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

Joonas Lehtinen, PhD
Vaadin Ltd - CEO
joonas@vaadin.com

twitter: #vaadin @joonaslehtinen

vaadin }>

vaadin }>

vaadin }>

Vaadin is a UI framework for desktop-like web apps

vaadin }>

New configs, taglibs and syntax!?!

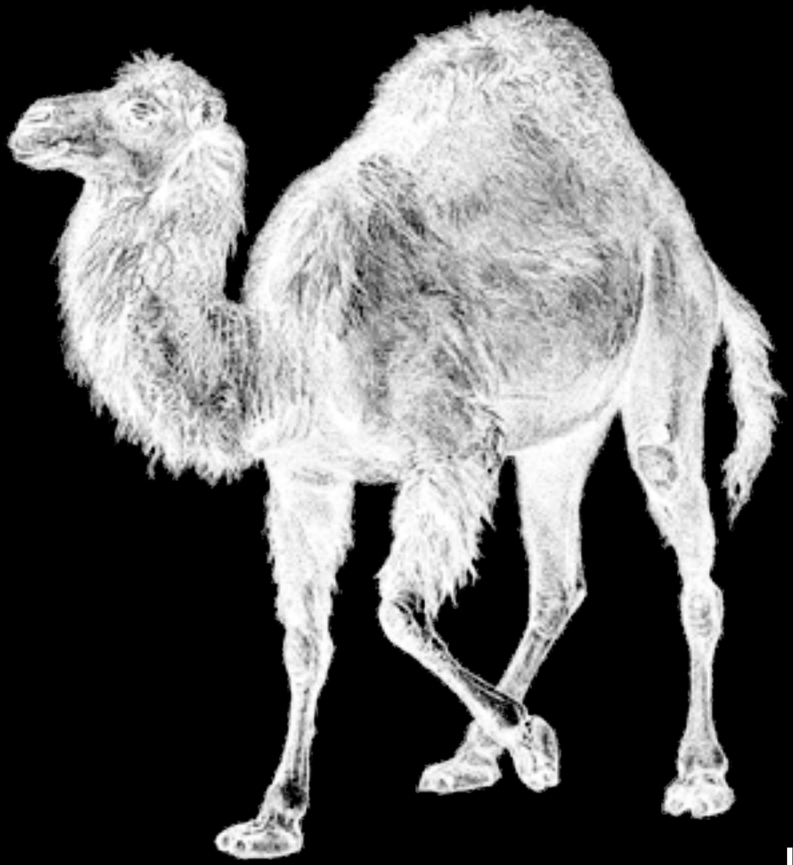
JavaScript, DOM, Applet, plugins?

No!

{ This is Java. Nothing else.

java } html >

vaadin }>



healthcare portal, 100 kloc of perl, ..

1998

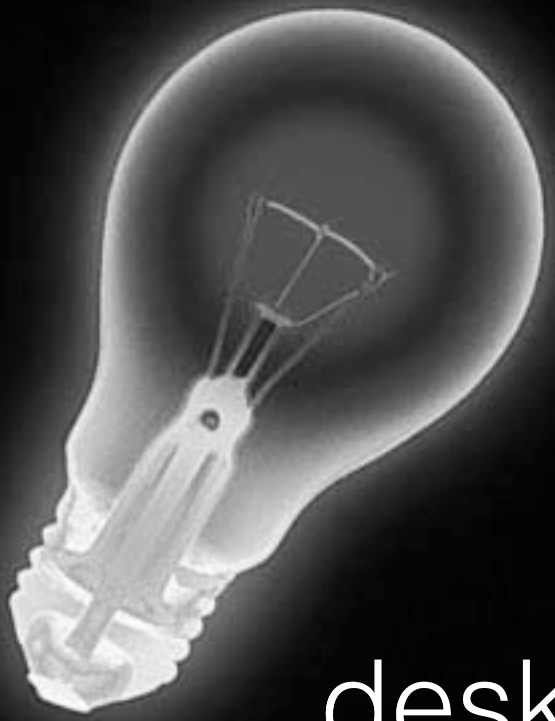
vaadin }>

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

vaadin }>

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

vaadin }>



desktop programming paradigm for web!

vaadin }>

IT MILL

found — 2000

millstone — 2002

ajax — 2005

google web toolkit — 2008

vaadin }>

re-released as

2009

vaadin }>

vaadin }>

Vaadin is now
21 months young
and 10 years old

vaadin }>

Apache License

Contents

Server-side RIA

What is it? Pros & cons?

In practice

Coding a Vaadin application step-by-step

Vaadin

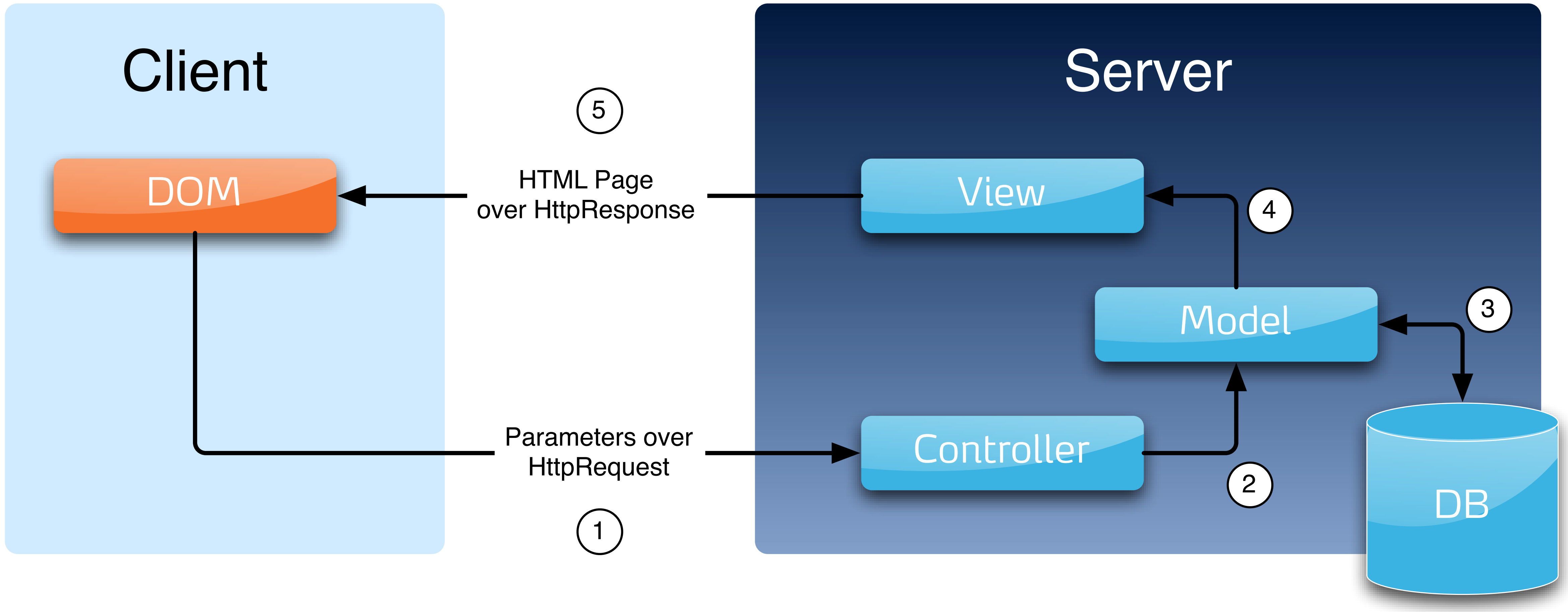
Big picture, Extending, Getting started

Discussion

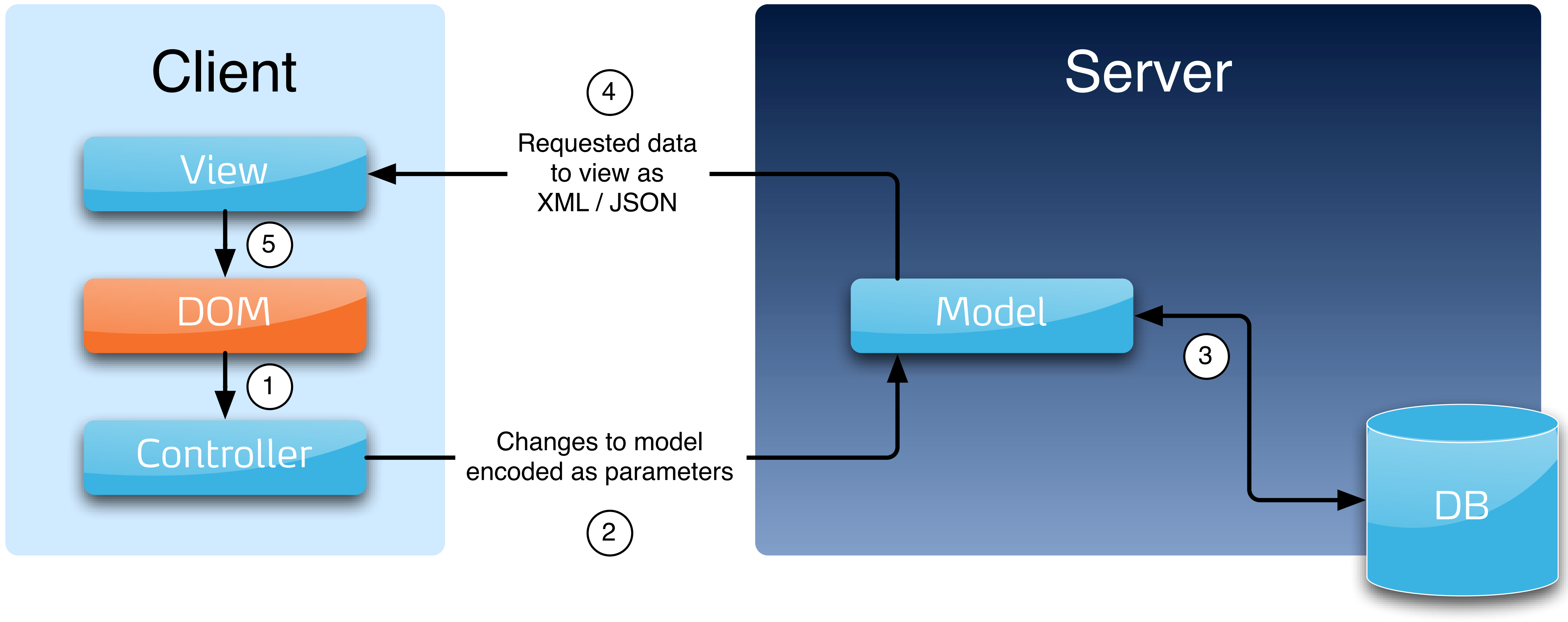
vaadin }>

Server-side RIA

“Web 1.0”


Client
Server
DOM
View
Model
Controller
DB
5
HTML Page
over HttpResponse
4
3
Parameters over
HttpRequest
2
1

Client-side RIA

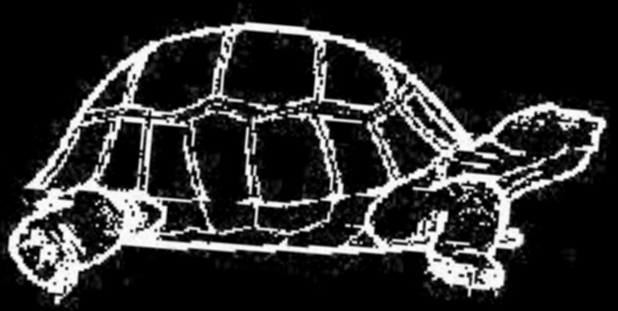

Client
View
5
DOM
1
Controller
4
Requested data to view as XML / JSON
Changes to model encoded as parameters
2
Server
Model
3
DB

challenge
web is not easy

different
features
in different
browsers

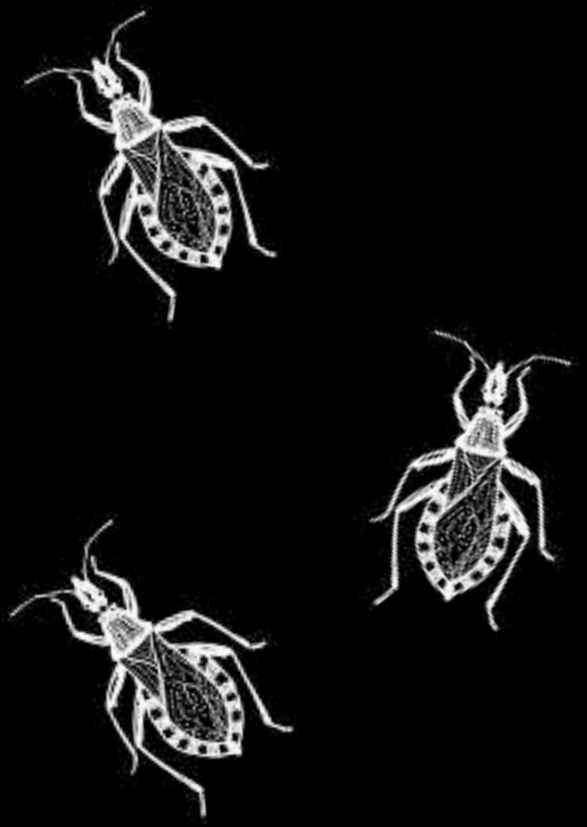
vaadin }>

different
performance
in different
browsers



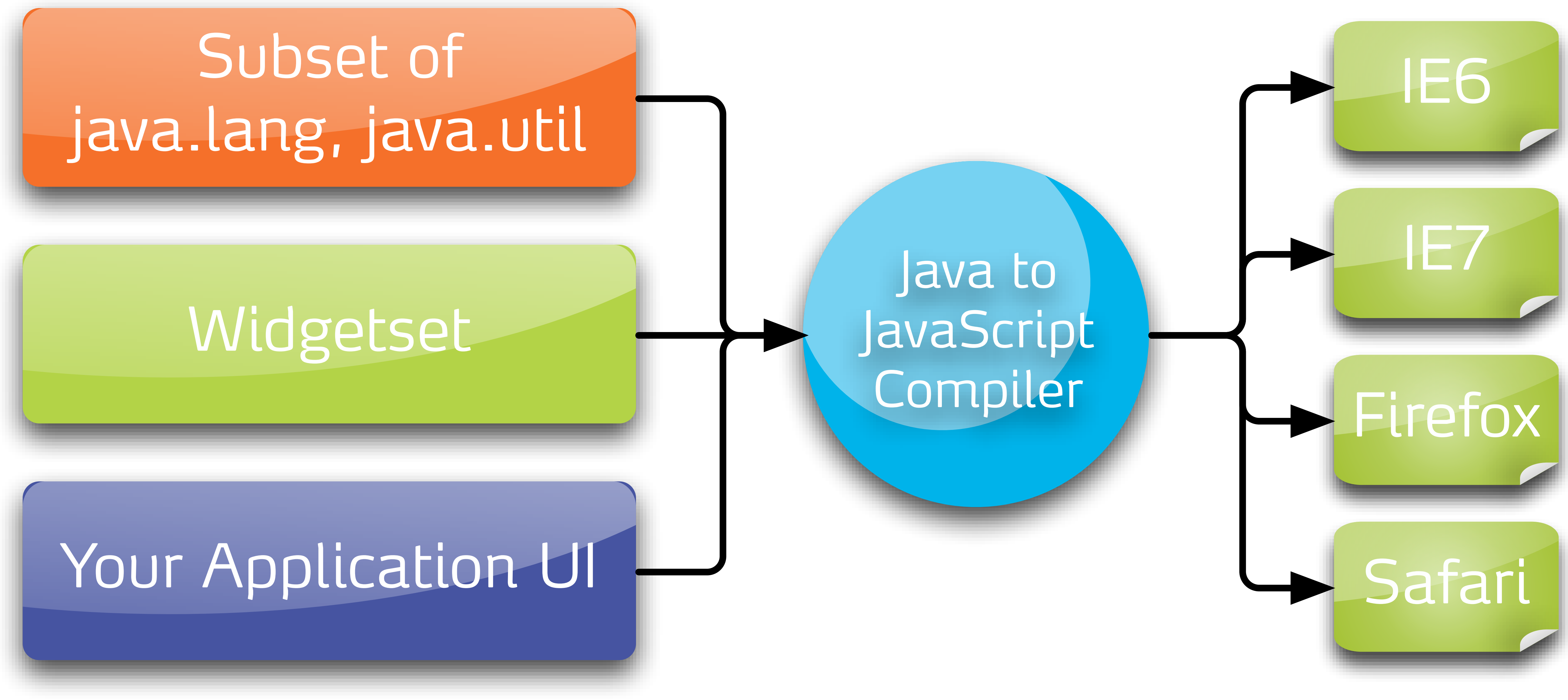
vaadin }>

different
bugs
in different
browsers



vaadin }>

Google Web Toolkit


Subset of java.lang, java.util
Widgetset
Your Application UI
Java to JavaScript Compiler
IE6
IE7
Firefox
Safari

