

# Rich Web Applications in Server-side Java without Plug-ins or JavaScript

Joonas Lehtinen, PhD  
Vaadin Ltd - CEO  
[joonas@vaadin.com](mailto:joonas@vaadin.com)

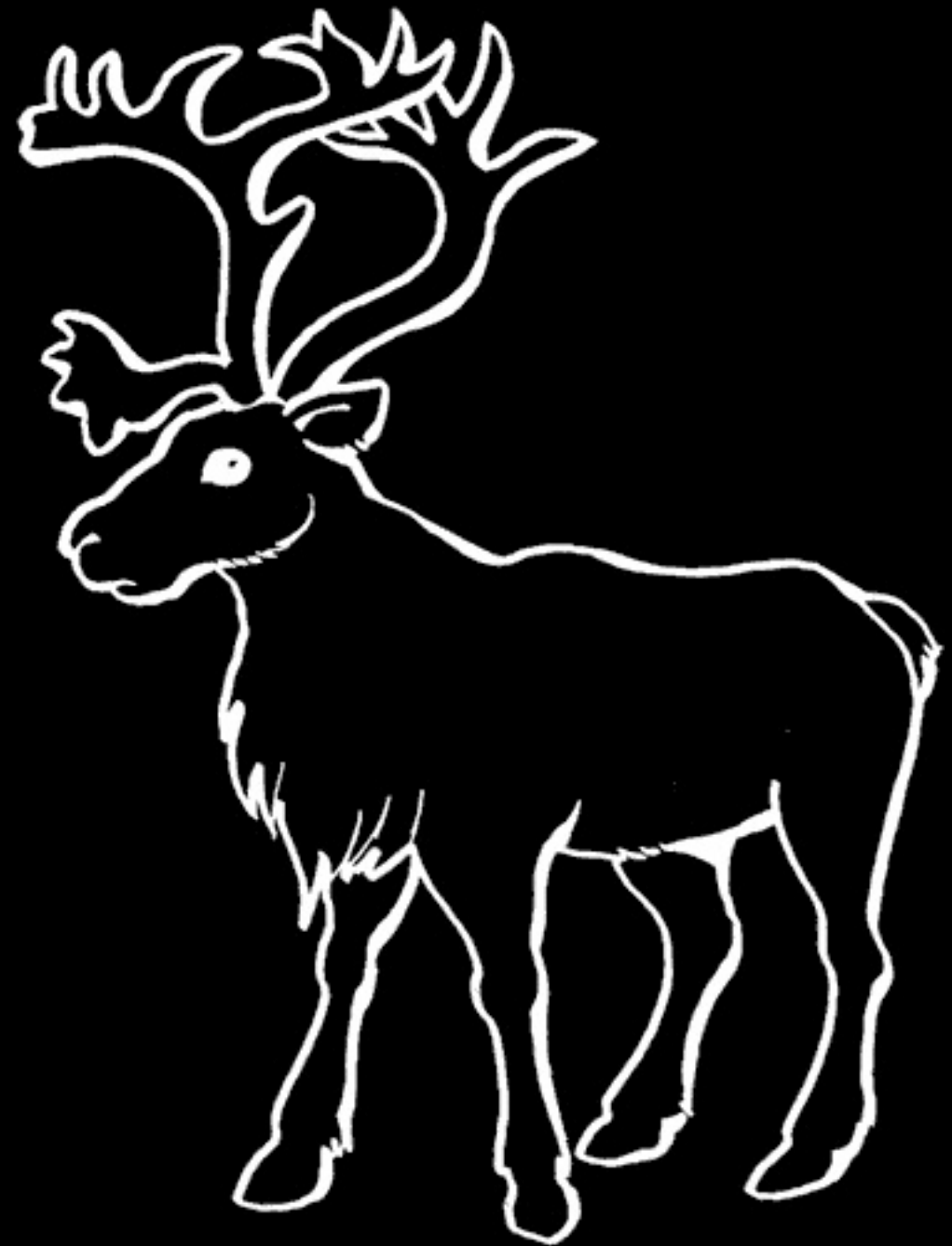
**vaadin** }>



**vaadin** }>

vaadin }>

vaadin }>





כ

v

v





**Vaadin is a  
UI framework  
for desktop-like  
web apps**

**Vaadin is a  
UI framework  
for desktop-like  
web apps**

**New configs,  
taglibs and  
syntax!?!**

**JavaScript,  
DOM, Applet,  
plugins?**



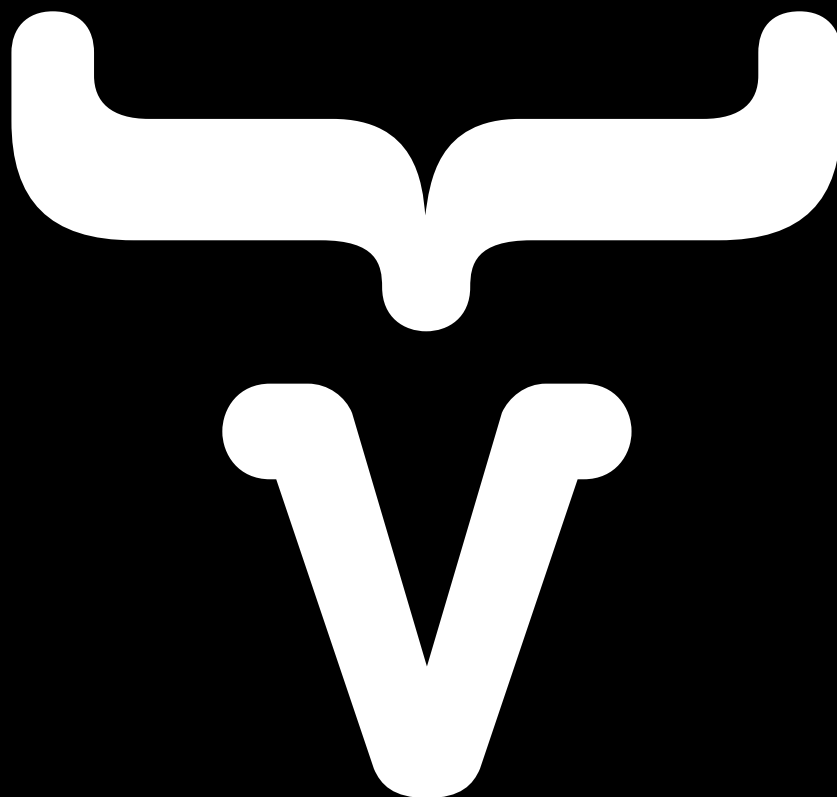


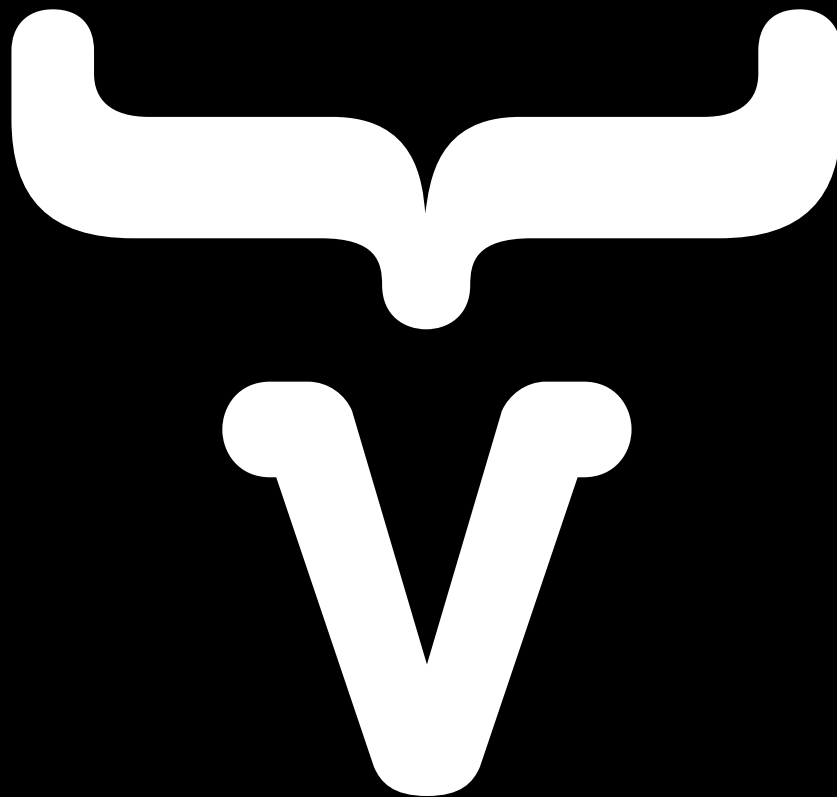
New configs,  
taglibs and  
syntax!?!  
sy

JavaScript,  
DOM, Ajax, et,  
plugins?  
**No!**

*This is Java.  
Nothing else.*









java } html >

java } html >



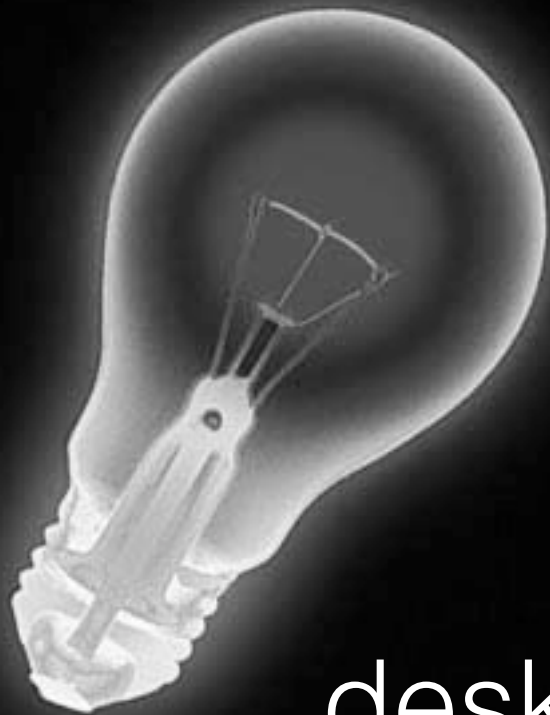
1998

healthcare portal, 100 kloc of perl, ..

web 1.0, netscape, ie5, ie6, ...



thinking of  
object oriented design, desktop, Java, U and I ...



desktop programming paradigm for web!



re-released as

2009

**vaadin }>**

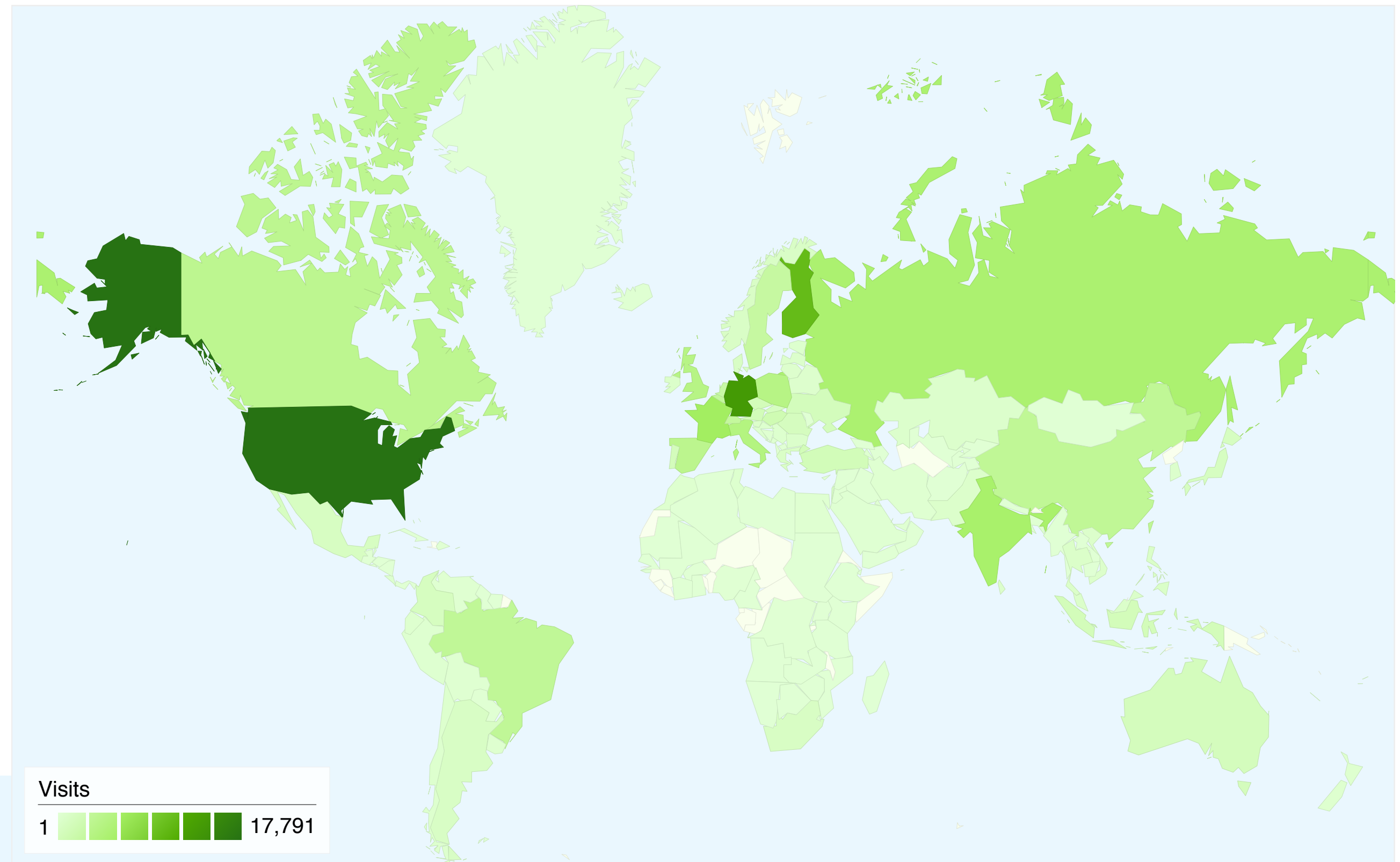


Vaadin is now  
21 months young  
and 10 years old

# Previous Month

Jan 12, 2011 - Feb 11, 2011

Comparing to: Site



133,294 visits came from 175 countries/territories

# Previous Month

Jan 12, 2011 - Feb 11, 2011

Comparing to: Site

1.	United States	17,791
2.	Germany	13,612
3.	Finland	10,438
4.	France	6,075
5.	India	5,557
6.	Russia	5,286
7.	Italy	4,444
8.	Poland	4,194
9.	United Kingdom	4,190
10.	Spain	3,717
16.	Sweden	2,688

Visits

1  17,791

133,294 visits came from 175 countries/territories

**is there a**



**?**





**Apache  
License**

# Contents

## Server-side RIA

What is it? Pros & cons?

