


# CaptainCasa

## XI. Community Meeting





# Partner

  
 greenfield-solutions GmbH

[Startseite](#) | [CaptainCasa](#) | [Kontakt](#) | [Über mich](#) | [Impressum](#)

Herzlich Willkommen!

In der greenfield-solutions GmbH haben Sie einen Partner, der Ihre Ideen und Anforderungen an hochwertige und komplexe Lösungen mit Fachkenntnis, Erfahrung und Herzblut umsetzt und den Betrieb in Ihrem Haus begleitet.

Als Service Partner der CaptainCasa GmbH arbeite ich mit dem seit Jahren etablierten JSF-Framework und bin aktives Mitglied innerhalb der Community.  
 Sie kennen CaptainCasa noch nicht? Informieren Sie sich hier: [CaptainCasa](#)

"Die Entwicklung komplexer Java Anwendungen ist für mich kein Hexenwerk - es ist mein Handwerk!"

Roland Heinrichs  
 Geschäftsführer

Suche nach:

Copyright greenfield-solutions GmbH  
 Theme By SiteOrigin



# Partner

**Joachim Unger**

**IT-Beratung**

**Start**

**Technologien**

**Schwerpunkte**

**Über mich**

**Kontakt**

**Kunden**

**Profil (PDF)**

**Impressum**

Hier folgt eine Auflistung der mir bestens vertrauten Technologien.

**Sprachen**

- Java / J2EE
- XSL / XSLT / FO
- HTML / HTML5 / CSS / JavaScript
- JSP / JSF (myFaces, Tomahawk)
- PL/SQL
- PL/pgSQL
- C / C++ / C# / .NET

**Datenbanken**

- Oracle
- PostgreSQL
- IBM DB2
- Microsoft SQL-Server
- Adabas
- MySQL

**Data Warehouse**

- Pentaho Mondrian
- ReportServer
- JasperReports
- BIRT
- JFreeChart

**Tools**

- Eclipse

**GUI**

- Swing, JavaFX,

**Anwendungs-Frameworks**

Letzte Aktualisierung  
24.05.2014

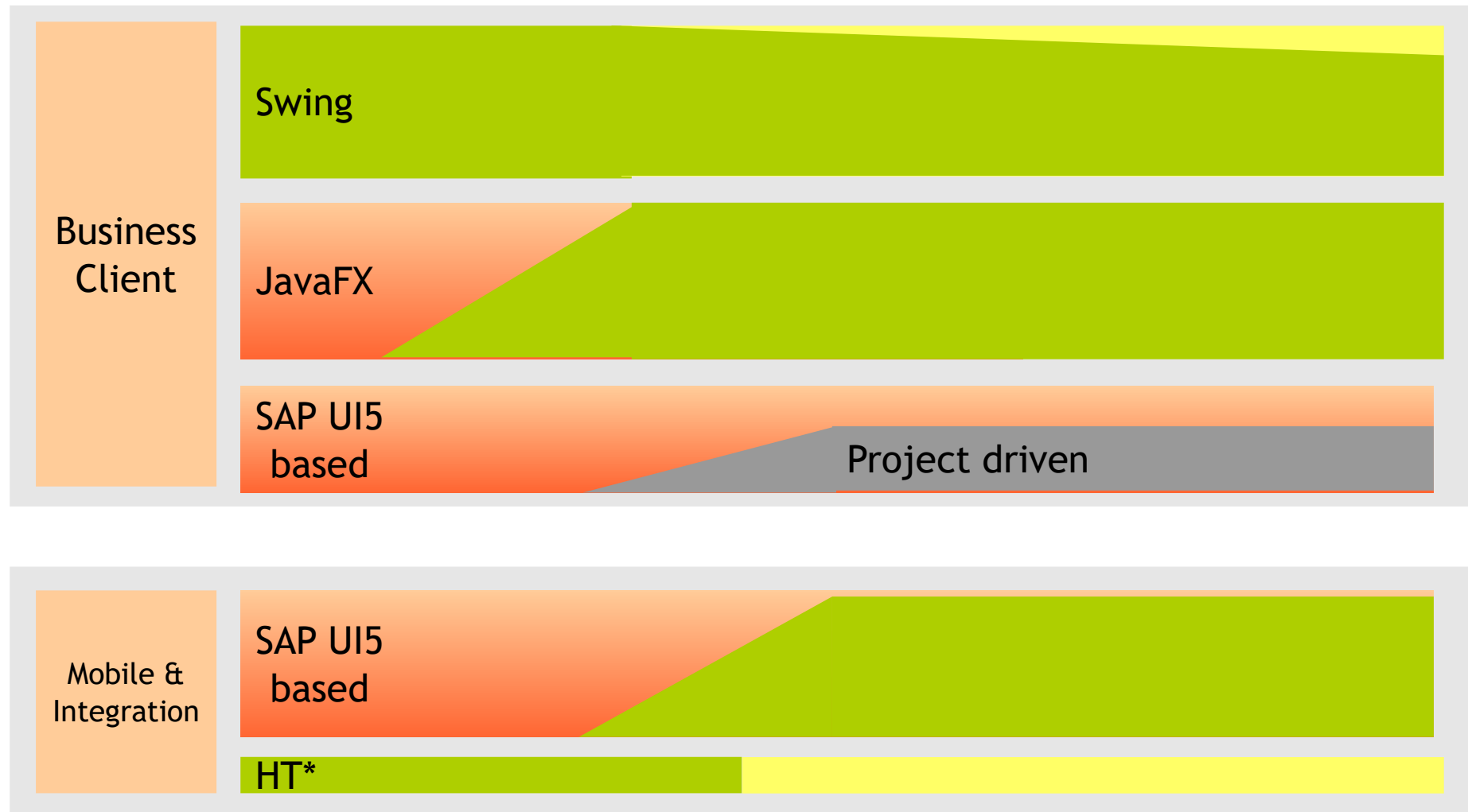
# 2014 Kein Wandertag....



# Agenda

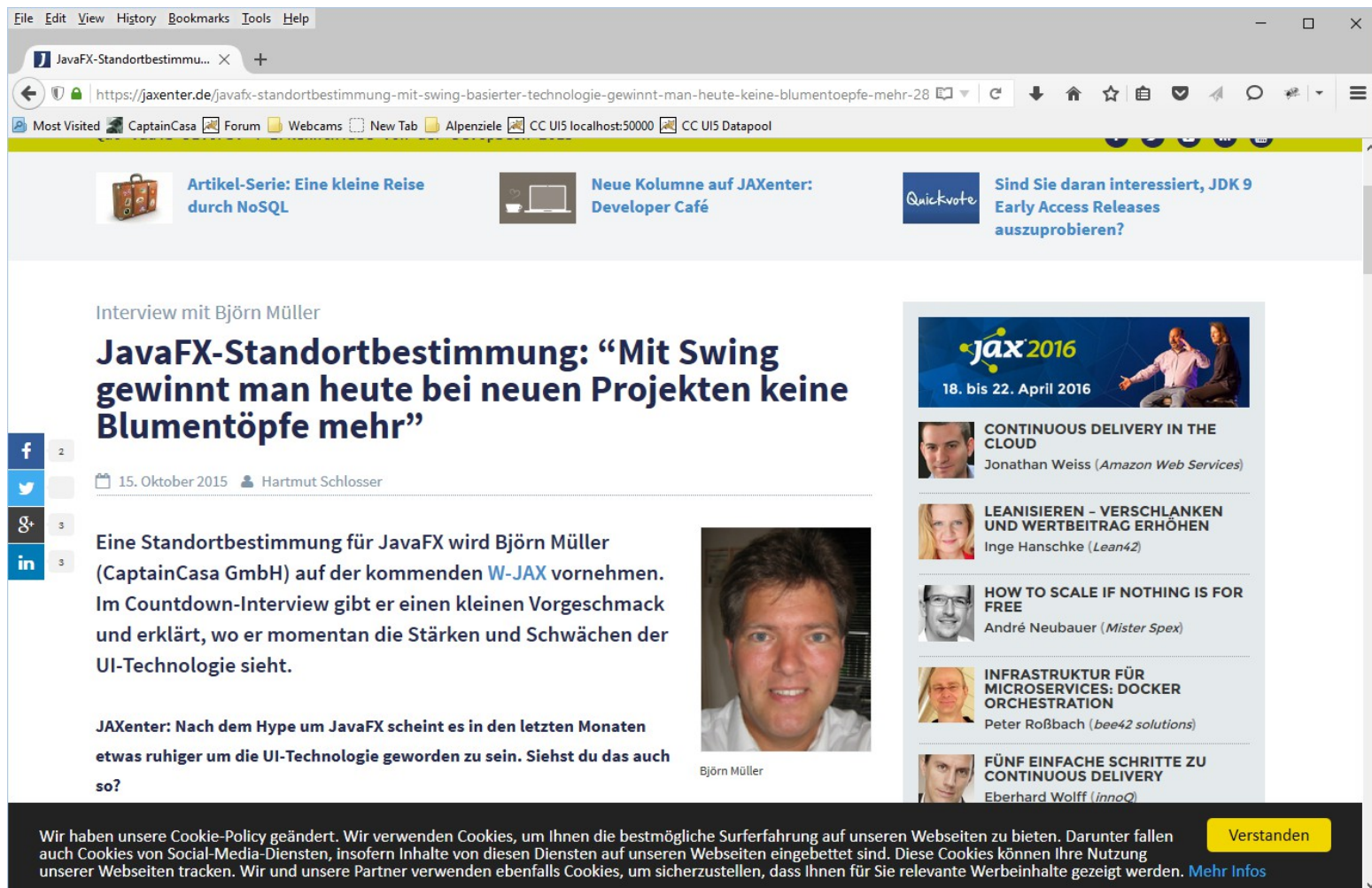
- 10:00h Standortbestimmung / Roadmap (I)
- 10:20h SAP/Open UI5
- 11:05h 2 Tips und 1 Szenario
- **11:25h Kaffeepause**
- 11:40h Page Bean Components
- 12:10h 2 Tips und 1 Szenario
- 12:30h New Components/Features 2015(I)
- **13:00h Mittagspause**
- 13:40h Ergonomisches UI Design
- 14:20h 2 Tips und 1 Szenario
- 14:40h New Components/Features 2015(II)
- 15:10h Spezialthemen (Zertifikat, SAML Portal)
- **15:30h Kaffeepause**
- 16:00h Standortbestimmung / Roadmap (II)

# Roadmap 2015





# Java Swing



The screenshot shows a web browser window with the following content:

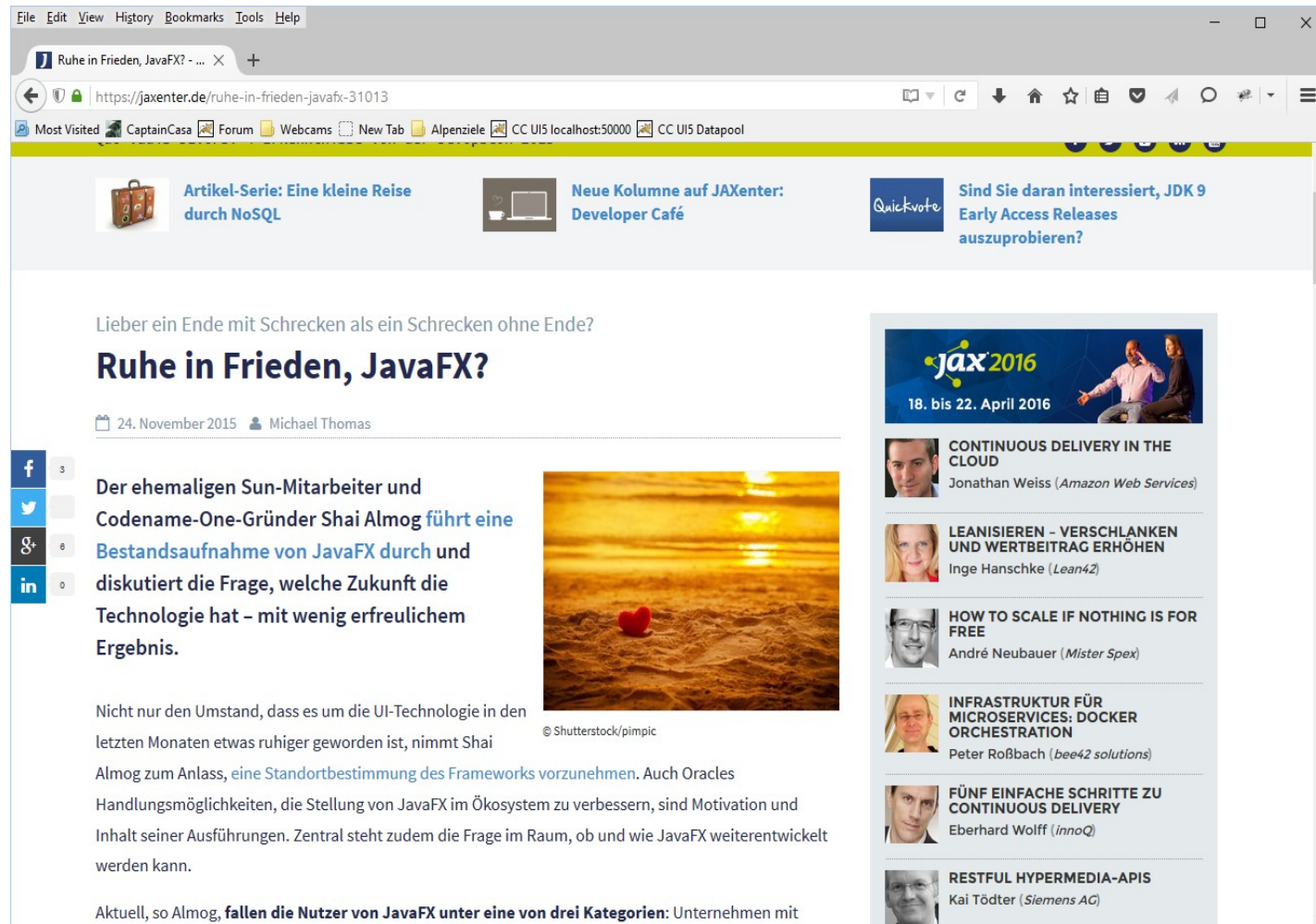
- Browser Address Bar:** <https://jaxenter.de/javafx-standortbestimmung-mit-swing-basierter-technologie-gewinnt-man-heute-keine-blumentoepe-mehr-28>
- Navigation Bar:** Includes "Most Visited" and "CaptainCasa" links.
- Article Header:**
  - Interview mit Björn Müller
  - JavaFX-Standortbestimmung: "Mit Swing gewinnt man heute bei neuen Projekten keine Blumentöpfe mehr"**
  - 15. Oktober 2015 | Hartmut Schlosser
- Article Text:**