simpler

- Java only
- forget the web

vaadin }>

cost-effective
stop debugging
JavaScript spaghetti

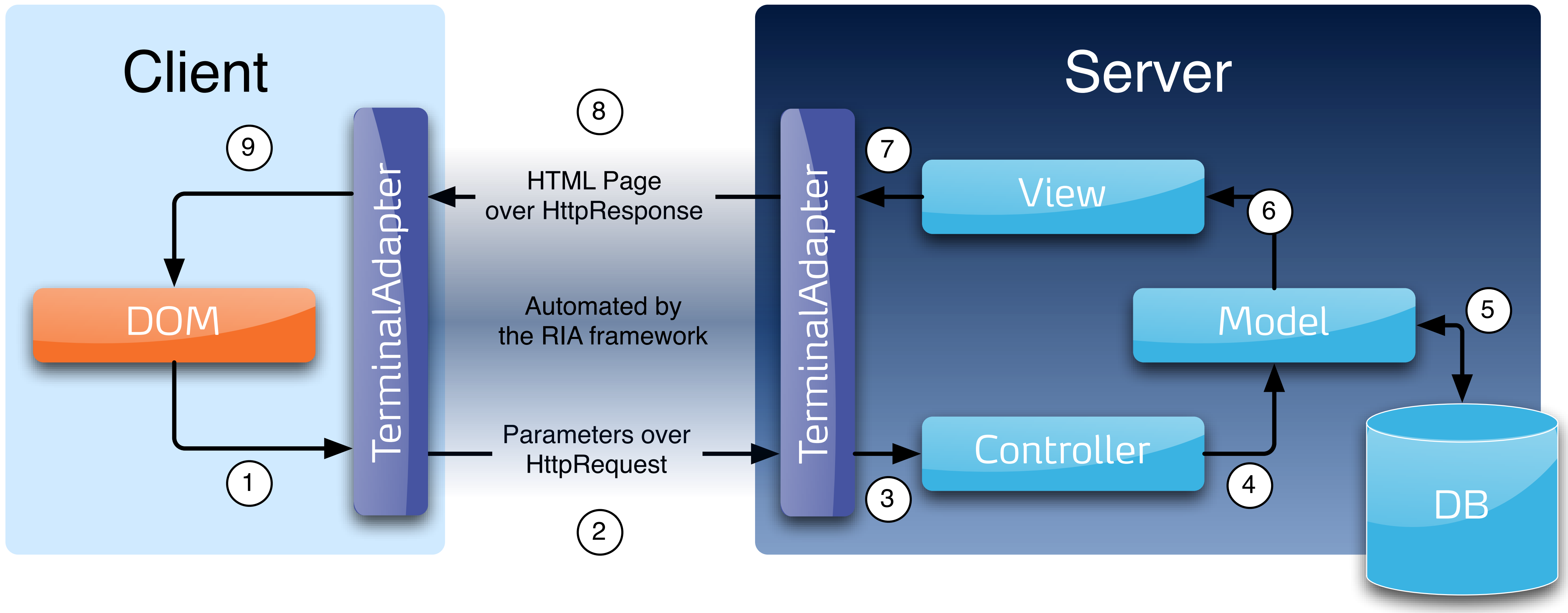
vaadin }>

modular
extensible

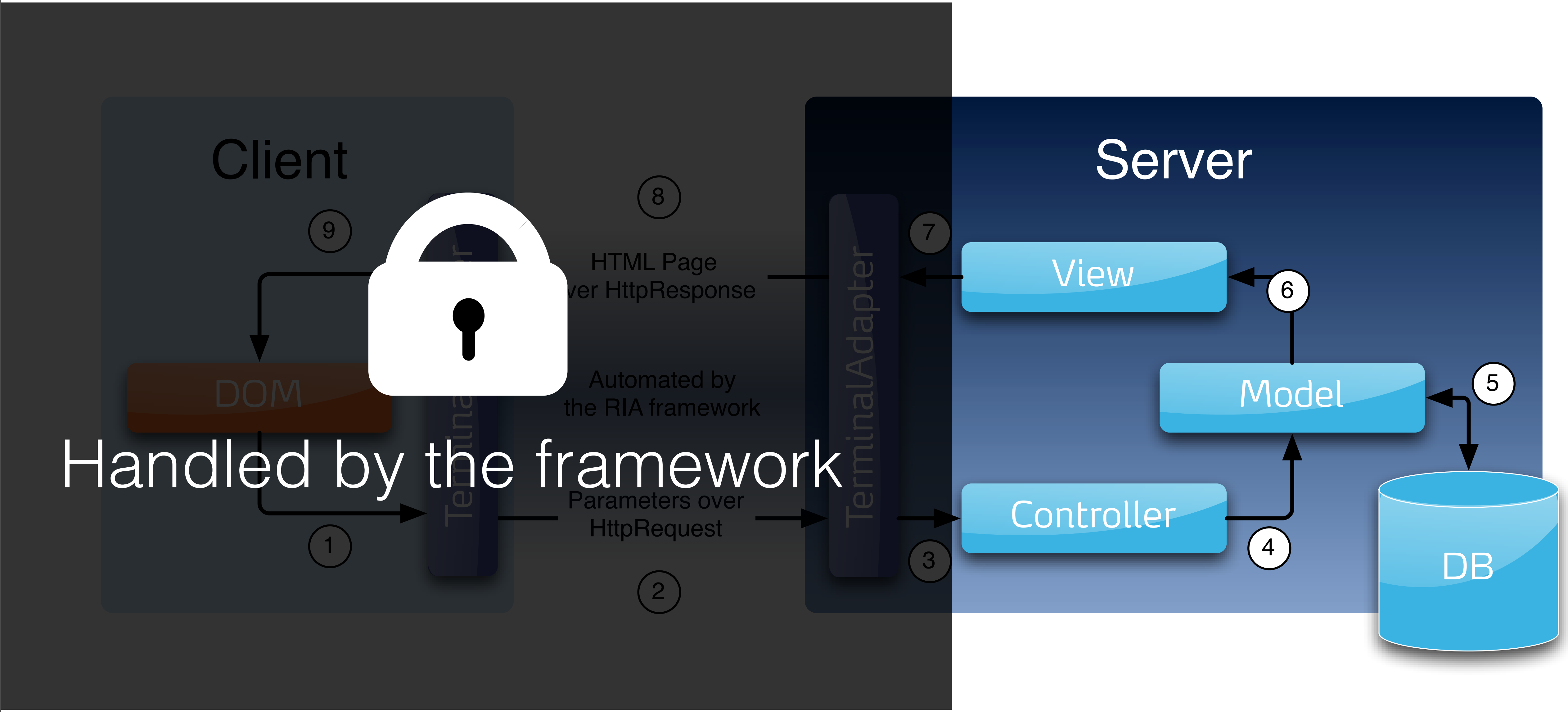
Building wonderful apps doesn’t require writing **fat** web clients.

vaadin }>

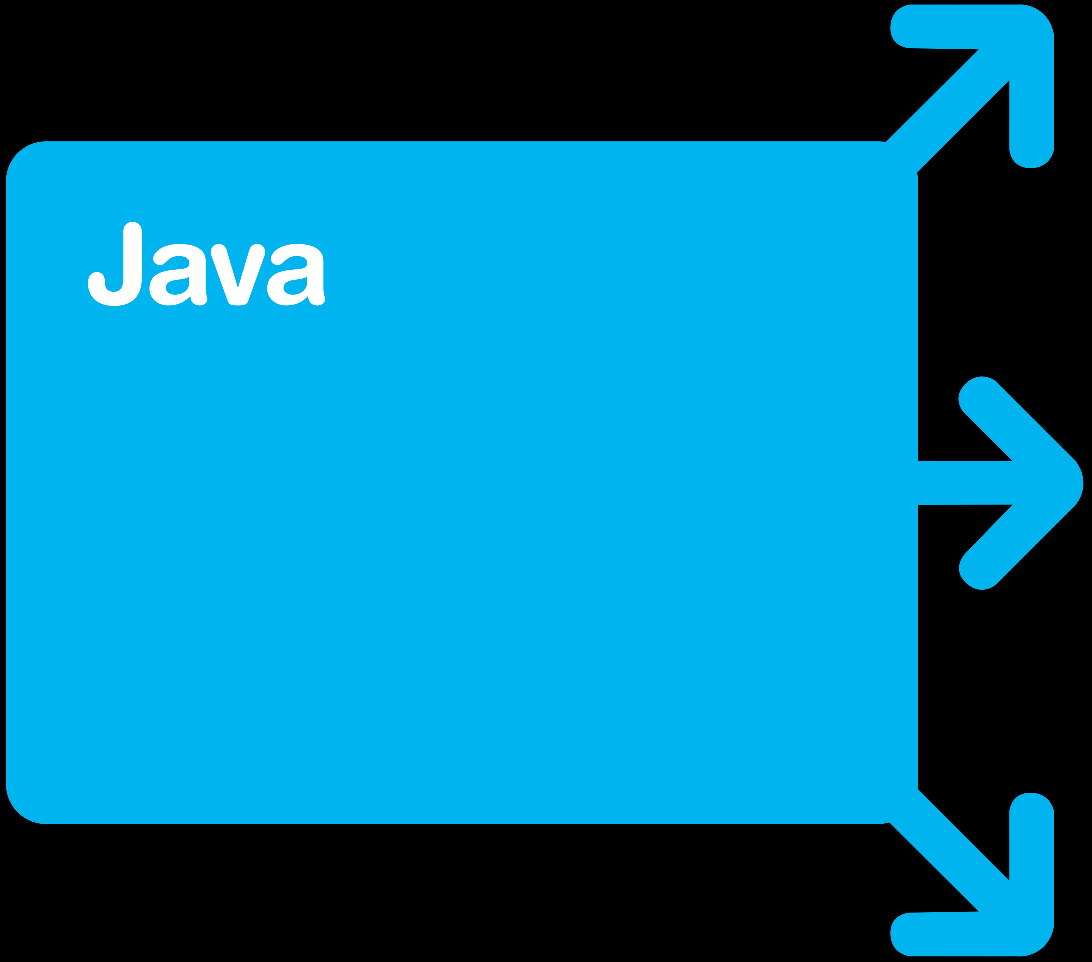
Server-side RIA



Server-side RIA



the benefits of Java



simpler
forget the web

cost-effective
no JavaScript
debugging

modular
extensible

vaadin }>

even simpler

- forget the client-side
- synchronous
- server resources

vaadin }>

more flexible

- all Java tools and libraries
- any JVM language

Scala Groovy

vaadin }>

more secure

- code stays in server
- less web services

not as scalable

UI state is stored in the server memory

vaadin }>

Measured 12.000 active concurrent users per server for a ticketing app

[Amazon EC2-large; limited by storage layer]

vaadin }>

no offline mode

server is always required

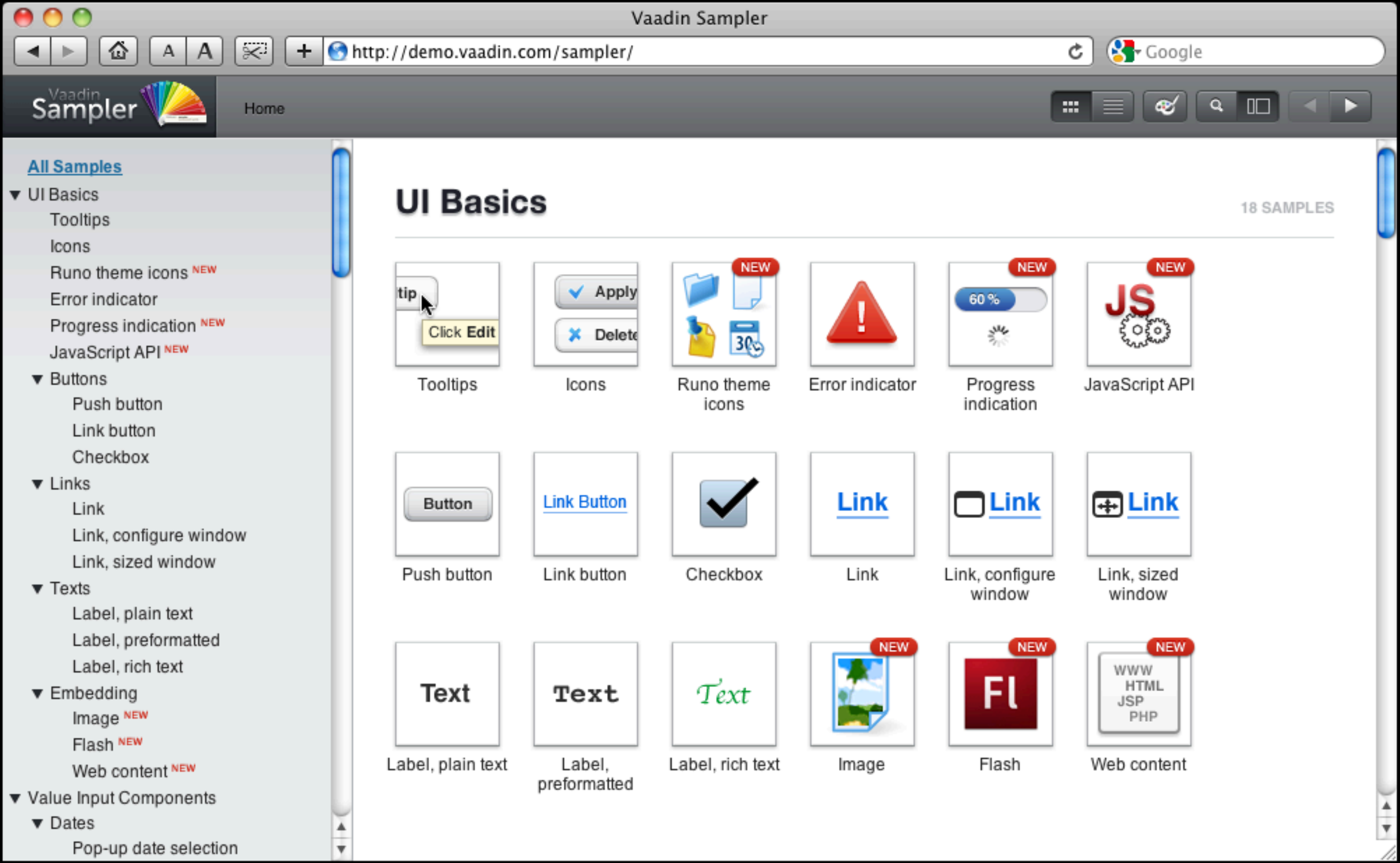
vaadin }>

#1 benefit development is really fast

vaadin }>

Vaadin Framework

Great UI Components



Vaadin Sampler

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

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 (NEW) · Error indicator · Progress indication (NEW) · JavaScript API (NEW)