# Contents

## Server-side RIA

What is it? Pros & cons?



## In practice

Coding a Vaadin  
application step-by-step



# Contents

## Server-side RIA

What is it? Pros & cons?



## In practice

Coding a Vaadin  
application step-by-step



## Vaadin

Big picture, Extending, Getting started

# Contents

## Server-side RIA

What is it? Pros & cons?

## Discussion



## In practice

Coding a Vaadin  
application step-by-step

## Vaadin

Big picture, Extending, Getting started



# Server-side RIA



# Rich Internet Application

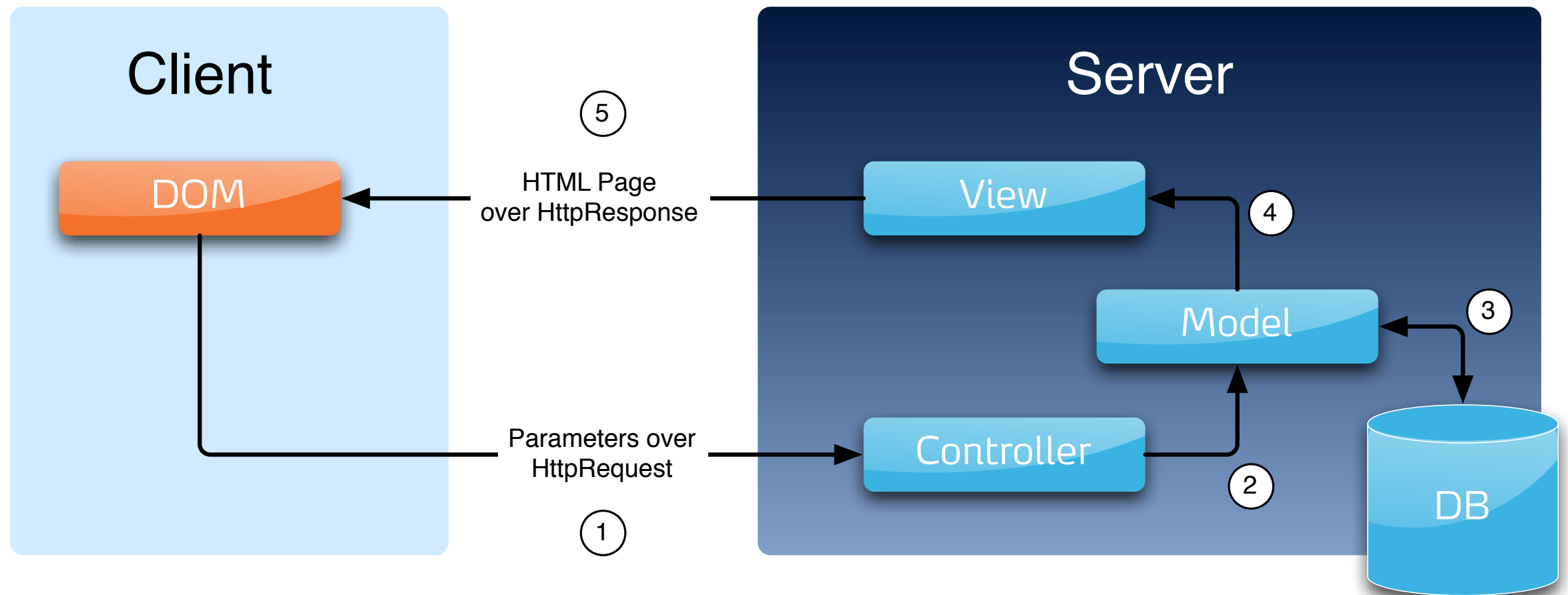
application that makes  
developers **rich**