Eine Standortbestimmung für JavaFX wird Björn Müller (CaptainCasa GmbH) auf der kommenden **W-JAX** vornehmen. Im Countdown-Interview gibt er einen kleinen Vorgeschmack und erklärt, wo er momentan die Stärken und Schwächen der UI-Technologie sieht.
- Author Photo:** A portrait of Björn Müller.
- Author Name:** Björn Müller
- Text Below Photo:**

**JAXenter:** Nach dem Hype um JavaFX scheint es in den letzten Monaten etwas ruhiger um die UI-Technologie geworden zu sein. Siehst du das auch so?
- Right Sidebar:**
  - jax2016** 18. bis 22. April 2016
  - CONTINUOUS DELIVERY IN THE CLOUD** Jonathan Weiss (*Amazon Web Services*)
  - LEANISIEREN - VERSCHLANKEN UND WERTBEITRAG ERHÖHEN** Inge Hanschke (*Lean42*)
  - HOW TO SCALE IF NOTHING IS FOR FREE** André Neubauer (*Mister Spex*)
  - INFRASTRUKTUR FÜR MICROSERVICES: DOCKER ORCHESTRATION** Peter Roßbach (*bee42 solutions*)
  - FÜNF EINFACHE SCHRITTE ZU CONTINUOUS DELIVERY** Eberhard Wolff (*InnoQ*)
- Footer:** Cookie policy notice with a "Verstanden" button.



# JavaFX




The screenshot shows a web browser window with the URL <https://jaxenter.de/ruhe-in-frieden-java-fx-31013>. The page features a navigation bar with several links: "Artikel-Serie: Eine kleine Reise durch NoSQL", "Neue Kolumne auf JAXenter: Developer Café", and "Quickvote Sind Sie daran interessiert, JDK 9 Early Access Releases auszuprobieren?". The main content area displays the article title "Ruhe in Frieden, JavaFX?" by Michael Thomas, dated 24. November 2015. The article text discusses the state of JavaFX and mentions Shai Almog. A social media sharing bar is visible on the left. On the right, there is a sidebar with a "jax 2016" event banner and a list of featured articles by various authors.

Lieber ein Ende mit Schrecken als ein Schrecken ohne Ende?

## Ruhe in Frieden, JavaFX?

24. November 2015 Michael Thomas

Der ehemaligen Sun-Mitarbeiter und Codename-One-Gründer Shai Almog führt eine Bestandsaufnahme von JavaFX durch und diskutiert die Frage, welche Zukunft die Technologie hat – mit wenig erfreulichem Ergebnis.



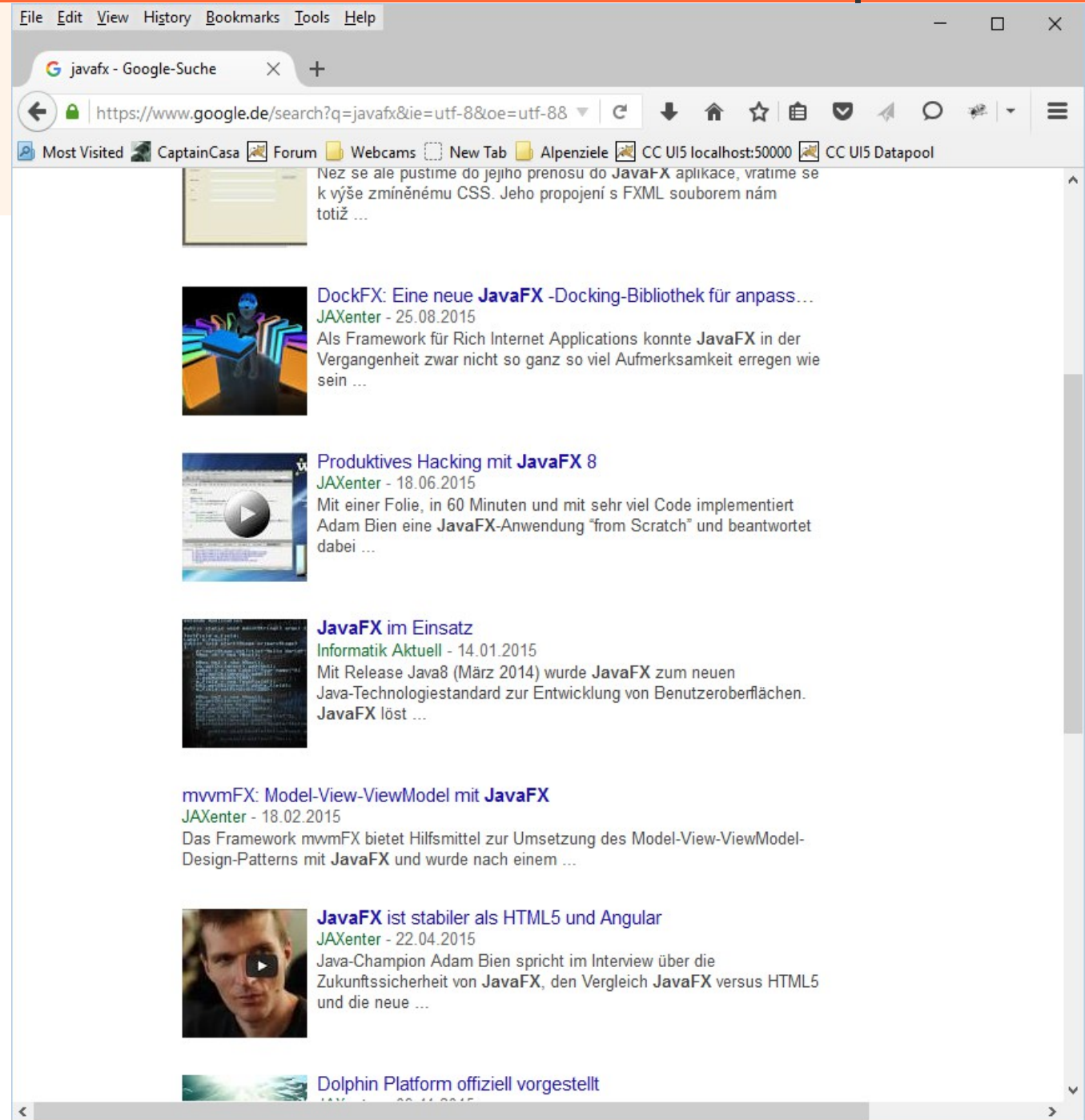
Nicht nur den Umstand, dass es um die UI-Technologie in den letzten Monaten etwas ruhiger geworden ist, nimmt Shai Almog zum Anlass, eine Standortbestimmung des Frameworks vorzunehmen. Auch Oracles Handlungsmöglichkeiten, die Stellung von JavaFX im Ökosystem zu verbessern, sind Motivation und Inhalt seiner Ausführungen. Zentral steht zudem die Frage im Raum, ob und wie JavaFX weiterentwickelt werden kann.

Aktuell, so Almog, fallen die Nutzer von JavaFX unter eine von drei Kategorien: Unternehmen mit

**jax 2016**  
18. bis 22. April 2016

- CONTINUOUS DELIVERY IN THE CLOUD**  
Jonathan Weiss (*Amazon Web Services*)
- LEANISIEREN – VERSCHLANKEN UND WERTBEITRAG ERHÖHEN**  
Inge Hanschke (*Lean42*)
- HOW TO SCALE IF NOTHING IS FOR FREE**  
André Neubauer (*Mister Spex*)
- INFRASTRUKTUR FÜR MICROSERVICES: DOCKER ORCHESTRATION**  
Peter Roßbach (*bee42 solutions*)
- FÜNF EINFACHE SCHRITTE ZU CONTINUOUS DELIVERY**  
Eberhard Wolff (*innQ*)
- RESTFUL HYPERMEDIA-APIs**  
Kai Tödter (*Siemens AG*)

# JavaFX



The screenshot shows a Google search for 'javafx' in German. The search results include several articles and videos:

- Nez se ale pustime do jejho prenosu do JavaFX aplikace, vratime se k výše zmíněnému CSS. Jeho propojení s XML souborem nám totiž ...**
- DockFX: Eine neue JavaFX -Docking-Bibliothek für anpass...**  
JAXenter - 25.08.2015  
Als Framework für Rich Internet Applications konnte JavaFX in der Vergangenheit zwar nicht so ganz so viel Aufmerksamkeit erregen wie sein ...
- Produktives Hacking mit JavaFX 8**  
JAXenter - 18.06.2015  
Mit einer Folie, in 60 Minuten und mit sehr viel Code implementiert Adam Bien eine JavaFX-Anwendung "from Scratch" und beantwortet dabei ...
- JavaFX im Einsatz**  
Informatik Aktuell - 14.01.2015  
Mit Release Java8 (März 2014) wurde JavaFX zum neuen Java-Technologiestandard zur Entwicklung von Benutzeroberflächen. JavaFX löst ...
- mvmFX: Model-View-ViewModel mit JavaFX**  
JAXenter - 18.02.2015  
Das Framework mvmFX bietet Hilfsmittel zur Umsetzung des Model-View-ViewModel-Design-Patterns mit JavaFX und wurde nach einem ...
- JavaFX ist stabiler als HTML5 und Angular**  
JAXenter - 22.04.2015  
Java-Champion Adam Bien spricht im Interview über die Zukunftssicherheit von JavaFX, den Vergleich JavaFX versus HTML5 und die neue ...
- Dolphin Platform offiziell vorgestellt**  
JAXenter - 20.11.2015

# HTML - JavaFX

- JavaFX
  - Much UI, „few“ users
- HTML5
  - (Much) UI, „many“ users



# HTML - JavaFX

- JavaFX
  - Browser is not relevant
- HTML5
  - Browser is my constant enemy





# HTML - JavaFX

- JavaFX
  - „Complex“ Installation
  - „Zero“ Maintenance
- HTML5
  - „Zero“ Installation
  - „Complex“ Maintenance



# SAP / Open UI5



# SAP UI5

- Client side JavaScript library
  - SAP as developer
  - ...open sourced („9x%“)
- Commitment
  - SAP message
  - SAP applications

The screenshot shows the SAPUI5 Developer Guide website. The page title is "SAPUI5: UI Development Toolkit for HTML5". The navigation menu includes "DEVELOPER GUIDE", "CONTROLS", "EXPLORED", "API REFERENCE", "DEMO APPS", and "ICONS". The main content area features a search bar, a table of contents, and a "Note" section. At the bottom, there are three buttons: "Read Me First", "Tutorials", and "Developing Apps".

UI development toolkit for HTML5 - Demo Kit

Legal Disclosure | Privacy | Terms of Use | About | 1:

DEVELOPER GUIDE CONTROLS EXPLORED API REFERENCE DEMO APPS ICONS

Filter

What's New in SAPUI5

SAPUI5: UI Development Toc

- Read Me First
- Tutorials
- Essentials
- Developing Apps
- Extending Apps
- Developing Content
- Glossary
- Analysis Path Framework (AF)

## SAPUI5: UI Development Toolkit for HTML5

You can use the UI development toolkit for HTML5 (SAPUI5) to create apps with rich user interfaces for modern Web business applications.

Apps developed with SAPUI5 present one consistent user experience. They are responsive across browsers and devices - they run on smartphones, tablets, and desktops. The UI controls automatically adapt themselves to the capabilities of each device.

SAPUI5 offers powerful development concepts:

- Built-in extensibility concepts at code and application level
- Data binding types and Model-View-Controller (MVC)
- Feature-rich UI controls for handling complex UI patterns and predefined layouts for typical use cases.
- Full translation support
- Keyboard interaction support and accessibility features

And many more...

SAP Fiori applications are built with SAPUI5. SAP Fiori apps follow the SAP Fiori Design Guidelines to ensure consistent design and a high level of design quality. Explore the guidelines: <https://experience.sap.com/fiori-design/>.