Push button · Link button · Checkbox · Link · Link, configure window · Link, sized window

Label, plain text · Label, preformatted · Label, rich text · Image (NEW) · Flash (NEW) · Web content (NEW)

vaadin }>

Combined power of

- Server-side RIA
- Google Web Toolkit

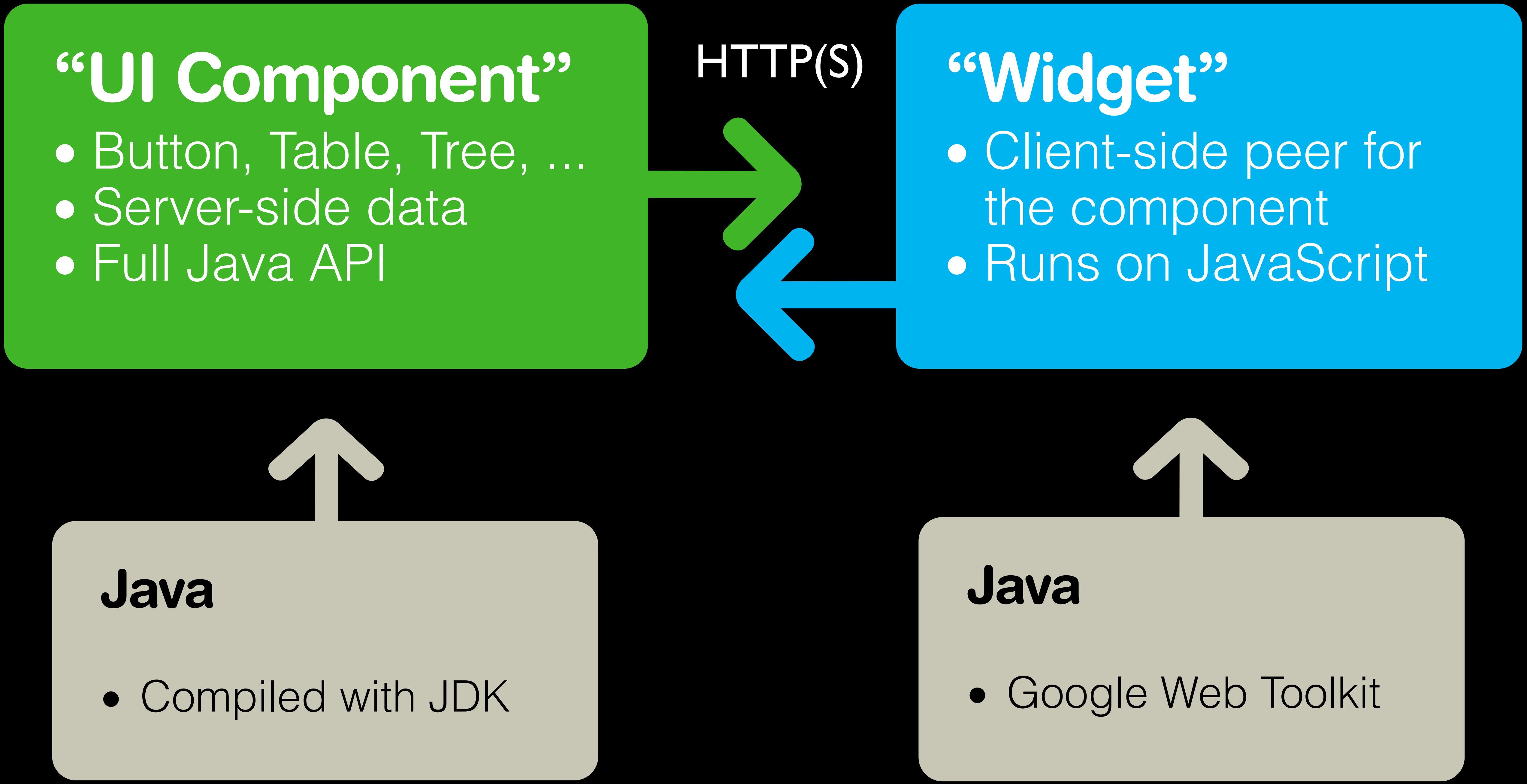
vaadin }>

Combined power of

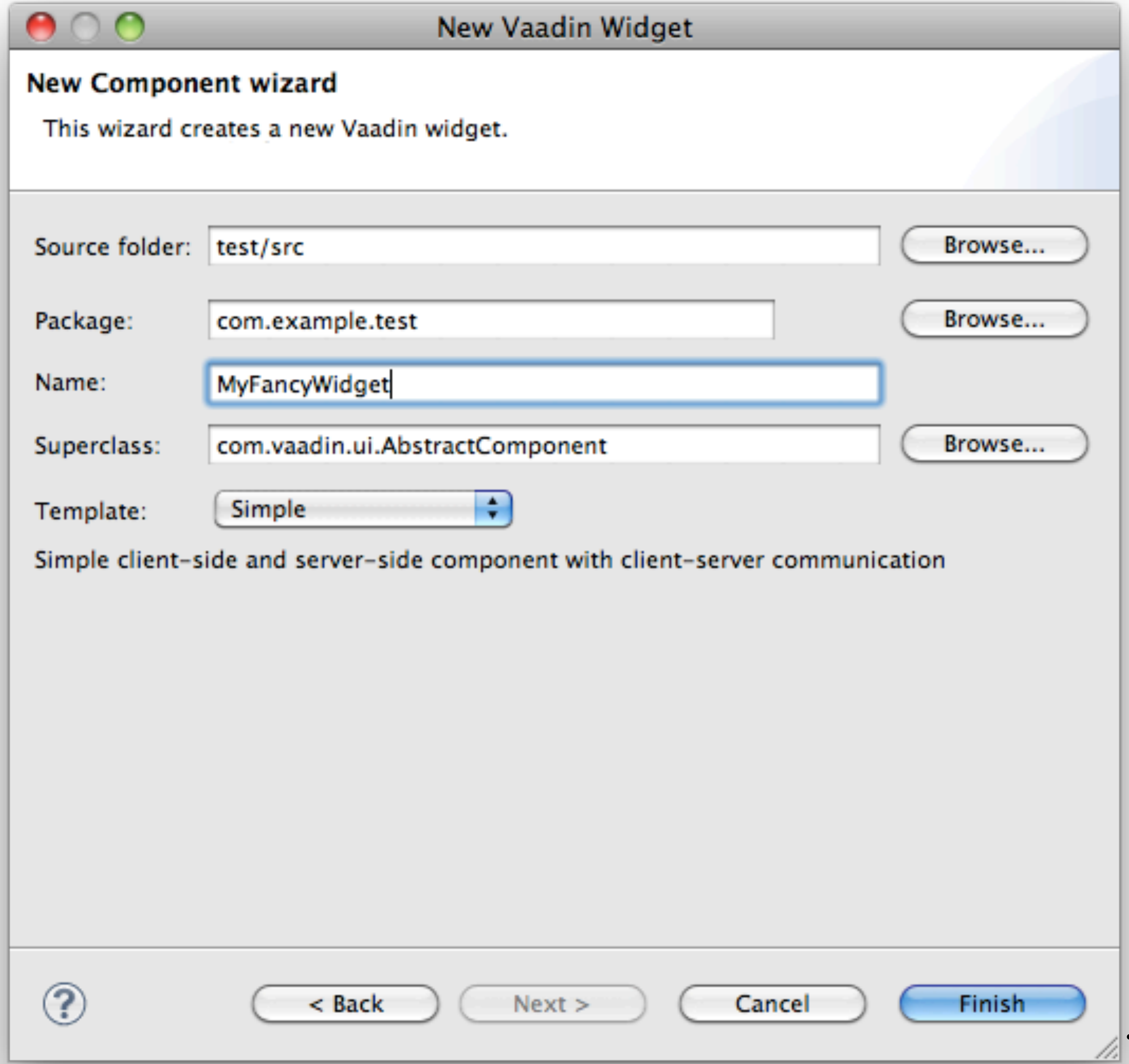
- Server-side RIA
- Google Web Toolkit

vaadin }>

Vaadin UI component architecture



Creating new UI components is really easy


New Vaadin Widget

New Component wizard

This wizard creates a new Vaadin widget.

Source folder:	test/src
Package:	com.example.test
Name:	MyFancyWidget
Superclass:	com.vaadin.ui.AbstractComponent
Template:	Simple

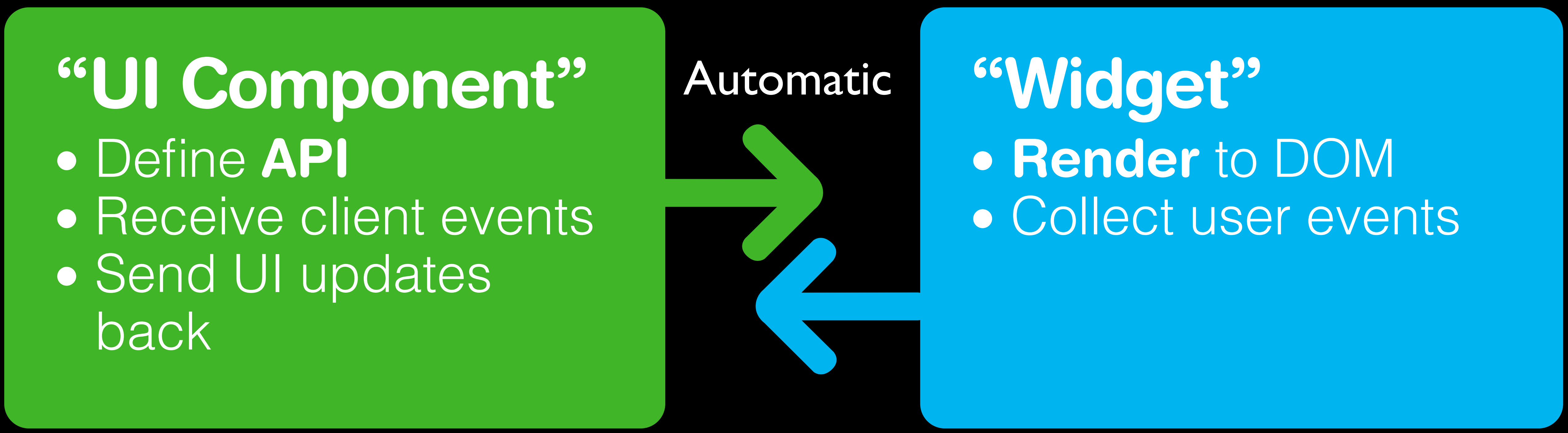
Simple client-side and server-side component with client-server communication

vaadin }>

Implement two classes

Server-side

Client-side

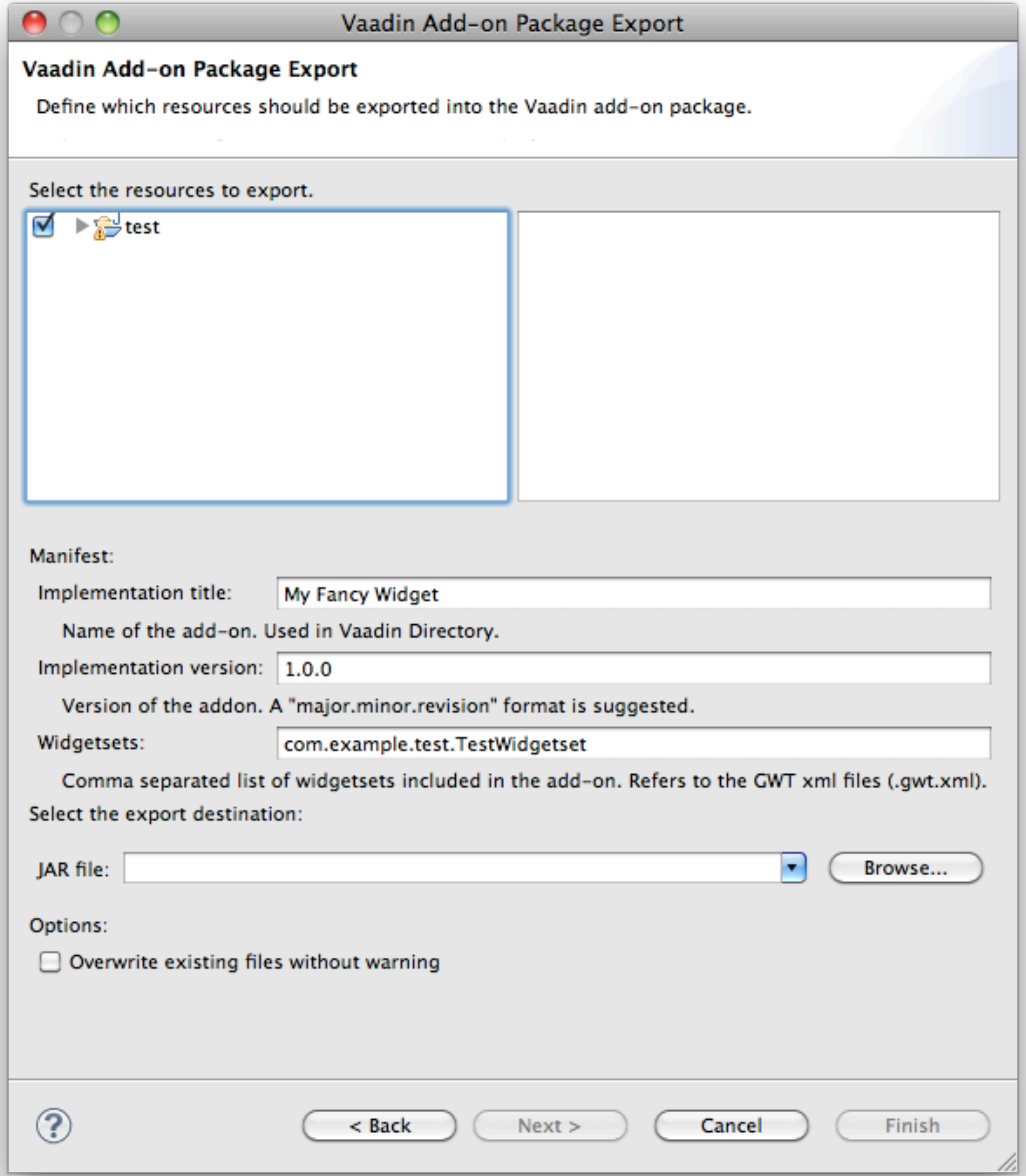


“UI Component”

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

“Widget”