if they bill by **hour**

# Architecture



# “Web 1.0”



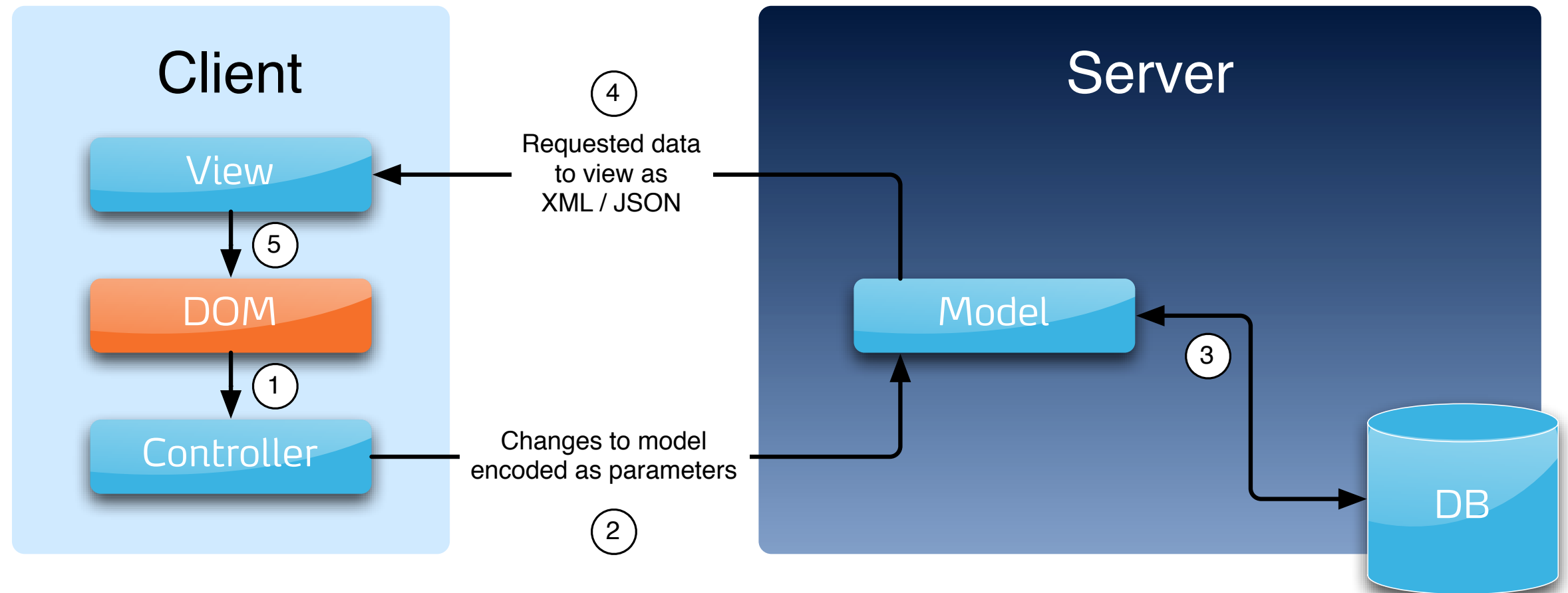




client-side ria

**fat client**

# Client-side RIA



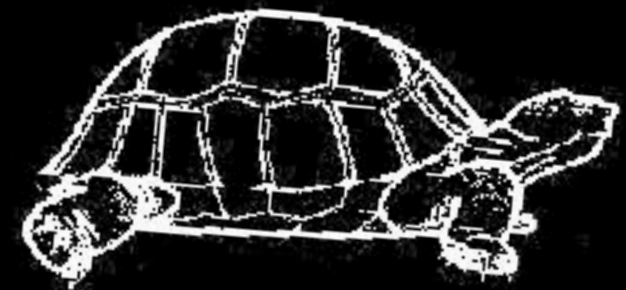
challenge  
web is  
not easy



different  
**features**  
in different  
browsers

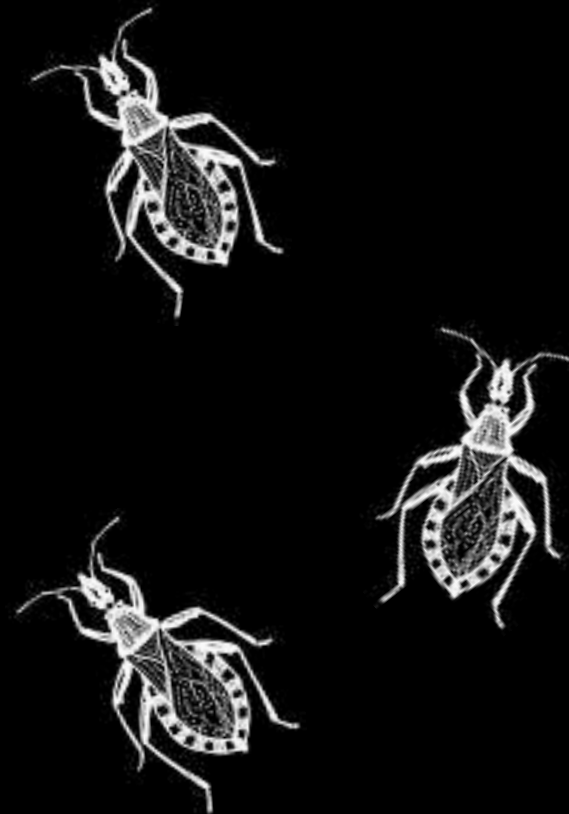


different  
**performance**  
in different  
browsers



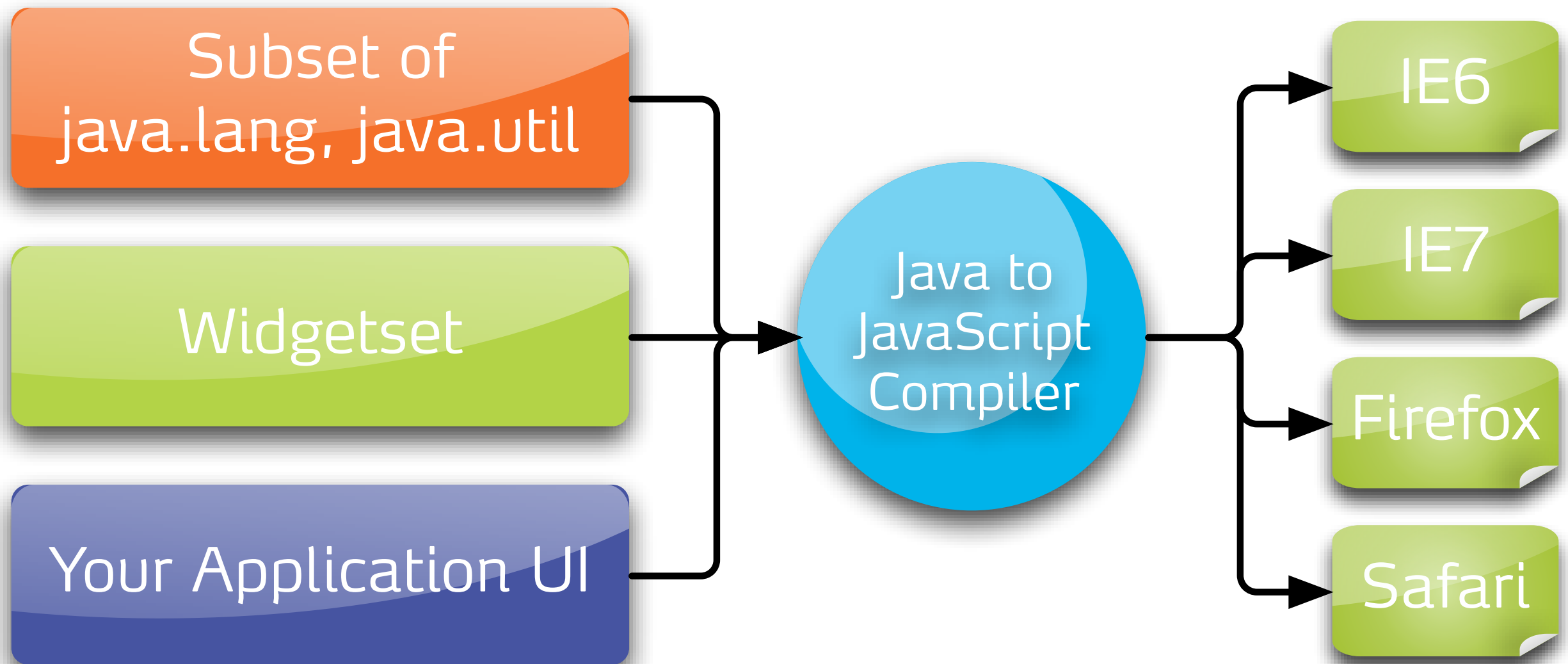


# different **bugs** in different browsers





Google Web Toolkit

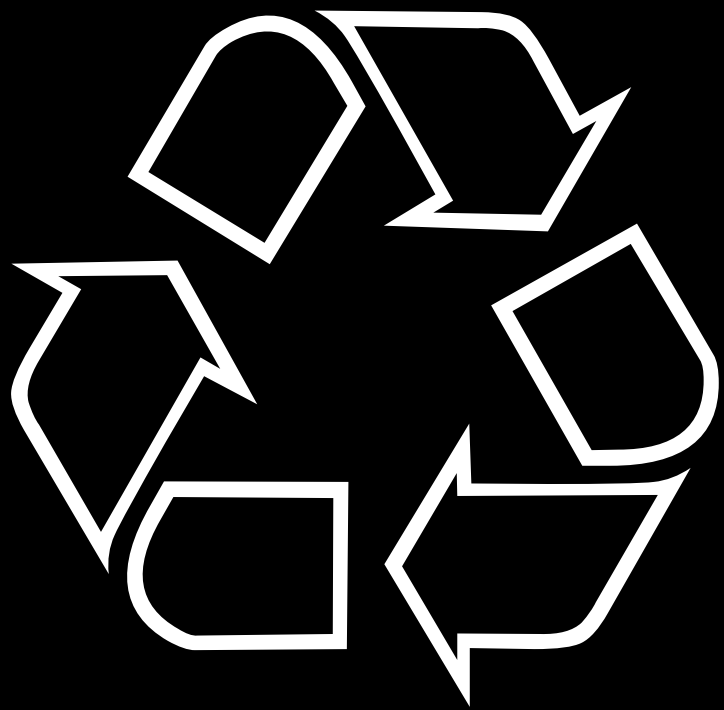




# simpler

- Java only
- forget the web

**cost-effective**  
stop debugging  
JavaScript spaghetti



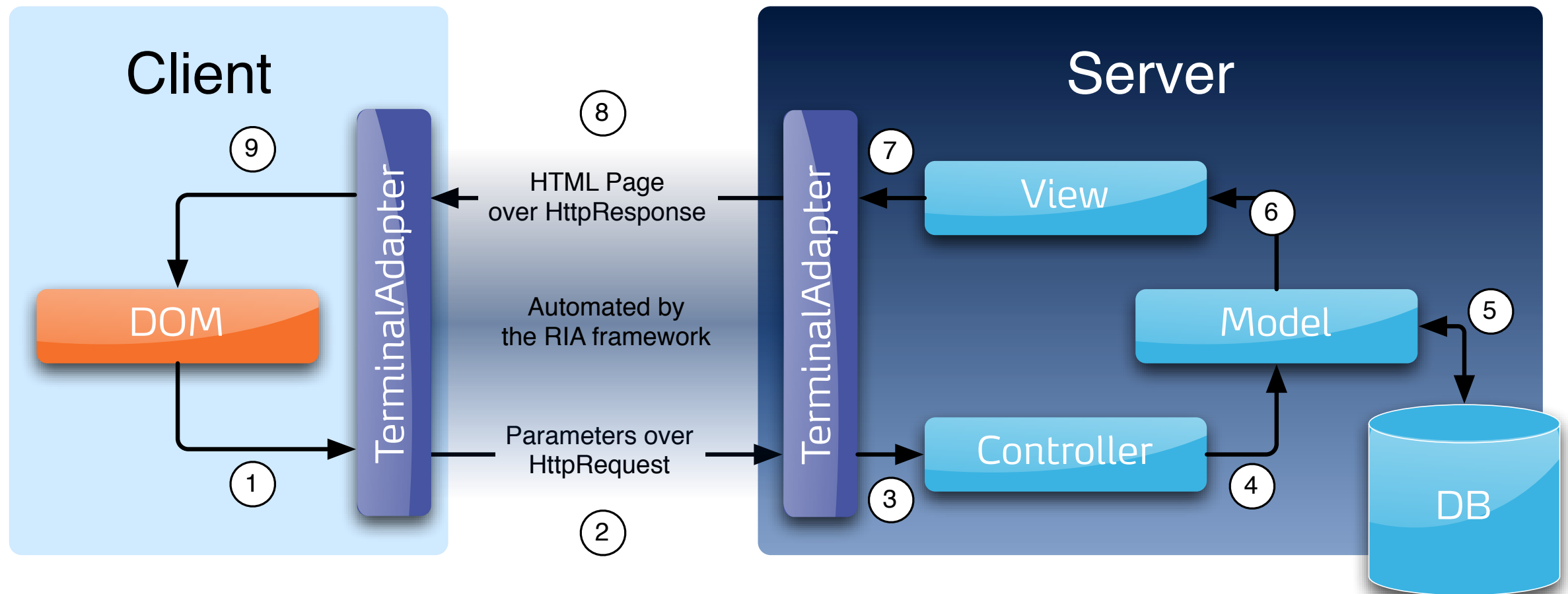
**modular  
extensible**

Building wonderful  
apps doesn't  
require writing **fat**  
web clients.

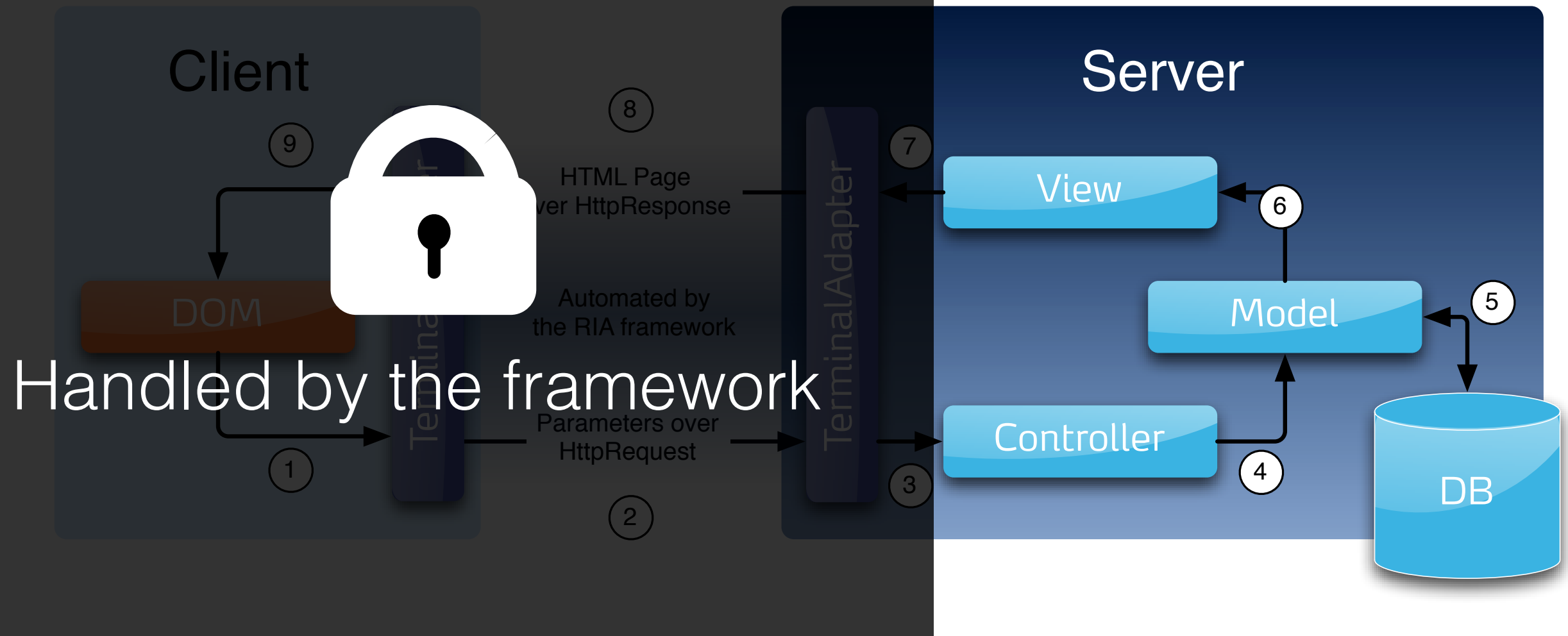
server-side ria  
thin client



# Server-side RIA



# Server-side RIA



# the benefits of Java

Java



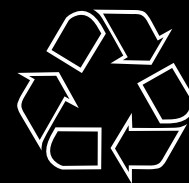
**simpler**

forget the web

**cost-effective**

no JavaScript  
debugging

**modular  
extensible**





# even simpler

- forget the client-side
- synchronous
- server resources

# more flexible

- all Java tools and libraries
- any JVM language

# more flexible

- all Java tools and libraries
- any JVM language

Scala

Groovy

# more secure

- code stays in server
- less web services

**not as scalable**

UI state is stored in  
the server memory

Measured 12.000  
active concurrent  
users per server  
for a ticketing app

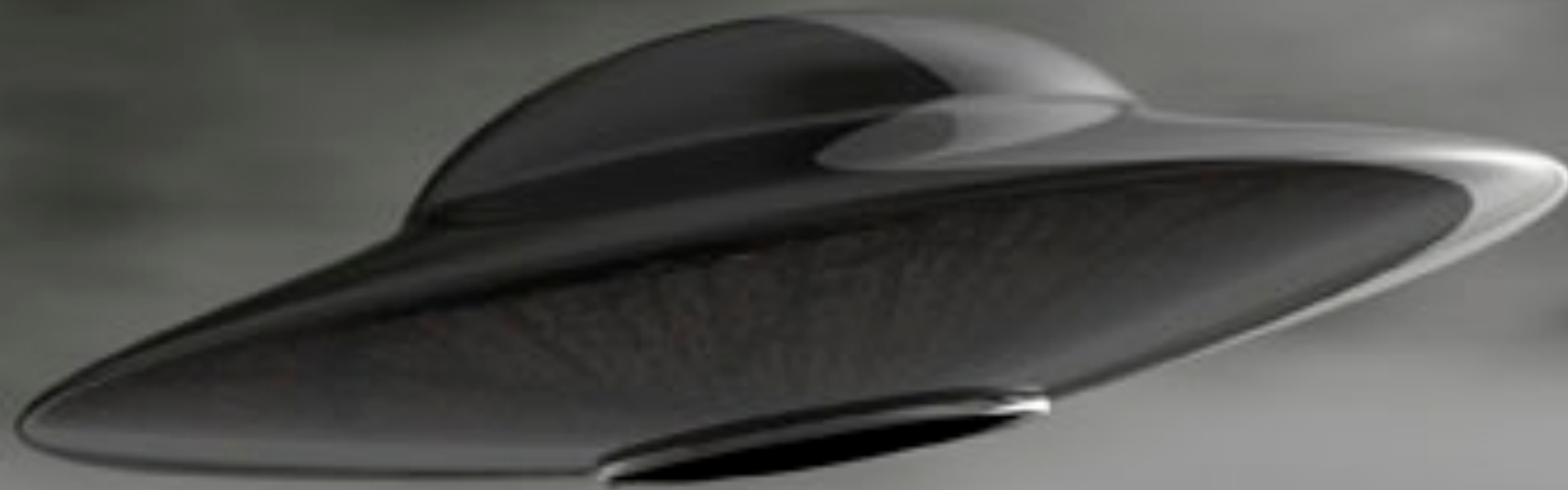
[Amazon EC2-large; limited by storage layer]

**no offline mode**  
server is always  
required



**#1 benefit**  
development  
is really fast





# Seeing is believing

**What we can do  
in 15 minutes?**

Docmgr Application

A

A

+

>

http://localhost:8080/DocMgr/

javaforum joonas

NAME	ICON	SIZE	LAST MODIFIED
Application.vwindowDetachEvent.html	icon/mies/mie.gif	7691	Tue Nov 24 13:36:39 EET 2009
Application.WindowDetachListener.html	icon/files/file.gif	8324	Tue Nov 24 13:36:39 EET 2009
Application.html	icon/files/file.gif	40555	Tue Nov 24 13:36:39 EET 2009
ApplicationContext.TransactionListener.html	icon/files/file.gif	12628	Tue Nov 24 13:36:47 EET 2009
ApplicationContext.html	icon/files/file.gif	12289	Tue Nov 24 13:36:47 EET 2009
ApplicationResource.html	icon/files/file.gif	11406	Tue Nov 24 13:36:47 EET 2009

Methods in [com.vaadin](#) that return [Application](#)

<a href="#">Application</a>	<b>Application.UserChangeEvent.<a href="#">getApplication()</a></b> Gets the application where the user change occurred.
<a href="#">Application</a>	<b>Application.WindowDetachEvent.<a href="#">getApplication()</a></b> Gets the application from which the window was detached.
<a href="#">Application</a>	<b>Application.WindowAttachEvent.<a href="#">getApplication()</a></b> Gets the application to which the window was attached.

