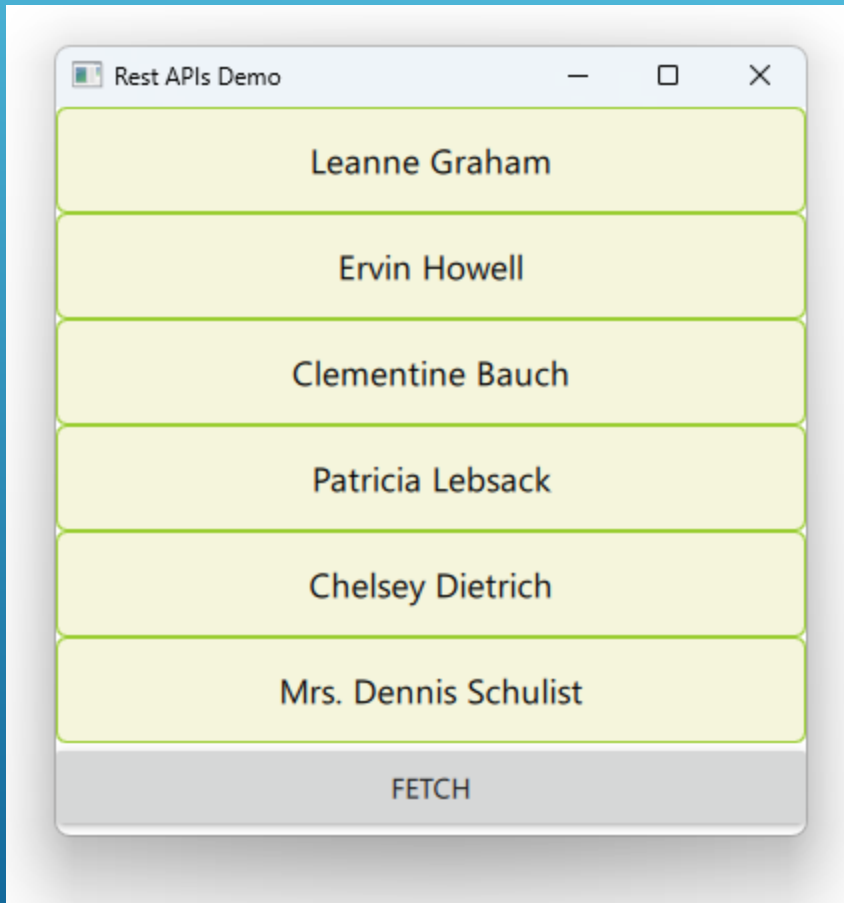


Notes to self

- . Getting data from a remote rest api
- . We use : `https://jsonplaceholder.typicode.com/users`
- . The fetching of the data is exactly the same as in the last lecture
- . Once we get the data we parse it into json :
 - . `var object = JSON.parse(response)`
- . And loop through the json array we get from the internet
- . Adding json objects to the listmodel :
 - . `//Loop over the json object capturing joke data`
`object.forEach(function(userdata){`
`mListModelId.append({"userdata" : userdata.name})`
`})`
- . Improvise.

XMLHttpRequest



Fetch Data through callback

```
function fetchData( url , callback){
    var xhr = new XMLHttpRequest();
    xhr.onreadystatechange = function() {
        if (xhr.readyState === XMLHttpRequest.HEADERS_RECEIVED) {
            print('HEADERS_RECEIVED');
        } else if (xhr.readyState === XMLHttpRequest.DONE) {
            print('DONE')
            if ( xhr.status == 200)
            {
                console.log("resource found" + xhr.responseText.toString())
                callback(xhr.responseText.toString())
            } else
            {
                callback(null)
            }
        }
    }
    xhr.open("GET",url );
    xhr.send();
}
```

Trigger the API call

```
Button{
    id : buttonId
    Layout.fillWidth: true
    text : "Fetch"
    onClicked: {
        fetchData("https://jsonplaceholder.typicode.com/users", function(response){
            if ( response){
                //Parse the data
                //Turn the returned JSON string into a JSON object that you can parse
                var object = JSON.parse(response)

                //Loop over the json object capturing joke data
                object.forEach(function(userdata){
                    mListModelId.append({"userdata" : userdata.name})
                })
            }else{
                console.log("Something went wrong")
            }
        })
    }
}
```

ListView and ListModel

```
ListModel{  
    id : mListModelId  
}  
  
ListView{  
    id : mListViewId  
    model: mListModelId  
    delegate: delegateId  
    Layout.fillWidth: true  
    Layout.fillHeight: true  
}
```

Delegate

```
Component {  
    id : delegateId  
    Rectangle {  
        id : rectangleId  
        color: "beige"  
  
        Text {  
            id : textId  
            anchors.centerIn: parent  
            text : userdata //Or modelData  
            //text : modelData  
        }  
    }  
}
```

CMake

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