Quick & Easy Desktop Development with NetBeans and its HTML/JAVA API

Ioannis (John) Kostaras
FOSDEM 2-3 February 2019
Context

➢ (Apache) NetBeans
➢ Rich Client Platform
➢ Desktop Applications
Prerequisites

➢ (Apache) NetBeans 8.2 or later
➢ JDK 8
Apache NetBeans

➢ An integrated development environment
➢ Mainly for the Java programming language
➢ Support for many other programming languages:
  ▪ Groovy/Grails, PHP, Python, Ruby/Rails, HTML5/CSS, JavaScript, Scala, C/C++, …
➢ Support for a plethora of version control systems:
  ▪ Git, Mercurial, Subversion, …
NetBeans Rich Client Platform

➢ A platform to develop desktop applications
➢ NetBeans IDE is based on NetBeans RCP

NetBeans IDE

NetBeans Platform

Extras: Visual Library, Palette

GUI
Action System, Window System, Nodes, Explorer Views

Core
Module System, FileSystem, Lookup, Startup
Window System, Explorer Views

➢ Based on Java Swing

<table>
<thead>
<tr>
<th>NetBeans RCP</th>
<th>Java Swing</th>
</tr>
</thead>
<tbody>
<tr>
<td>TopComponent</td>
<td>JFrame</td>
</tr>
<tr>
<td>OutlineView</td>
<td>JTable</td>
</tr>
<tr>
<td>BeanTreeView</td>
<td>JTree</td>
</tr>
<tr>
<td>ListView</td>
<td>JList</td>
</tr>
<tr>
<td>ChoiceView</td>
<td>JComboBox</td>
</tr>
<tr>
<td>IconView</td>
<td>-</td>
</tr>
</tbody>
</table>
JavaFX Integration

➢ Add JavaFX content to the `TopComponent` using Swing component `JFXPanel`

➢ The `JFXPanel` component is a Swing `JComponent` specifically implemented to embed JavaFX content in a Swing application. `JFXPanel` starts up the JavaFX runtime for you.

➢ It also transparently forwards all input (mouse, key) and focus events to the JavaFX runtime.

➢ It allows both Swing and JavaFX to run concurrently.
public final class MyTopComponent extends TopComponent {
    private static JFXPanel fxPanel;
    private void init() {
        fxPanel = new JFXPanel();
        add(fxPanel, BorderLayout.CENTER);
        Platform.setImplicitExit(false);
        Platform.runLater(() -> createScene());
    }
    private void createScene() {
        try {
            Parent root =
                FXMLLoader.load(getClass().getResource("MyJavaFX.fxml"));
            Scene scene = new Scene(root, Color.LIGHTBLUE);
            fxPanel.setScene(scene);
        } catch (IOException ex) {
            Exceptions.printStackTrace(ex);
        }
    }
}
JavaFX – TopComponent Interaction

NetBeans Platform

TopComponent

JavaFX Controller

JFXPanel

Public methods

FXML

Scene Graph

FXML Loader

FXML Loader

HTML/JAVA UI
Portable UI (HTML 5)

basic building blocks and advanced high level concepts for communication between JavaScript and Java

Based on

Dukescr ipt
What is DukeScript

➢ A new technology for creating cross-platform mobile, desktop and web applications.
➢ Allows you to write your logic in Java and render the result to a number of clients, which can be web browser, portable devices etc.
➢ DukeScript applications are plain Java applications that internally use HTML5 technologies and JavaScript for rendering.
➢ This way developers only need to write clean Java code and can still leverage the latest developments in modern UI technology.
How does it work

HTML 5 Renderer

DukeScript

JVM

android.webkit.WebView

DukeScript

dalvik

HTML 5 Browser

DukeScript

bck2brwsr

A JVM implemented in JavaScript

HTML/JAVA UI
Pros & Cons

+ Write in Java
+ Write once run everywhere (web, JavaFX, Android, iOS, …)
+ API similar to JavaFX
- Not a lot of documentation available
- Need to learn a new API
Technologies to master

- HTML(5)
- CSS(3)
- JavaScript
- Knockout.js
- DukeScript
- Model-View-ViewModel (MVVM)
Model-View-ViewModel (MVVM)
Sample application

- DukeScript (HTML/JAVA UI) has a clean separation of design and development.
- With DukeScript it is possible to completely outsource the UI design to a designer with no knowledge of DukeScript, or a specific set of tools.
- DukeScript uses HTML5 for the framework's UI and there are plenty of tools to build HTML UIs with the help of CSS and it is a well known technology to UI designers.
@Model(className = "Tasks", targetType = ",", properties = {
  @Property(name = "text", type = String.class)
})

public final class TasksCntrl {
  @ComputedProperty
  static String templateName() {
    return "window";
  }
...}

<div data-bind="template: templateName"></div>
<script type="text/html" id="window">
  <input data-bind="value: text"></input>
  <button data-bind="click: showDialog, enable: text">Ask!</button>
</script>
Binding Contexts

Properties available only in View:

- `$root`: refers to the top-level ViewModel
- `$data`: refers to the ViewModel object of the current context (can be omitted)
- `$parent`: refers to the parent ViewModel object (useful for nested loops)
- `$index`: contains the current item's index in the array

https://knockoutjs.com/documentation/binding-context.html
Annotations for the ViewModel

- **ObservableArrays:**
  ```java
  @Property(name = "tasks", type = Task.class, array = true)
  <div data-bind="foreach: tasks" >
  
  @ComputedProperty: observable properties derived from other properties:
  ```
  ```java
  @ComputedProperty public static static int
  numberOfTasksWithAlert(List<Task> tasks) {
  return listTasksWithAlert(tasks).size();
  }
  ```
Annotations for the ViewModel

➢ @Function:

```java
@Function public static void removeTask(Tasks tasks, Task data) {
    tasks.getTasks().remove(data);
}
```
## A Todo application (main Tasks window)

A Todo application (main Tasks window) is a GUI application that displays the user's tasks. The tasks are displayed in a table format with columns for priority, description, alert status, and due date.

<table>
<thead>
<tr>
<th>Priority</th>
<th>Description</th>
<th>Alert?</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Finish TodoDS article!</td>
<td>true</td>
<td>2017-03-10</td>
</tr>
<tr>
<td>10</td>
<td>Book conference room!</td>
<td>false</td>
<td>2017-04-01</td>
</tr>
</tbody>
</table>

There are 1 task(s) with alerts today.
A Todo application (Create/Edit Task window)
References

➢ Dukescript


➢ Kostaras I. (2016), *TodoDS*

➢ Kostaras I. (2015), *Port Your Java Applets*


➢ https://bits.netbeans.org/10.0/javadoc/

➢ http://137.254.56.27/html+java/1.6/index.html
Q&A

HTML/JAVA UI