Constructors in [com.vaadin](#) with parameters of type [Application](#)

<b><a href="#">Application.UserChangeEvent</a></b> ( <a href="#">Application</a> source, <a href="#">Object</a> newUser, <a href="#">Object</a> prevUser) Constructor for user change event.
---

Uses of [Application](#) in [com.vaadin.service](#)

Methods in [com.vaadin.service](#) with parameters of type [Application](#)

void	<b>ApplicationContext.TransactionListener.<a href="#">transactionEnd</a></b> ( <a href="#">Application</a> application, <a href="#">Object</a> transactionData) Invoked at the end of every transaction.
void	<b>ApplicationContext.TransactionListener.<a href="#">transactionStart</a></b> ( <a href="#">Application</a> application, <a href="#">Object</a> transactionData) Invoked at the beginning of every transaction.

vaadin >



# Vaadin Framework



# Great UI Components



# Great UI Components

Vaadin Sampler

<http://demo.vaadin.com/sampler/>

Google

Home

All Samples

▼ UI Basics

Tooltips

Icons

Runo theme icons NEW

Error indicator

Progress indication NEW

JavaScript API NEW

▼ Buttons

Push button

Link button

Checkbox

▼ Links

Link

Link, configure window

Link, sized window

▼ Texts

Label, plain text

Label, preformatted

Label, rich text

▼ Embedding

Image NEW

Flash NEW

Web content NEW

▼ Value Input Components

▼ Dates

Pop-up date selection

UI Basics

18 SAMPLES

Tooltips

Icons

Runo theme icons

Error indicator

Progress indication

JavaScript API

Push button

Link button

Checkbox

Link

Link, configure window

Link, sized window

Label, plain text

Label, preformatted

Label, rich text

Image

Flash

Web content





**Combined power of**

- **Server-side RIA**
- **Google Web Toolkit**

# Combined power of

- Server-side RIA
- Google Web Toolkit

# Vaadin UI component architecture

# Vaadin UI component architecture

## “UI Component”

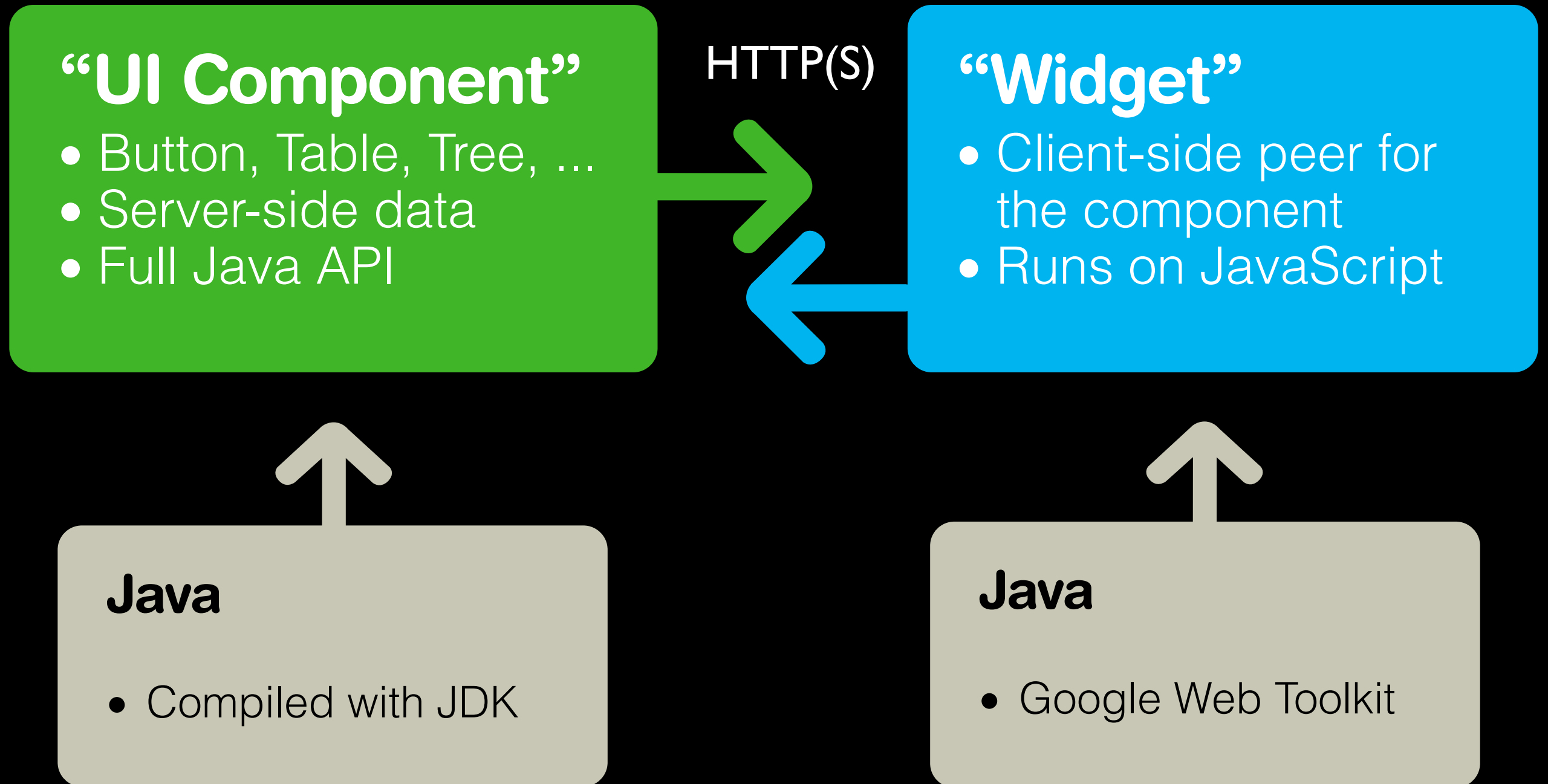
- Button, Table, Tree, ...
- Server-side data
- Full Java API

HTTP(S)

## “Widget”

- Client-side peer for the component
- Runs on JavaScript

# Vaadin UI component architecture



# Vaadin UI component architecture

## “UI Component”

- Button, Table, Tree, ...
- Server-side data
- Full Java API

HTTP(S)

## “Widget”

- Client-side peer for the component
- Runs on JavaScript

## Java

- Compiled with JDK

## Java

- Google Web Toolkit

Creating new UI  
components is  
really easy

**New Vaadin Widget**

**New Component wizard**  
This wizard creates a new Vaadin widget.

Source folder:

Package:

Name:

Superclass:

Template:

Simple client-side and server-side component with client-server communication



# Implement two classes

# Implement two classes

## Server-side

### “UI Component”

- Define **API**
- Receive client events
- Send UI updates back

Automatic

## Client-side


### “Widget”

- **Render** to DOM
- Collect user events

## Vaadin Add-on Package Export

Define which resources should be exported into the Vaadin add-on package.

Select the resources to export.

☒ ▶  test

Manifest:

Implementation title:

Name of the add-on. Used in Vaadin Directory.


Implementation version:

Version of the addon. A "major.minor.revision" format is suggested.

Widgetsets:


Comma separated list of widgetsets included in the add-on. Refers to the GWT xml files (.gwt.xml).

Select the export destination:

JAR file:  

Options:

☐ Overwrite existing files without warning



# Upload New Add-on

Select a category to post your new add-on to.

Note, that if you're updating a previous add-on, that is done by editing the add-on from the list above.



## UI Components



Server-side and/or client-side  
UI components



## Themes



Themes for Vaadin applications



## Data Components



Components related to the  
Vaadin data model, e.g.  
Container or Validator  
implementations



## Tools



Tools for Vaadin developers



## Miscellaneous



Other Vaadin add-ons

Upload Add-on Package

## Directory

Browse

All

UI Components

Data Components

Themes

Tools

Miscellaneous

Guest

[Authoring](#)

[Subscribe RSS](#)

[Help](#)

[FAQ](#)

[Feedback](#)

Most Recent Highest Rated Top Downloads

Showing

CERTIFIED STABLE BETA EXPERIMENTAL

« Previous

Next »

1

2

3

4

5

6

7

8

9

10

11

12

13

14

158 Results

### EasyUploads

In [UI Components](#) by [Matti Tahvonen](#)

Use file uploads as fields in Form, upload multiple files at once - easily!

Version 0.4.2 BETA

★★★★★ 2

↓ 304

### OpenLayers Wrapper

In [UI Components](#) by [Matti Tahvonen](#)

Vaadin server side components that wrap essential OpenLayers objects

Version 0.4.0 EXPERIMENTAL

★★★★★ 2

↓ 142

### I18N4Vaadin

In [Miscellaneous](#) by [Petter Holmström](#)

A small add-on for creating localized applications

Version 0.9.0 BETA

★★★★★ 1

↓ 10

### Navigator

In [UI Components](#) by [Joonas Lehtinen](#)

Navigator is an easy to use view manager that supports lazy initialization, bookmarking and multiple browser windows.

Version 0.3 EXPERIMENTAL

★★★★★ 3

↓ 234

### ConfirmDialog

In [UI Components](#) by [Run Uilder](#)

A versatile confirm dialog for Vaadin

Version 1.1.0 BETA

★★★★★ 6

↓ 676

### CustomField

In [UI Components](#) by [Henri Sara](#)

A form field whose presentation and logic can be customized

Version 0.8.2 BETA

★★★★★ 8

↓ 1075

### Drawer

In [UI Components](#) by [Henrik Paul](#)

An animated component to hide or show information

### Transactional Container

In [Data Components](#) by [Tommi Laukkanen](#)

Transactional Container offers same base features as Lazy Query Container but reads and writes all data



## Directory

### Browse

#### All

[UI Components](#)[Data Components](#)[Themes](#)[Tools](#)[Miscellaneous](#)[Guest](#)[Authoring](#)[Subscribe RSS](#)[Help](#)[FAQ](#)[Feedback](#)

## PaperStack

In [UI Components](#) by [Tomi Virkki](#) ★★★★★ 11 ↓ 194

[Report this add-on](#)**Version**

0.8.1 (latest)

**Maturity**

EXPERIMENTAL

**License**[Apache License 2.0](#)**Vaadin**

6.2 upwards

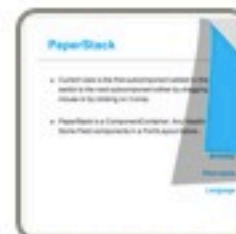
**Browser Compatibility**

### Overview

PaperStack is a component container whose subcomponents are presented sequentially, one subcomponent at a time. User can switch between the subcomponents by mouse dragging the upper right corner of a view revealing the underlying subcomponent simultaneously. The transition effect simulates leafing through a stack of papers.

### Highlights

```
1 package org.vaa
2
3 import com.vaad
4 import com.vaad
5
6 public class MyA
```

[Code Example](#)[Screenshot 2](#)[Screenshot 1](#)

### Release notes

0.8.1:

**Download Now**

Version 0.8.1 (86 kB)

**Maven POM**

### Related Links

- [Discussion Forum](#)
- [Online Demo](#)
- [Source Code](#)

### Share

 | [More...](#)[Permalink to this add-on:](#)<http://vaadin.com/addon/paperstack>

## Directory

### Browse

#### All

#### UI Components

#### Data Components

#### Themes

#### Tools

#### Miscellaneous

Guest

[Authoring](#)

[Subscribe RSS](#)

[Help](#)

[FAQ](#)

[Feedback](#)

## PaperStack

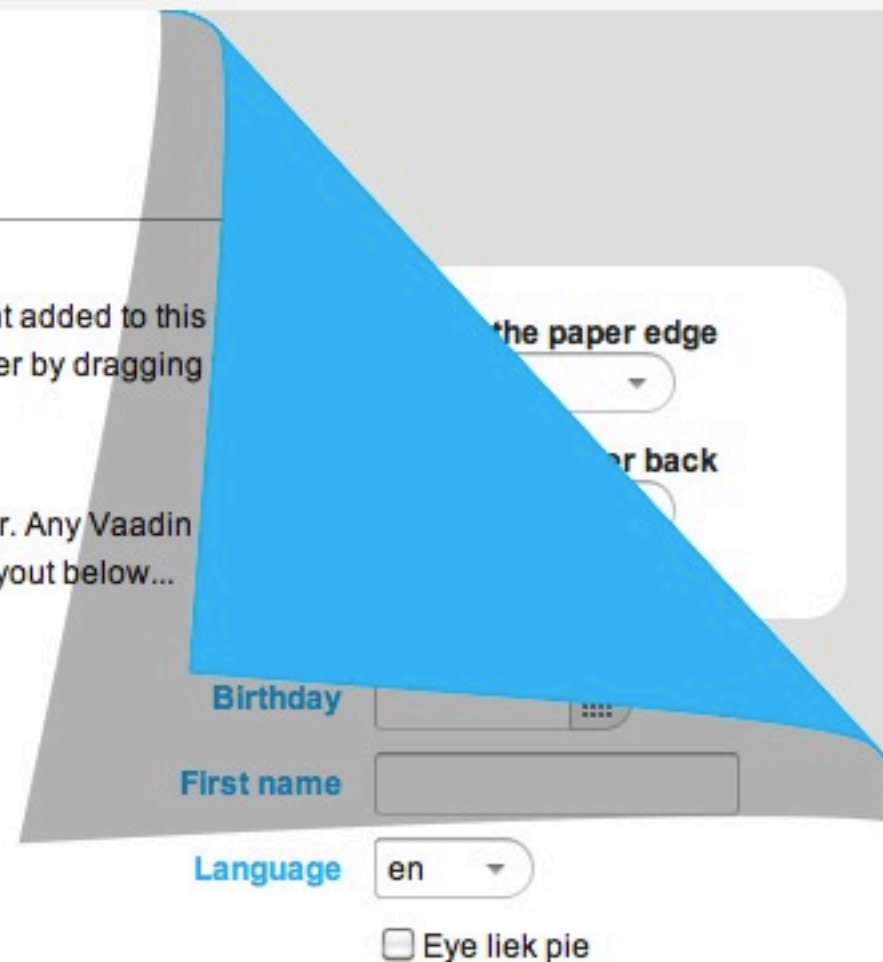
by [Tom Middeldijk](#) 44 5 404

[Report this add-on](#)

### PaperStack - Screenshot 1

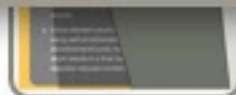
## PaperStack

- Current view is the first subcomponent added to this switch to the next subcomponent either by dragging mouse or by clicking on it once.
- PaperStack is a ComponentContainer. Any Vaadin Some Field components in a FormLayout below...