For more information about SAP Fiori, see <http://www.sap.com/fiori>

**Note**  
The following image contains links to more information.

Read Me First Tutorials Developing Apps

Start developing apps right away. Read our best practices and learn about security and

# CaptainCasa Usage of UI5

- Same way of usage as default CaptainCasa
  - Server side UI5 rendering
- Components are 1:1 reflections of UI5 components
  - „No“ discussion on components!
  - What you get is what you get!





# Demo



# CaptainCasa UI5 Positioning

- CaptainCasa UI5 is an add-on to CaptainCasa
  - CaptainCasa UI5 is not positioned as an autarc framework for server-side SAP UI5 rendering
- „Mobile first“
  - The mobile components are/were transferred first
  - Desktop components on request



# Page Bean Components



# JSF Components are heavyweight

The screenshot displays a JSF component editor interface. On the left, the 'Component Tree' shows a hierarchy of components: 'Components' (root), 'trowtitlebar', 'trowheader', 'trowbodypane', 'trow', 'tfoldablepane / w:100%' (selected), 'trow', 'tfoldablepane / w:100% h:100%', 'trow', and 'trowstatusbar'. On the right, the 'Live View' shows the rendered component, titled 'Test Page Bean Component'. It features a yellow header 'Some other data', a blue header 'Data', and a table with the following data:

	North	East
CDs	1,000,00	2,000,0
PCs	0,00	1,0
Laptops	2,00	3,0
Tablets	4,00	5,0
Smartphones	0,00	1,0
Watches	2,00	3,0
Printers	4,00	5,0

Below the table, there is another blue header 'Some other data' and a text input field.

- Fine for „real client components“
- Not OK for „composite components“



# Page Beans are lightweight

The screenshot displays the CaptainCasa Layout Editor Toolset interface. The main workspace is divided into several panels:

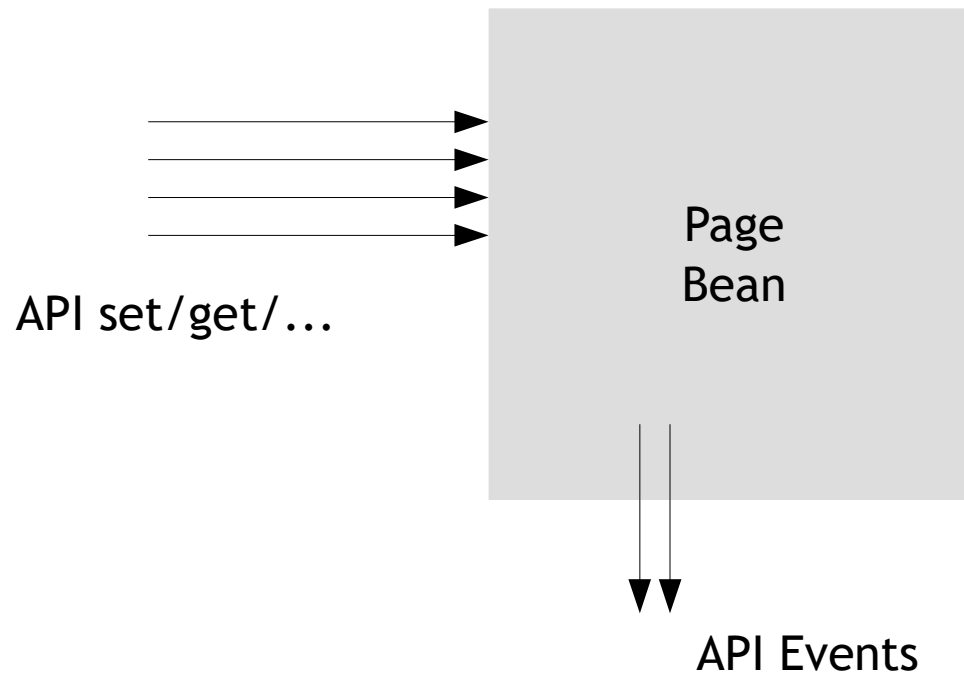
- Component Tree:** Shows the hierarchical structure of the page components, including a table and a chart.
- Live View:** Displays the rendered page. It features a table with data for PCs, Laptops, and Tablets across North and South regions, and a bar chart visualizing this data.
- Bean Browser:** Shows the managed server beans, including a Dispatcher.
- Style & Macro:** Provides options for styling and macro application.
- Messages:** A panel for handling messages.

The table in the Live View is highlighted with a red box and contains the following data:

	North	South
PCs	0,00	1,00
Laptops	2,00	3,00
Tablets	4,00	5,00

The bar chart visualizes this data, with bars for PCs (orange), Laptops (yellow), and Tablets (green) for both North and South regions. The Y-axis ranges from 0.0 to 5.5.

# Page Beans are „nice“ Components



## But so far...

- PageBeans are just normal page implementations
  - Code (=> .class)
  - Page (=> .jsp)
  - Resources (=> .png)
  - Literals (=> .properties)
- „Perfect“ for modularization within a project
- Not perfect for cross-project modularization
  - ...too much to copy...
  - ...too many side conditions...



# Page Bean Components

- Page Bean Components are PageBeans, that are...
  - Autarc components (.jar)
  - Easily transferable (import .jar)
  - As lightweight as a library
- All resources (.class, .jsp, .png, .properties) are kept in one .jar file



# Page Bean Components

- Page Bean Components are part of the Tools
  - Simple configuration

The screenshot displays the CaptainCasa Enterprise Client interface. On the left, the 'Component Tree' shows a hierarchy of components. A component named '#@TestUI.gridChart' is highlighted, and an arrow points to it from the bottom left. On the right, the 'Live View' shows a 'Test Page Bean Component' with a 'Data' section. This section contains a table and a bar chart. The table has columns for 'North' and 'East' and rows for 'PCs', 'Laptops', and 'Tablets'. The bar chart displays the data for 'North' and 'East' for each device type.

	North	East
PCs	0,00	1,00
Laptops	2,00	3,00
Tablets	4,00	5,00

Legend: PCs (Red), Laptops (Orange), Tablets (Green)

...well...

- Some dependencies are left...
  - CaptainCasa version
  - [ Styling (style.xml, fx.css) ]





# Positioning

- Internal usage at your side
  - Centralized generic components
  - Centralized application components
- CaptainCasa  $\Leftrightarrow$  „You“
  - Development of Page Bean Components is welcome!
  - Open source etc.
  - Addon library to CaptainCasa



# 2015 Components and Features

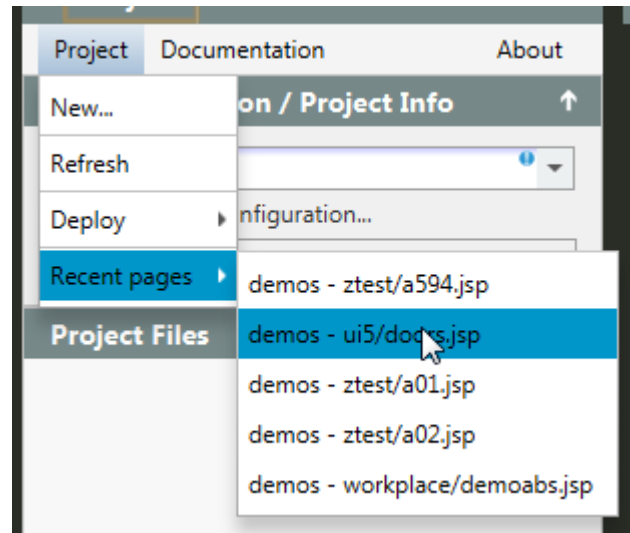


# PASSWORD

Unencrypted	<input type="password" value="..."/>
MD5 Encrypted	<input type="text"/>
SHA Encrypted	<input type="text"/>
<input type="button" value="Show Content"/>	



... ;-)



## Improvement - ComponentNode Classes now return “this”

For each component (e.g. FIELD) there is a corresponding ComponentNode-class (e.g. FIELDNode) for dynamically arranging components by using (ROW)DYNAMICCONTENT.

Up to now set-ers of this class did not return anything. Now they return the instance of the component. This allows a more compact and nicer way of writing the corresponding code:

```
Before:  
FIELDNode fn = new FIELDNode();  
fn.setWidth(...);  
fn.setEnabled(...);  
fn.setText(...);
```

```
Now:  
FIELDNode fn = new FIELDNode()  
                .setWidth(...)  
                .setEnabled(...)  
                .setText(...);
```

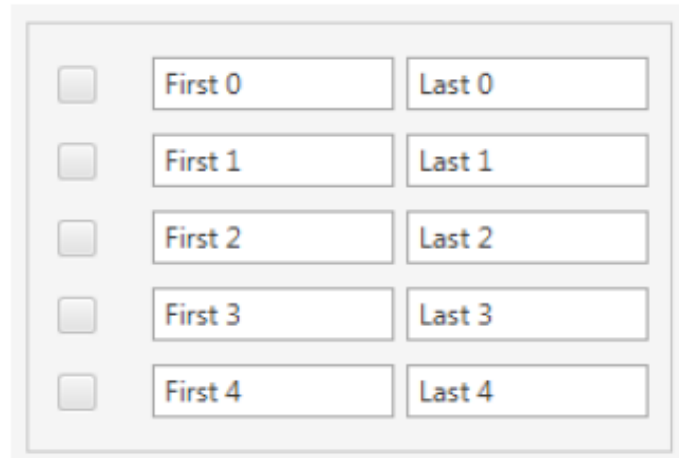
## Improvement - New component: REPEAT

We believe that this is really some important component - and a bit ask ourselves why this component is not part of CaptainCasa since the framework's beginning... ;-)

The FIXGRID is nice - but clearly is a component which follows the principle of “outside-in” sizing: the size of the cells is driven by the grid, the component(s) inside the cell have to follow. This has some number of advantages - but also makes it difficult to use the grid in situations where you want to use it in a much more flexible way - e.g. with rows of different height that is driven by the content of the cells.

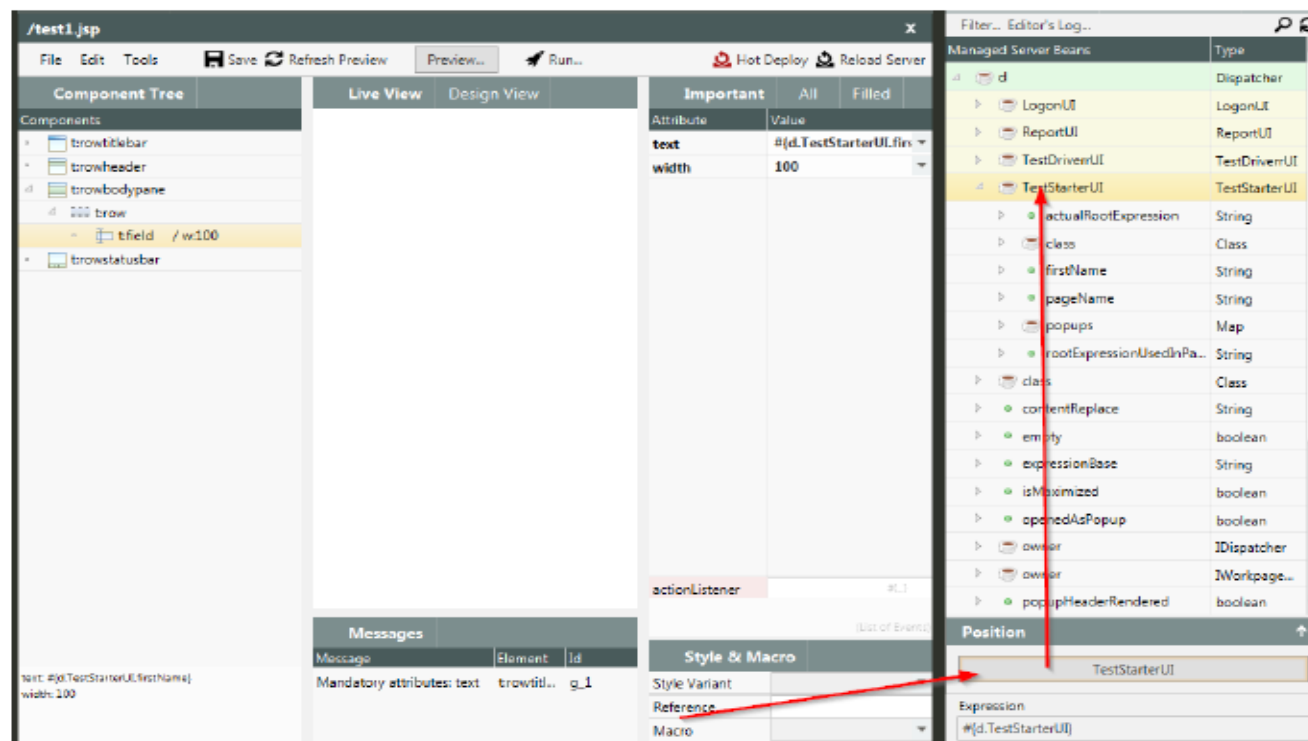
The REPEAT component overcomes these limitations - though being a very simple component. It is bound to a list and just repeats its content for each list item.

Example:



<input type="checkbox"/>	First 0	Last 0
<input type="checkbox"/>	First 1	Last 1
<input type="checkbox"/>	First 2	Last 2
<input type="checkbox"/>	First 3	Last 3
<input type="checkbox"/>	First 4	Last 4





This button shows the name of the corresponding class. Whenever pressing the button then the beans browser tree will be opened so that the class is opened. - In nearly all scenarios a page is assigned to one class only, so this button accelerated the navigation to this class significantly.

