

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

Ioannis (John) Kostaras
FOSDEM 2-3 February 2019



Context

HTML/JAVA UI

- (Apache) NetBeans
- Rich Client Platform
- Desktop Applications



Prerequisites

- (Apache) NetBeans 8.2 or later
- JDK 8

Apache NetBeans

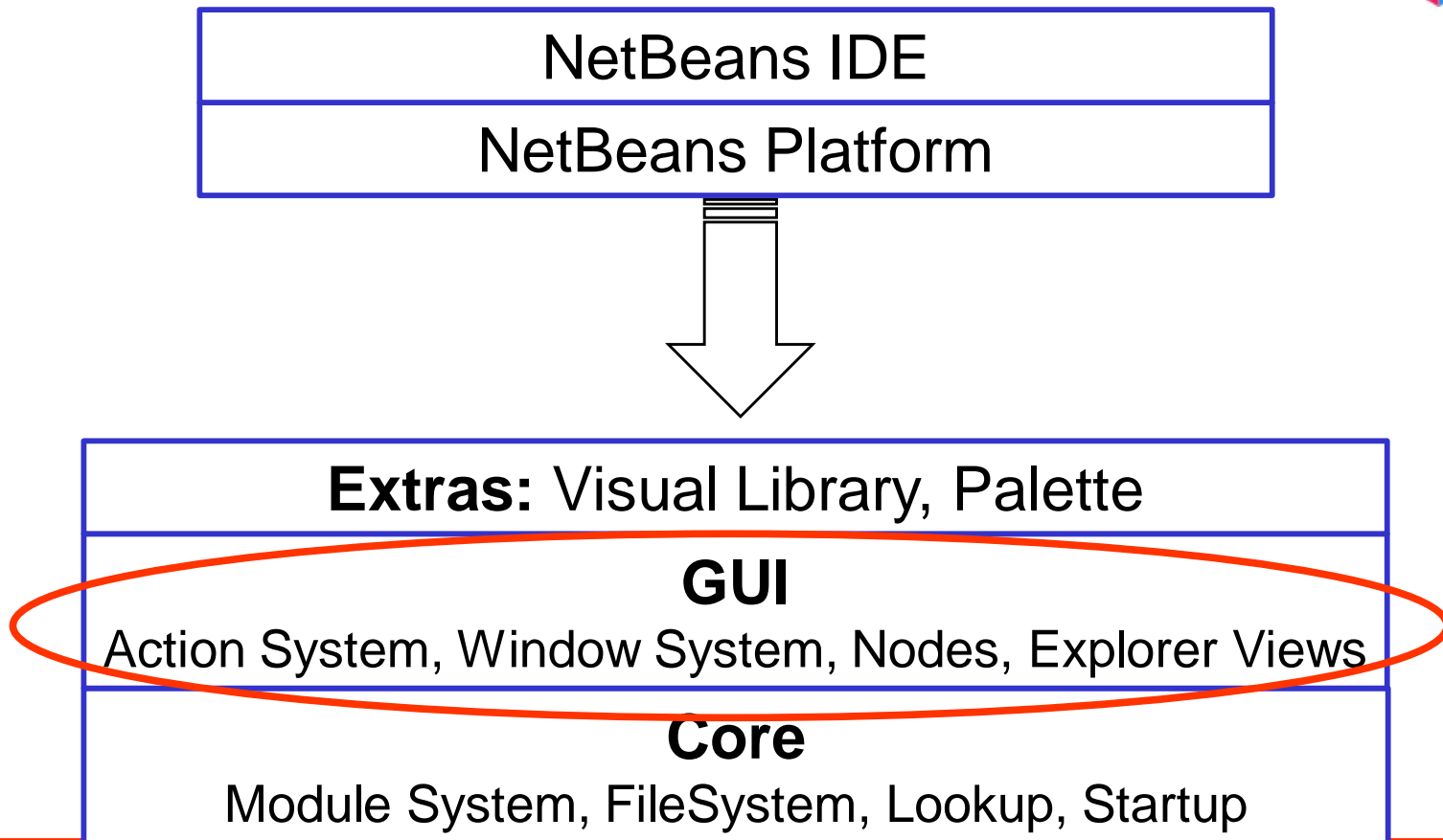
- An integrated development environment
- Mainly for the Java programming language
- Support for many other programming languages:
 - Groovy/Grails, PHP, Python, Ruby/Rails, HTML5/CSS, JavaScript, Scala, C/C++, ...
- Support for a plethora of version control systems:
 - Git, Mercurial, Subversion, ...