5  
6 `public class MyA`

Code Example



Screenshot 2



Screenshot 1

### Release notes

0.0.1:



## Directory

[Browse](#)[All](#)[UI Components](#)[Data Components](#)[Themes](#)[Tools](#)[Miscellaneous](#)[Guest](#)[Authoring](#)[Subscribe RSS](#)[Help](#)[FAQ](#)[Feedback](#)

## PaperStack

In [UI Components](#) by [Tomi Virkki](#) ★★★★★ 11 ↓ 194

[Report this add-on](#)

Version

0.8.1 (latest)

Maturity

EXPERIMENTAL

Browser Compatibility



Download Now

Version 0.8.1 (86 kB)



Maven POM



### PaperStack - Code Example

```
PaperStack paperStack = new PaperStack();

paperStack.addComponent(new Label("Hello!"));

InlineDateField inlineDateField = new InlineDateField();
inlineDateField.setResolution(DateField.RESOLUTION_DAY);
paperStack.addComponent(inlineDateField, "#999");

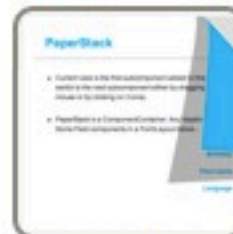
mainWindow.addComponent(paperStack);
```

```
1 package org.vaa
2
3 import com.vaad
4 import com.vaad
5
6 public class MyA
```

Code Example



Screenshot 2



Screenshot 1

### Release notes

0.8.1:

### Related Links

[Discussion Forum](#)

[Online Demo](#)

[Source Code](#)

e

More...

Permalink to this add-on:

<http://vaadin.com/addon/paperstack>



## Directory

### Browse

#### All

[UI Components](#)[Data Components](#)[Themes](#)[Tools](#)[Miscellaneous](#)[Guest](#)[Authoring](#)[Subscribe RSS](#)[Help](#)[FAQ](#)[Feedback](#)

## PaperStack

In [UI Components](#) by [Tomi Virkki](#) ★★★★★ 11 ↓ 194[Report this add-on](#)

Version

0.8.1 (latest)

Maturity

EXPERIMENTAL

License

[Apache License 2.0](#)

Vaadin

6.2 upwards

Browser Compatibility



### Overview

PaperStack is a component container whose subcomponents are presented sequentially, one subcomponent at a time. User can switch between the subcomponents by mouse dragging the upper right corner of a view revealing the underlying subcomponent simultaneously. The transition effect simulates leafing through a stack of papers.

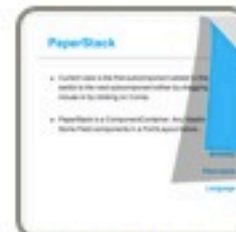
### Highlights

```
1 package org.vaa
2
3 import com.vaad
4 import com.vaad
5
6 public class MyA
```

Code Example



Screenshot 2



Screenshot 1

### Release notes

0.8.1:

[Download Now](#)

Version 0.8.1 (86 kB)

[Maven POM](#)

```
<dependency>
  <groupId>org.vaadin.addons</groupId>
  <artifactId>paperstack</artifactId>
  <version>0.8.1</version>
</dependency>
```

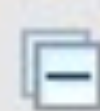
```
<repository>
  <id>vaadin-addons</id>
  <url>http://maven.vaadin.com/vaadin-
  addons</url>
</repository>
```

### Related Links

- [Discussion Forum](#)
- [Online Demo](#)
- [Source Code](#)

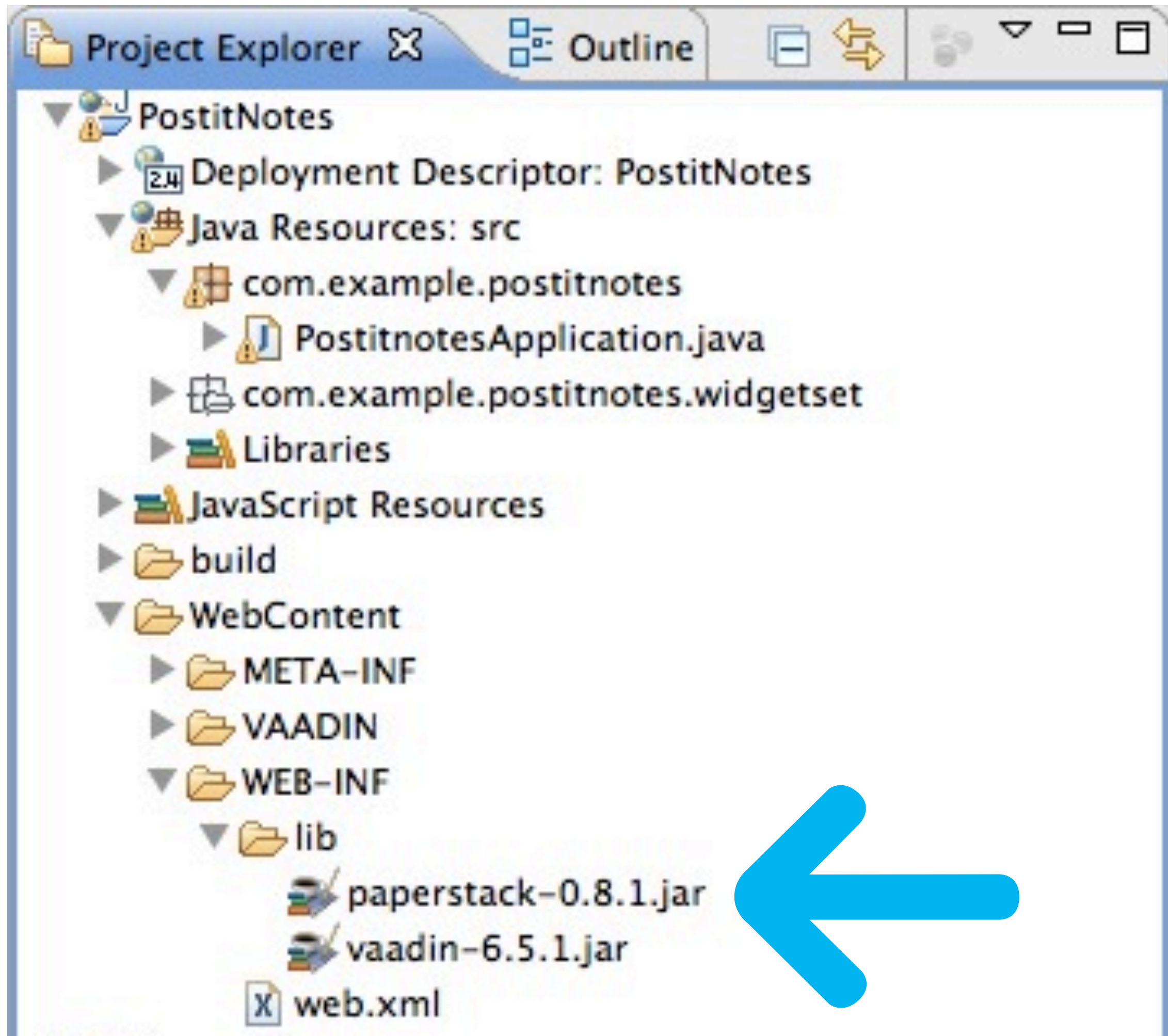
### Share

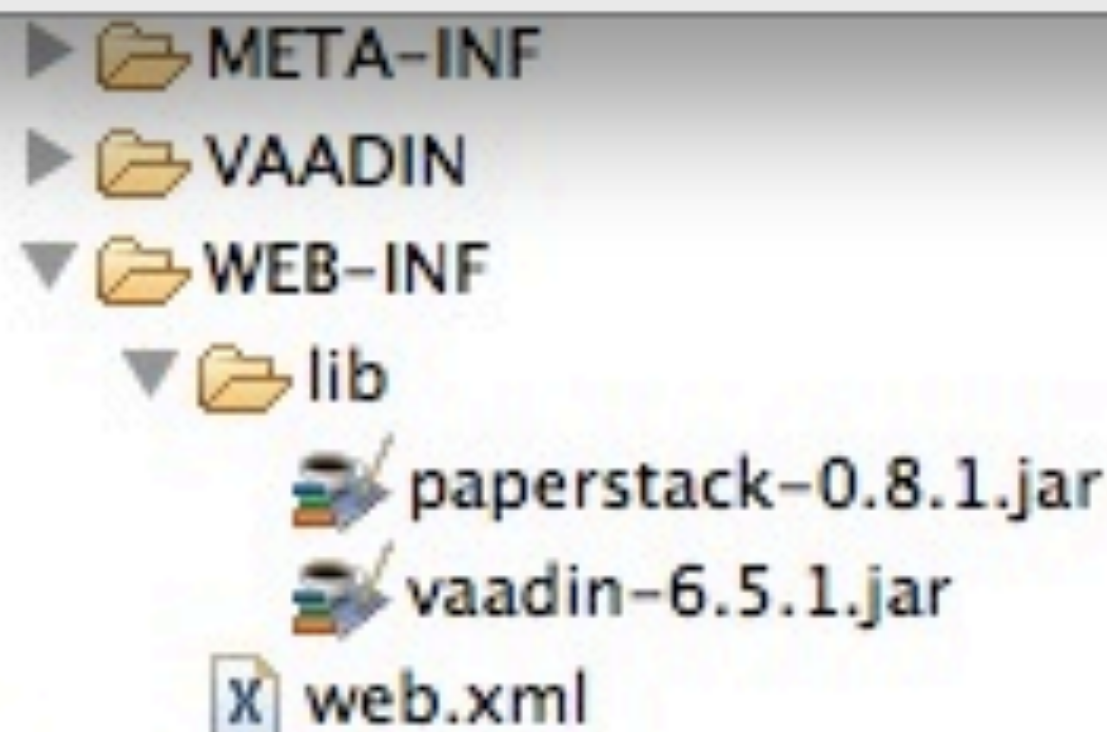
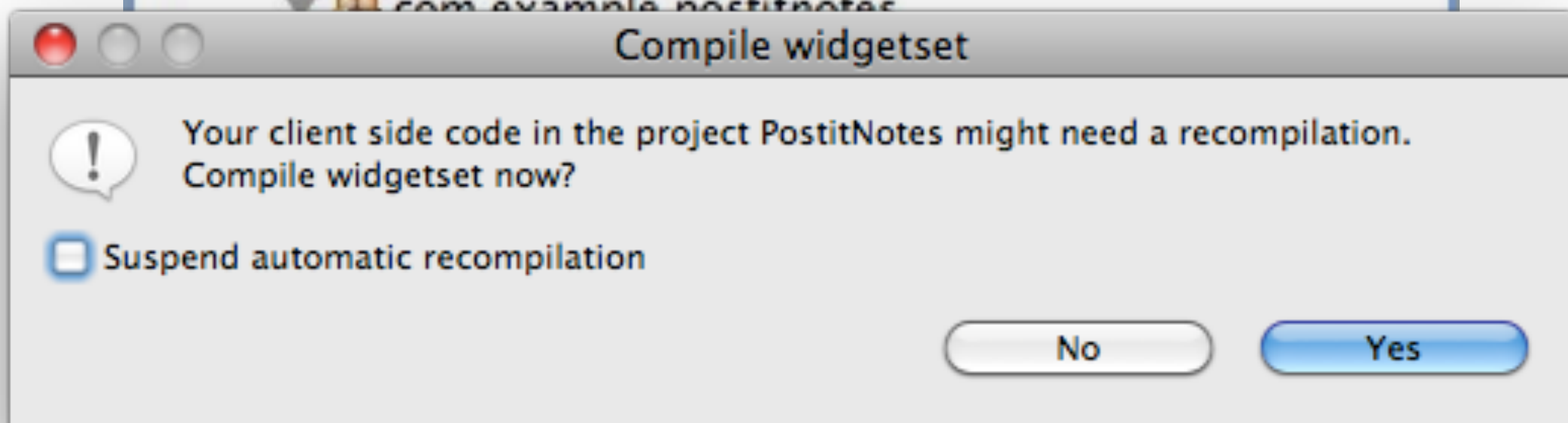
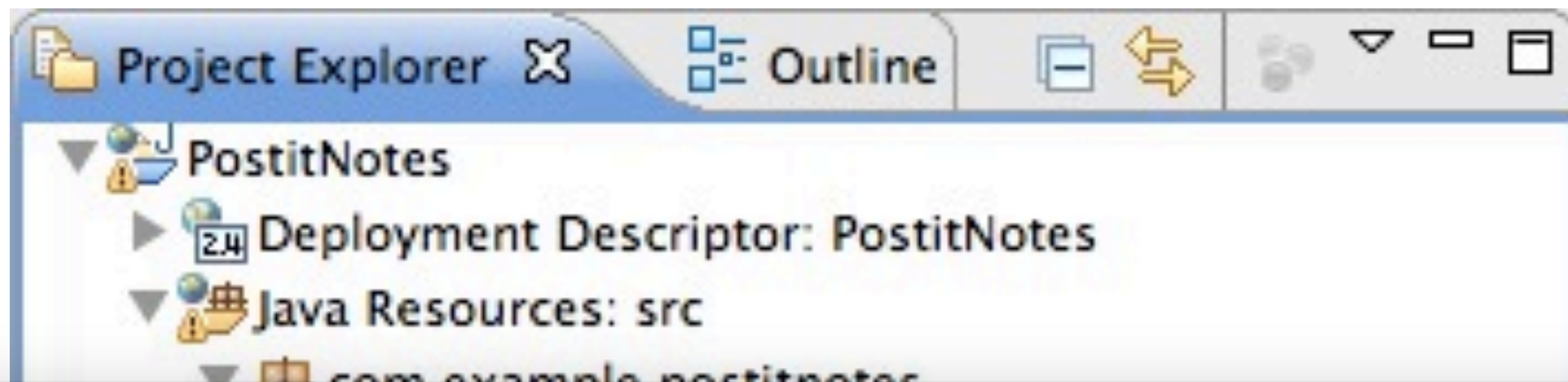
[Permalink to this add-on:](#)



