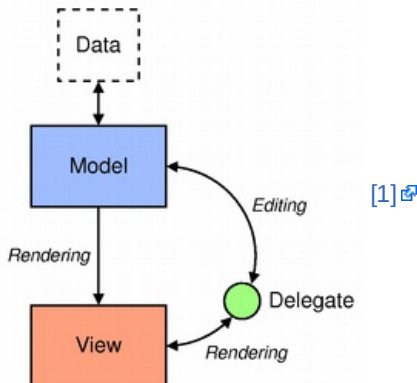


# Filtering lists in Qt

This article shows how to filter a list using [QSortFilterProxyModel](#), part of the Qt model/view architecture.

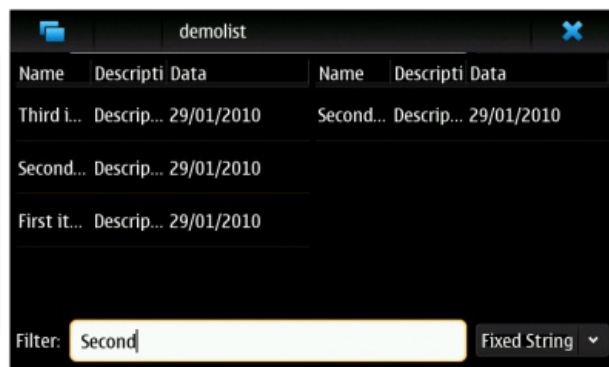
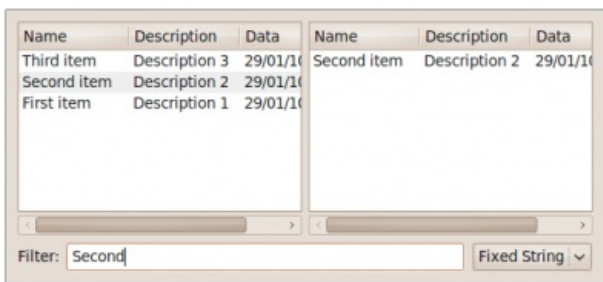
[Qt model/view architecture](#) provides separated objects for presenting and handling data sources, offering great flexibility for programmers when dealing with data. In particular, the class [QSortFilterProxyModel](#) is useful for filtering/sorting data sources, mapping the source model indexes into new indexes in a transparent way.



In the next example is demonstrated how to use [QSortFilterProxyModel](#). Two models are created. The first model, called `model`, is an instance of [QStandardItemModel](#) and it is used as the primary data source. The second model, called `proxy`, is an instance of [QSortFilterProxyModel](#) and it uses `model` as a data source. The list `lsta` will use `model` as model and the list `lstb` will use `proxy` as model.

```
model = new QStandardItemModel(0,3,this);
proxy = new QSortFilterProxyModel;
// ...
proxy->setSourceModel(model);
lsta->setModel(model);
lstb->setModel(proxy);
```

The user can filter `lstb` contents using regular expressions, wildcard or fixed string.



You can download the entire source code from [File:MBA-demolist.zip](#). It was tested in GNU/Linux and Maemo 5 running Qt 4.6.1 but it can be compiled for other platforms as well.