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



Vaadin Add-on Package Export

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: My Fancy Widget

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

Implementation version: 1.0.0

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

Widgetsets: com.example.test.TestWidgetset

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

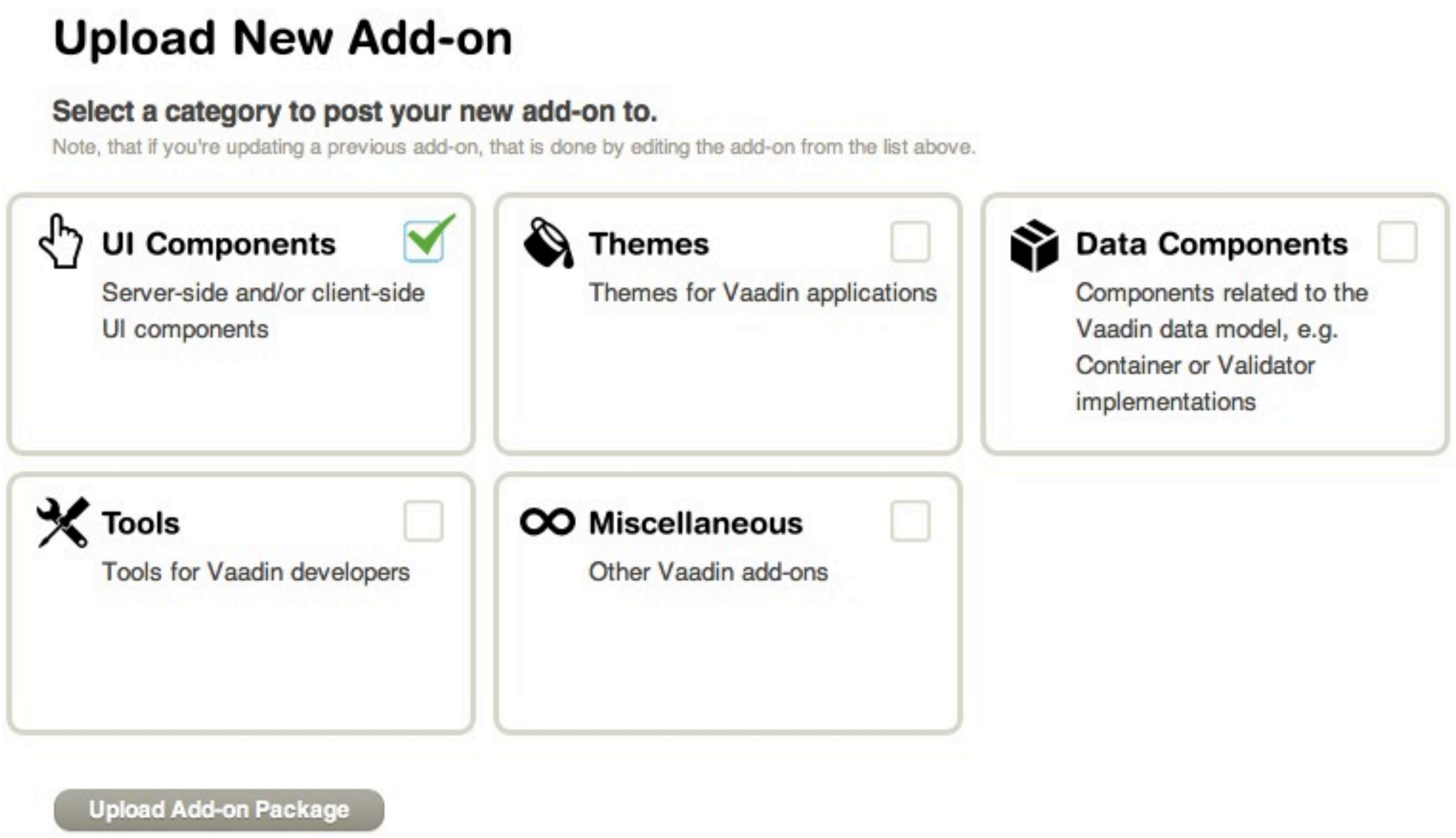
Select the export destination:

JAR file:

Options:

☐ Overwrite existing files without warning

vaadin }>



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.

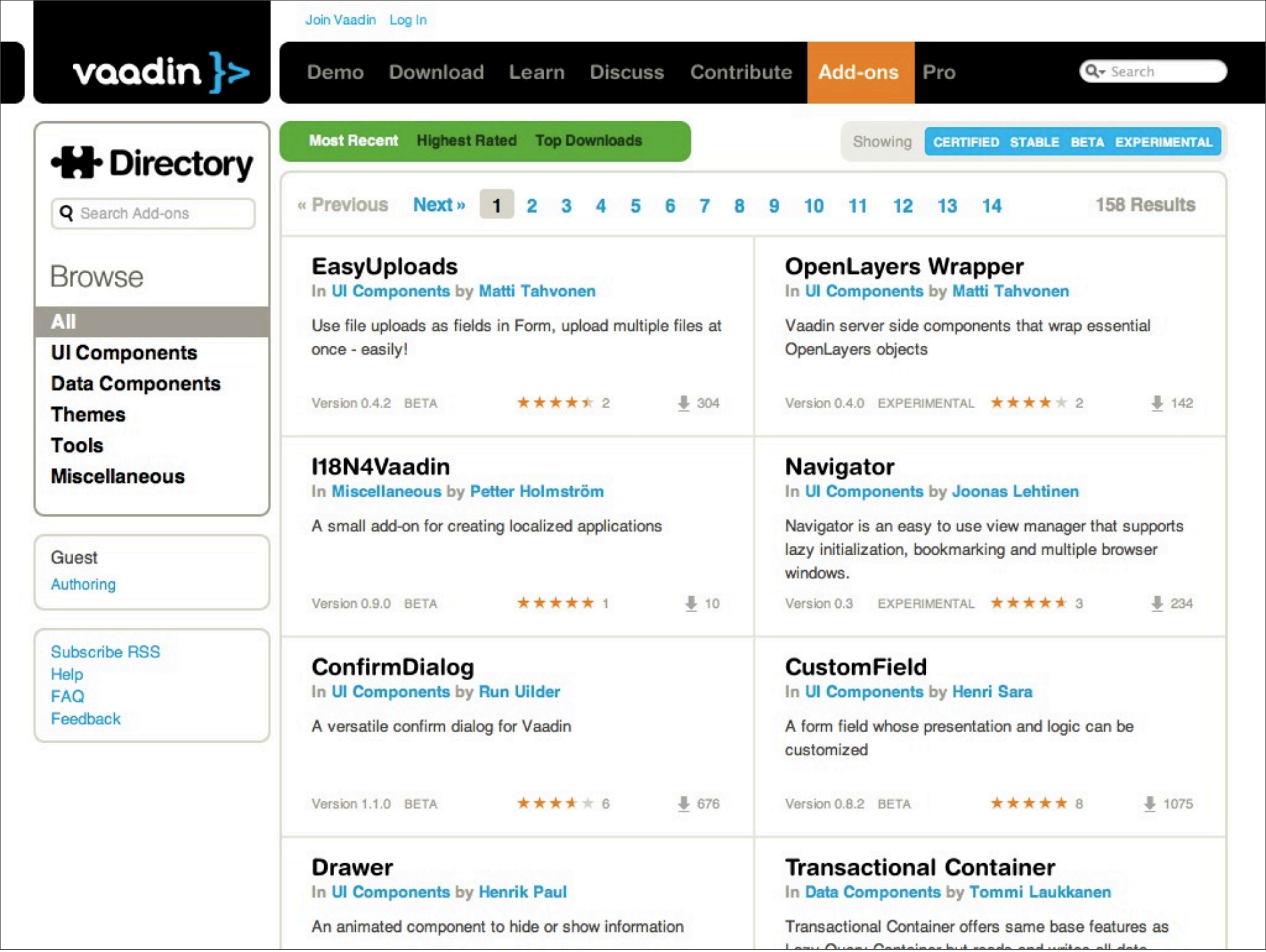
UI Components ✓
Server-side and/or client-side UI components

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


Join Vaadin Log In

Demo Download Learn Discuss Contribute Add-ons Pro Search

Directory

Search Add-ons

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 ★★★★½ 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

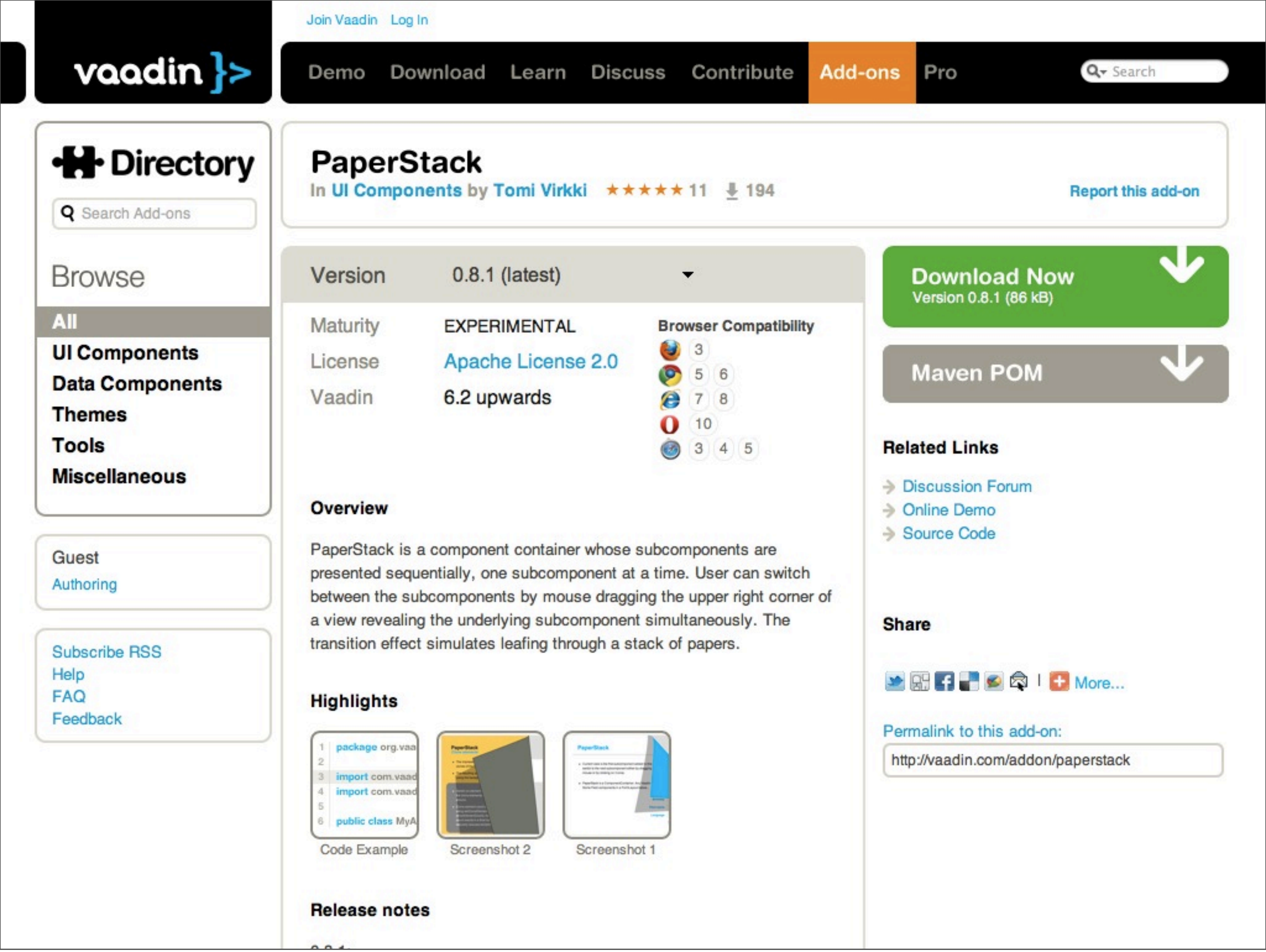
Navigator
In UI Components by Joonas Lehtinen
Navigator is an easy to use view manager that supports lazy initialization, bookmarking and multiple browser windows.
Version 0.3 EXPERIMENTAL ★★★★★ 3 234

ConfirmDialog
In UI Components by Run Uilder
A versatile confirm dialog for Vaadin
Version 1.1.0 BETA ★★★½☆ 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



Join Vaadin Log In

vaadin }>

Demo Download Learn Discuss Contribute Add-ons Pro Search

Directory

Search Add-ons

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
3
5 6
7 8
10
3 4 5

Download Now
Version 0.8.1 (86 kB)

Maven POM

Related Links

- Discussion Forum
- Online Demo
- Source Code

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

```
package org.vaa
import com.vaad
import com.vaad
public class MyA
```

Code Example Screenshot 2 Screenshot 1

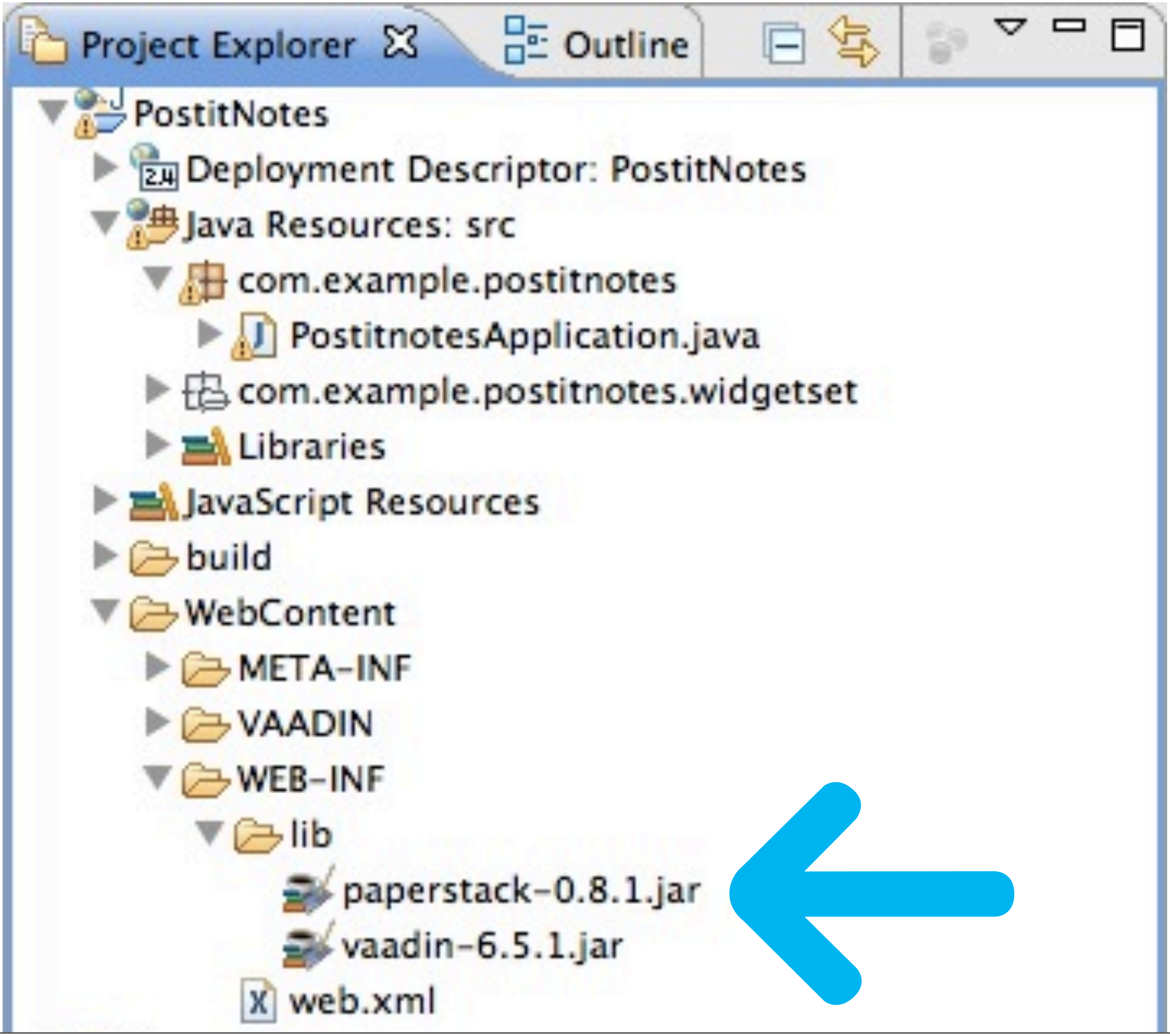
Share

More...