NetBeans Rich Client Platform

HTML/JAVA UI

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



Window System, Explorer Views

➤ Based on Java Swing

NetBeans Window System / Explorer Views

Java Swing

NetBeans RCP	Java Swing
TopComponent	JFrame
OutlineView	JTable
BeanTreeView	JTree
ListView	JList
ChoiceView	JComboBox
IconView	-

- 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.

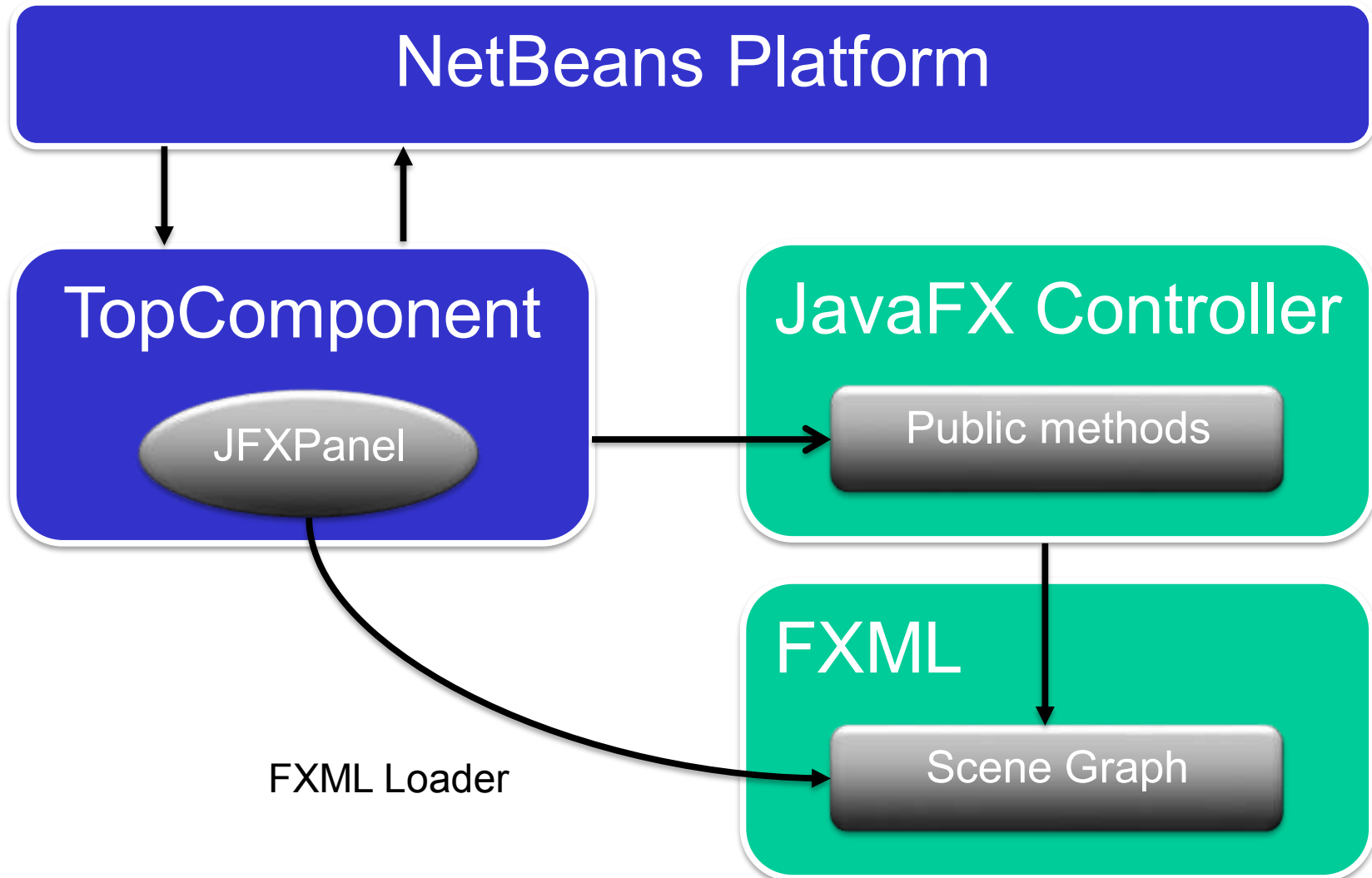
JavaFX Integration

LITMI / IAN / A III

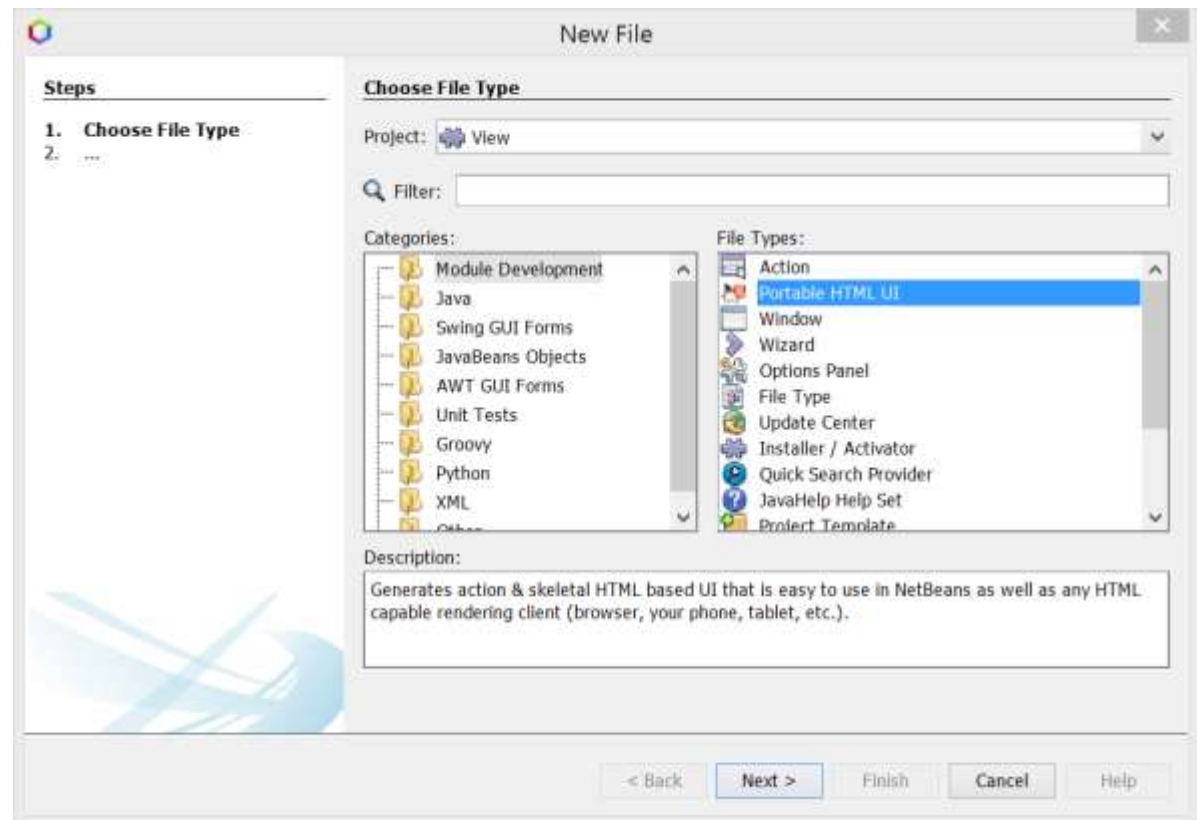
