPreferences.Editor editor = preferences.edit();
191                 editor.putInt(SettingConstants.DISTANCE_SORTING, checkedId);
192                 editor.apply();
193             }
194         });
195     }
```

6.31.3 Documentazione dei membri dato

6.31.3.1 df

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

6.31.3.2 EXTRA_MESSAGE

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

6.31.3.3 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

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

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

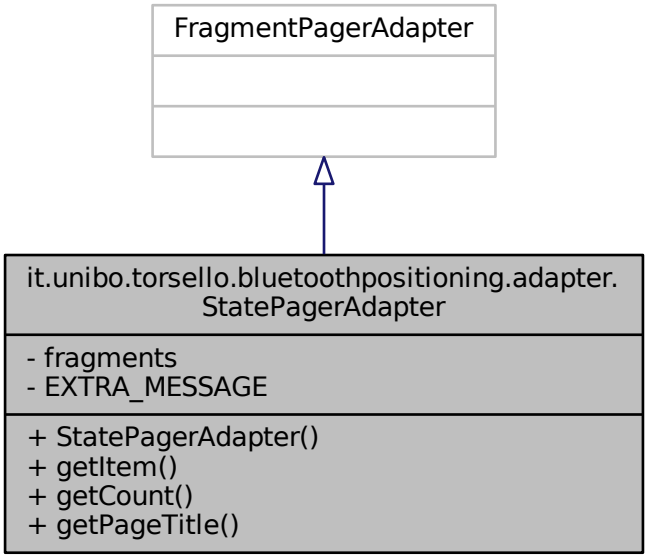
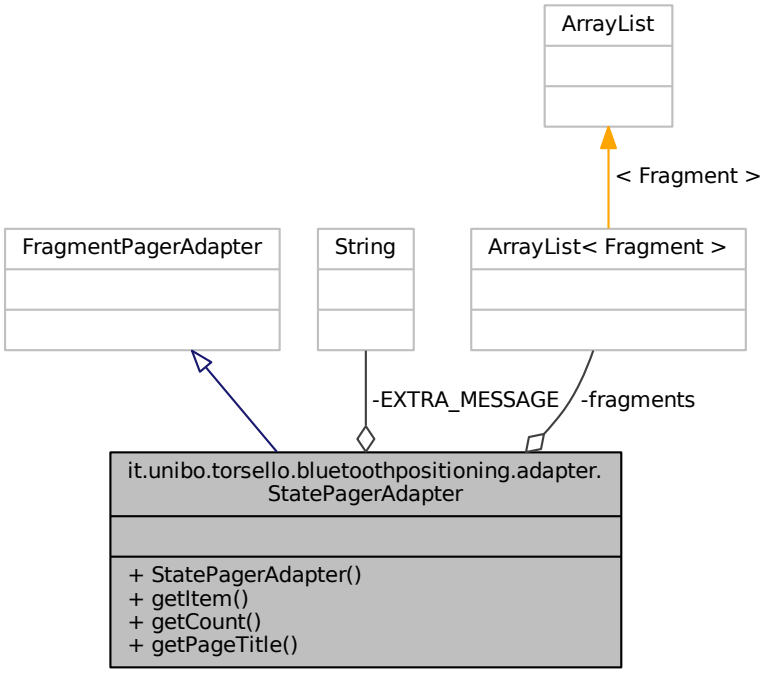


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.adapter.StatePagerAdapter:



Membri pubblici

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

Attributi privati

- ArrayList< Fragment > fragments

Attributi privati statici

- static final String EXTRA_MESSAGE = "EXTRA_MESSAGE"

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 StatePagerAdapter()

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

```
21                                                                              {
22         super(fm);
23         this.fragments = fragments;
24     }
```

6.32.3 Documentazione delle funzioni membro

6.32.3.1 getCount()

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

```
32                           {
33         return this.fragments.size();
34     }
```

6.32.3.2 getItem()

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

```
27                                       {
28         return this.fragments.get(position);
29     }
```

6.32.3.3 getPageTitle()

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