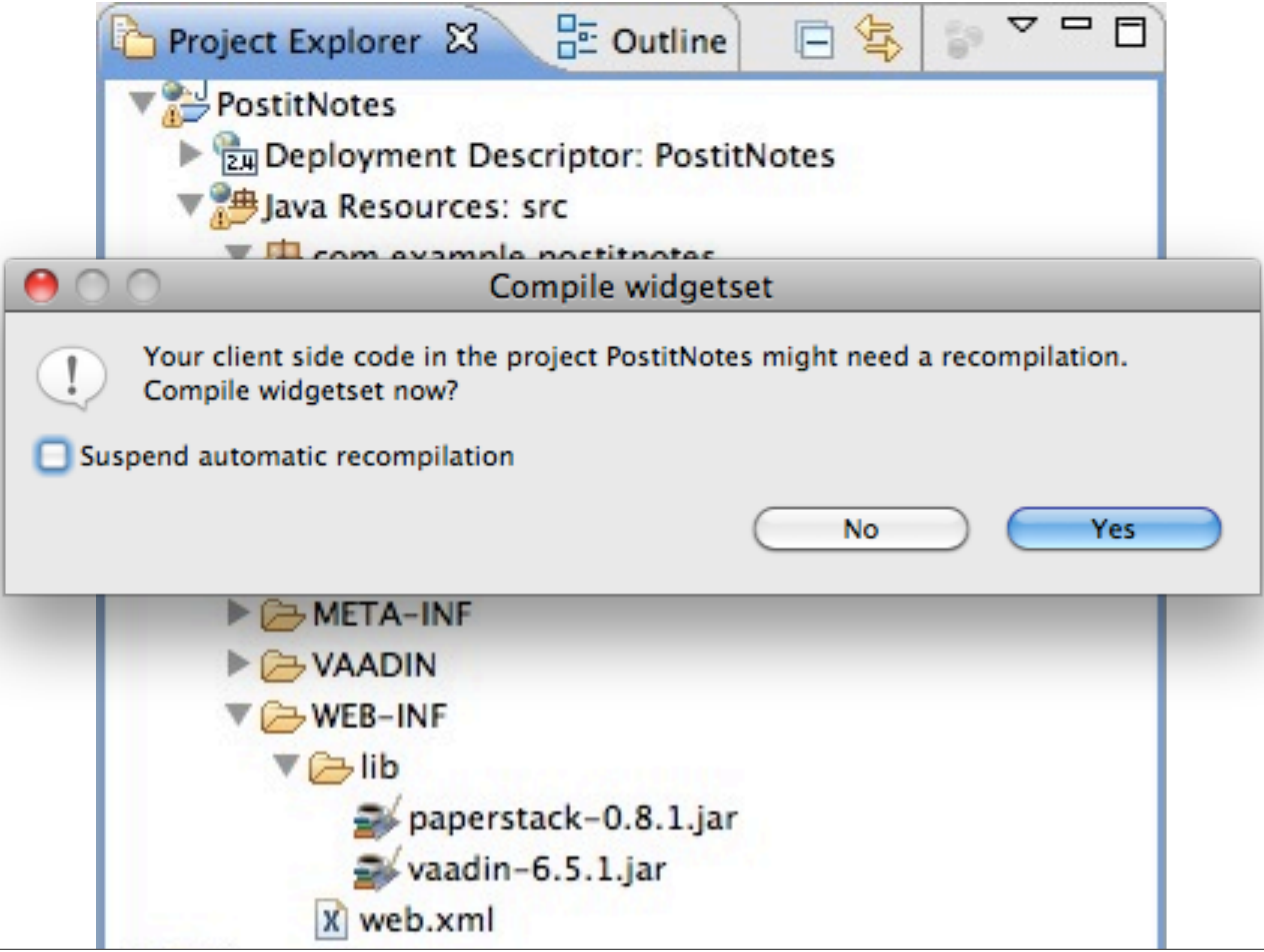
Permalink to this add-on:

http://vaadin.com/addon/paperstack

Release notes



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



Project Explorer
Outline
PostitNotes
Deployment Descriptor: PostitNotes
Java Resources: src
com.example.postitnotes

Compile widgetset

Your client side code in the project PostitNotes might need a recompilation.
Compile widgetset now?

Suspend automatic recompilation

No Yes

META-INF
VAADIN
WEB-INF
lib
paperstack-0.8.1.jar
vaadin-6.5.1.jar
web.xml

First class Java citizen

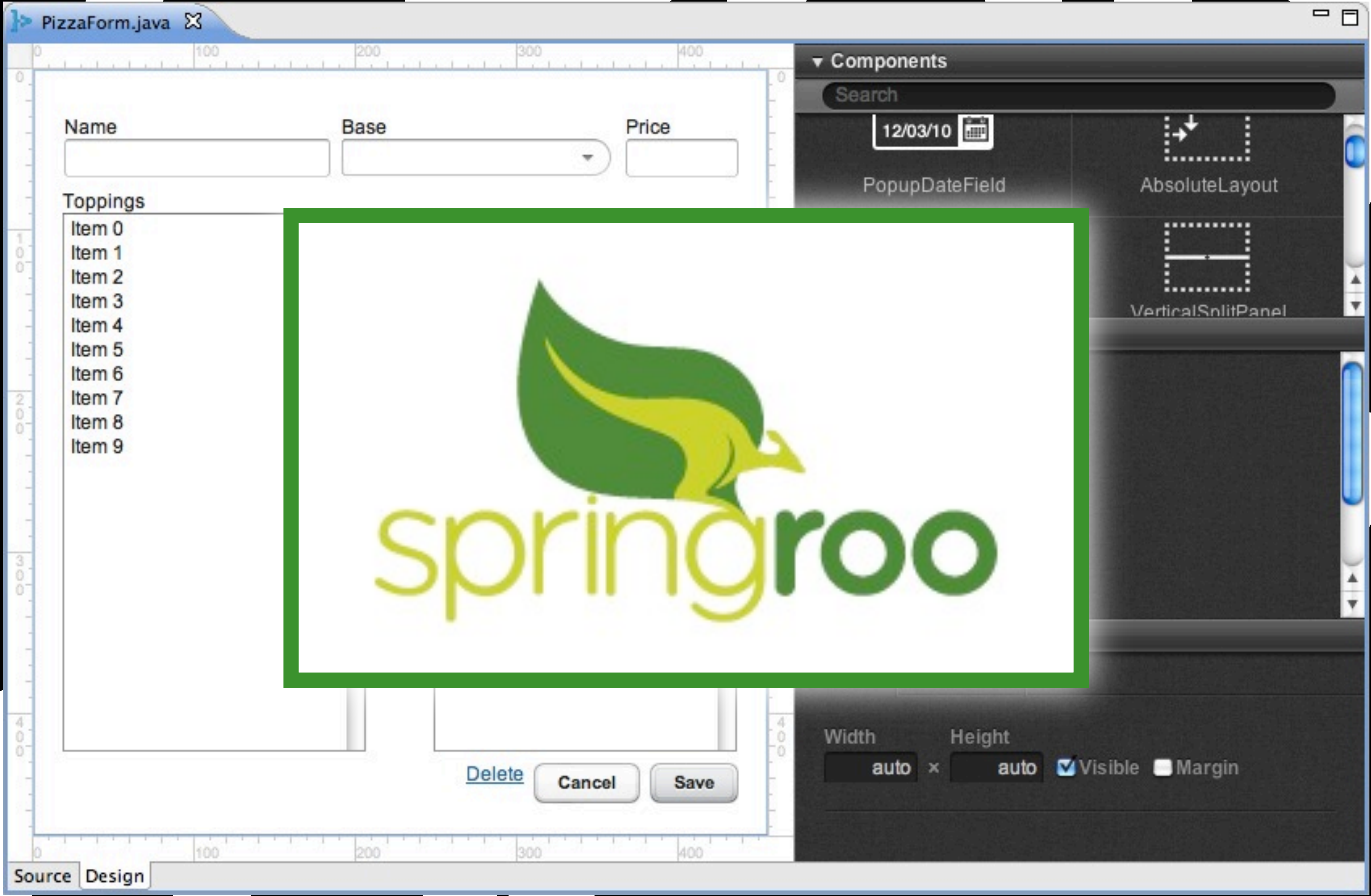
vaadin }>

First class Java citizen

vaadin }>

Servlet
Portlet
App Engine

Eclipse
Maven
Netbeans
Spring Roo


PizzaForm.java

Name

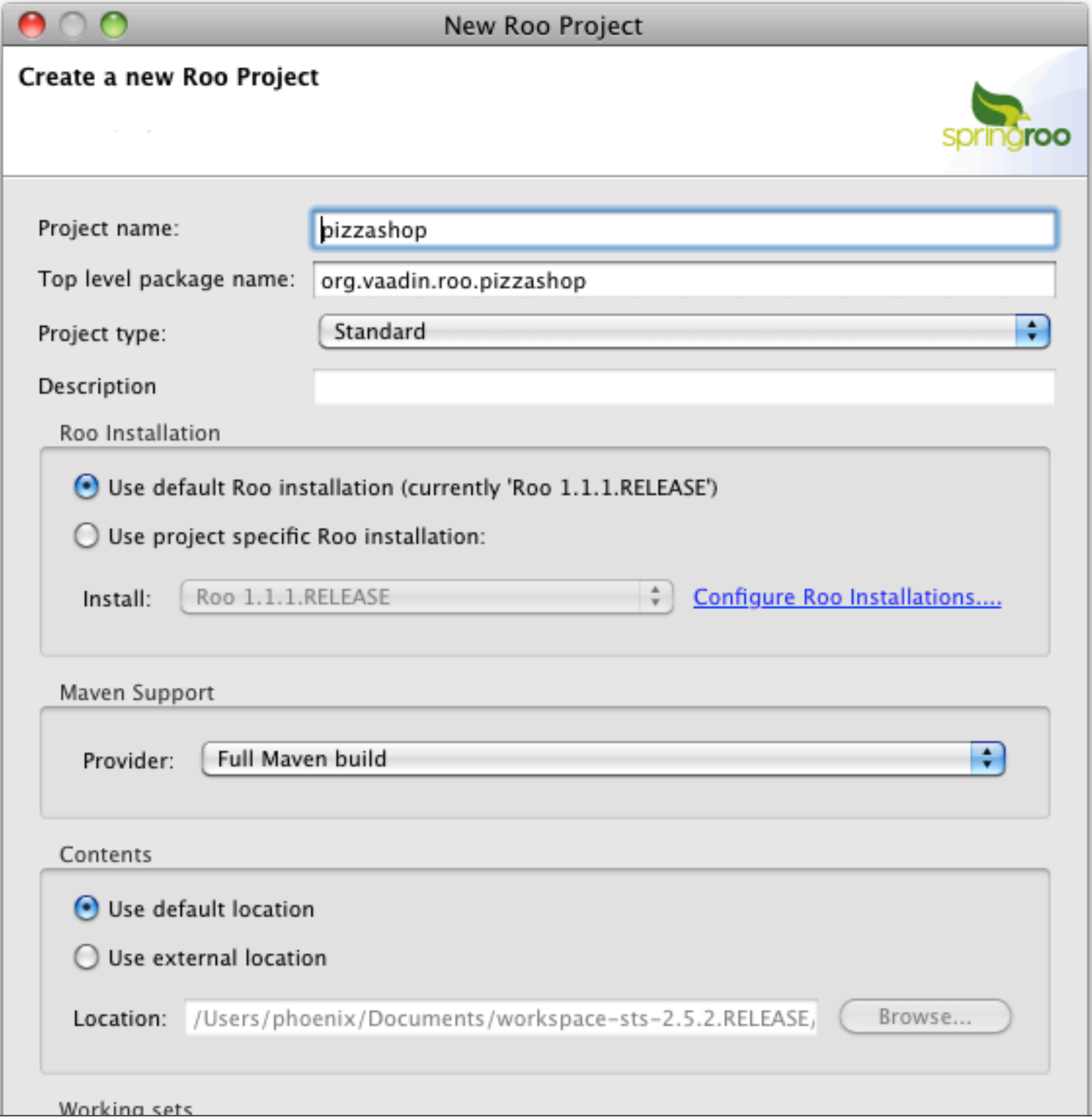
Base

Price

Toppings

Item 0
Item 1
Item 2
Item 3
Item 4
Item 5
Item 6
Item 7
Item 8
Item 9

vaadin }>



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

Maven Support

Provider: Full Maven build

Contents

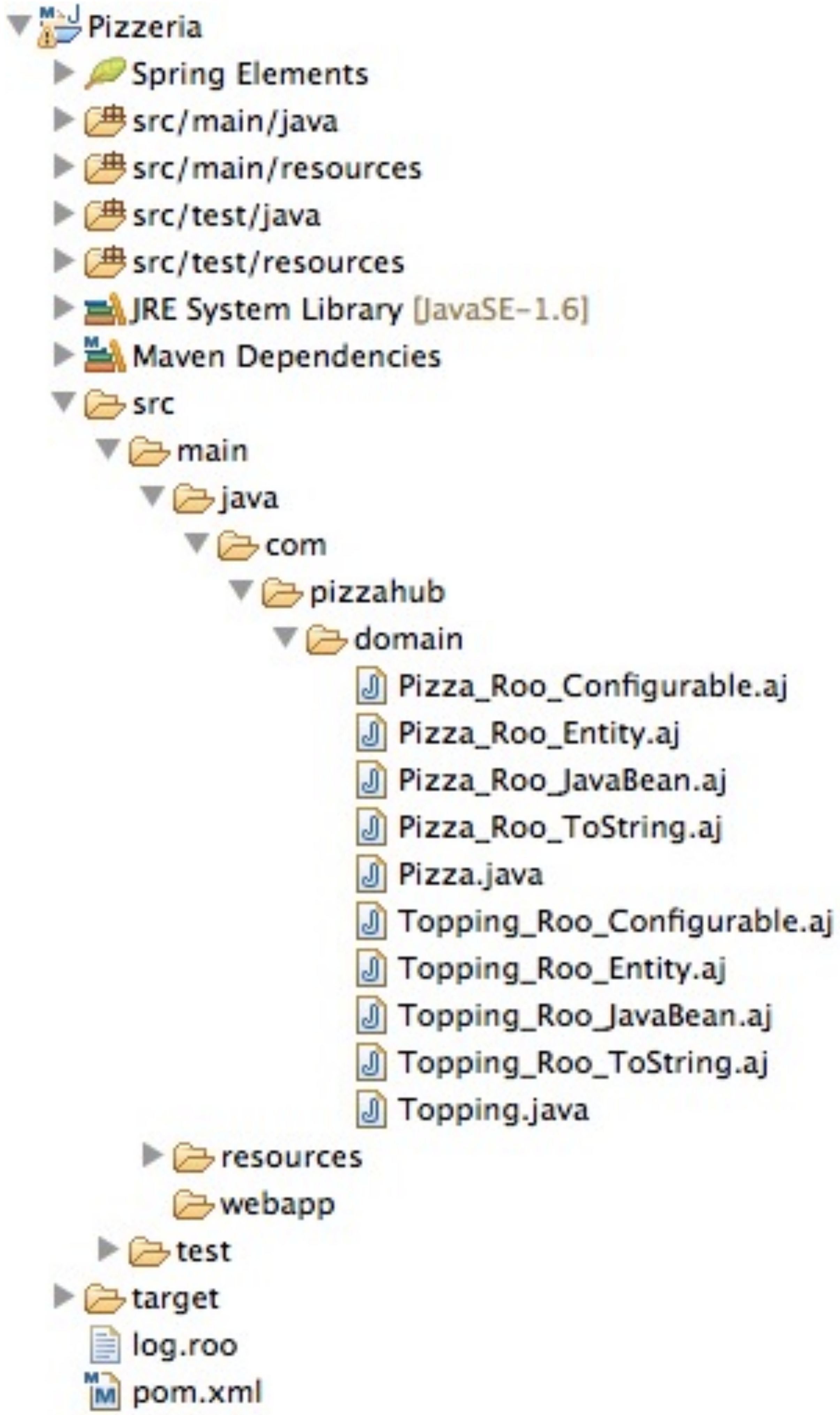
[✓ Use default location]