```
public final class MyTopComponent extends TopComponent {
    private static JFXPanel fxPanel;
    private void init() {
        fxPanel = new JFXPanel();
        add(fxPanel, BorderLayout.CENTER);
        Platform.setImplicitExit(false);
        Platform.runLater(() -> createScene());
    }
    private 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

HTML/JAVA UI



- Portable UI (HTML 5)
- basic building blocks and advanced high level concepts for communication between JavaScript and Java
- Based on
- Dukescript



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/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/JAVA UI

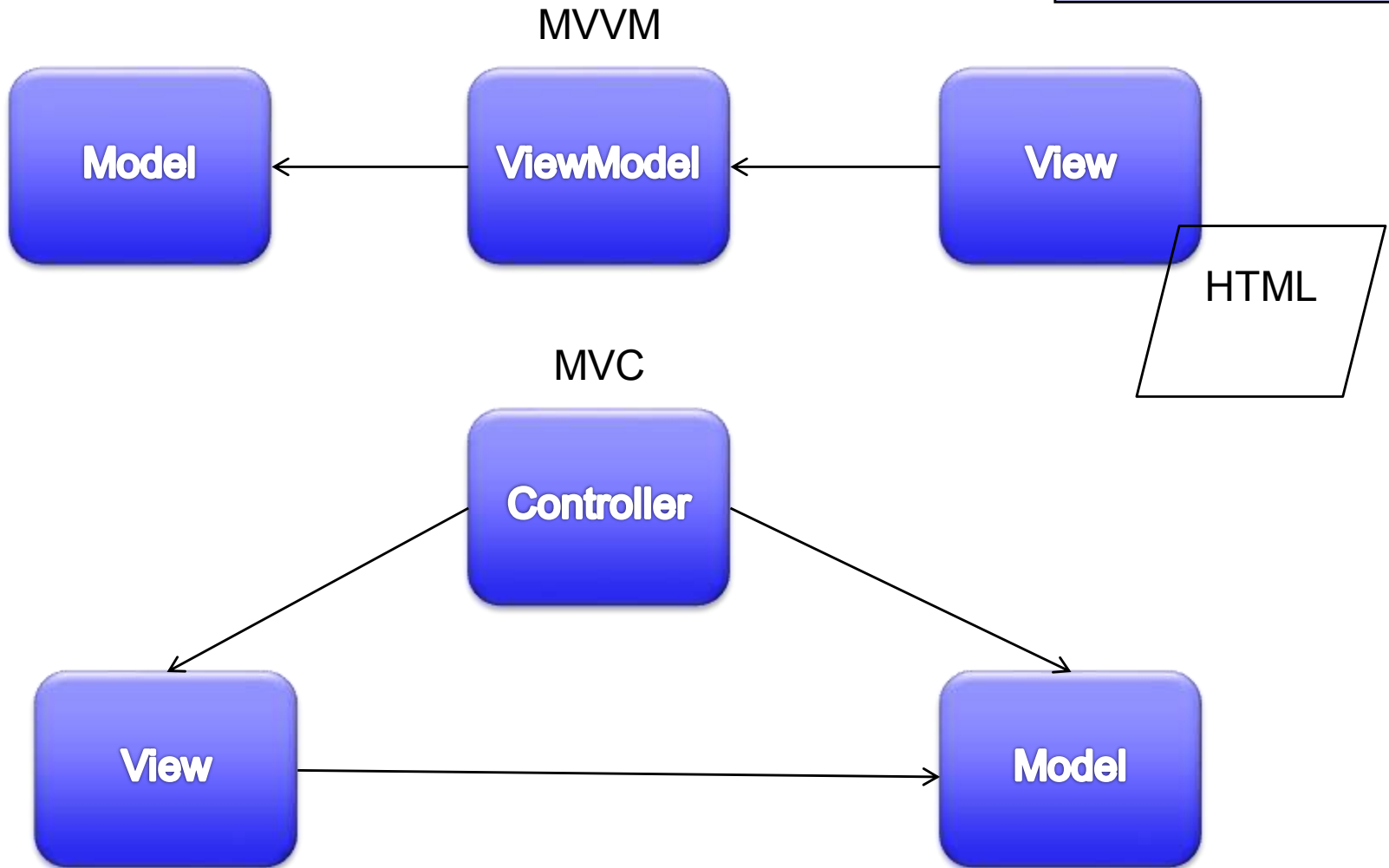
- HTML(5)
- CSS(3)
- JavaScript
- Knockout.js
- DukeScript
- Model-View-ViewModel (MVVM)

databinding



Model-View-ViewModel (MVVM)

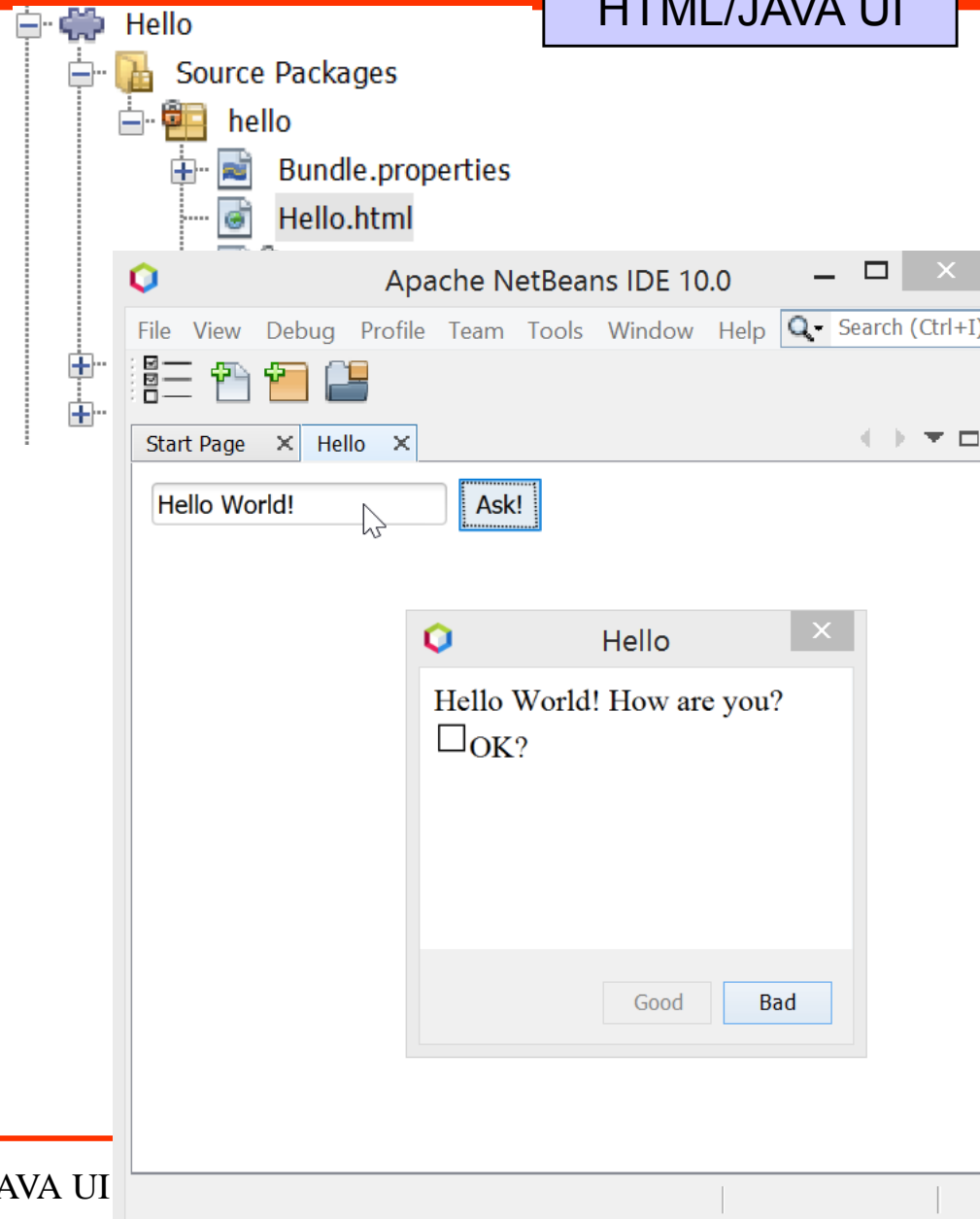
HTML/JAVA UI



Sample application

HTML/JAVA UI

- 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.



Code