```
37                                                 {
38         return fragments.get(position).getArguments().getString(
      EXTRA_MESSAGE);
39     }
```

6.32.4 Documentazione dei membri dato

6.32.4.1 EXTRA_MESSAGE

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

6.32.4.2 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

6.33 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.fragment.UsbMeasurementFragment

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

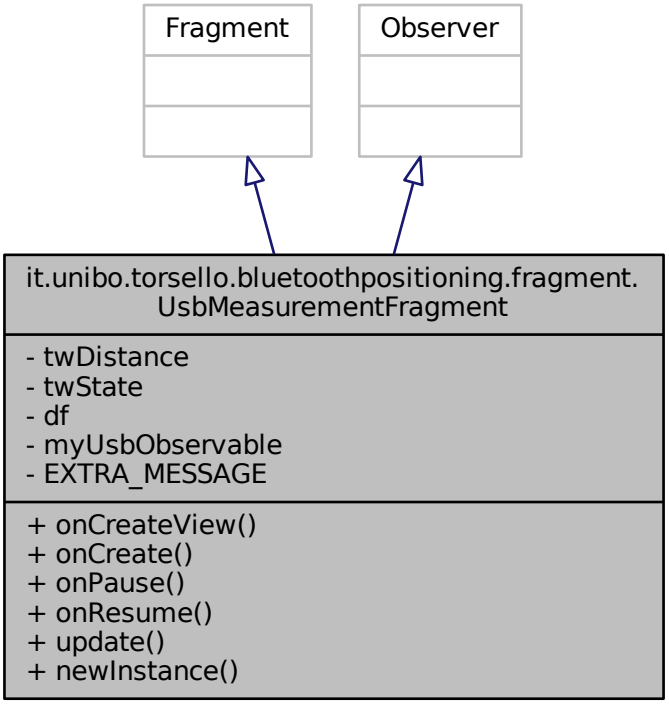
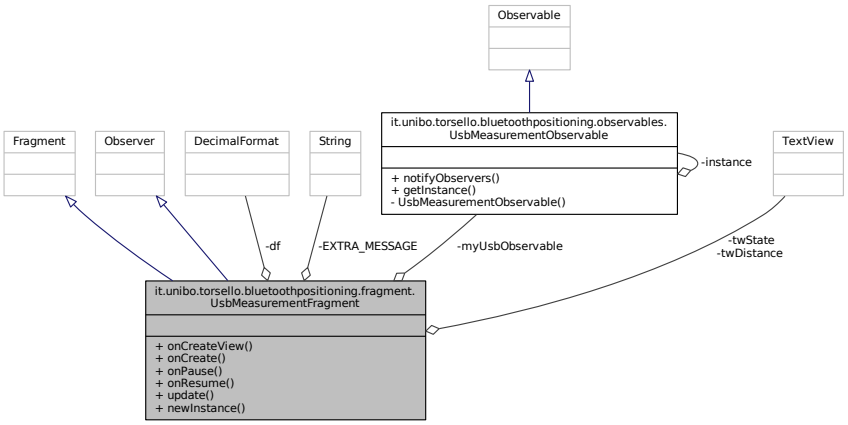


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.fragment.UsbMeasurementFragment:



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"

6.33.1 Descrizione dettagliata

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

6.33.2 Documentazione delle funzioni membro

6.33.2.1 newInstance()

```
static UsbMeasurementFragment it.unibo.torsello.bluetoothpositioning.fragment.UsbMeasurement↩
Fragment.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     }
```

6.33.2.2 onCreate()

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

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

6.33.2.3 onCreateView()

```
View it.unibo.torsello.bluetoothpositioning.fragment.UsbMeasurementFragment.onCreateView (
              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     }
```

6.33.2.4 onPause()

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

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

6.33.2.5 onResume()

```
void it.unibo.torsello.bluetoothpositioning.fragment.UsbMeasurementFragment.onResume ( )
```

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

6.33.2.6 update()

```
void it.unibo.torsello.bluetoothpositioning.fragment.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     }
```

6.33.3 Documentazione dei membri dato

6.33.3.1 df

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

6.33.3.2 EXTRA_MESSAGE