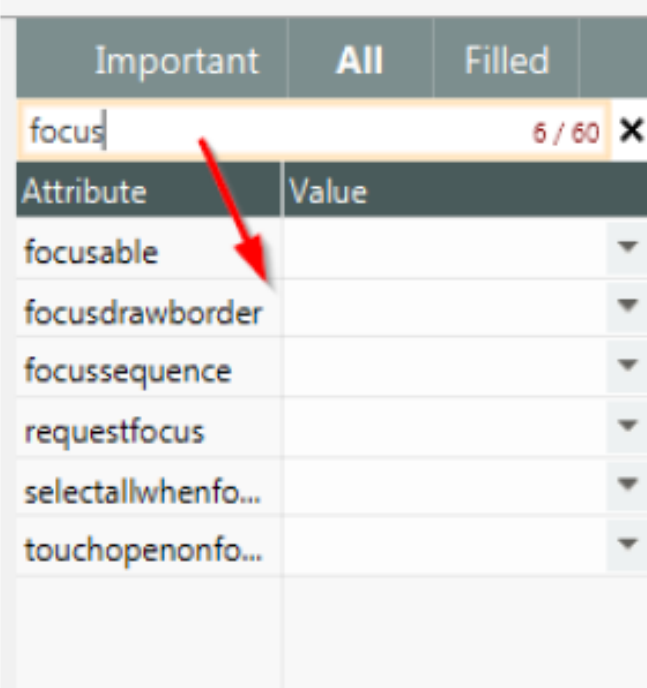
### Package abbreviations:

In the project configuration you now can define package abbreviations that are used inside the bean browser's tree.

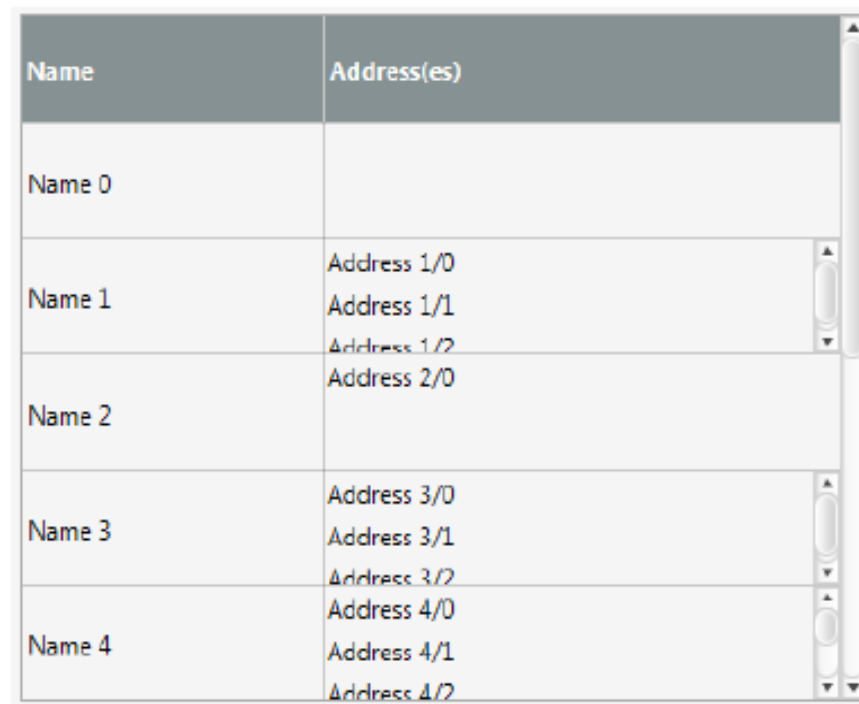
```
<project ...>
  ...
  <packageabbreviation packagename="managedbeans" abbreviation="mb"/>
  ...
</project>
```

## Improvement - Attribute Search in the Layout Editor

In the “All” and “Filled” view of the attributes of a component you now can search for attributes by name:



Important	All	Filled
focus 6 / 60 X		
Attribute	Value	
focusable		▼
focusdrawborder		▼
focussequence		▼
requestfocus		▼
selectallwhenfo...		▼
touchopenonfo...		▼

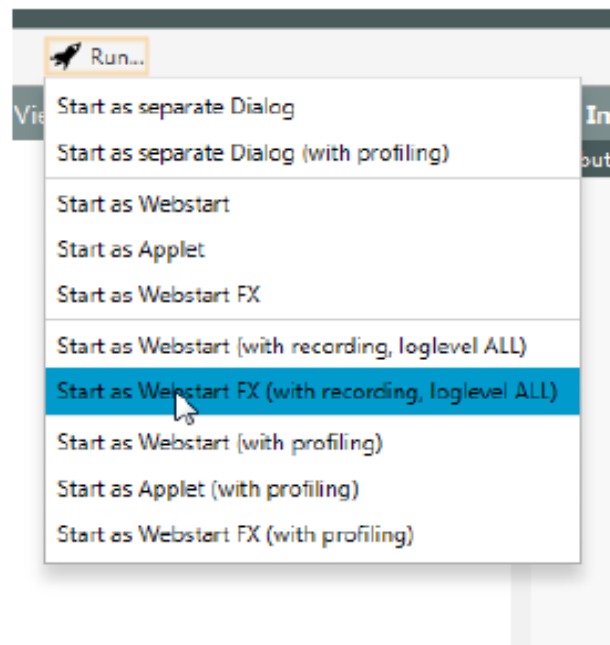


Name	Address(es)
Name 0	
Name 1	Address 1/0 Address 1/1 Address 1/2
Name 2	Address 2/0
Name 3	Address 3/0 Address 3/1 Address 3/2
Name 4	Address 4/0 Address 4/1 Address 4/2

## Improvement - Tools: Starting of Page in loglevel=ALL Mode

The client is able to log all the http traffic into a log file. This log file can then be used as “dummy http server”, so that certain client situations can be recorded. CaptainCasa can then replay these logfiles and as consequence do some error analysis independent from your server. This function had to be invoked by certain client parameters up to now.

You now can start this function directly from the Layout Editor:



### Improvement - Swing Client: PDF Renderer

Some more functions in the Swing PDF Renderer:



## Improvement - Swing/FX Client: adapting Client Literals

All messages and literals that are shown inside the client are read through some central function. This function was now extended to also allow some influence on client side in order to use different texts than the one that are defined in the client side property files.

The extension is described in the following chapter of the Developer's Guide:

### Updating Client side Literals

Inside the client processing there are a couple of literals that are used as "fix literals" of the client. Examples:

- If there is a connection problem then the client will show a corresponding popup asking the user if to cancel the current processing.
- Button texts and tool tip texts of certain components (e.g. PDFRENDERER)

In general these texts are only available on client side. CaptainCasa stores them in property files that are included in the client side jar-files.

You may update these literals and replace them by own text definitions by implementing the client side interface "[IClientLiteralsExtension](#)":

```
public interface IClientLiteralsExtension
{
    public String getLit(String literal);
}
```

The class that you add to the client processing needs to be passed as client parameter "[clientliteralextensionclassname](#)".

Every time the client needs some text for a certain literal id, it will first check the interface - and only will then use the property definitions if the interface returns null.

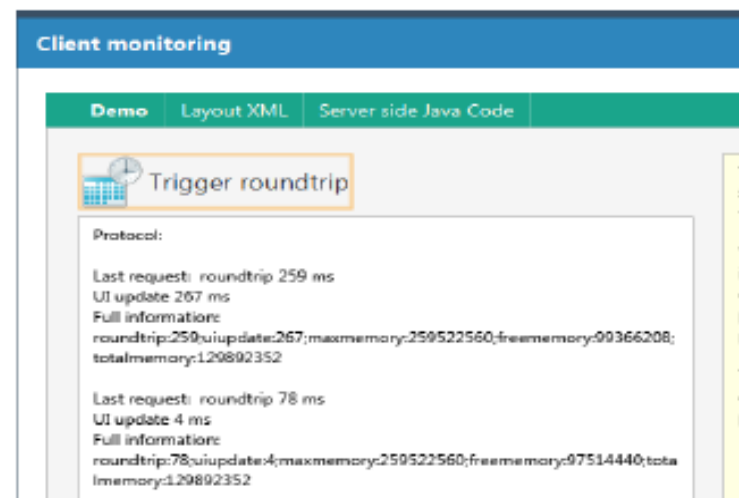


## Improvement - Swing/FX-Client: Component CLIENTMONITOR

The component CLIENTMONITOR transfers basic monitoring information of the client to the server. The information is sent with every request and includes:

- The roundtrip duration of the request (measured from client's perspective)
- The UI processing duration for the last roundtrip
- Memory information

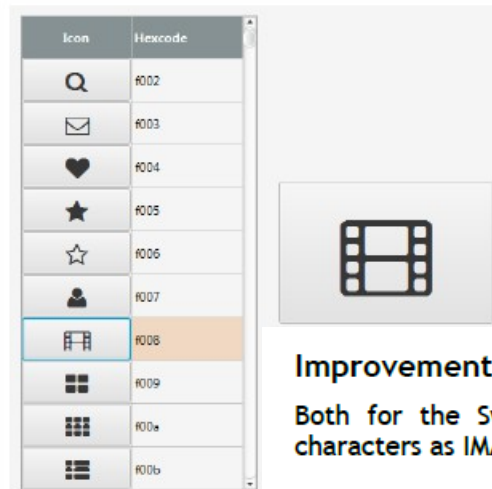
Please check the example, that is available in the “New Functions” section of the demo workplace.



The screenshot displays a web application interface titled "Client monitoring". It features a navigation bar with "Demo", "Layout XML", and "Server side Java Code". A "Trigger roundtrip" button is visible. The main content area shows two sets of performance data:

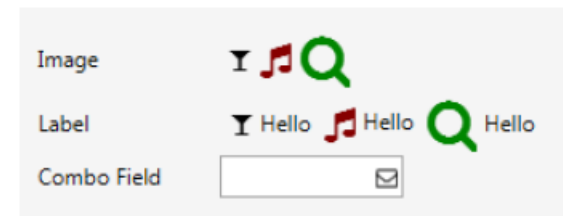
```
Protocol:
Last request: roundtrip 259 ms
UI update 267 ms
Full information:
roundtrip:259;uiupdate:267;maxmemory:250522560;freememory:90366208;
totalmemory:129892352

Last request: roundtrip 78 ms
UI update 4 ms
Full information:
roundtrip:78;uiupdate:4;maxmemory:250522560;freememory:97514440;total
memory:129892352
```



### Improvement - Directly define AwesomeFont-Character as IMAGE

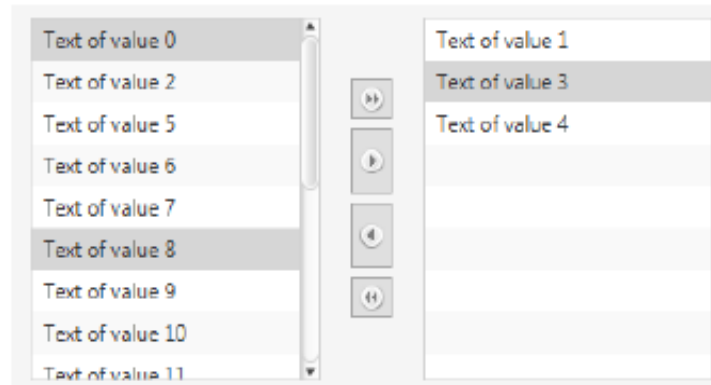
Both for the Swing-Client and for the FX-Client you now can define AwesomeFont characters as IMAGE attribute:



The image shows the corresponding example (Demo Workplace => New Functions), in which e.g. IMAGE-IMAGE, LABEL-IMAGE and COMBOFIELD-IMAGE are directly defined via AwesomeFont characters. There is a certain syntax that you have to define within the IMAGE attribute - example: "<t:image id="g\_2" image="/awesomefont(f000,#000000,12)" >". Please check details within the demo.

### Improvement - FX Client: new Component PICKLIST

The component PICKLIST allows to select one or several values out of a list of several values:



CaptainCasa Enterprise Client FX

### Structured Data Pool Manager - demo

Data Pool Configure data structure...

Reread data Export...

	Number	Description	Category	Priority	Uregency	Status	Responsible
	all	all	all	all	all	all	all
1	0001	Evaluate data pool manager	request	A	A	finished	
2	0002	Publsh data pool manager	request	A	A	in process	
3	0003	Add documentation	bug	B	A	in process	

Add item

Copy into clipboard Send...

#### Item Data

Background:

---

Number:

Description:

Category:

Priority:

Uregency:

Status:

Responsible:

#### Attachments

Add

#### Discussion / Comments

**demo**, 17.08.2015 14:16:24

The corresponding web site is <http://www.CaptainCasa.com>

Add

### Improvement - Popup position/size-Persistence can be switched off

If a dialog has a “persistId” (passed by `Modal/ModelessPopup.setPersistId(..)`) then its position and size is stored automatically on server side.

This mechanism now can be explicitly switched on/off by configuring in “system.xml”:

```
<system>
...
... <dialogs persistenceenabled="true" useworkpageidaspersistid="true"/>
...
...
</system>
```

The default value for both attributes is “true”.