Use external location

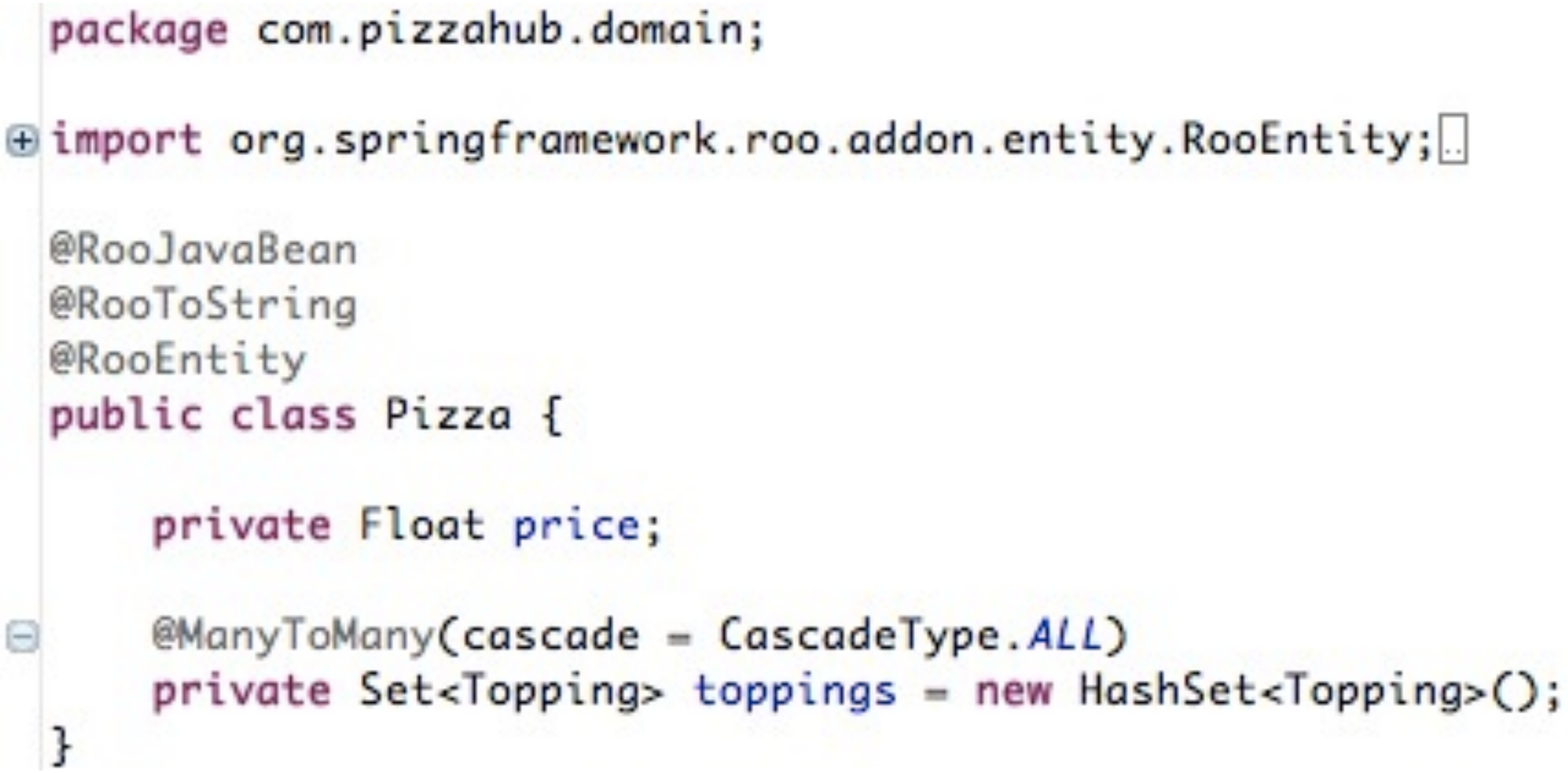
Location: /Users/phoenix/Documents/workspace-sts-2.5.2.RELEASE/

```
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
```

vaadin }>

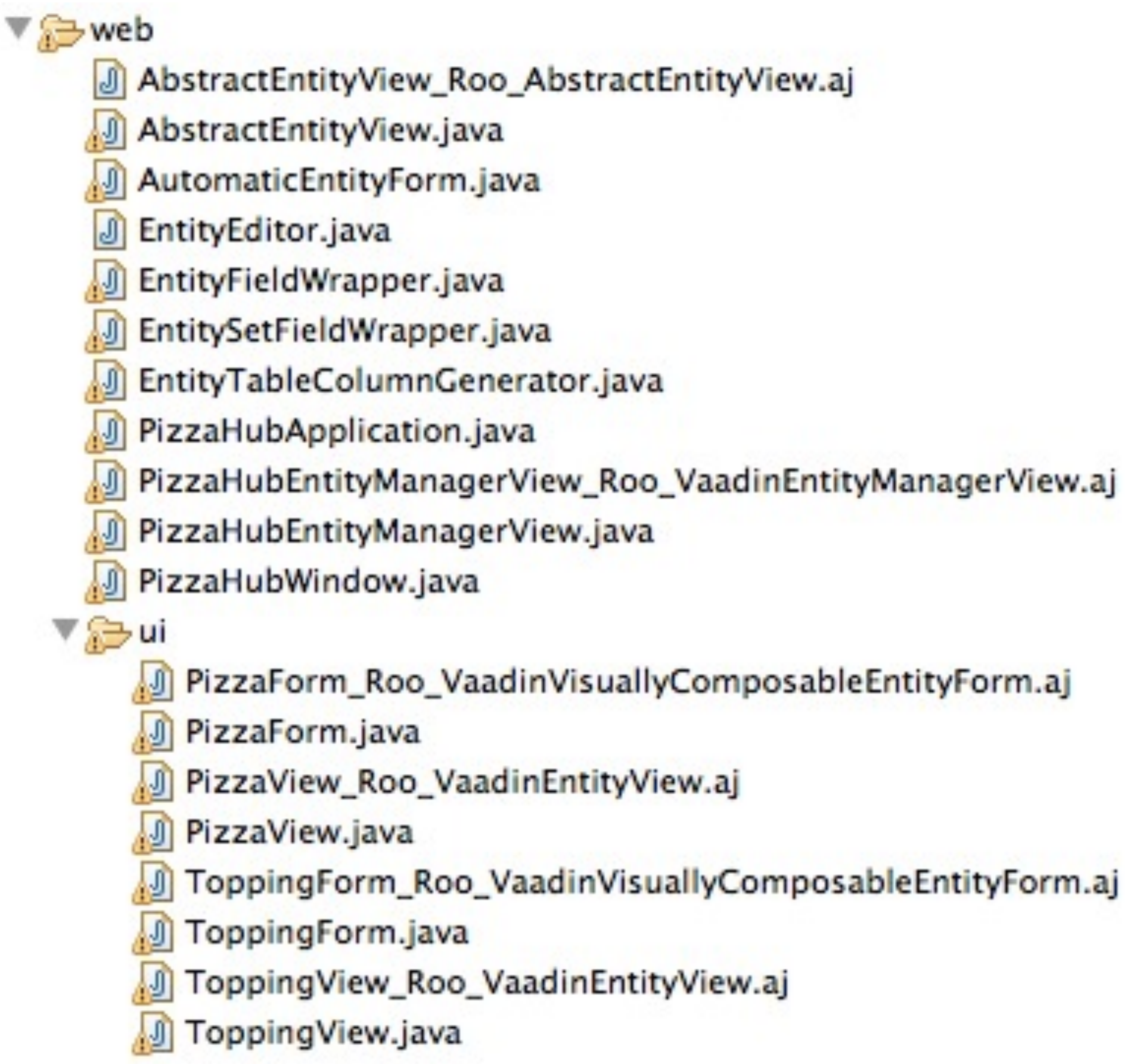


```
Pizzeria
  Spring Elements
  src/main/java
  src/main/resources
  src/test/java
  src/test/resources
  JRE System Library [JavaSE-1.6]
  Maven Dependencies
  src
    main
      java
        com
          pizzahub
            domain
              Pizza_Roo_Configurable.aj
              Pizza_Roo_Entity.aj
              Pizza_Roo_JavaBean.aj
              Pizza_Roo_ToString.aj
              Pizza.java
              Topping_Roo_Configurable.aj
              Topping_Roo_Entity.aj
              Topping_Roo_JavaBean.aj
              Topping_Roo_ToString.aj
              Topping.java
      resources
      webapp
    test
  target
  log.roo
  pom.xml
```

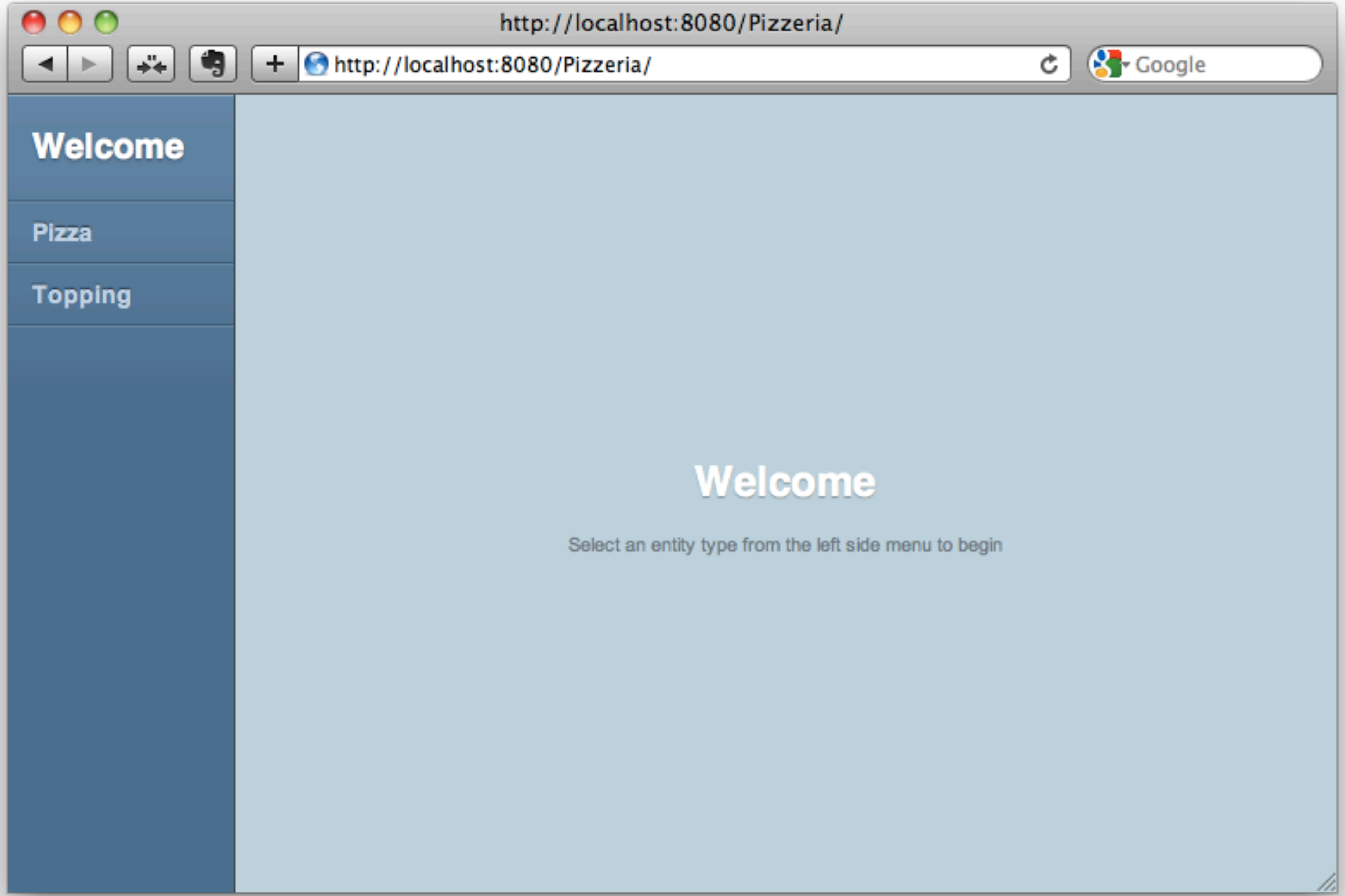


```
package com.pizzahub.domain;

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
```