```
final String it.unibo.torsello.bluetoothpositioning.fragment.UsbMeasurementFragment.EXTRA_ME↩
SSAGE = "EXTRA_MESSAGE"  [static], [private]
```

6.33.3.3 myUsbObservable

```
UsbMeasurementObservable it.unibo.torsello.bluetoothpositioning.fragment.UsbMeasurement↩
Fragment.myUsbObservable  [private]
```

6.33.3.4 twDistance

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

6.33.3.5 twState

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

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

- UsbMeasurementFragment.java

6.34 Riferimenti per la classe it.unibo.torsello.bluetoothpositioning.observables.UsbMeasurementObservable

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

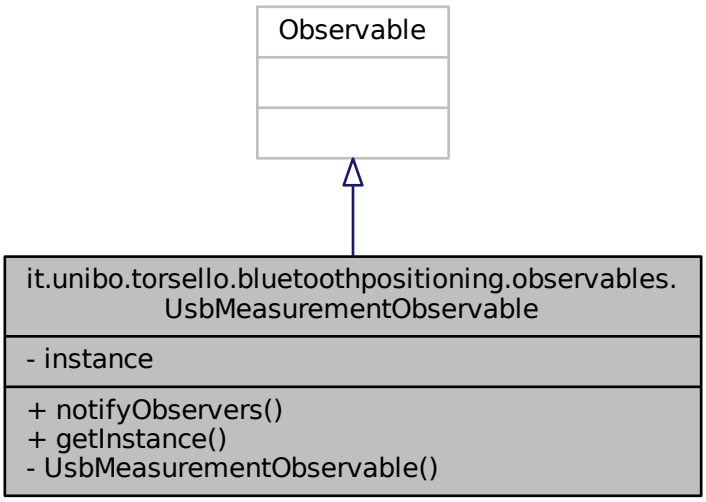
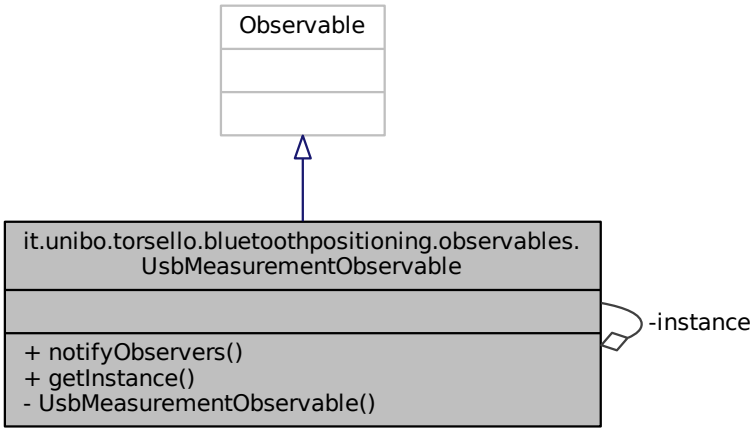


Diagramma di collaborazione per it.unibo.torsello.bluetoothpositioning.observables.UsbMeasurementObservable:



Membri pubblici

- void notifyObservers (Object data)

Membri pubblici statici

- static UsbMeasurementObservable getInstance ()

Membri privati

- UsbMeasurementObservable ()

Attributi privati statici

- static UsbMeasurementObservable instance = new UsbMeasurementObservable()

6.34.1 Descrizione dettagliata

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

6.34.2 Documentazione dei costruttori e dei distruttori

6.34.2.1 UsbMeasurementObservable()

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

```
13                                      {
14         super();
15
16     }
```

6.34.3 Documentazione delle funzioni membro

6.34.3.1 getInstance()

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

```
18                                                      {
19         return instance;
20     }
```

6.34.3.2 notifyObservers()

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