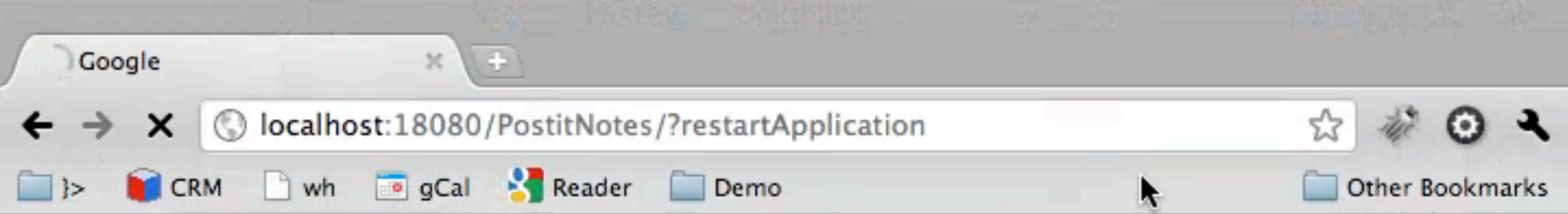
- ▼ PostitNotes
  - ▶ Deployment Descriptor: PostitNotes
  - ▼ Java Resources: src
    - ▼ com.example.postitnotes
      - ▶ PostitnotesApplication.java
      - ▶ com.example.postitnotes.widgetset
      - ▶ Libraries
    - ▶ JavaScript Resources
    - ▶ build
  - ▼ WebContent
    - ▶ META-INF
    - ▶ VAADIN
    - ▼ WEB-INF
      - ▼ lib
        - paperstack-0.8.1.jar
        - vaadin-6.5.1.jar
      - web.xml







1 class  
34 lines of code






First class  
**Java** citizen



First class  
**Java** citizen



Servlet  
Portlet  
App Engine

A large, thick, gray closing curly brace (}) is positioned on the right side of the slide, spanning the vertical height of the text above it.


Eclipse  
Maven  
Netbeans  
Spring Roo





New Roo Project

Create a new Roo Project



Project name:

pizzashop

Top level package name:

org.vaadin.roo.pizzashop

Project type:

Standard

Description

Roo Installation

☒ Use default Roo installation (currently 'Roo 1.1.1.RELEASE')

☐ Use project specific Roo installation:

Install:

Roo 1.1.1.RELEASE

[Configure Roo Installations....](#)

Maven Support

Provider:

Full Maven build

Contents

☒ Use default location

☐ Use external location

Location:

/Users/phoenix/Documents/workspace-sts-2.5.2.RELEASE,

Browse...

Working sets

**persistence setup** --provider HIBERNATE  
--database HYPERSONIC\_IN\_MEMORY

**entity** --class ~.domain.Topping

**field** string --fieldName name --notNull

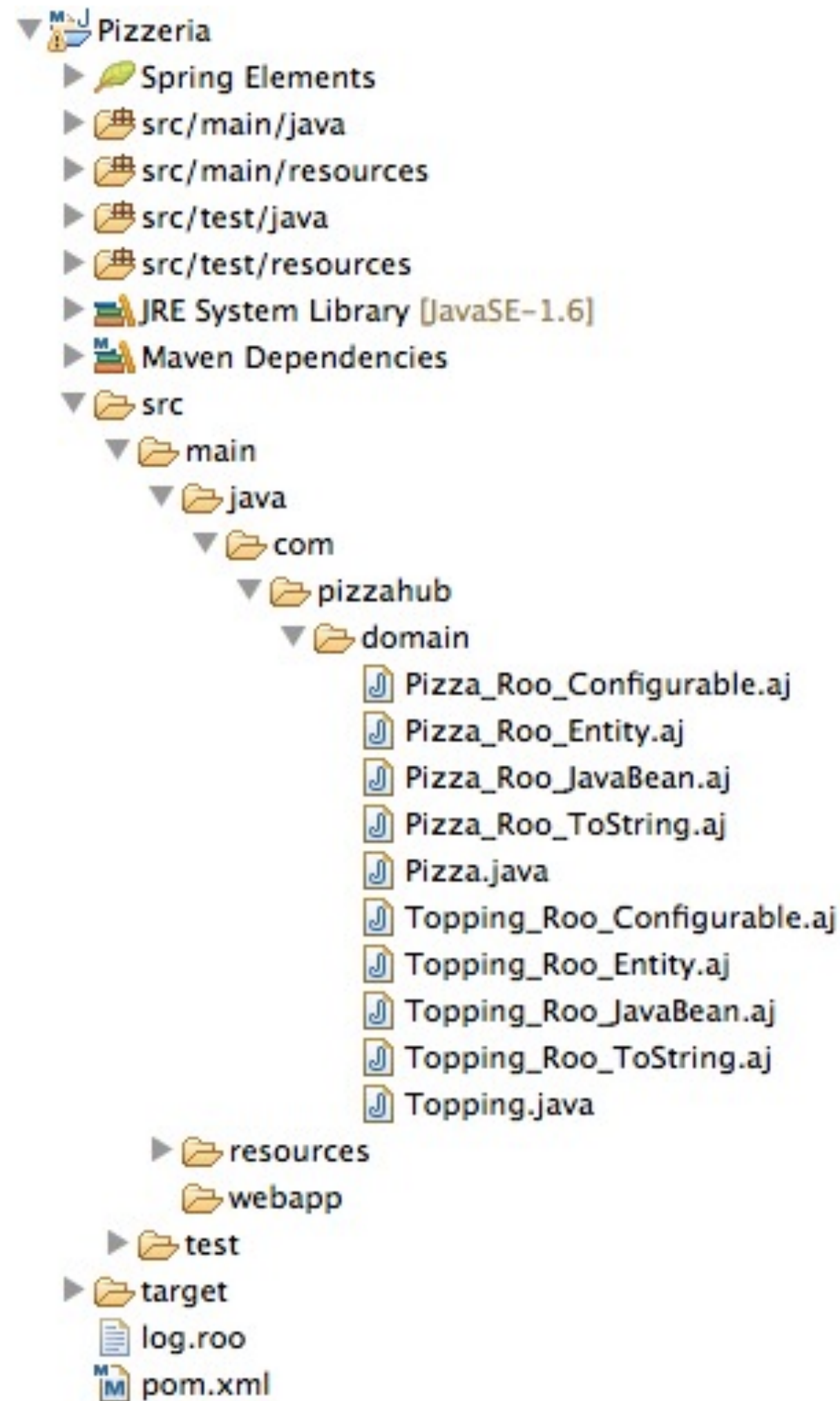
**entity** --class ~.domain.Pizza

**field** number --fieldName price

--type java.lang.Float

**field** set --fieldName toppings

--type ~.domain.Topping





```
package com.pizzahub.domain;

+ import org.springframework.roo.addon.entity.RooEntity;

@RooJavaBean
@RooToString
+ @RooEntity
public class Pizza {

    private Float price;

    @ManyToMany(cascade = CascadeType.ALL)
    private Set<Topping> toppings = new HashSet<Topping>();
}
```



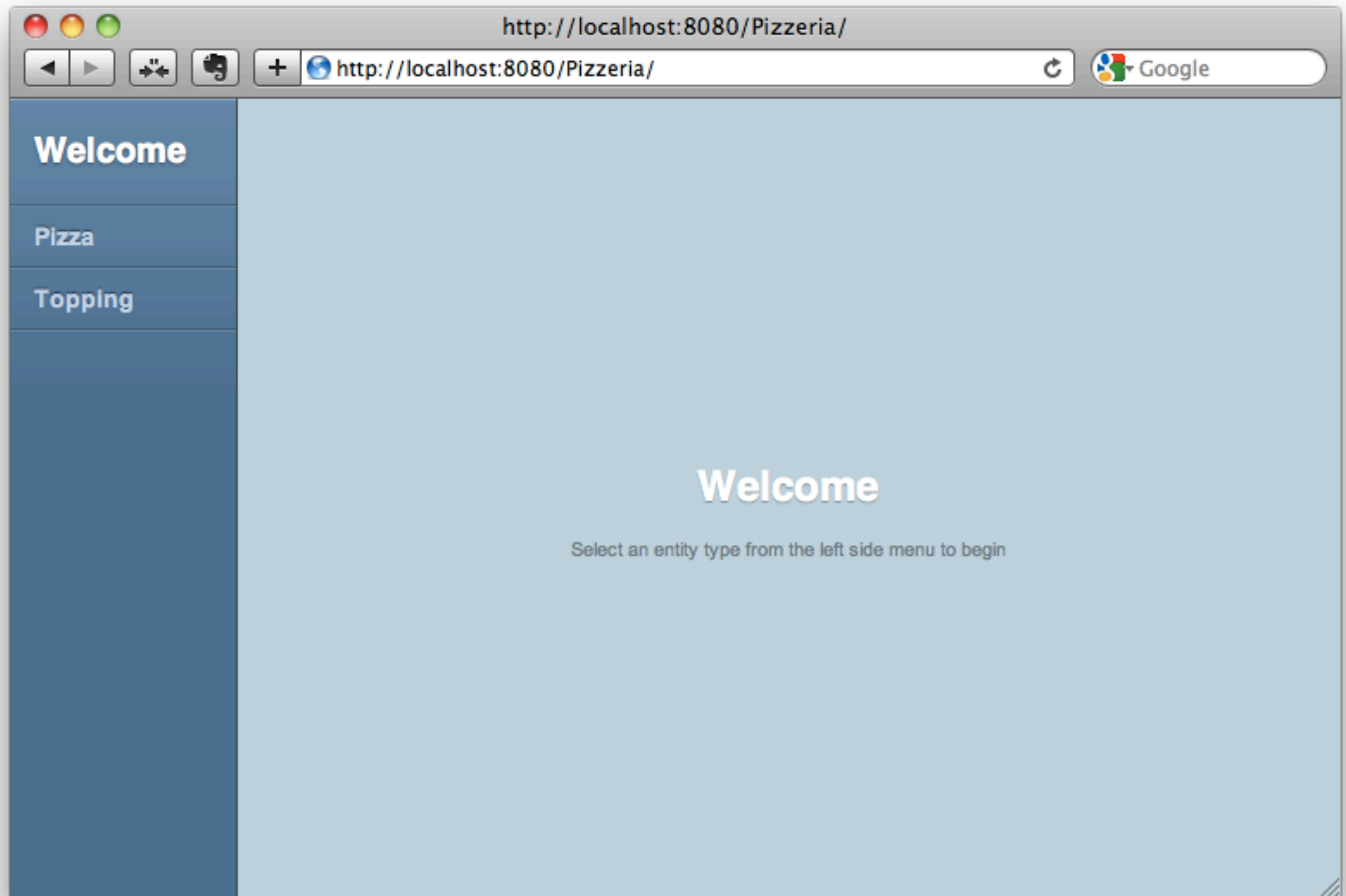
## vaadin setup

```
--applicationPackage ~.web  
--baseName PizzaShop  
--themeName pizza  
--useJpaContainer false
```

## vaadin generate all

```
--package ~.web.ui  
--visuallyComposable true
```

- ▼ web
  - AbstractEntityView\_Roo\_AbstractEntityView.aj
  - AbstractEntityView.java
  - AutomaticEntityForm.java
  - EntityEditor.java
  - EntityFieldWrapper.java
  - EntitySetFieldWrapper.java
  - EntityTableColumnGenerator.java
  - PizzaHubApplication.java
  - PizzaHubEntityManagerView\_Roo\_VaadinEntityManagerView.aj
  - PizzaHubEntityManagerView.java
  - PizzaHubWindow.java
- ▼ ui
  - PizzaForm\_Roo\_VaadinVisuallyComposableEntityForm.aj
  - PizzaForm.java
  - PizzaView\_Roo\_VaadinEntityView.aj
  - PizzaView.java
  - ToppingForm\_Roo\_VaadinVisuallyComposableEntityForm.aj
  - ToppingForm.java
  - ToppingView\_Roo\_VaadinEntityView.aj
  - ToppingView.java



Welcome

Pizza + New

Topping

ID	PRICE	TOPPINGS
1	10.0	Onion, Olive, Bacon, Pepper
2	8.0	Onion, Pepper, Tomato, Cheese, Mushroom