- “persistenceenabled” => this is the global switch to switch on/off the persistence
- “useworkpageidaspersistid” => if set to “true” then automatically the id of a workpage will be used as persistId as well within the workplace management.

### Improvement - FILEDOWNLOAD\*: Transfer of Client File Name

When using “IBufferedContent” (or better: when deriving from DefaultBufferedContent) for providing server side download information then you now also receive the client file name that the user selected for download.

Background: you now may download some different data dependent on the extension - e.g. an Excel file on “.xls” and a text file on “.txt”.





```
package org.eclnt.jsfserver.elements.util;

public interface IFIXGRIDExportInfoProvider
{
    public String findBackgroundColorForDataCell(int rowIndex,
                                                String dataCellValue,
                                                ColumnInfo column);

    public String findAlignmentForDataCell(int rowIndex,
                                           String dataCellValue,
                                           ColumnInfo column);

    public String findTextColorForDataCell(int rowIndex,
                                           String dataCellValue,
                                           ColumnInfo column);
}
```

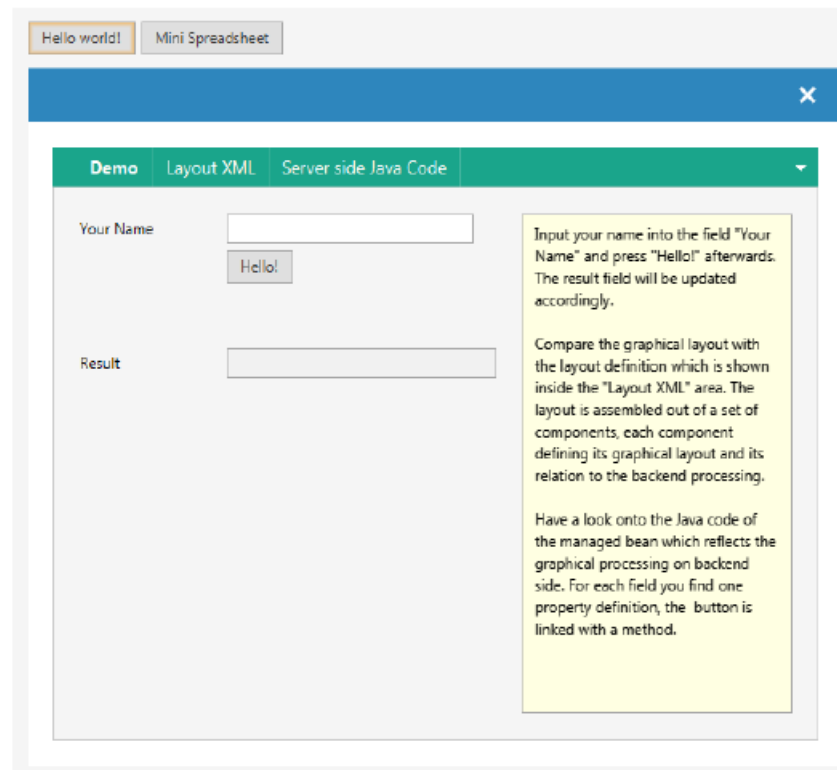


component.



### Improvement - New PageBeanComponent - CCMultiPageContainer

In certain scenarios you want to integrate CaptainCasa clients of different server systems into one user interface. This is the purpose of the component CCMultiPageContainer.



### Improvement - FX Client: attribute ANIMATECHANGEOSIZE

All components now provide a new attribute ANIMATECHANGEOSIZE. If set to true then changes of the component size (e.g. the WIDTH or HEIGHT is set by a server side property) are applied smoothly - and not in one step. Please check the corresponding demo in the demo workplace (section "New Functions").



## Improvement - FX Client: New component ROWADAPTIVEAREA

An additional component in the area of “adaptive layout” is available: ROWADAPTIVEAREA. It allows to flexibly arrange “blocks” of components dependent on the available width:

- “Wide” screen:

Name and Title	Address	Communication
Some label <input type="text"/>	Some label <input type="text"/>	Some label <input type="text"/>
Some label <input type="text"/>	Some label <input type="text"/>	Some label <input type="text"/>
Some label <input type="text"/>	Some label <input type="text"/>	Some label <input type="text"/>
Some label <input type="text"/>		
Some label <input type="text"/>		

## Improvement - FX Client: New component ROWADAPTIVELINE

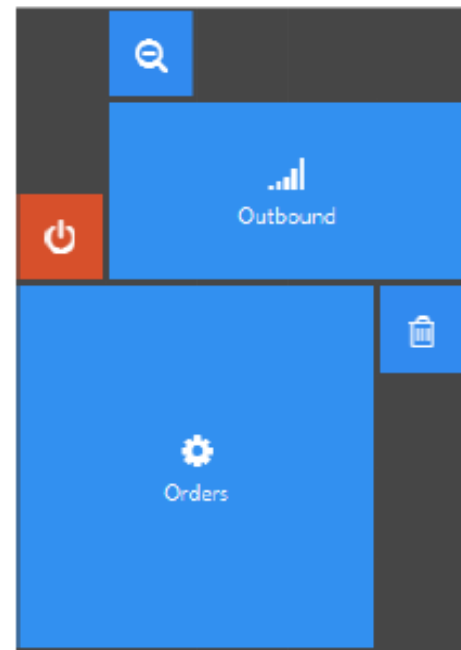
The components behaves like a row which switches from a one-line mode into a two-lines mode if a certain width is not available. You are able to define the following screen scenario with one layout:

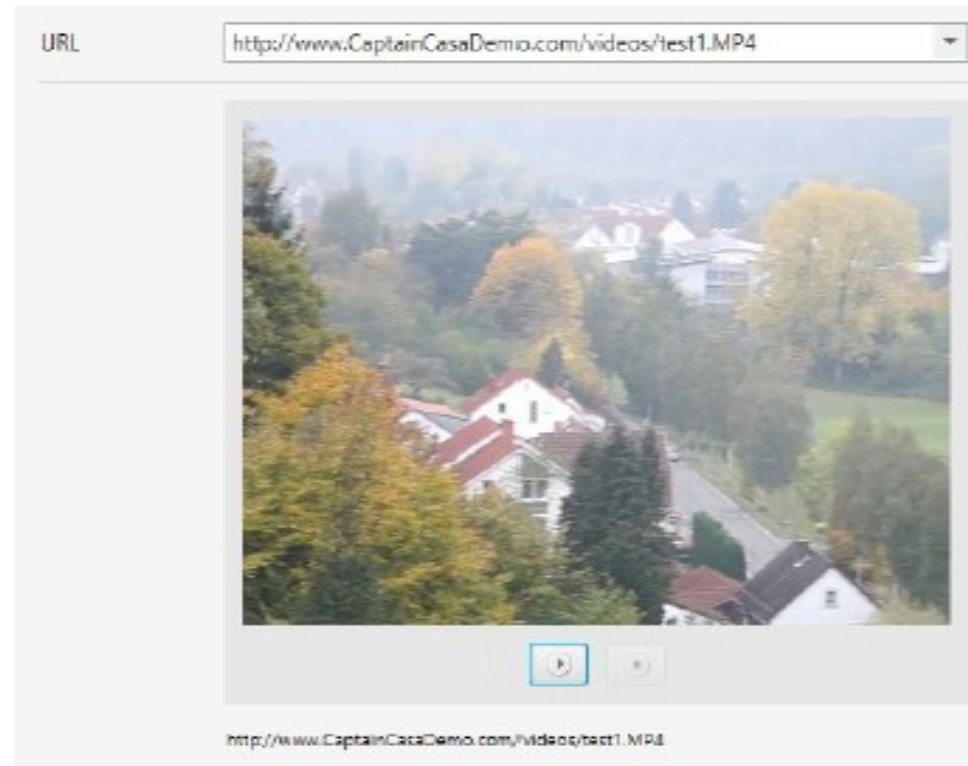
- “Much” space available

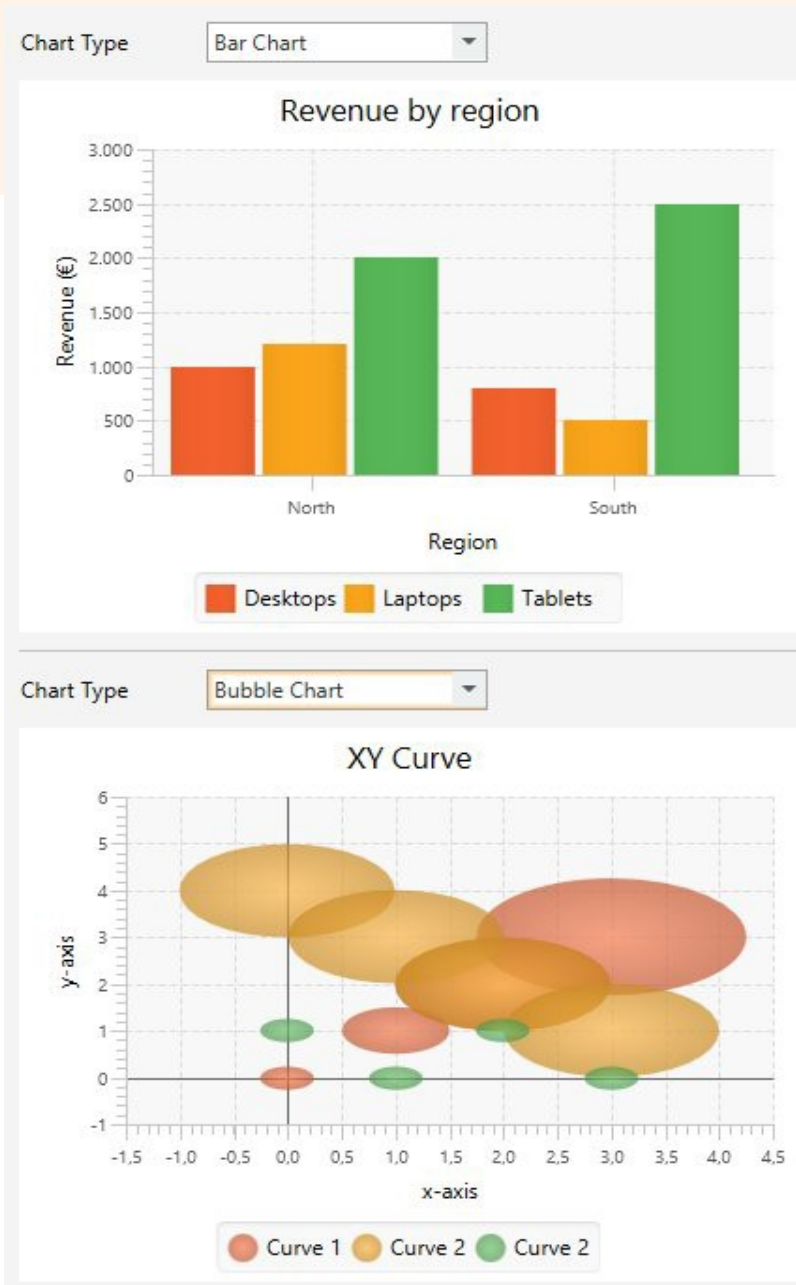
First name	<input type="text"/>	
Last Name	<input type="text"/>	
Street	<input type="text"/>	<input type="text"/>
Town	<input type="text"/>	<input type="text"/>

### Improvement - Page Bean Component “CCMatrixDashboard”

In addition to the “CCDashboard” component there is now a “CCMatrixDashboard” component that arranges page beans within a matrix:









## Improvement - FX Client: Performance for FIELD, FORMATTEDFIELD, COMBOFIELD

There is a new option in the three components - CREATIONHINT. If set to “toggle” then the components are not immediately fully rendered, but are fully rendered on request only. This means: the content of the component is rendered as simple text output, if the user tabs or clicks into the component (focus is gained) then the component internally switches from simple text output to actually rendering the corresponding field.

Background: the TextField component in FX is a quite heavy one...

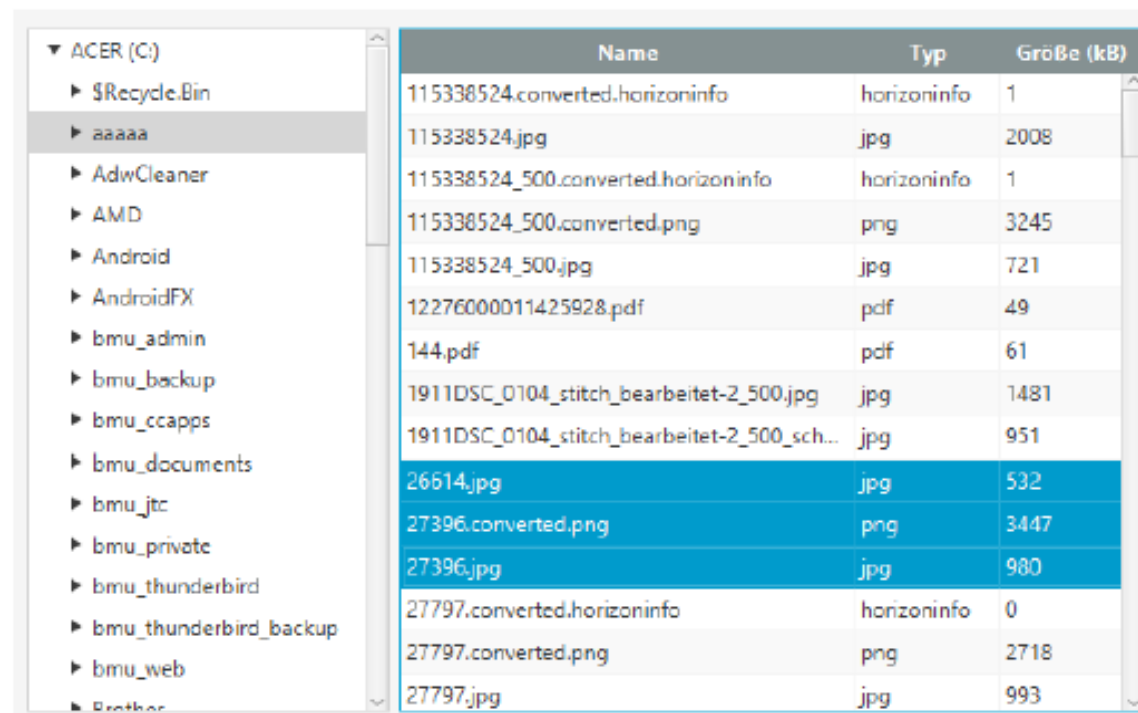
In “toggle” mode the component does NOT support all features of the normal mode. The “toggle” mode is optimized for usage within a grid cell. It e.g. does not support borders.