```
@Model(className = "Tasks", targetId = "", 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>
```

- 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`:

```
@Property(name = "tasks", type =  
Task.class, array = true)
```

```
<div data-bind="foreach: tasks" >
```

- `@ComputedProperty`: observable properties derived from other properties:

```
@ComputedProperty public static int  
numberOfTasksWithAlert(List<Task> tasks) {  
return listTasksWithAlert(tasks).size();  
}
```

Annotations for the ViewModel

➤ @Function:

```
@Function public static void removeTask(Tasks
tasks, Task data) {
    tasks.getTasks().remove(data);
}
```

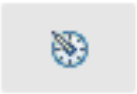
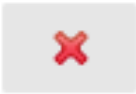
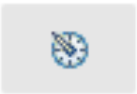
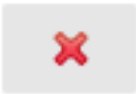
A Todo application (main Tasks window)

HTML/JAVA UI

TasksWindow ×

Tasks



Priority	Description	Alert?	Due Date	
5	Finish TodoDS article!	true	2017-03-10	  <input type="checkbox"/>
10	Book conference room!	false	2017-04-01	  <input type="checkbox"/>

There are 1 task(s) with alerts today.

A Todo application (Create/Edit Task window)

TasksWindow

Tasks

Create/Edit Task

Description: Finish TodoDS article!

Priority: 5

Due Date: 2017-03-10

Show alert: 0 days before

Obs: Obs

Completed Task

Save Clear Cancel

HTML/JAVA UI

References

- [Dukescript](#)
- Lozano F. (2006), "A complete App using NetBeans 5", NetBeans Magazine, Issue 1, May,
http://netbeans.org/download/magazine/01/nb01_completeapp.pdf
- Epple A. (2016), *Java everywhere: Write Once Run Everywhere with DukeScript*, [LeanPub](#).
- Epple A. (2015), "Java Everywhere: Write Once Run Anywhere with DukeScript", [JavaCodeGeeks](#).
- Epple A. (2015), "[Common Misconceptions about DukeScript](#)".
- Kostaras I. (2016), [TodoDS](#)
- Kostaras I. (2015), [Port Your Java Applets](#)
- Hodson R. (2012), *Knockout.js Succintly*, Syncfusion.
- <https://bits.netbeans.org/10.0/javadoc/>
- <http://137.254.56.27/html+java/1.6/index.html>

Q&A