Price

6.0

Toppings

Onion  
Bacon  
Olive  
Pepper  
Salami  
Mushroom

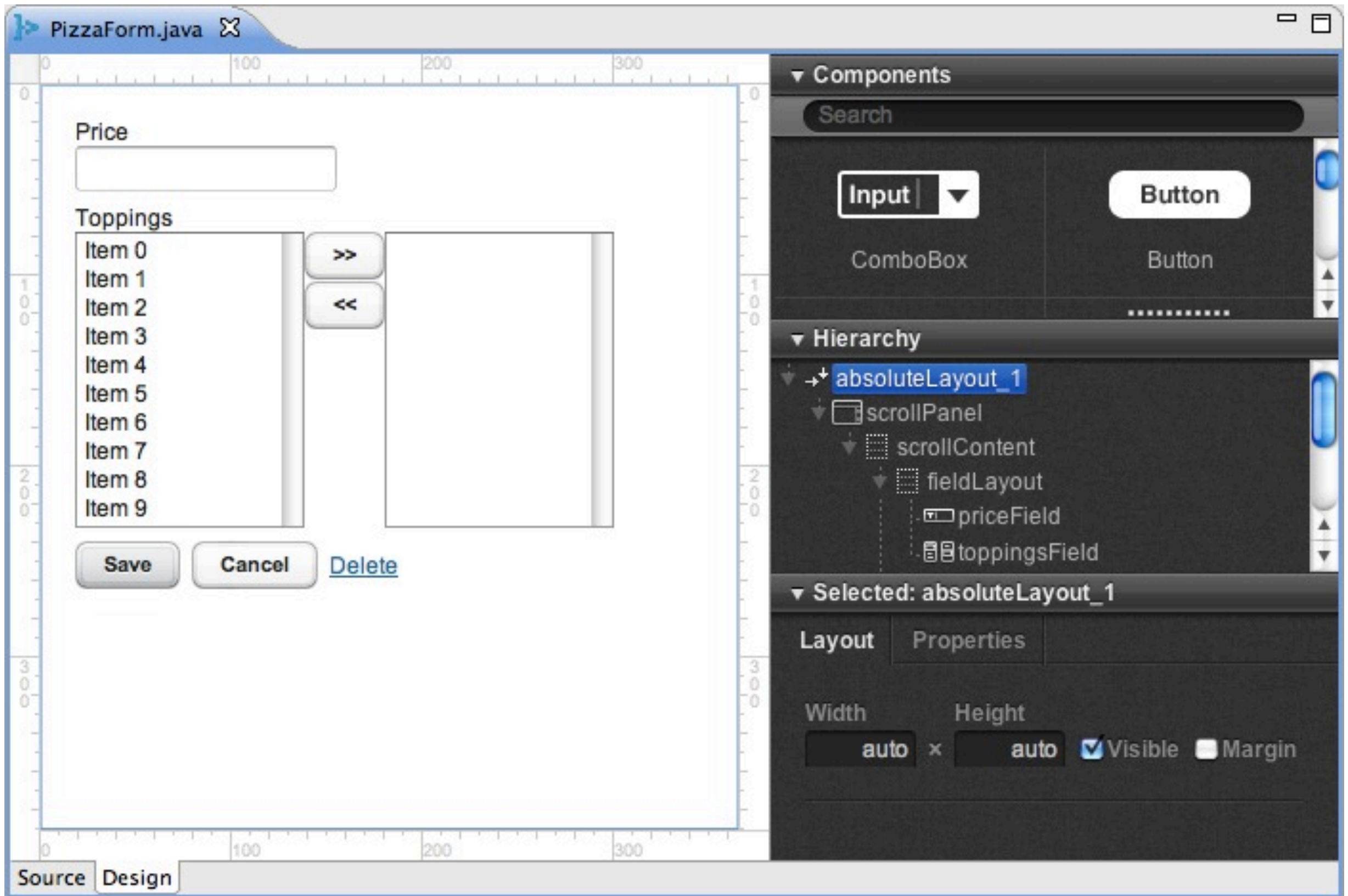
>>  
<<

Cheese  
Tomato

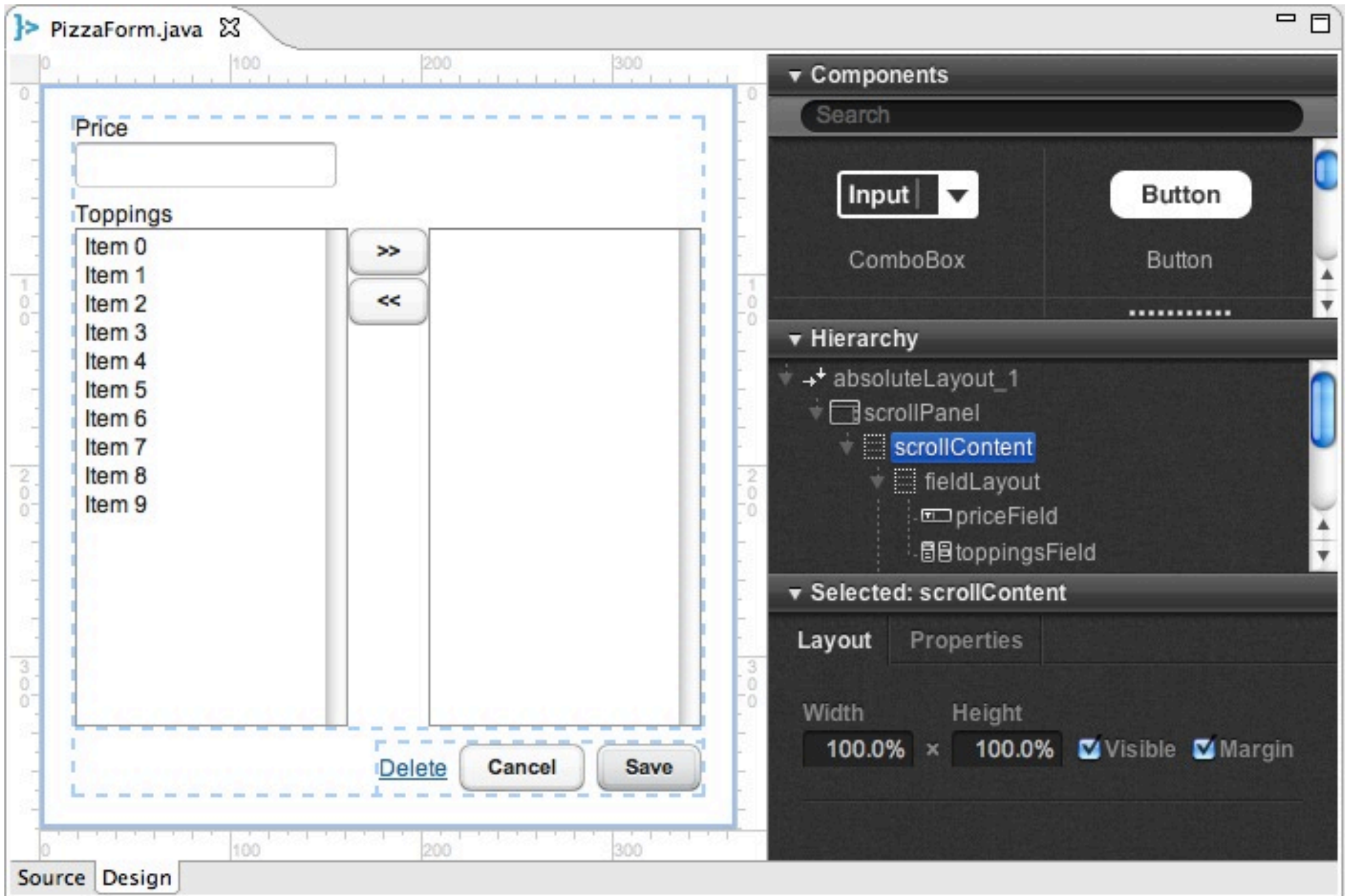
Save

Cancel

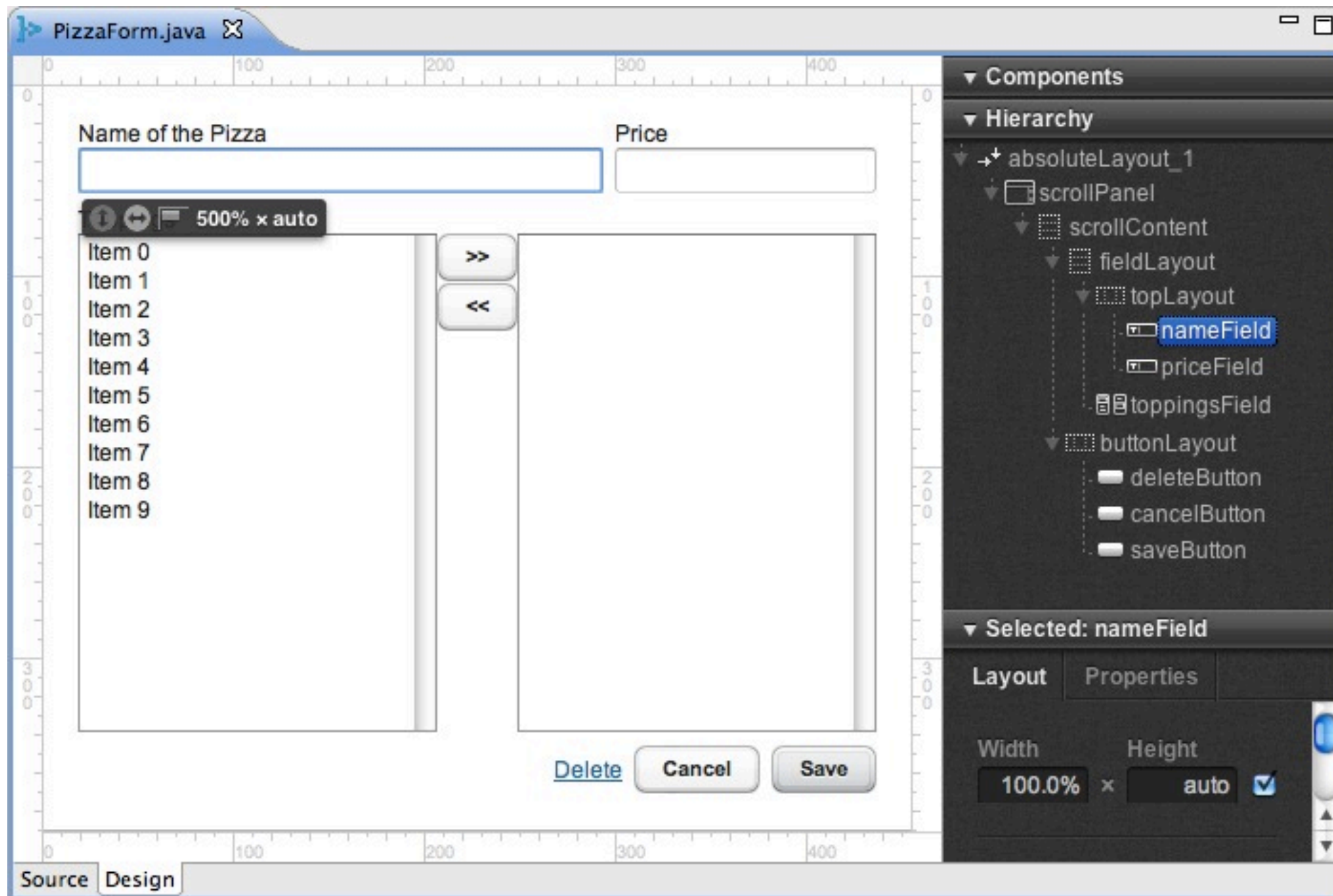
[Delete](#)







**field string --class ~.domain.Pizza**  
**--notNull --fieldName name --sizeMin 3**



ⓧ ⓧ ⓧ

⏪ ⏩

🔍

+

🌐 http://localhost:8080/Pizzeria/

↻

🔍 Google

Welcome

Pizza + New

Topping

ID	NAME	PRICE	TOPPINGS
1	Vaadin Special	8.0	mustard, onion, Olives, Tomato

Name of the Pizza \*  
Vaadin Special

Price  
8.0

Toppings

cheese  
Bacon  
Pepperoni  
Pepper

>>  
<<

onion  
mustard  
Olives  
Tomato

[Delete](#)