(We internally applied the performance improvements to our layout editor - seeing a significantly better build-up-time when editing pages....)



## Improvement - New component CLIENTFILEEXPLORER

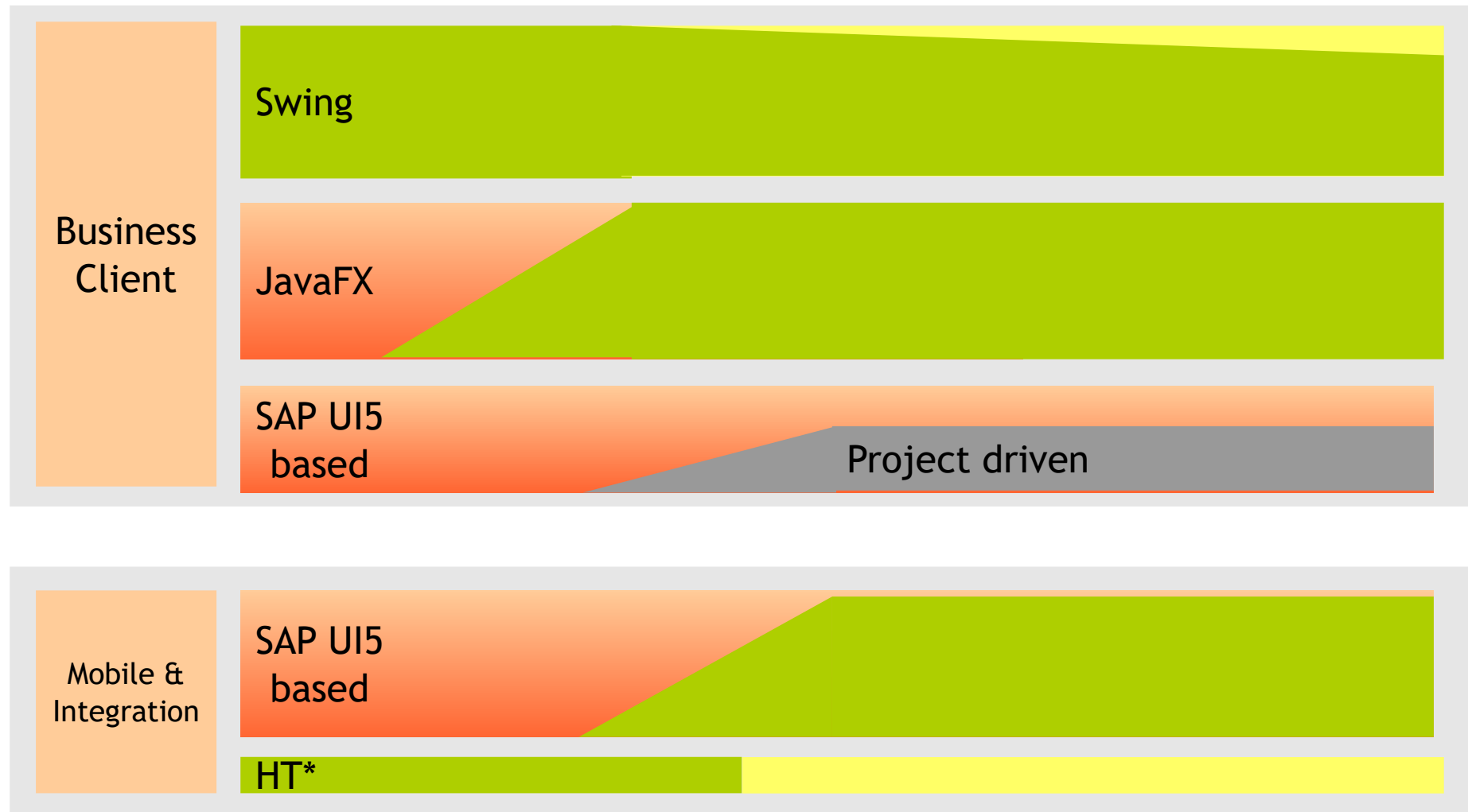
It is now possible to embed a “mini explorer” onto the client's file system:



# Road Map 2016



# Roadmap 2015



# Roadmap 2016

Business Client

Swing

JavaFX

Mobile & Integration

SAP UI5 based

HT\*

Application Components

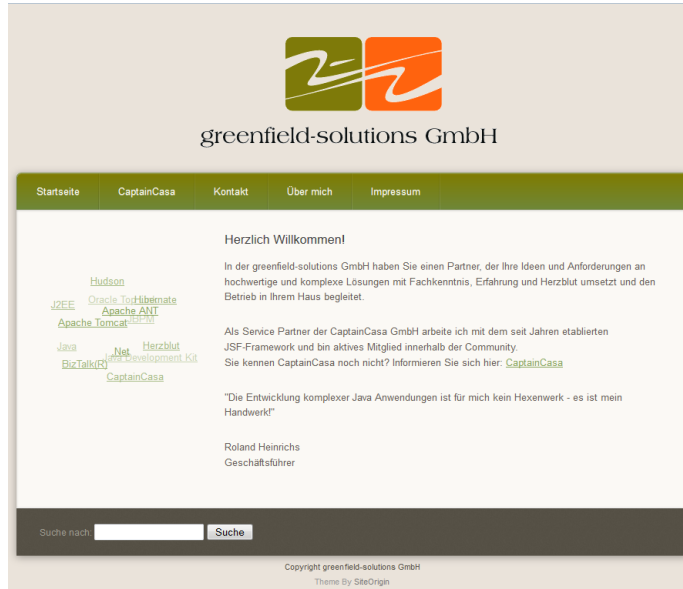



# CaptainCasa XI. Community Meeting





# Partner



  
**greenfield-solutions GmbH**

[Startseite](#)   [CaptainCasa](#)   [Kontakt](#)   [Über mich](#)   [Impressum](#)

**Herzlich Willkommen**

In der greenfield-solutions GmbH haben Sie einen Partner, der Ihre Ideen und Anforderungen an hochwertige und komplexe Lösungen mit Fachkenntnis, Erfahrung und Herzblut umsetzt und den Betrieb in Ihrem Haus begleitet.

Als Service Partner der CaptainCasa GmbH arbeite ich mit dem seit Jahren etablierten JSF-Framework und bin aktives Mitglied innerhalb der Community.  
 Sie kennen CaptainCasa noch nicht? Informieren Sie sich hier: [CaptainCasa](#)

"Die Entwicklung komplexer Java Anwendungen ist für mich kein Hexenwerk - es ist mein Handwerk!"

Roland Heinrichs  
 Geschäftsführer

Suche nach:

Copyright greenfield-solutions GmbH  
 Theme By SiteOrigin





# Partner

**Joachim Unger**

**IT-Beratung**

**Start**

**Technologien**

**Schwerpunkte**

**Über mich**

**Kontakt**

**Kunden**

**Profil (PDF)**

**Impressum**

Hier folgt eine Auflistung der mir bestens vertrauten Technologien.

**Sprachen**

- Java / J2EE
- XSL / XSLT / FO
- HTML / HTML5 / CSS / JavaScript
- JSP / JSF (myFaces, Tomahawk)
- PL/SQL
- PL/pgSQL
- C / C++ / C# / NET

**Datenbanken**

- Oracle
- PostgreSQL
- IBM DB2
- Microsoft SQL-Server
- Adabas
- MySQL

**Data Warehouse**

- Pentaho
- Mondrian
- ReportServer
- JasperReports
- BIRT
- JFreeChart

**Tools**

- Eclipse

**GUI**

- Swing, JavaFX,

**Anwendungs-Frameworks**

Letzte Aktualisierung  
24.05.2014



## 2014 Kein Wandertag....

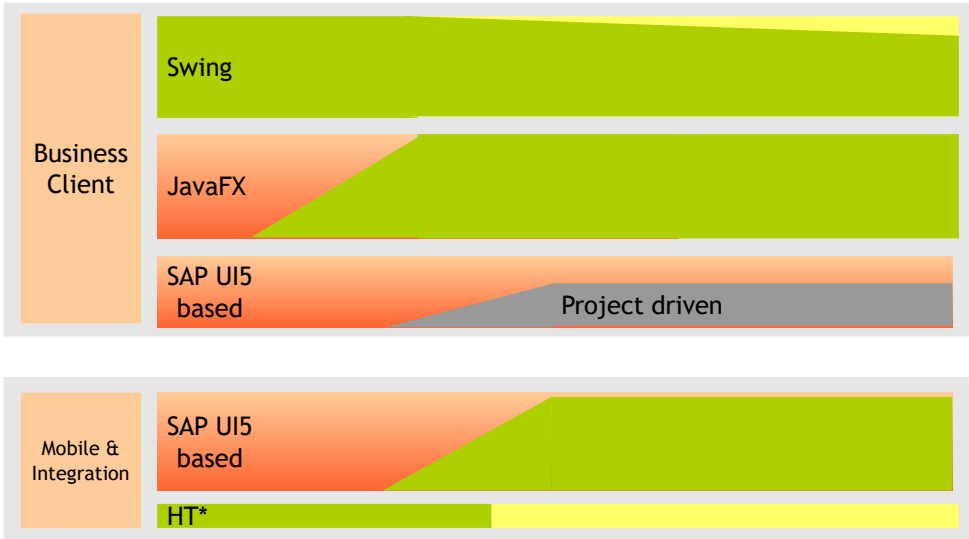


## Agenda

- 10:00h Standortbestimmung / Roadmap (I)
- 10:20h SAP/Open UI5
- 11:05h 2 Tips und 1 Szenario
- 11:25h Kaffeepause
- 11:40h Page Bean Components
- 12:10h 2 Tips und 1 Szenario
- 12:30h New Components/Features 2015(I)
- 13:00h Mittagspause
- 13:40h Ergonomisches UI Design
- 14:20h 2 Tips und 1 Szenario
- 14:40h New Components/Features 2015(II)
- 15:10h Spezialthemen (Zertifikat, SAML Portal)
- 15:30h Kaffeepause
- 16:00h Standortbestimmung / Roadmap (II)



# Roadmap 2015



# Java Swing

The screenshot shows a web browser window with the following content:

- Browser Address Bar:** <https://javenter.de/javafx-standortbestimmung-mit-swing-basierter-technologie-gewinnt-man-heute-keine-blumentöpfe-mehr-28>
- Page Header:**
  - Artikel-Serie: Eine kleine Reise durch NoSQL
  - Neue Kolumne auf JAXenter: Developer Café
  - Quickfix: Sind Sie daran interessiert, JDK 9 Early Access Releases auszuprobieren?
- Main Article:**
  - Interview mit Björn Müller
  - JavaFX-Standortbestimmung: "Mit Swing gewinnt man heute bei neuen Projekten keine Blumentöpfe mehr"**
  - 15. Oktober 2015 | Hartmut Schlosser
  - Author: Björn Müller (CaptainCasa GmbH)
  - Summary: "Eine Standortbestimmung für JavaFX wird Björn Müller (CaptainCasa GmbH) auf der kommenden W-JAX vornehmen. Im Countdown-Interview gibt er einen kleinen Vorgeschmack und erklärt, wo er momentan die Stärken und Schwächen der UI-Technologie sieht."
  - Text snippet: "JAXenter: Nach dem Hype um JavaFX scheint es in den letzten Monaten etwas ruhiger um die UI-Technologie geworden zu sein. Siehst du das auch so?"
- Right Sidebar (JAX 2016):**
  - 18. bis 22. April 2016
  - CONTINUOUS DELIVERY IN THE CLOUD (Jonathan Weiss)
  - LEANISIEREN – VERSCHLANKEN UND WERTBEITRAG ERHÖHEN (Inge Hanschke)
  - HOW TO SCALE IF NOTHING IS FOR FREE (André Neubauer)
  - INFRASTRUKTUR FÜR MICROSERVICES: DOCKER ORCHESTRATION (Pieter Roßbach)
  - FÜNF EINFACHE SCHRITTE ZU CONTINUOUS DELIVERY (Eberhard Wolff)
- Footer:** Cookie policy notice with a "Verstanden" button.