```
23                                          {
24         setChanged();
25         super.notifyObservers(data);
26     }
```

6.34.4 Documentazione dei membri dato

6.34.4.1 instance

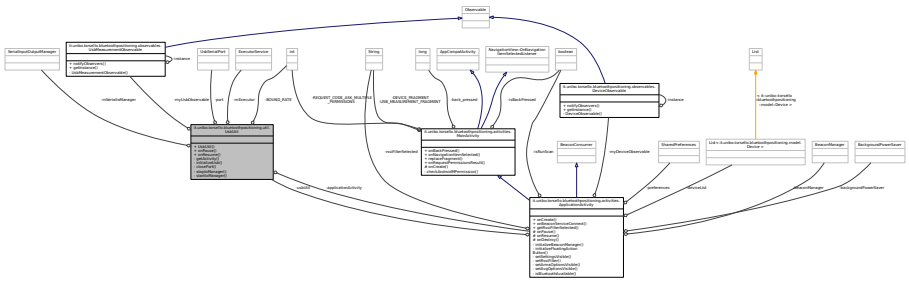
```
UsbMeasurementObservable it.unibo.torsello.bluetoothpositioning.observables.UsbMeasurement↩
Observable.instance = new UsbMeasurementObservable()  [static], [private]
```

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

- UsbMeasurementObservable.java

6.35 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

Attributi privati statici

- static final int BOUND_RATE = 115200

6.35.1 Descrizione dettagliata

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

6.35.2 Documentazione dei costruttori e dei distruttori

6.35.2.1 UsbUtil()

```
it.unibo.torsello.bluetoothpositioning.util.UsbUtil.UsbUtil (
            ApplicationActivity applicationActivity )
```

```
40                                                          {
41         this.applicationActivity = applicationActivity;
42         myUsbObservable = UsbMeasurementObservable.getInstance();
43     }
```

6.35.3 Documentazione delle funzioni membro

6.35.3.1 closePort()

```
void it.unibo.torsello.bluetoothpositioning.util.UsbUtil.closePort ( )  [private]
```

```
117                                {
118         if (port != null) {
119             try {
120                 port.close();
121             } catch (IOException e) {
122                 // Ignore.
123             }
124             port = null;
125         }
126     }
```

6.35.3.2 getActivity()

```
ApplicationActivity it.unibo.torsello.bluetoothpositioning.util.UsbUtil.getActivity ( )  [private]
```

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

6.35.3.3 initializeUsb()

```
void it.unibo.torsello.bluetoothpositioning.util.UsbUtil.initializeUsb ( )  [private]
```

```
59                                  {
60
61          // Find all available drivers from attached devices.
62          UsbManager usbManager = (UsbManager) getActivity().getSystemService(Context.USB_SERVICE)
    ;
63          List<UsbSerialDriver> availableDrivers = UsbSerialProber.getDefaultProber().findAllDrivers(
    usbManager);
64          if (!availableDrivers.isEmpty()) {
65
66              // Open a connection to the first available driver.
67              UsbSerialDriver driver = availableDrivers.get(0);
68
69              if (usbManager.hasPermission(driver.getDevice())) {
70                  if (usbManager.openDevice(driver.getDevice()) != null) {
71                      // Read some data! Most have just one port (port 0).
72                      port = driver.getPorts().get(0);
73                  }
74              } else {
75                  Intent startIntent = new Intent(getActivity(), getClass());
76                  PendingIntent pendingIntent =
77                          PendingIntent.getService(getActivity(), 0, startIntent, 0);
78                  usbManager.requestPermission(driver.getDevice(), pendingIntent);
79
80 //                 PendingIntent mPendingIntent = PendingIntent.getBroadcast(getActivity(), 0, new
     Intent("intent"), 0);
81 //                 usbManager.requestPermission(driver.getDevice(), mPendingIntent);
82              }
83
84              if (port != null) {
85
86                  UsbDeviceConnection connection = usbManager.openDevice(port.getDriver().getDevice());
87
88                  if (connection != null) {
89                      try {
90                          port.open(connection);
91                          port.setParameters(BOUND_RATE, UsbSerialPort.DATABITS_8,
92                                  UsbSerialPort.STOPBITS_1, UsbSerialPort.PARITY_NONE);
93
94 //                       String details = "CD  - Carrier Detect" + port.getCD() + '\n' +
95 //                               "CTS - Clear To Send" + port.getCTS() + '\n' +
96 //                               "DSR - Data Set Ready" + port.getDSR() + '\n' +
97 //                               "DTR - Data Terminal Ready" + port.getDTR() + '\n' +
98 //                               "DSR - Data Set Ready" + port.getDSR() + '\n' +
99 //                               "RI  - Ring Indicator" + port.getRI() + '\n' +
100 //                               "RTS - Request To Send" + port.getRTS();
101
102                     } catch (IOException e) {
103                         myUsbObservable.notifyObservers(
    getActivity().getString(R.string.error_opening_device)
104                                 + " " + e.getMessage());
105                         myUsbObservable.notifyObservers(false);
106                         closePort();
107                         return;
108                     }
109
110                     stopIoManager();
111                     startIoManager();
112                 }
113             }
114         }
115     }
```

6.35.3.4 onPause()

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

