

Notes to self

- . Exploring SwipeView and PageIndicator
 - . We use them to allow the user to swipe through pages
 - . Use the docs and the Qt5 course to fill in the details.
 - . Improvise.

PageIndicator and SwipeView

XMPP ARCHITECTURE

Every node on the Xmpp network has an address. Client addresses, also called JIDs (Jabber IDs) look like : `user@server.com`

The server address is the part to the right of the @ symbol in a JID

Xmpp is a federated protocol : This means that you can send messages between several completely independent servers.

FEDERATION

In our architecture, `user1@server1.com` can send a message to `user1@server3.com`

server2.com will be smart enough to know that the message is not his, and forward it to server3.com. server3.com will deliver the message to the connected user

A more complete JID is of the form : `user@server.com/resource`

The resource part in a JID identifies the device you are using to connect to your XMPP server (pc1, pc2)

In Xmp, you can use one account `user1@server1.com` to connect from multiple devices (pc1, pc2)

- A JID of the form `user1@server1.com` is called a bare JID.
- A JID of the form `user1@server1.com/ps` is called a full JID

Xmpp allows us to exchange chunks of XML data between clients and servers

Before the exchange of XML chunks, an XMPP client has to

- Open a long lived TCP connection to the server
- Inside the TCP connection, negotiate an XML stream to the server

When the server accepts,

- It also opens an XML stream to the client

We have two streams in total : a stream from client to server (CS) and from server to client (SC)

SwipeView

```
SwipeView{
    id : swipeViewId
    Image {
        id : image1
        fillMode: Image.PreserveAspectFit
        //source: "https://www.learnqt.guide/images/qt_quick_fundamentals.png"
        source:"qrc:/images/1.png"
    }
    Image {
        id : image2
        fillMode: Image.PreserveAspectFit
        //source: "https://www.learnqt.guide/images/qt_quick_intermediate.png"
        source: "qrc:/images/2.png"
    }
    //...
}
```

PageIndicator

```
PageIndicator{  
    id : pageIndicatorId  
    anchors.bottom: parent.bottom  
    anchors.horizontalCenter: parent.horizontalCenter  
  
    //Page indicator specific properties  
    currentIndex: swipeViewId.currentIndex  
    interactive: true  
    count : swipeViewId.count  
}
```

All together

```
import QtQuick
import QtQuick.Window
import QtQuick.Controls

Window {
    visible: true
    width: 640
    height: 480

    SwipeView{
        id : swipeViewId
        //...
    }
    PageIndicator{
        //...
    }
}
```

CMake

```
find_package(Qt6 6.2 COMPONENTS Quick QuickControls2 REQUIRED)
...
target_link_libraries(app2-Button
    PRIVATE Qt6::Quick Qt6::QuickControls2)
```