# JavaFX

Lieber ein Ende mit Schrecken als ein Schrecken ohne Ende?

## Ruhe in Frieden, JavaFX?

24. November 2015 · Michael Thomas

Der ehemaligen Sun-Mitarbeiter und Codename-One-Gründer Shai Almog führt eine Bestandsaufnahme von JavaFX durch und diskutiert die Frage, welche Zukunft die Technologie hat – mit wenig erfreulichem Ergebnis.

Nicht nur der Umstand, dass es um die UI-Technologie in den letzten Monaten etwas ruhiger geworden ist, nimmt Shai Almog zum Anlass, eine Standortbestimmung des Frameworks vorzunehmen. Auch Oracles Handlungsmöglichkeiten, die Stellung von JavaFX im Ökosystem zu verbessern, sind Motivation und Inhalt seiner Ausführungen. Zentral steht zudem die Frage im Raum, ob und wie JavaFX weiterentwickelt werden kann.

Aktuell, so Almog, **fallen die Nutzer von JavaFX unter eine von drei Kategorien:** Unternehmen mit

**JAX 2016**  
18. bis 22. April 2016

- CONTINUOUS DELIVERY IN THE CLOUD**  
Jonathan Weiss (Amazon Web Services)
- LEANISIEREN – VERSCHLANKEN UND WERTBEITRAG ERHÖHEN**  
Inge Hanschke (Lea42)
- HOW TO SCALE IF NOTHING IS FOR FREE**  
André Neubauer (Mister Spex)
- INFRASTRUKTUR FÜR MICROSERVICES: DOCKER ORCHESTRATION**  
Peter Roßbach (bee42 solutions)
- FÜNF EINFACHE SCHRITTE ZU CONTINUOUS DELIVERY**  
Eberhard Wolff (innoc)
- RESTFUL HYPERMEDIA-APIs**  
Kai Tödter (Siemens AG)



## HTML - JavaFX

- JavaFX
  - Much UI, „few“ users
- HTML5
  - (Much) UI, „many“ users





## HTML - JavaFX

- JavaFX
  - Browser is not relevant
- HTML5
  - Browser is my constant enemy



## HTML - JavaFX

- JavaFX
  - „Complex“ Installation
  - „Zero“ Maintenance
- HTML5
  - „Zero“ Installation
  - „Complex“ Maintenance

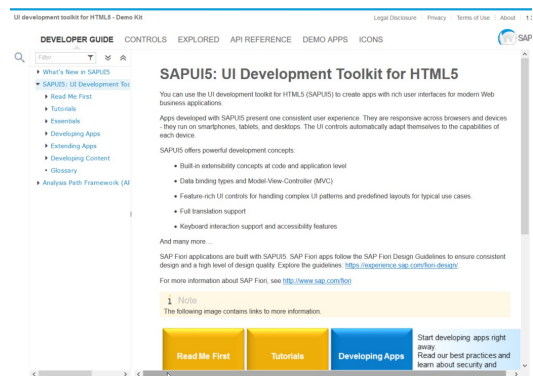




# SAP/Open UI5

## SAP UI5

- Client side JavaScript library
  - SAP as developer
  - ...open sourced („9x%“)
- Commitment
  - SAP message
  - SAP applications



## CaptainCasa Usage of UI5

- Same way of usage as default CaptainCasa
  - Server side UI5 rendering
- Components are 1:1 reflections of UI5 components
  - „No“ discussion on components!
  - What you get is what you get!



Demo



## CaptainCasa UI5 Positioning

- CaptainCasa UI5 is an addon to CaptainCasa
  - CaptainCasa UI5 is **not** positioned as autarc framework for server-side SAP UI5 rendering
- „Mobile first“
  - The mobile components are/were transferred first
  - Desktop components on request

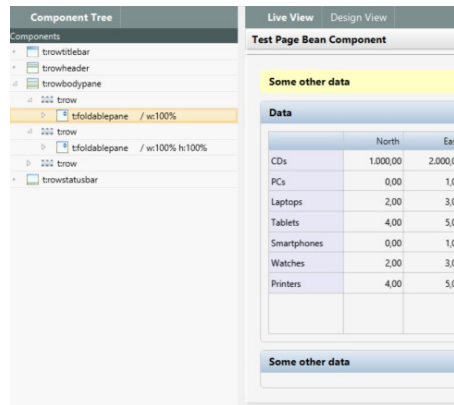




# Page Bean Components



## JSF Components are heavyweight



- Fine for „real client components“
- Not OK for „composite components“

# Page Beans are lightweight

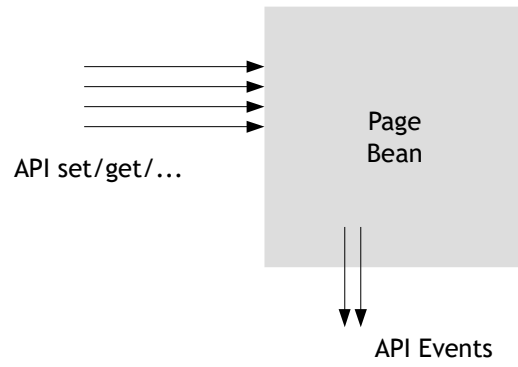
The screenshot shows the CaptainCasa Layout Editor Toolset interface. The main window displays a 'Live View' of a page with a bar chart and a table. The table data is as follows:

	North	South
PCs	0.00	1.00
Laptops	2.00	3.00
Tablets	4.00	5.00

The bar chart visualizes this data, with bars for PCs (green), Laptops (orange), and Tablets (green) for both North and South regions. The y-axis ranges from 0.0 to 5.5. The legend at the bottom of the chart identifies the colors: PCs (green), Laptops (orange), and Tablets (green).

The interface also includes a Component Tree on the left, a Bean Browser on the right, and a Messages panel at the bottom.

## Page Beans are „nice“ Components



## But so far...

- PageBeans are just normal page implementations
  - Code (=> .class)
  - Page (=> .jsp)
  - Resources (=> .png)
  - Literals (=> .properties)
- „Perfect“ for modularization within a project
- Not perfect for cross-project modularization
  - ...too much to copy...
  - ...too many side conditions...



## Page Bean Components

- Page Bean Components are PageBeans, that are...
  - Autarc components (.jar)
  - Easily transferable (import .jar)
  - As lightweight as a library
- All resources (.class, .jsp, .png, .properties) are kept in one .jar file



## Page Bean Components

- Page Bean Components are part of the Tools
  - Simple configuration

The screenshot displays the CaptainCasa Enterprise Client interface. On the left, the 'Component Tree' shows a hierarchy of components: 'Components' (root), 'trowTitlebar', 'trowheader', 'trowbodypane', 'trow', 'trow' (with 'efoldablepane / w:100% h:40'), 'trow' (with '> #id TestU.gridChart'), and 'trowstatusbar'. An arrow points from the 'trow' component containing the chart to the 'Live View' on the right.

The 'Live View' shows a 'Test Page Bean Component' with two sections: 'Some other data' and 'Data'. The 'Data' section contains a table and a bar chart.

	North	East
PCs	0,00	1,00
Laptops	2,00	3,00
Tablets	4,00	5,00

The bar chart displays the data from the table, with bars for PCs (red), Laptops (orange), and Tablets (green) for both North and East regions. The y-axis ranges from 0.0 to 5.5.

...well...

- Some dependencies are left...
  - CaptainCasa version
  - [ Styling (style.xml, fx.css) ]



## Positioning

- Internal usage at your side
  - Centralized generic components
  - Centralized application components
- CaptainCasa <=> „You“
  - Development of Page Bean Components is welcome!
  - Open source etc.
  - Addon library to CaptainCasa





# 2015 Components and Features



# PASSWORD

Unencrypted

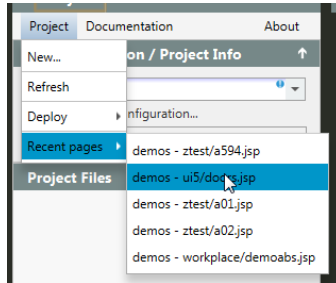
MD5 Encrypted

SHA Encrypted

Show Content



... ;-)



**Improvement - ComponentNode Classes now return "this"**

For each component (e.g. FIELD) there is a corresponding ComponentNode-class (e.g. FIELDNode) for dynamically arranging components by using (ROW)DYNAMICCONTENT.

Up to now set-ers of this class did not return anything. Now they return the instance of the component. This allows a more compact and nicer way of writing the corresponding code:

```
Before:
FIELDNode fn = new FIELDNode();
fn.setWidth(...);
fn.setEnabled(...);
fn.setText(...);

Now:
FIELDNode fn = new FIELDNode()
                .setWidth(...)
                .setEnabled(...)
                .setText(...);
```



### Improvement - New component: REPEAT

We believe that this is really some important component - and a bit ask ourselves why this component is not part of CaptainCasa since the framework's beginning... ;-)

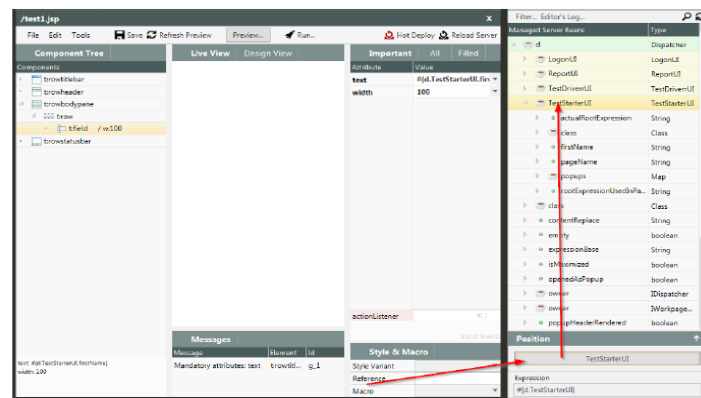
The FIXGRID is nice - but clearly is a component which follows the principle of "outside-in" sizing: the size of the cells is driven by the grid, the component(s) inside the cell have to follow. This has some number of advantages - but also makes it difficult to use the grid in situations where you want to use it in a much more flexible way - e.g. with rows of different height that is driven by the content of the cells.

The REPEAT component overcomes these limitations - though being a very simple component. It is bound to a list and just repeats its content for each list item.

Example:

<input type="checkbox"/>	First 0	Last 0
<input type="checkbox"/>	First 1	Last 1
<input type="checkbox"/>	First 2	Last 2
<input type="checkbox"/>	First 3	Last 3
<input type="checkbox"/>	First 4	Last 4





This button shows the name of the corresponding class. Whenever pressing the button then the beans browser tree will be opened so that the class is opened. - In nearly all scenarios a page is assigned to one class only, so this button accelerated the navigation to this class significantly.

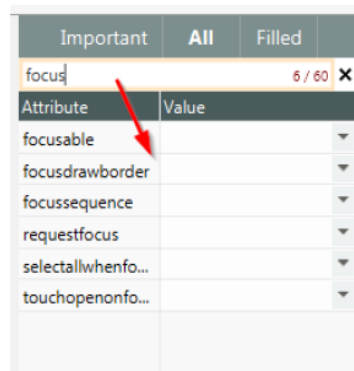
**Package abbreviations:**

In the project configuration you now can define package abbreviations that are used inside the bean browser's tree.

```
<project ...>
  ...
  <packageabbreviation packagename="managedbeans" abbreviation="mb"/>
  ...
</project>
```

### Improvement - Attribute Search in the Layout Editor

In the "All" and "Filled" view of the attributes of a component you now can search for attributes by name:



Important	All	Filled
focus 6 / 60 X		
Attribute	Value	
focusable		▼
focusdrawborder		▼
focussequence		▼
requestfocus		▼
selectallwhenfo...		▼
touchopenonfo...		▼





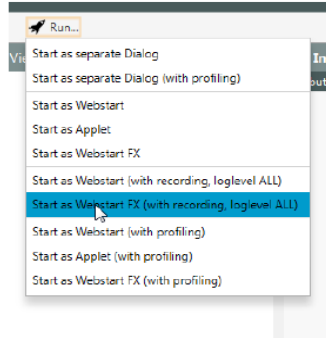
Name	Address(es)
Name 0	
Name 1	Address 1/0 Address 1/1 Address 1/2
Name 2	Address 2/0
Name 3	Address 3/0 Address 3/1 Address 3/2
Name 4	Address 4/0 Address 4/1 Address 4/2



### Improvement - Tools: Starting of Page in loglevel=ALL Mode

The client is able to log all the http traffic into a log file. This log file can then be used as "dummy http server", so that certain client situations can be recorded. CaptainCasa can then replay these logfiles and as consequence do some error analysis independent from your server. This function had to be invoked by certain client parameters up to now.

You now can start this function directly from the Layout Editor:



**Improvement - Swing Client: PDF Renderer**  
Some more functions in the Swing PDF Renderer:



### Improvement - Swing/FX Client: adapting Client Literals

All messages and literals that are shown inside the client are read through some central function. This function was now extended to also allow some influence on client side in order to use different texts than the one that are defined in the client side property files.

The extension is described in the following chapter of the Developer's Guide:

#### Updating Client side Literals

Inside the client processing there are a couple of literals that are used as "fix literals" of the client. Examples:

- If there is a connection problem then the client will show a corresponding popup asking the user if to cancel the current processing.
- Button texts and tool tip texts of certain components (e.g. PDFRENDERER)

In general these texts are only available on client side. CaptainCasa stores them in property files that are included in the client side jar files.

