JavaFX Now and Beyond, Desktop to Mobile

Paul Anderson
Gail Anderson
Anderson Software Group, Inc.

asgteach.com

© 2020 Anderson Software Group
So Who Are We?

- Training Company  
  - Java, JavaFX Courses

- JavaFX Authors  
  - Definitive Guide to JavaFX  
  - JavaFX Rich Client Programming on the NetBeans Platform

- LiveLesson Videos  
  - JavaFX Programming  
  - Java Reflection

© 2020 Anderson Software Group
 Agenda

- Why JavaFX on Mobile?
- Gluon Framework
- Gluon Charm Controls
- Gluon Connect and CloudLink
- Gluon Substrate and GraalVM
- Oracle Cloud Database
- Oracle REST Data Services
- Wrap Up, Q & A
Why JavaFX On Mobile?

- Critical Goal
  - Platform independent source code
  - “Write Once, Install Everywhere”

- Design Approach
  - Frameworks are a must
  - Hide platform dependencies and messy details

- JavaFX Advantages
  - Java UI, scene graph, nodes, FXML views
  - Properties, listeners, binding, event handlers
Gluon Framework
Gluon Substrate

Java Application

Dependencies (via Maven/Gradle)

Gluon Substrate (And Client plugins)

- Runs on Win/Linux/Mac
- Native image for desktop
- Native image for mobile/embedded

Java SDK OpenJDK static build

JavaFX SDK OpenJFX static build

GraalVM Native-image AOT

Gluon Mobile Mobile UI Device access
Gluon Mobile

- Development Tools
  - Scene Builder for layouts, CSS and skins
  - Maven for builds, IDE independent

- Gluon Library
  - Charm Controls, Material Design Guidelines
  - Local and cloud storage
  - Gluon Maps

- Hardware Control
  - Camera, accelerometer, GPS, gestures, …
Mobile App Structure

- MobileApplication
  - Main class for JavaFX mobile applications
  - Extends JavaFX Application class
  - Specify views as factories that are called on demand

- Views
  - View class invokes FXMLLoader for FXML
  - Presenter class is the FXML controller class

- Resources
  - fxml, css, images, licensing files
Gluon Charm: Look and Feel
Build and Run with Maven

- Configuration File
  ```
  pom.xml
  ```

- Desktop Native Target
  ```
  $ mvn client:build
  $ mvn client:run
  ```

- IOS Native Target
  ```
  $ mvn -Pios client:build
  $ mvn -Pios client:run
  ```
Gluon Connect

- **Client Side Library**
  - Maps data with observable properties and lists
  - Supports bidirectional data transfers
  - Provides notifications
  - Syncs data automatically

- **Supports Common Data Sources**
  - Gluon CloudLink
  - File provider
  - REST provider
Gluon Cloud Demo

BP Cloud

<table>
<thead>
<tr>
<th>Date</th>
<th>Blood Pressure</th>
<th>Heart Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sep 15, 2017</td>
<td>104/80 mmHg</td>
<td>65 bpm</td>
</tr>
<tr>
<td>May 3, 2017</td>
<td>134/80 mmHg</td>
<td>61 bpm</td>
</tr>
<tr>
<td>Apr 28, 2017</td>
<td>121/76 mmHg</td>
<td>55 bpm</td>
</tr>
<tr>
<td>Apr 28, 2017</td>
<td>120/80 mmHg</td>
<td>60 bpm</td>
</tr>
<tr>
<td>Apr 26, 2017</td>
<td>124/85 mmHg</td>
<td>61 bpm</td>
</tr>
<tr>
<td>Oct 5, 2015</td>
<td>121/77 mmHg</td>
<td>58 bpm</td>
</tr>
<tr>
<td>Oct 4, 2015</td>
<td>115/75 mmHg</td>
<td>57 bpm</td>
</tr>
<tr>
<td>-</td>
<td>127/78 mmHg</td>
<td>58 bpm</td>
</tr>
</tbody>
</table>
Oracle Database
# People Demo