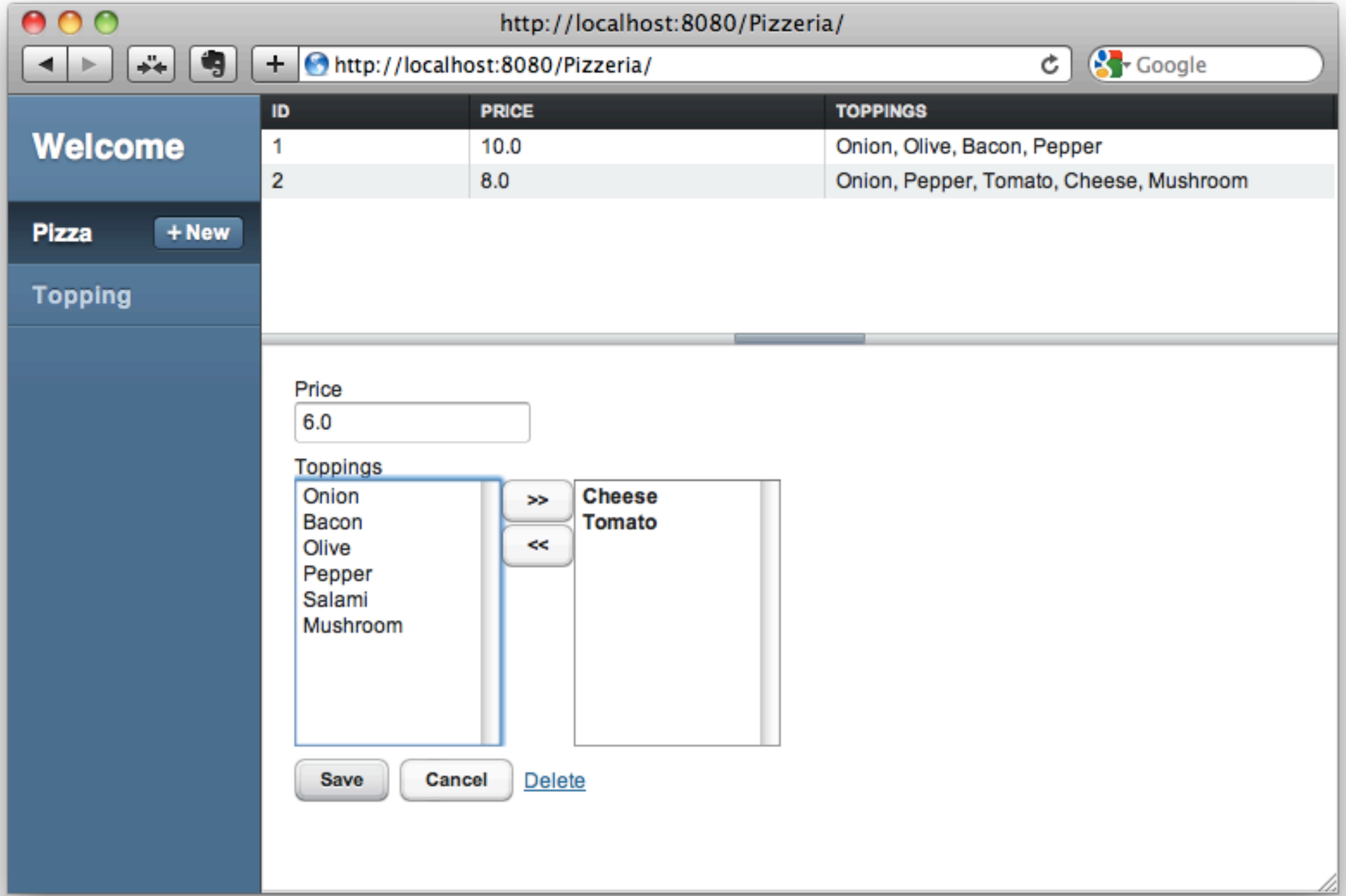


```
▼ web
    AbstractEntityView_Roo_AbstractEntityView.aj
    AbstractEntityView.java
    AutomaticEntityForm.java
    EntityEditor.java
    EntityFieldWrapper.java
    EntitySetFieldWrapper.java
    EntityTableColumnGenerator.java
    PizzaHubApplication.java
    PizzaHubEntityManagerView_Roo_VaadinEntityManagerView.aj
    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
```


http://localhost:8080/Pizzeria/
http://localhost:8080/Pizzeria/
Google
Welcome
Pizza
Topping

Welcome

Select an entity type from the left side menu to begin

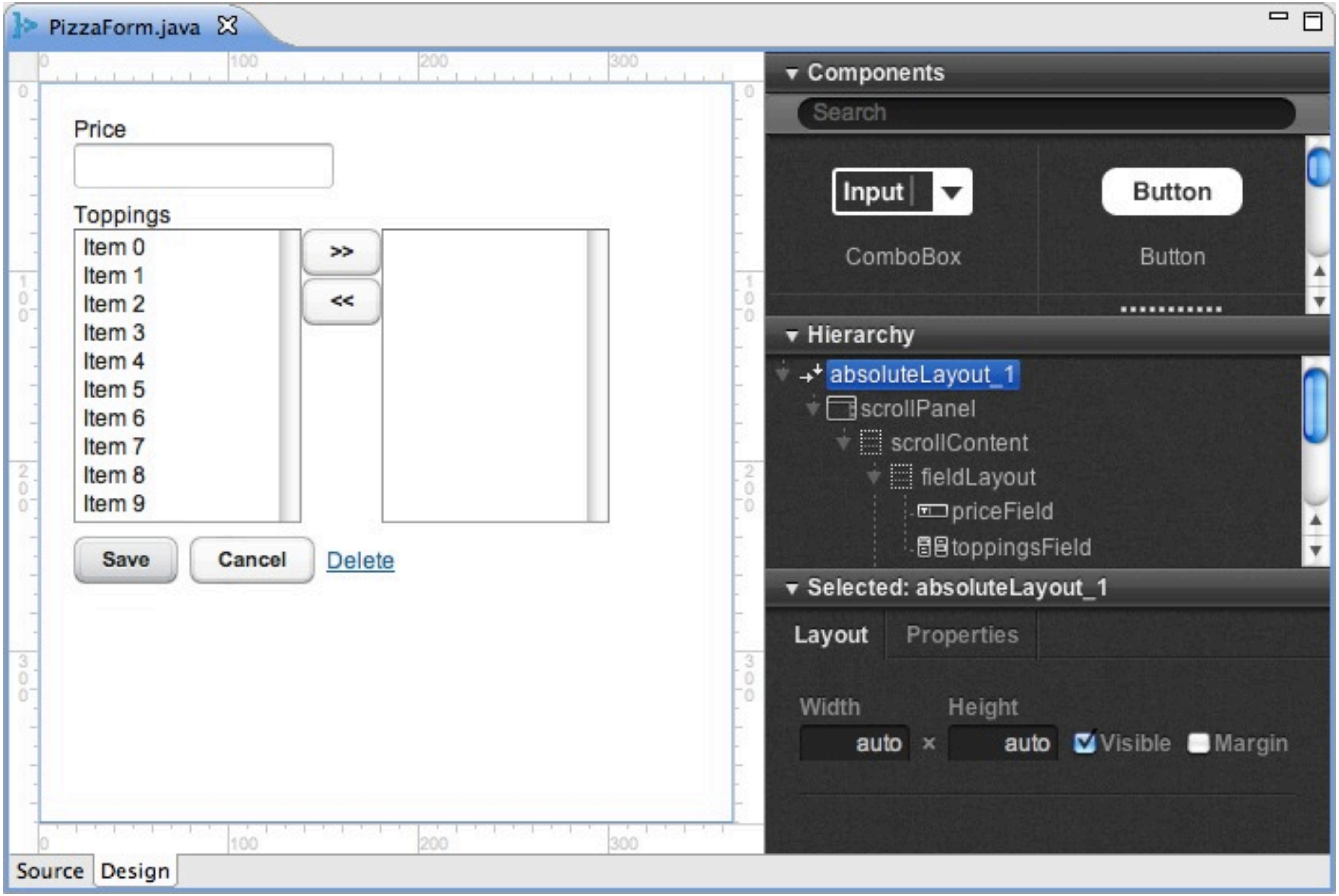

http://localhost:8080/Pizzeria/

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

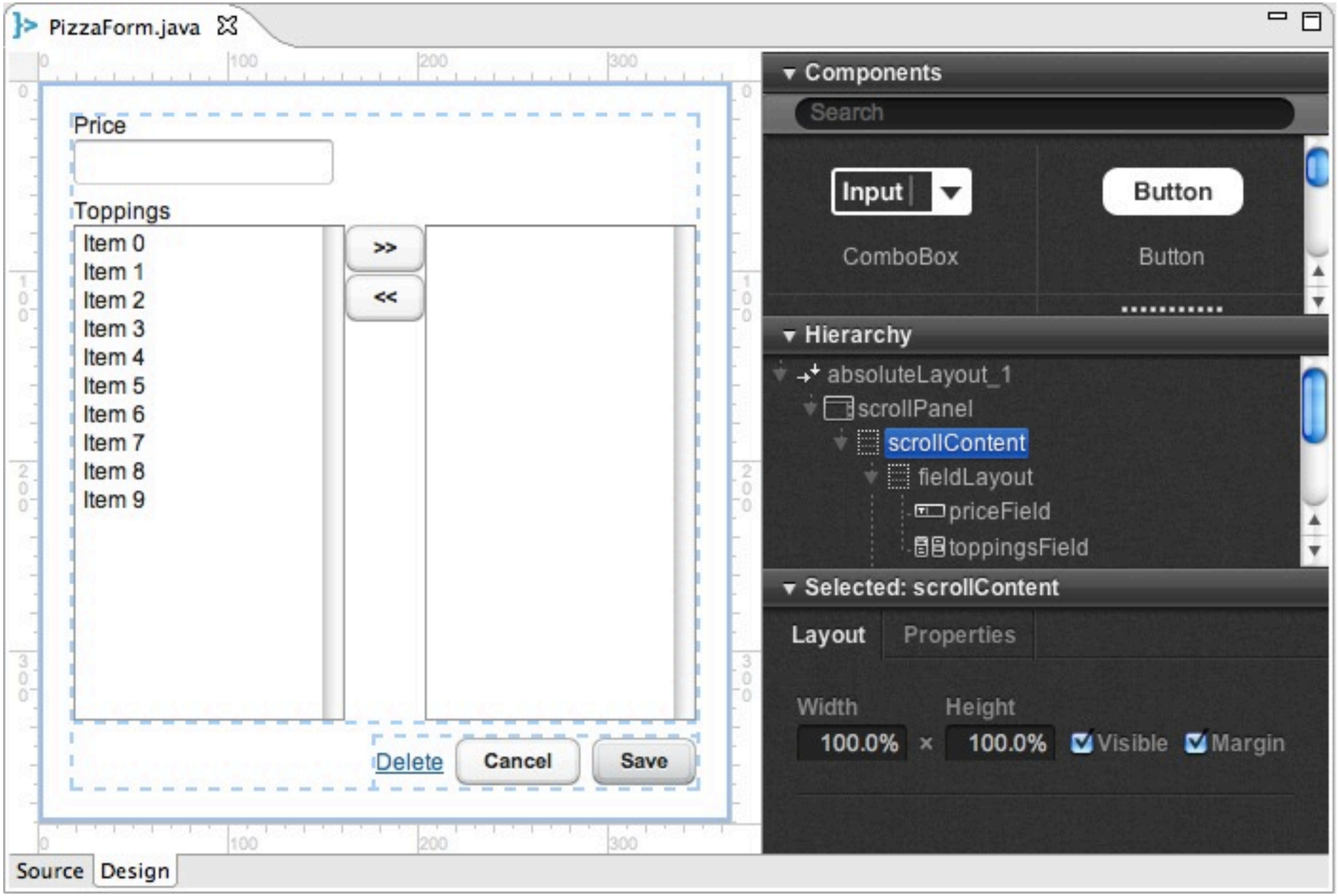
Price: 6.0

Toppings

Onion	>>	**Cheese**
Bacon	<<	**Tomato**
Olive		
Pepper		
Salami		
Mushroom		

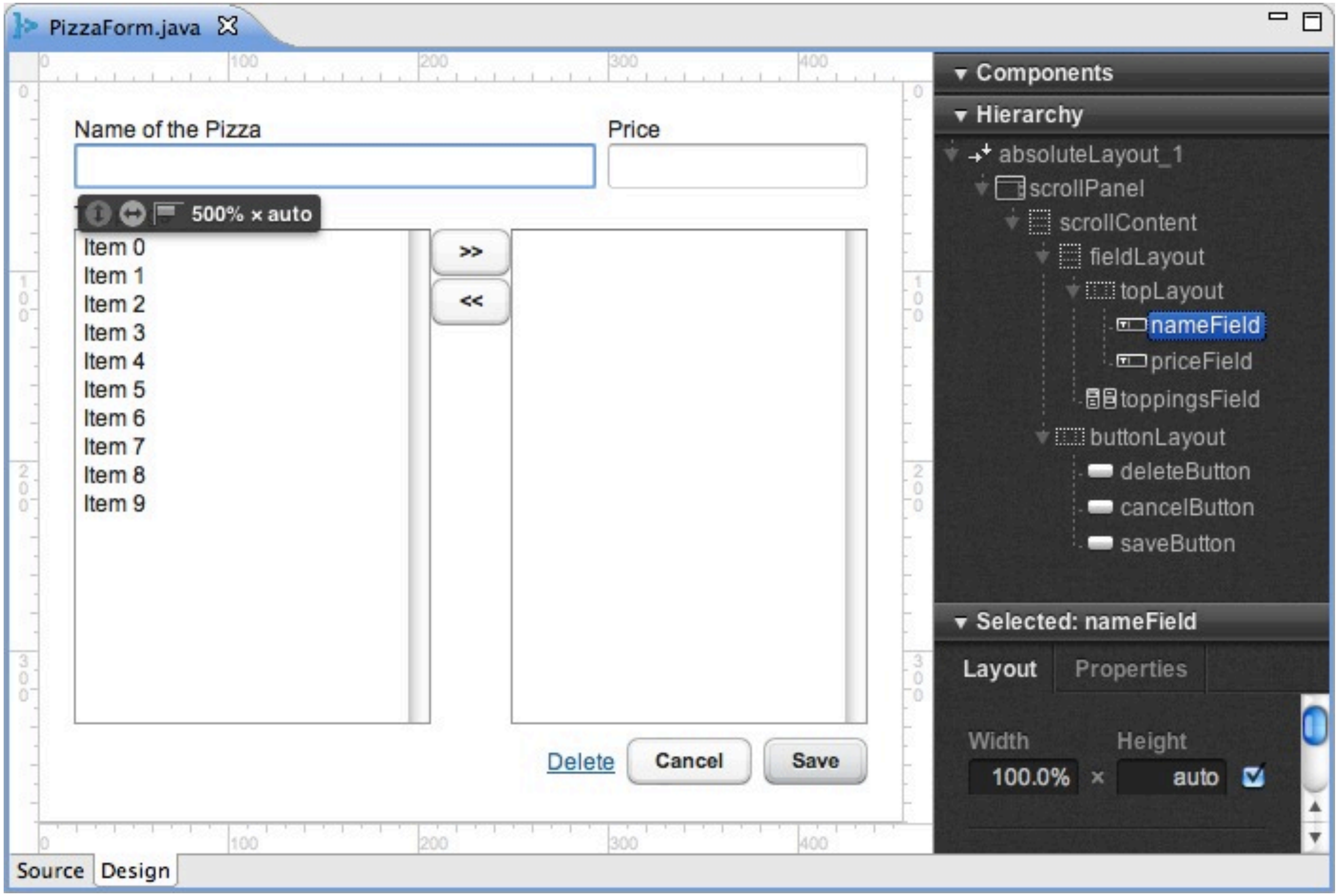

PizzaForm.java
Price
Toppings
Item 0
Item 1
Item 2
Item 3
Item 4
Item 5
Item 6
Item 7
Item 8
Item 9
>>
<<
Save
Cancel
Delete
Components
Search
Input
ComboBox
Button
Button
Hierarchy
absoluteLayout_1
scrollPanel
scrollContent
fieldLayout
priceField
toppingsField
Selected: absoluteLayout_1
Layout
Properties
Width
Height
auto
auto
Visible
Margin
Source
Design