You may update these literals and replace them by own text definitions by implementing the client side interface "JClientLiteralsExtension":

```
public interface JClientLiteralsExtension
{
    public String getLit(String literal);
}
```

The class that you add to the client processing needs to be passed as client parameter "clientliteralextensionclassname".

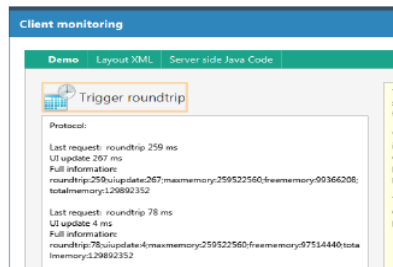
Every time the client needs some text for a certain literal id, it will first check the interface - and only will then use the property definitions if the interface returns null.

**Improvement - Swing/FX-Client: Component CLIENTMONITOR**

The component CLIENTMONITOR transfers basic monitoring information of the client to the server. The information is sent with every request and includes:

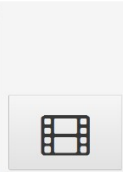
- The roundtrip duration of the request (measured from client's perspective)
- The UI processing duration for the last roundtrip
- Memory information

Please check the example, that is available in the "New Functions" section of the demo workplace.



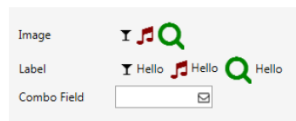


Icon	Hexcode
	#002
	#003
	#004
	#005
	#006
	#007
	#008
	#009
	#00a
	#00b



**Improvement - Directly define AwesomeFont-Character as IMAGE**

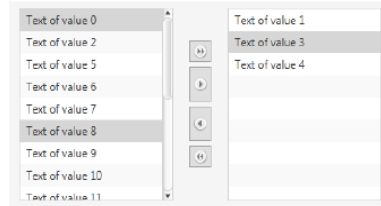
Both for the Swing-Client and for the FX-Client you now can define AwesomeFont characters as IMAGE attribute:



The image shows the corresponding example (Demo Workplace => New Functions), in which e.g. IMAGE-IMAGE, LABEL-IMAGE and COMBOFIELD-IMAGE are directly defined via AwesomeFont characters. There is a certain syntax that you have to define within the IMAGE attribute - example: "<:image id="g\_2" image="/awesomefont(f000,#000000,12)" >". Please check details within the demo.

**Improvement - FX Client: new Component PICKLIST**

The component PICKLIST allows to select one or several values out of a list of several values:



CaptainCasa Enterprise Client FX

### Structured Data Pool Manager - demo

Data Pool Configure data structure...

Reread data Export...

Number	Description	Category	Priority	Uregency	Status	Responsible	
all	all	all	all	all	all	all	
1	0001	Evaluate data pool manager	request	A	A	finished	
2	0002	Publsh data pool manager	request	A	A	in process	
3	0003	Add documentation	bug	B	A	in process	

Add Item

Copy into clipboard Send...

#### Item Data

Background:

Number:

Description:

Category:

Priority:

Uregency:

Status:

Responsible:

#### Attachments

Add

#### Discussion / Comments

**demo** 17.08.2015 14:16:24  
The corresponding web site is <http://www.CaptainCasa.com>

Add



### Improvement - Popup position/size-Persistence can be switched off

If a dialog has a "persistid" (passed by Modal/ModelessPopup.setPersistid(..)) then its position and size is stored automatically on server side.

This mechanism now can be explicitly switched on/off by configuring in "system.xml":

```
<system>
...
... <dialogs persistenceenabled="true" useworkpageidaspersistid="true"/>
...
...
</system>
```

The default value for both attributes is "true".

- "persistenceenabled" => this is the global switch to switch on/off the persistence
- "useworkpageidaspersistid" => if set to "true" then automatically the id of a workpage will be used as persistid as well within the workplace management.



**Improvement - FILEDOWNLOAD\*: Transfer of Client File Name**

When using "IBufferedContent" (or better: when deriving from DefaultBufferedContent) for providing server side download information then you now also receive the client file name that the user selected for download.

Background: you now may download some different data dependent on the extension - e.g. an Excel file on ".xls" and a text file on ".txt".



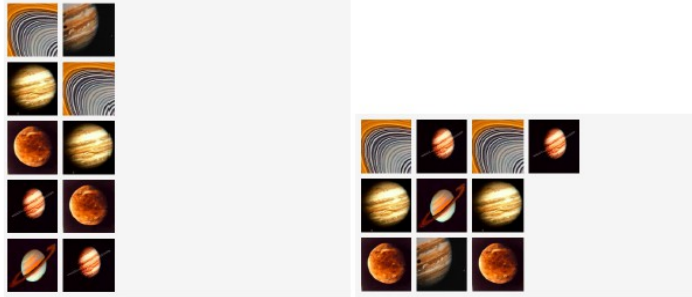
```
package org.ecInt.jsfserver.elements.util;
public interface IFIXGRIDExportInfoProvider
{
    public String findBackgroundColorForDataCell(int rowIndex,
                                                String dataCellValue,
                                                ColumnInfo column);

    public String findAlignmentForDataCell(int rowIndex,
                                           String dataCellValue,
                                           ColumnInfo column);

    public String findTextColorForDataCell(int rowIndex,
                                           String dataCellValue,
                                           ColumnInfo column);
}
```

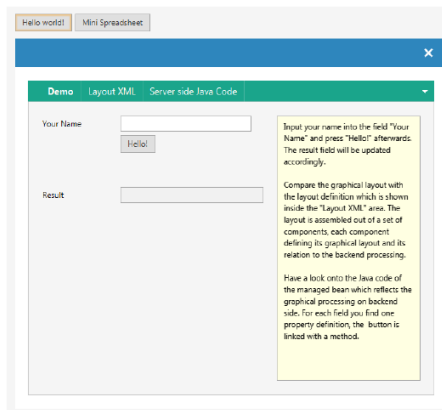


component.



**Improvement - New PageBeanComponent - CCMultiPageContainer**

In certain scenarios you want to integrate CaptainCasa clients of different server systems into one user interface. This is the purpose of the component CCMultiPageContainer.



**Improvement - FX Client: attribute ANIMATECHANGEOSIZE**

All components now provide a new attribute ANIMATECHANGEOSIZE. If set to true then changes of the component size (e.g. the WIDTH or HEIGHT is set by a server side property) are applied smoothly - and not in one step. Please check the corresponding demo in the demo workplace (section "New Functions").



**Improvement - FX Client: New component ROWADAPTIVEAREA**

An additional component in the area of “adpative layout” is available: ROWADAPTIVEAREA. It allows to flexibly arrange “blocks” of components dependent on the available width:

- “Wide” screen:

Name and Title		Address		Communication	
Some label	<input type="text"/>	Some label	<input type="text"/>	Some label	<input type="text"/>
Some label	<input type="text"/>	Some label	<input type="text"/>	Some label	<input type="text"/>
Some label	<input type="text"/>	Some label	<input type="text"/>	Some label	<input type="text"/>
Some label	<input type="text"/>	Some label	<input type="text"/>	Some label	<input type="text"/>
Some label	<input type="text"/>	Some label	<input type="text"/>	Some label	<input type="text"/>

**Improvement - FX Client: New component ROWADAPTIVELINE**

The components behaves like a row which switches from a one-line mode into a two-lines mode if a certain width is not available. You are able to define the following screen scenario with one layout:

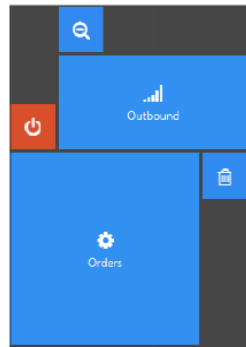
- “Much” space available

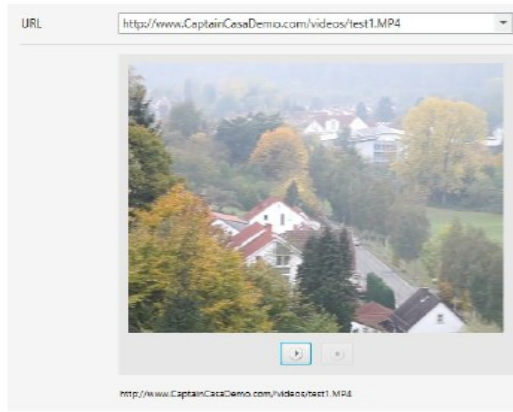
First name	<input type="text"/>	
Last Name	<input type="text"/>	
Street	<input type="text"/>	<input type="text"/>
Town	<input type="text"/>	<input type="text"/>



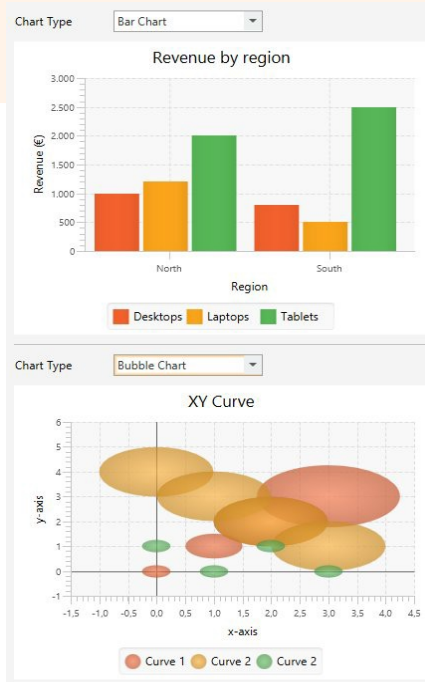
**Improvement - Page Bean Component “CCMatrixDashboard”**

In addition to the “CCDashboard” component there is now a “CCMatrixDashboard” component that arranges page beans within a matrix:









### Improvement - FX Client: Performance for FIELD, FORMATTEDFIELD, COMBOFIELD

There is a new option in the three components - CREATIONHINT. If set to "toggle" then the components are not immediately fully rendered, but are fully rendered on request only. This means: the content of the component is rendered as simple text output, if the user tabs or clicks into the component (focus is gained) then the component internally switches from simple text output to actually rendering the corresponding field.

Background: the TextField component in FX is a quite heavy one...

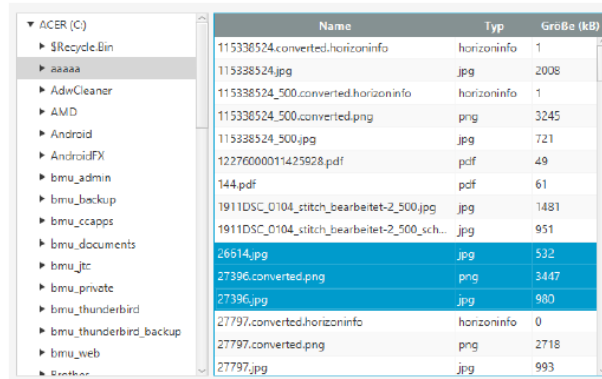
In "toggle" mode the component does NOT support all features of the normal mode. The "toggle" mode is optimized for usage within a grid cell. It e.g. does not support borders.

(We internally applied the performance improvements to our layout editor - seeing a significantly better build-up-time when editing pages....)



**Improvement - New component CLIENTFILEEXPLORER**

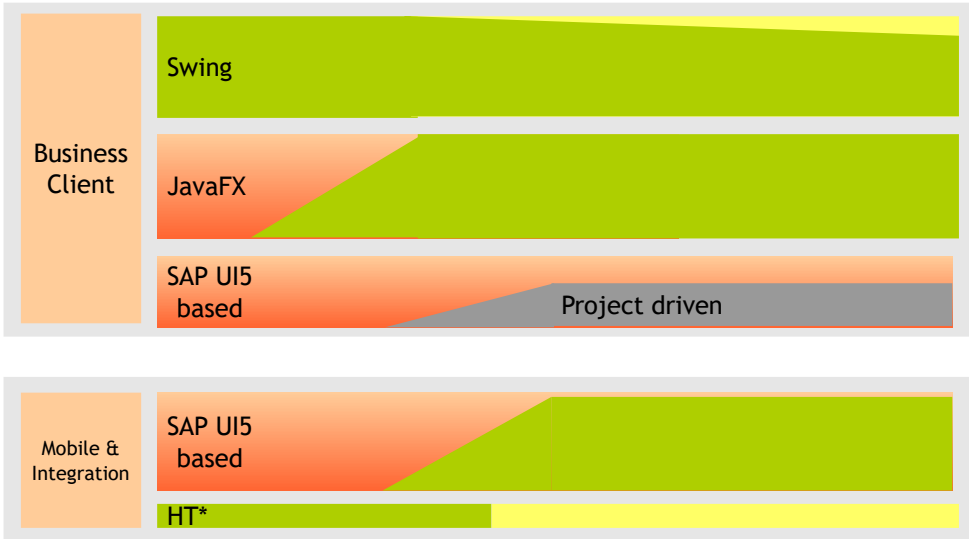
It is now possible to embed a “mini explorer” onto the client’s file system:



# Road Map 2016



# Roadmap 2015



# Roadmap 2016

Business Client	Swing
	JavaFX

Mobile & Integration	SAP UI5 based
	HT*



Application Components	
------------------------	--