<table>
<thead>
<tr>
<th>Dept</th>
<th>Name</th>
<th>Position</th>
<th>Hire Date</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>CLARK</td>
<td>MANAGER</td>
<td>1981-06-09</td>
<td>3000</td>
</tr>
<tr>
<td></td>
<td>Harry</td>
<td>Wizard</td>
<td>2018-10-17</td>
<td>4200</td>
</tr>
<tr>
<td></td>
<td>Hermione</td>
<td>Wizard</td>
<td>2018-10-12</td>
<td>4500</td>
</tr>
<tr>
<td></td>
<td>KING</td>
<td>PRESIDENT</td>
<td>1981-11-17</td>
<td>3000</td>
</tr>
<tr>
<td></td>
<td>MILLER</td>
<td>CLERK</td>
<td>1982-01-23</td>
<td>1300</td>
</tr>
<tr>
<td>20</td>
<td>FORD</td>
<td>ANALYST</td>
<td>1981-12-02</td>
<td>3000</td>
</tr>
<tr>
<td></td>
<td>JONES</td>
<td>MANAGER</td>
<td>1981-04-02</td>
<td>2975</td>
</tr>
<tr>
<td></td>
<td>SCOTT</td>
<td>ANALYST</td>
<td>1987-04-19</td>
<td>3000</td>
</tr>
</tbody>
</table>
## Remote Functions with REST

<table>
<thead>
<tr>
<th>REST Method</th>
<th>Remote Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>getEmployees()</td>
</tr>
<tr>
<td>GET</td>
<td>getEmployee()</td>
</tr>
<tr>
<td>POST</td>
<td>createEmployee()</td>
</tr>
<tr>
<td>PUT</td>
<td>updateEmployee()</td>
</tr>
<tr>
<td>DELETE</td>
<td>deleteEmployee()</td>
</tr>
</tbody>
</table>
API Dashboard

API Management

Remote Functions

Function Information
- **Type**: HTTP Request
- **Name**: getEmployee
- **Internal only**: false
- **Method**: GET
- **End Point**: http://54.217.216.239/ords/pdb1/myuser1/testmodule9/emp/$empno
- **Request Body**: none

Function Parameters

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Value</th>
<th>Test Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable param</td>
<td>empno</td>
<td>7521</td>
<td>7369</td>
</tr>
</tbody>
</table>
Remote Functions

- GET Method Definition

```java
public class RemoteService {
    public <T> GluonObservableObject<T>
        getEmployees(Class<T> clazz) {
        RemoteFunctionObject function =
            RemoteFunctionBuilder
                .create("getEmployees")
                .object();
        return function.call(clazz);
    }
    // other remote functions...
}
```
Remote Functions

- Dependency Injection

```java
@Inject
private RemoteService remoteService;

@FXML
private
CharmListView<Employee, Integer> charmListView;

private
GluonObservableObject<WorkForce> workforce;

private ChangeListener<? super Boolean>
listener = null;
```
Remote Functions

GET Method Invocation

```java
workforce = remoteService.getEmployees(WorkForce.class);
...
workforce.initializedProperty().addListener(
    listener = (ObservableValue<? extends Boolean> obsrv, Boolean ov, Boolean nv) -> {
        if (nv) {
            charmListView.setItems(FXCollections.observableArrayList(
                workforce.get().getItems()));
        }
    });
```
Remote Functions

DELETE Method Definition

```java
public class RemoteService {
    public <T> GluonObservableObject<T> deleteEmployee(Class<T> clazz, String value) {
        RemoteFunctionObject function = RemoteFunctionBuilder.create("deleteEmployee")
            .param("empno", value)
            .object();
        return function.call(clazz);
    }
}
```

Summary

- **JavaFX Advantages**
  - Platform independent source code
  - Observables, binding & background tasks to sync UI
  - Flexible skinning to fit mobile form

- **Useful Frameworks**
  - Gluon/Maven framework for mobile deployment
  - Based on OpenJDK and OpenJFX
  - Afterburner framework for dependency injection
  - Oracle Cloud Database and ORDS
Wrap Up

Thanks for Coming!
- paul@asgteach.com
- gail@asgteach.com

Source Code
- asgteach.com
  - DEVNEXUS 2020 Code
    - Click to Download
- Q & A

@paul_asgteach
@gail_asgteach