```
49                          {
50          stopIoManager();
51          closePort();
52      }
```

6.35.3.5 onResume()

```
void it.unibo.torsello.bluetoothpositioning.util.UsbUtil.onResume ( )
```

```
54                          {
55         initializeUsb();
56     }
```

6.35.3.6 startIoManager()

```
void it.unibo.torsello.bluetoothpositioning.util.UsbUtil.startIoManager ( )  [private]
```

```
135                                 {
136         if (port != null) {
137
138             SerialInputOutputManager.Listener mListener =
139                     new SerialInputOutputManager.Listener() {
140
141                         @Override
142                         public void onRunError(Exception e) {
143                             myUsbObservable.notifyObservers(false);
144                             myUsbObservable.notifyObservers(
      getActivity().getString(R.string.usb_device_not_connected));
145                             myUsbObservable.notifyObservers(0D);
146                         }
147
148                         @Override
149                         public void onNewData(final byte[] data) {
150                             try {
151                                 myUsbObservable.notifyObservers(true);
152                                 myUsbObservable.notifyObservers(
      getActivity().getString(R.string.usb_device_connected));
153                                 String value = new String(data, "UTF-8").trim();
154                                 double distanceEstimate = Double.valueOf(value) / 100;
155                                 myUsbObservable.notifyObservers(
      distanceEstimate);
156                             } catch (NumberFormatException | UnsupportedEncodingException nfe) {
157                                 nfe.printStackTrace();
158                             }
159                         }
160                     };
161
162             mSerialIoManager = new SerialInputOutputManager(port, mListener);
163             mExecutor.submit(mSerialIoManager);
164         }
165     }
```

6.35.3.7 stopIoManager()

```
void it.unibo.torsello.bluetoothpositioning.util.UsbUtil.stopIoManager ( )  [private]
```

```
128                                 {
129         if (mSerialIoManager != null) {
130             mSerialIoManager.stop();
131             mSerialIoManager = null;
132         }
133     }
```

6.35.4 Documentazione dei membri dato

6.35.4.1 applicationActivity

```
ApplicationActivity it.unibo.torsello.bluetoothpositioning.util.UsbUtil.applicationActivity
[private]
```

6.35.4.2 BOUND_RATE

```
final int it.unibo.torsello.bluetoothpositioning.util.UsbUtil.BOUND_RATE = 115200  [static],
[private]
```

6.35.4.3 mExecutor

```
final ExecutorService it.unibo.torsello.bluetoothpositioning.util.UsbUtil.mExecutor = Executors.↩
newSingleThreadExecutor()  [private]
```

6.35.4.4 mSerialIoManager

```
SerialInputOutputManager it.unibo.torsello.bluetoothpositioning.util.UsbUtil.mSerialIoManager
[private]
```

6.35.4.5 myUsbObservable