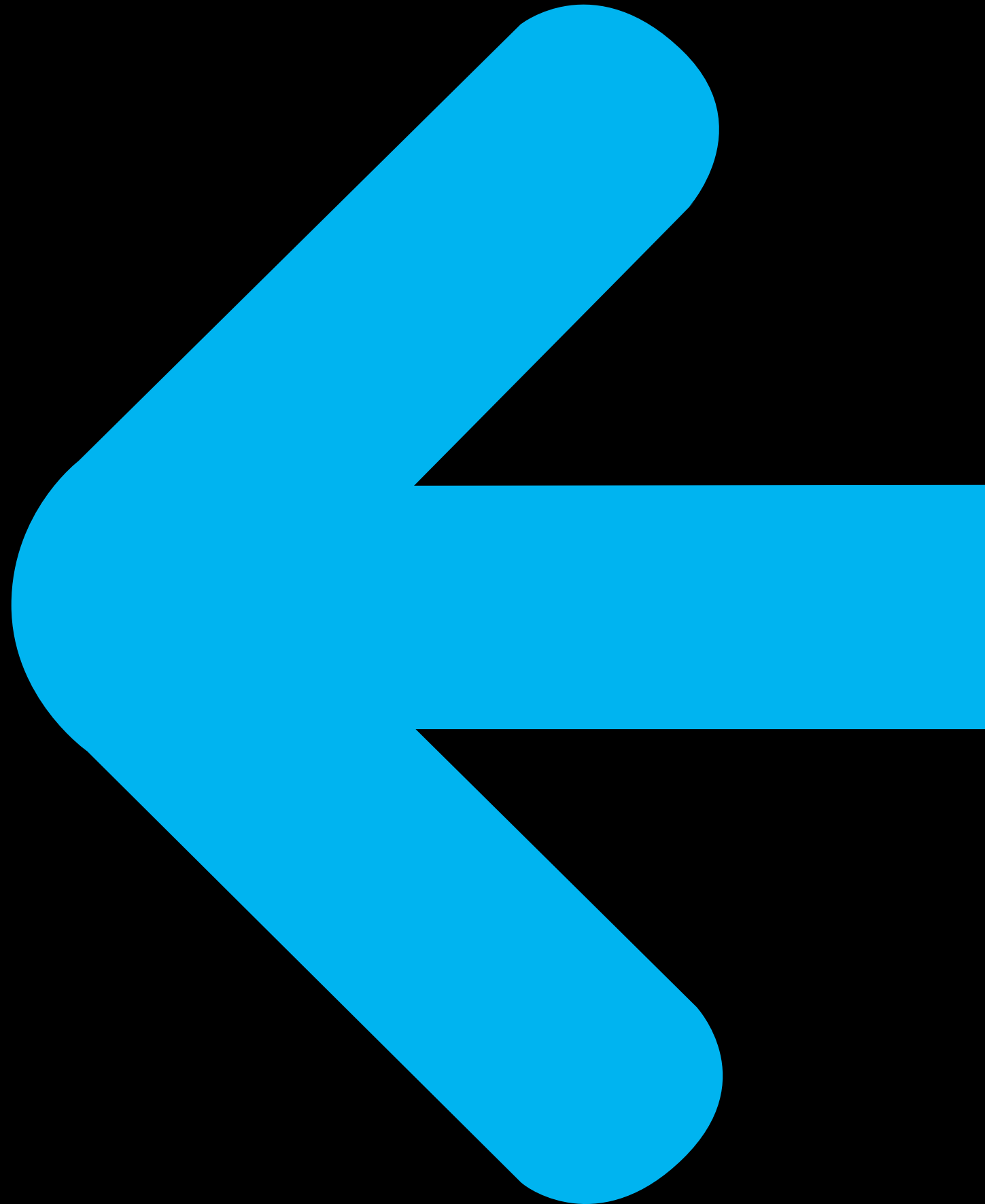
Cancel

Save



# getting started

getting  
started



# vaadin }> thinking of U and I

**Vaadin** is a Java framework for building **modern web applications** that look great, perform well and make you and your users happy.



## Demo

[Feature Sampler »](#)

[More Demos »](#)

[Code examples »](#)



## Download

Now Available

**Vaadin 6.5.1**

[Download Vaadin »](#)

[Add-ons - Tooling »](#)

[Source Code - Apache License »](#)



## Learn

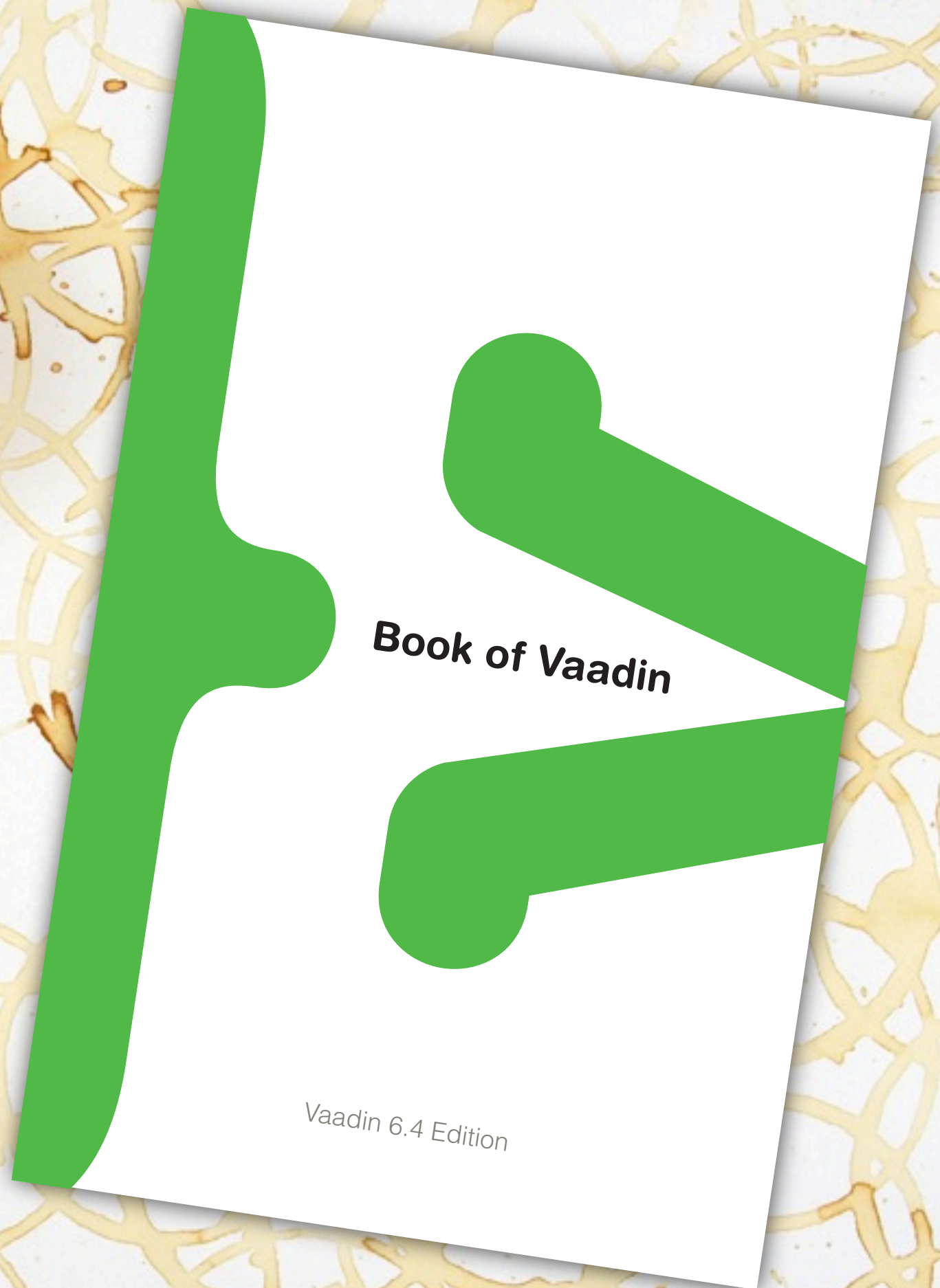
[30 Seconds to Vaadin »](#)

[Book of Vaadin - Tutorial »](#)

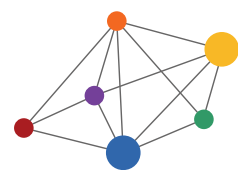
[FAQ - Forum - Recent posts »](#)



**GET A  
Free  
Copy**





**CONTENTS INCLUDE:**

- About Vaadin
- Creating An Application
- Components
- Layout Components
- Themes
- Data Binding and more...

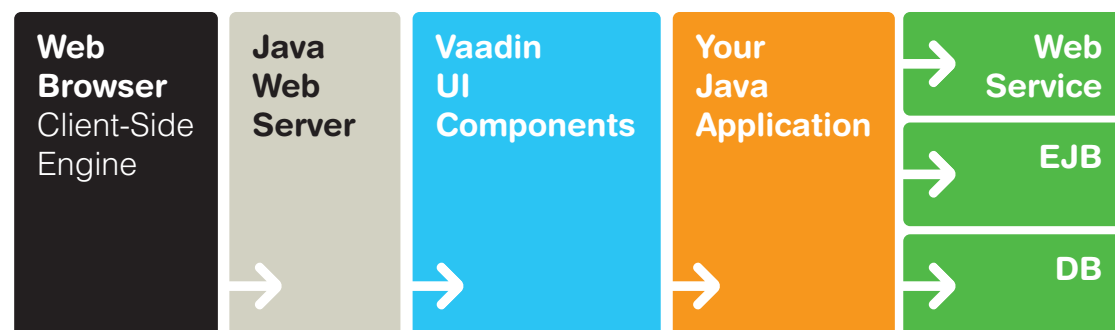
# Getting Started with Vaadin

By Marko Grönroos

## ABOUT VAADIN

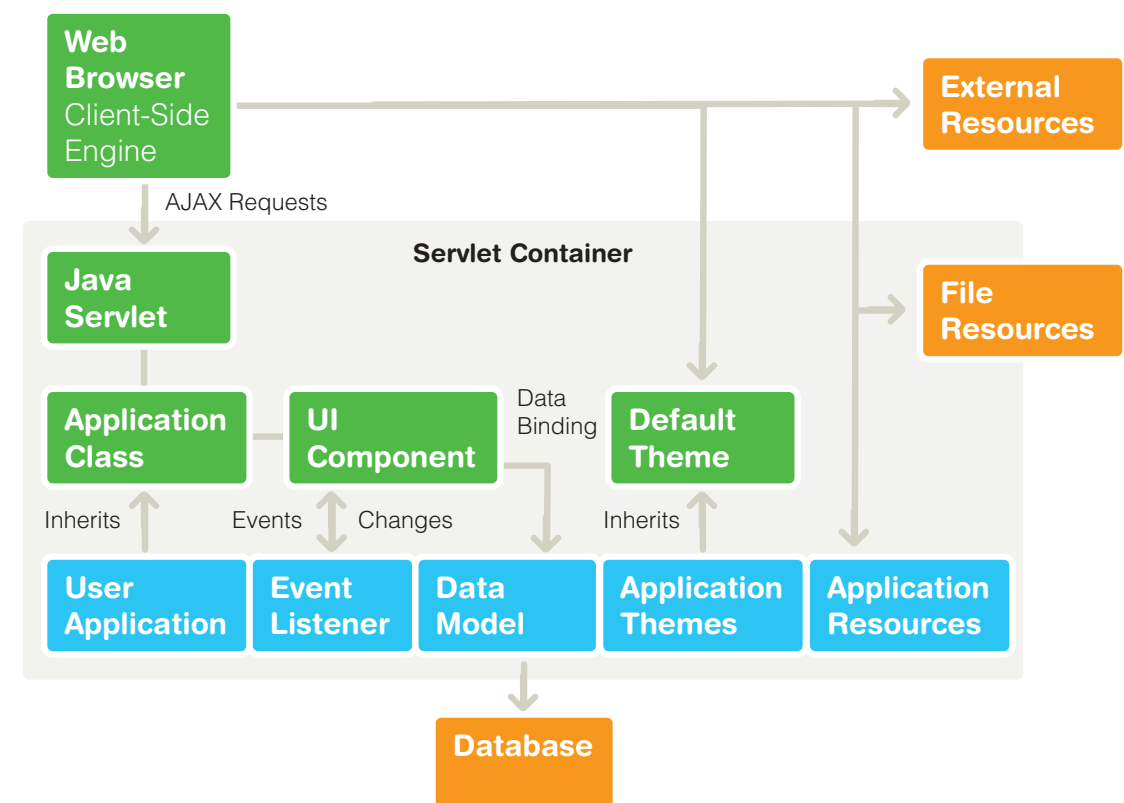
Vaadin is a server-side Ajax web application development framework that allows you to build web applications just like with traditional desktop frameworks, such as AWT or Swing. An application is built from user interface components contained hierarchically in layout components.

In the server-driven model, the application code runs on a server, while the actual user interaction is handled by a client-side engine running in the browser. The client-server communications and any client-side technologies, such as HTML and JavaScript, are invisible to the developer. As the client-side engine runs as JavaScript in the browser, there is no need to install plug-ins. Vaadin is released under the Apache License 2.0.



**Figure 1:** Vaadin Client-Server Architecture

If the built-in selection of components is not enough, you can extend the framework with the Google Web Toolkit (GWT).



**Figure 2:** Architecture for Vaadin Applications



You can get a reference to the application object from any component attached to the application with `getApplication()`

## Event Listeners

In the event-driven model, user interaction with user interface components triggers server-side events, which you can handle

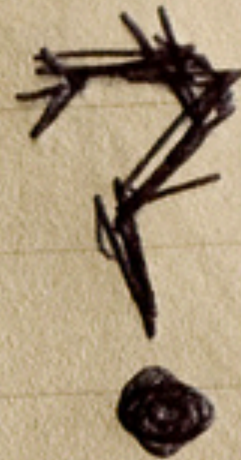
A world map showing the density of forum activity. The United States and parts of Europe are highlighted in dark green, indicating the highest activity levels. Other regions like Canada, Australia, and parts of Europe and Asia are in light green, while Africa and South America are in yellow, indicating lower activity levels. The text is overlaid on the map.

# Forums with 1000 msgs/m

**Ask the**  
[really active, world wide]  
**Community**



# Questions Comments



[joonas@vaadin.com](mailto:joonas@vaadin.com)

[vaadin.com/joonas](http://vaadin.com/joonas)

twitter: joonaslehtinen