# Qt Mobility：Location&Map例子详解

## 介绍

Qt Mobility提供了丰富的位置（Location）和地图（Map）相关的API，开发者可已很方便的调用这些API实现位置与地图相关的应用。

本文的例子展示了如何获取当前经纬度（室内、室外）、显示地图、在地图中标记当前位置（随位置变化不断更新）、将当前位置信息发送至服务器、根据经纬度获取当前地址信息、两点间路线规划等功能。

## 相关API

QGeoPositionInfoSource：获取当前位置信息。

QGeoPositionInfo：位置信息。

QGraphicsGeoMap：显示地图数据，并负责与地图的交互。

QGeoServiceProvider：地理信息提供者。

QGeoMappingManager：为地图显示和交互提供支持。

QGeoMapPixmapObject：地图对象，本例中用他来在地图上添加自己的位置标记。

QGeoSearchManager：地理信息搜索，根据经纬度查询地址信息等操作。

QGeoRoutingManager：路线相关操作。

## 程序说明

### 获取当前位置（室内/室外）

下面的代码片段展示了如何获取当前位置信息，程序中同时启动了非GPS和GPS两种模式获取当前位置，因为当用户在室内或者高大建筑物周围时，GPS可能无法接受到卫星数据，所以此时可以使用非GPS获取当前位置：

**注意：**

1、QGeoPositionInfoSource::AllPositioningMethods目前在室内无法获取经纬度信息

2、手机中“设置->应用程序设置->定位->定位方法”中，需要设置“辅助GPS”、“集成GPS”、“基于网络”，才能获取较好的用户体验。

非GPS模式：

```
m_AGpsPosiInfo->setPreferredPositioningMethods(QGeoPositionInfoSource::NonSatellitePositioningMethods);
connect( m_AGpsPosiInfo, SIGNAL(positionUpdated(QGeoPositionInfo)), this, SLOT(receivedAPosition(QGeoPositionInfo)) );
connect( m_AGpsPosiInfo, SIGNAL(updateTimeout()), this, SLOT(noneGpsTimeOut()) );
```

GPS模式：

```
m_GpsPosiInfo->setPreferredPositioningMethods(QGeoPositionInfoSource::SatellitePositioningMethods);
connect( m_GpsPosiInfo, SIGNAL(positionUpdated(QGeoPositionInfo)), this, SLOT(receivedGpsPosition(QGeoPositionInfo)) );
connect( m_GpsPosiInfo, SIGNAL(updateTimeout()), this, SLOT(gpsTimeOut()) );
```

### 在地图中标记当前位置

下面的函数在获取到当前位置信息后调用，它构造一个QGeoMapPixmapObject对象，并将其加入MapWidget中，这样就可以在地图上显示了。

```
void MainWindow::addMarkers()
{
    QPair<QPixmap, QPoint>  markerPair = makeStandardMarker( 0, "", QPen(), QPen(QColor(Qt::white)), QBrush(QColor(0, 0, 136)) );
    QPoint                  markerOffset = markerPair.second;
    m_MarkIcon = new QGeoMapPixmapObject( m_PositionInfo.coordinate(), markerOffset, markerIcon );
    m_mapWidget->addMapObject( m_MarkIcon );
}
```

### 根据当前位置查询地址信息


(c) Copyright Nokia 2013. All rights reserved.

QGeoSearchReply中保存了所有的查询结果，下面代码中，当查询结束时会调用到reverseFinished槽，在reverseFinished槽中可以从QGeoSearchReply对象中获取结果的地址信息。

```
void MainWindow::reverseGeoCode()
{
    QGeoSearchReply *reply = NULL;
    reply = m_SearchManager->reverseGeocode( m_PositionInfo.coordinate() );
    connect(reply, SIGNAL(finished()), this,SLOT(reverseFinished()));
    connect(reply,SIGNAL(error(QGeoSearchReply::Error, QString)), this,SLOT(resultsError(QGeoSearchReply::Error, QString)));
}
```

## 路线规划

下面的函数完成从当前位置到另一个位置两点目的地间路线规划的功能，QGeoRouteRequest请求完成后，routeFinished槽被调用，然后可以通过QGeoRouteReply对象获取到路线信息，并显示在地图上。

```
void MainWindow::startRouteToCBD()
{
    QList<QGeoCoordinate> waypoints;
    waypoints.append( m_PositionInfo.coordinate() );
    // Beijing CBD
    waypoints.append( QGeoCoordinate(39.90914369936543, 116.46184444514802) );
    QGeoRouteRequest req(waypoints);
    QGeoRouteReply *reply = m_RoutingManager->calculateRoute(req);
    QObject::connect(reply, SIGNAL(finished()), this,SLOT(routeFinished()));
    QObject::connect(reply,SIGNAL(error(QGeoRouteReply::Error, QString)), this,SLOT(routeResultsError(QGeoRouteReply::Error, QString
}
```

将获取到的QGeoMapRouteObject对象添加到MapWidget中，即可在地图上看到我们所请求的两点目的地之间的路线了。

```
void MainWindow::routeFinished()
{
    QGeoRouteReply *reply = static_cast<QGeoRouteReply*>(sender());    if ( reply == NULL )
    {
        return;
    }
    if ( reply->routes().size() < 1 )
    {
        m_MsgBox->setText( tr("no route") );
        return;
    }
    QGeoMapRouteObject *route = new QGeoMapRouteObject(reply->routes().at(0));
    QColor routeColor(Qt::blue);
    routeColor.setAlpha(127); //semi-transparent
    QPen pen(routeColor);
    pen.setWidth(7);
    pen.setCapStyle(Qt::RoundCap);
    route->setPen(pen);
    m_mapWidget->addMapObject(route);
    disconnect(reply, SIGNAL(finished()), this,SLOT(routeFinished()));
    disconnect(reply,SIGNAL(error(QGeoRouteReply::Error, QString)), this,SLOT(resultsError(QGeoRouteReply::Error, QString)));
    reply->deleteLater();
}
```

## 将当前位置信息发送至服务器

QHttp提供了对Http的支持，使用它可以很方便的将位置信息以get或者post的方式上传给服务器。

```
void HttpTools::sendPosToSrv(const QGeoCoordinate& geoCoordinate)
{
    if ( m_IsRequestFinished )
    {
        m_IsRequestFinished = false;
        QString msm ,name, prefix, latitude,longitude, zorder;
        prefix = "yourprefix/add_locations.php?";
        name="1111";
        latitude.setNum( geoCoordinate.latitude() );
        longitude.setNum( geoCoordinate.longitude() );
        zorder.setNum( geoCoordinate.altitude() );
        msm = prefix + "user_id=" + name + "&xx1=" + latitude + "&yy1="+longitude+"&zz1="+zorder;
        QByteArray datacontent = QTextCodec::codecForName("GB2312")->fromUnicode(msm);          m_Http->setHost( "www.yourserver.com
        m_Http->get( msm );
        qDebug() << datacontent << "\n";
    }
    else
    {
        qDebug() << "http is busy";
    }
 }
```

# 代码例子

Qt SDK 1.1 + N8/E7 测试通过。 File:CxtMap.zip

# 相关链接

- Qt Mobility 开发
- Qt 开发
- Qt Quick (Chinese)


(C) Copyright Nokia 2013. All rights reserved.

(c) Copyright Nokia 2013. All rights reserved.