vaadin }>

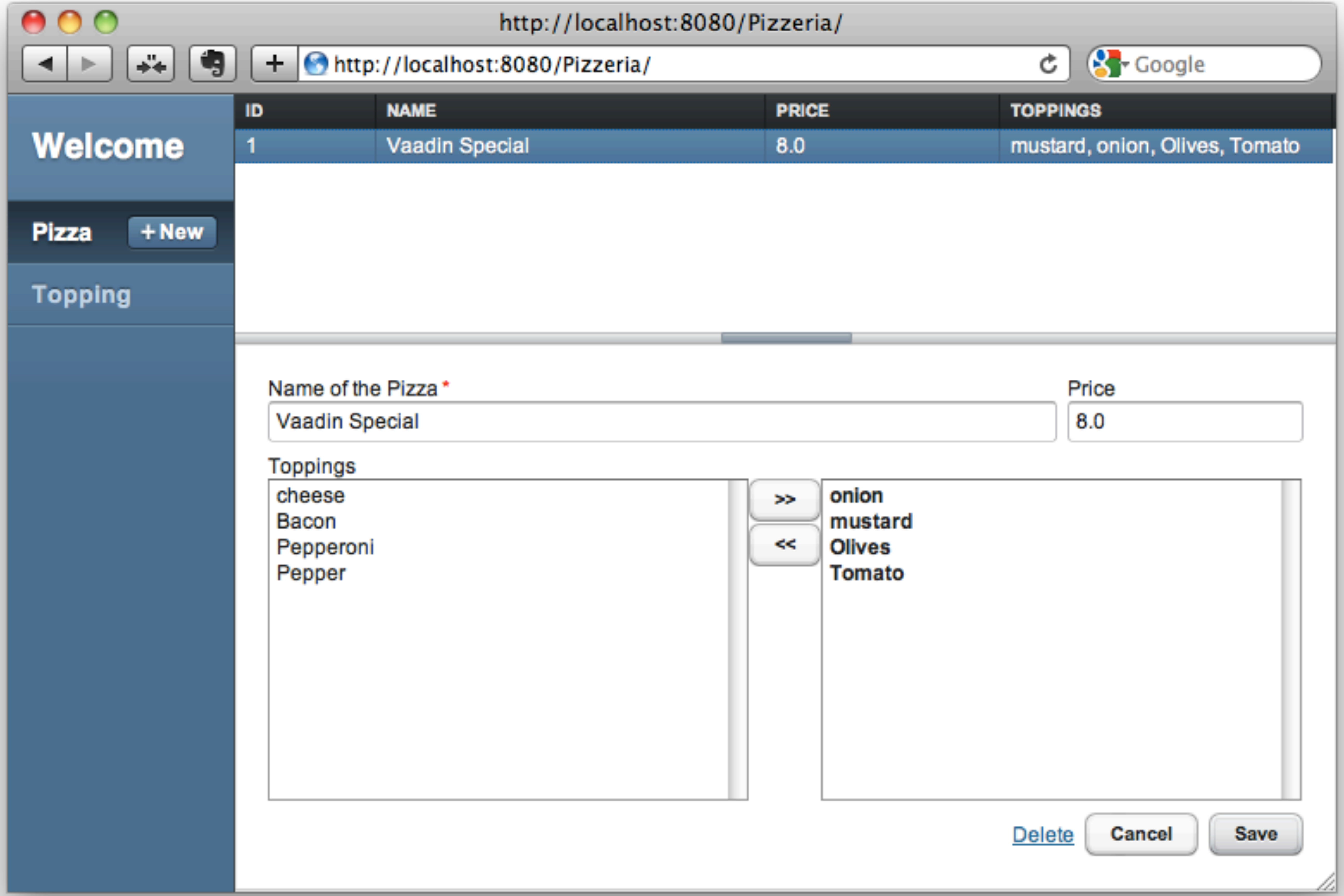


Width 100.0% × Height 100.0% ✓ Visible ✓ Margin

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



vaadin }>



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

Name of the Pizza *: Vaadin Special

Price: 8.0

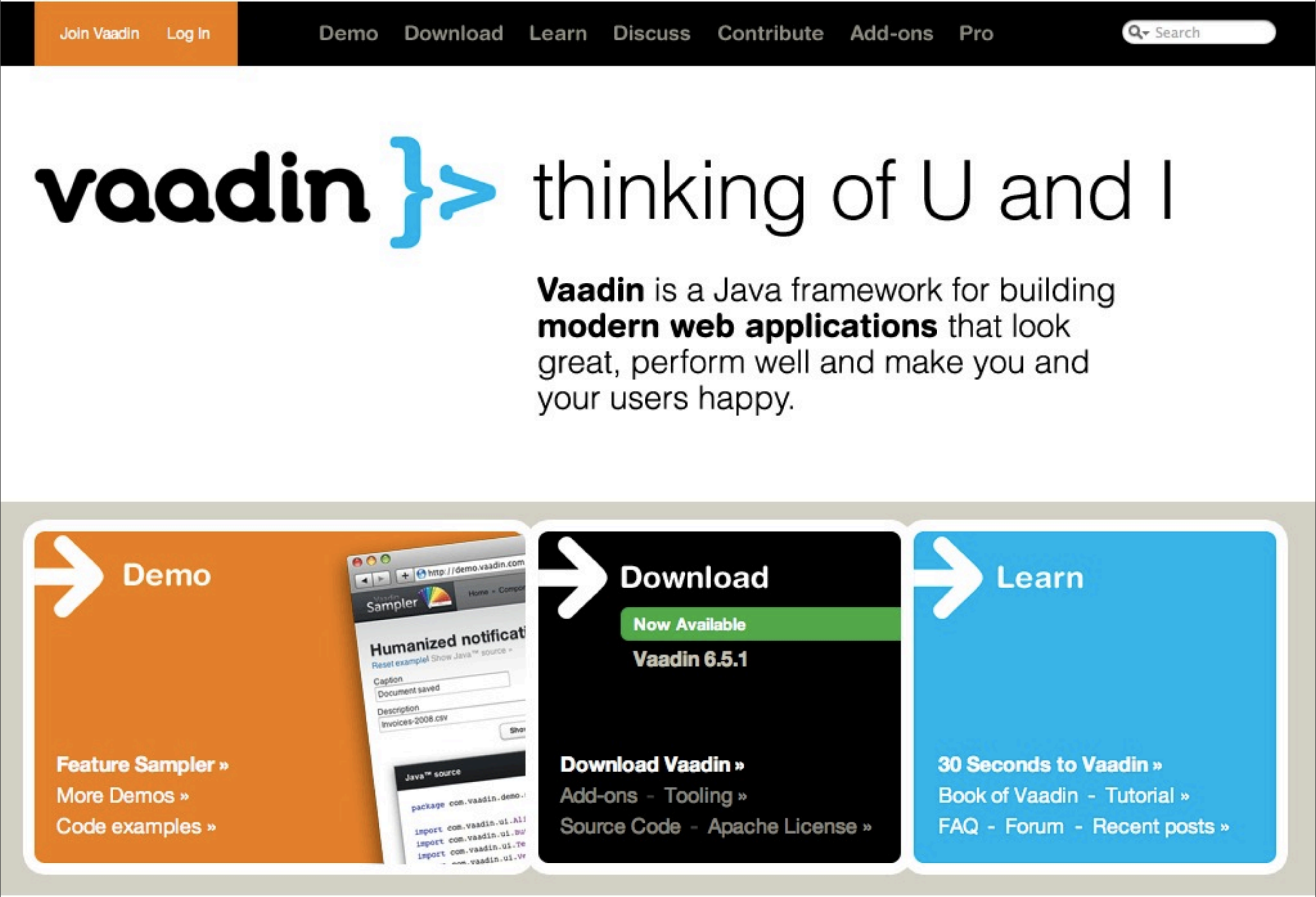
Toppings

cheese
Bacon
Pepperoni
Pepper

onion
mustard
Olives
Tomato

getting started

vaadin }>



Join Vaadin Log In

Demo Download Learn Discuss Contribute Add-ons Pro

Search

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

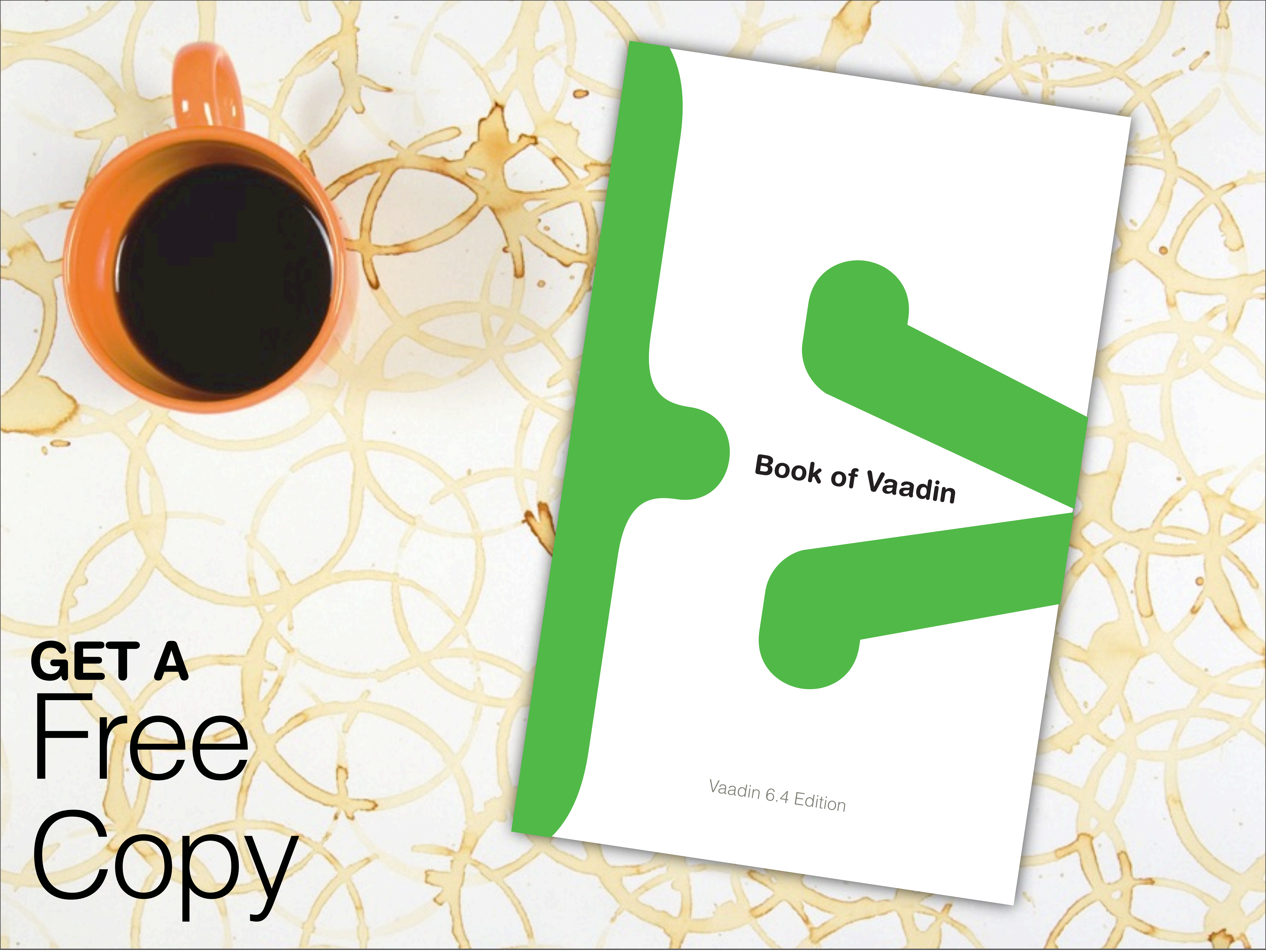
Now Available
Vaadin 6.5.1

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



#85

DZone Refcardz

brought to you by... vaadin }>

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.

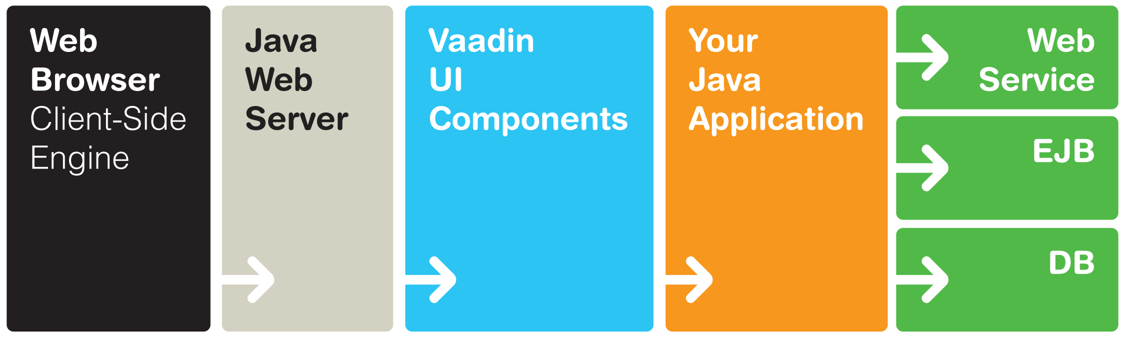


Figure 1: Vaadin Client-Server Architecture

If the built-in selection of components is not enough, you can

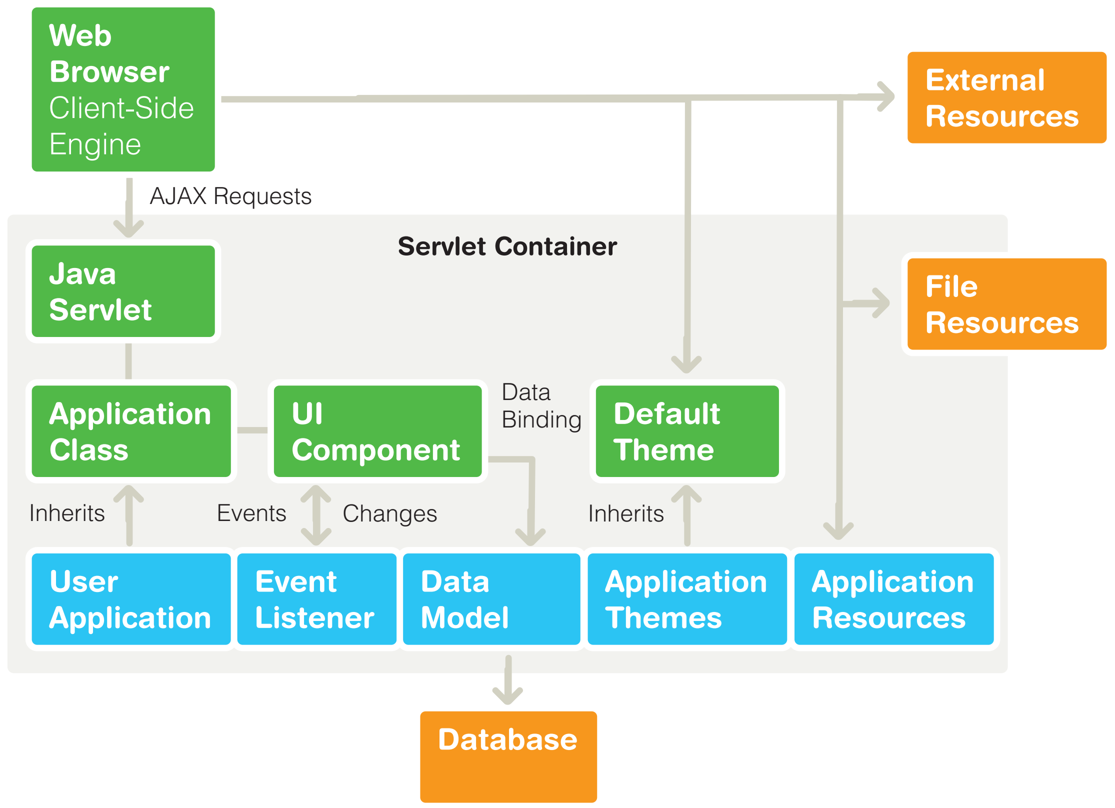


Figure 2: Architecture for Vaadin Applications

Hot Tip 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

Forums with
1000 msgs/m

Ask the
[really active, world wide]
Community

Questions
Comments

joonas@vaadin.com

vaadin.com/joonas

twitter: joonaslehtinen