```
UsbMeasurementObservable it.unibo.torsello.bluetoothpositioning.util.UsbUtil.myUsbObservable
[private]
```

6.35.4.6 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

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 CameraPreviewUtil.java

Composti

- class it.unibo.torsello.bluetoothpositioning.util.CameraPreviewUtil

Package

- package it.unibo.torsello.bluetoothpositioning.util

7.4 Riferimenti per il file ChartUtil.java

Composti

- class it.unibo.torsello.bluetoothpositioning.util.ChartUtil

Package

- package it.unibo.torsello.bluetoothpositioning.util

7.5 Riferimenti per il file Device.java

Composti

- class it.unibo.torsello.bluetoothpositioning.model.Device

Package

- package it.unibo.torsello.bluetoothpositioning.model

7.6 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

7.7 Riferimenti per il file DeviceChartFragment.java

Composti

- class it.unibo.torsello.bluetoothpositioning.fragment.DeviceChartFragment

Package

- package it.unibo.torsello.bluetoothpositioning.fragment

7.8 Riferimenti per il file DeviceConstants.java

Composti

- class it.unibo.torsello.bluetoothpositioning.constant.DeviceConstants

Package

- package it.unibo.torsello.bluetoothpositioning.constant

7.9 Riferimenti per il file DeviceDetailFragment.java

Composti

- class it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailFragment

Package

- package it.unibo.torsello.bluetoothpositioning.fragment

7.10 Riferimenti per il file DeviceDetailInner0Fragment.java

Composti

- class it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment
- interface it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.OnRecordingResume
- interface it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner0Fragment.OnRecordingReport

Package

- package it.unibo.torsello.bluetoothpositioning.fragment

7.11 Riferimenti per il file DeviceDetailInner1Fragment.java

Composti

- class it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner1Fragment

Package

- package it.unibo.torsello.bluetoothpositioning.fragment

7.12 Riferimenti per il file DeviceDetailInner2Fragment.java

Composti

- class it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailInner2Fragment

Package

- package it.unibo.torsello.bluetoothpositioning.fragment

7.13 Riferimenti per il file DeviceDetailReportFragment.java

Composti

- class it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailReportFragment

Package

- package it.unibo.torsello.bluetoothpositioning.fragment

7.14 Riferimenti per il file DeviceDetailResumeFragment.java

Composti

- class it.unibo.torsello.bluetoothpositioning.fragment.DeviceDetailResumeFragment

Package

- package it.unibo.torsello.bluetoothpositioning.fragment

7.15 Riferimenti per il file DeviceListFragment.java

Composti

- class it.unibo.torsello.bluetoothpositioning.fragment.DeviceListFragment

Package

- package it.unibo.torsello.bluetoothpositioning.fragment

7.16 Riferimenti per il file DeviceObservable.java

Composti

- class it.unibo.torsello.bluetoothpositioning.observables.DeviceObservable

Package

- package it.unibo.torsello.bluetoothpositioning.observables

7.17 Riferimenti per il file Estimation.java

Composti

- class it.unibo.torsello.bluetoothpositioning.distanceEstimation.Estimation

Package

- package it.unibo.torsello.bluetoothpositioning.distanceEstimation

7.18 Riferimenti per il file FABBehavior.java

Composti

- class it.unibo.torsello.bluetoothpositioning.extra.FABBehavior

Package

- package it.unibo.torsello.bluetoothpositioning.extra

7.19 Riferimenti per il file KFilter.java

Composti

- class it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter.KFilter

Package

- package it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter

7.20 Riferimenti per il file KFilter2.java

Composti

- class it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter2.KFilter2

Package

- package it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter2

7.21 Riferimenti per il file KFilterBuilder.java

Composti

- class it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter.KFilterBuilder

Package

- package it.unibo.torsello.bluetoothpositioning.filters.kalmanFilter

7.22 Riferimenti per il file KFilterConstants.java

Composti

- class it.unibo.torsello.bluetoothpositioning.constant.KFilterConstants

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.filters.MyArmaRssiFilter

Package

- package it.unibo.torsello.bluetoothpositioning.filters

7.25 Riferimenti per il file ReportUtils.java

Composti

- class it.unibo.torsello.bluetoothpositioning.util.ReportUtils

Package

- package it.unibo.torsello.bluetoothpositioning.util

7.26 Riferimenti per il file SaveImageTask.java

Composti

- class it.unibo.torsello.bluetoothpositioning.task.SaveImageTask

Package

- package it.unibo.torsello.bluetoothpositioning.task

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.UsbMeasurementFragment

Package

- package it.unibo.torsello.bluetoothpositioning.fragment

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

Indice analitico