

Vaadin 7 Now and in the Future

Artur Signell
VP, R&D
Vaadin

vaadin }>

”Is Vaadin 7 done yet?”

YES

934 tickets closed during **16 months** of development

Oldest ticket created **3/2008**

Newest ticket **2/2013**

3939 commits made by **23 authors**

93 % by **6 persons**

> 1 000 000 lines of code touched

6946 new Java files from GWT

876 new Java files since Vaadin 6.8

Vaadin 7.0.0 contains **1 455 192 lines** of
Java code

29% of the code is **test code**

**What has changed since
Vaadin 6?**

It's OK to speak about
HTML and **CSS**

It's OK to talk about
servlets and **HTTP sessions**

It's OK to do things on
the client side

**Much work on internal
code and structure**

**Give the user more power,
more possibilities**

.. instead of hiding things

7

Features

UI, Browser Tabs, Navigation

Mapping URI fragments to views

```
public class MyUI extends UI {  
    public void init(...) {  
        Navigator n =  
            new Navigator(this, this);  
        n.addView("persons",  
                PersonView.class);  
    }  
}
```

Automatic support for multiple browser tabs

Navigation API to make bookmarkable apps easy

URI, fragment, parameters, screen size available during init

Support for UI selection based on browser data

Forms and Data Binding

FieldGroup for data binding

Typed fields

Converters for formatting and parsing

Bean Validation

CustomField

Two phase commit

```
public class Person {  
    @Size(min = 5, max = 50)  
    private String name;
```

```
    @Min(0)  
    @Max(100)  
    private int age;
```

```
}
```

```
public class MyForm extends  
VerticalLayout {  
    TextField ageField;
```

```
    ...  
    fieldGroup.bind(ageField, "age");
```

```
    ...  
}
```

Creating and Customizing Widgets

```
public class ClickExtension
extends
AbstractExtensionConnector {
protected void extend(... target) {
...
targetWidget.addDomHandler(
new ClickHandler() {
public void onClick(...) {
Window.alert("Click on: " +
targetWidget);
}
...
}
```

GWT built-in

Client side uses widget and connector

Support for Javascript components

Extensions to add functionality without inheritance

Improved Communication

State shared
automatically between
server and client

Static typing and
complex types

Only changes sent

RPC in both directions

```
public class MyComponent
extends AbstractComponent {
    ...
    getState().text = "hello";
    ...
}

public class MyConnector ... {
    ...
    Window.alert("Server side says:
    " + getState().text);
    ...
}
```

Theming Improvements

Sassy CSS

```
$color: #ffecc6;
.v-tooltip {
  color: $color;
  &.dark {
    color: darken($color, 10%)
  }
}
```

Removed inline pixel sizes and reduced inline styles

Use border/margin/padding wherever you want

CSS

```
.v-tooltip {
  color: #ffecc6;
}
.v-tooltip.dark {
  color: #ffdb94;
}
```

Multiple themes on the same page

Mixing component styles from different themes

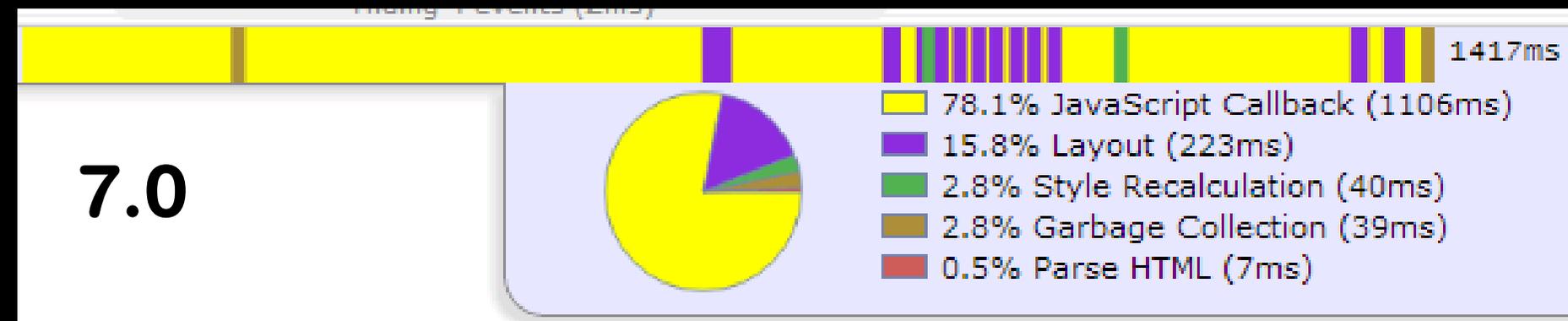
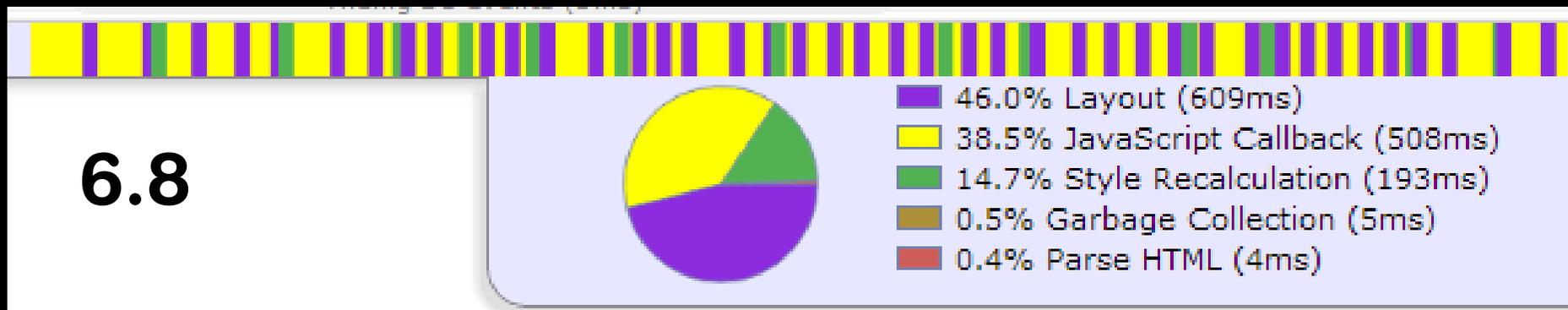
Performance

No or minimal layout calculations

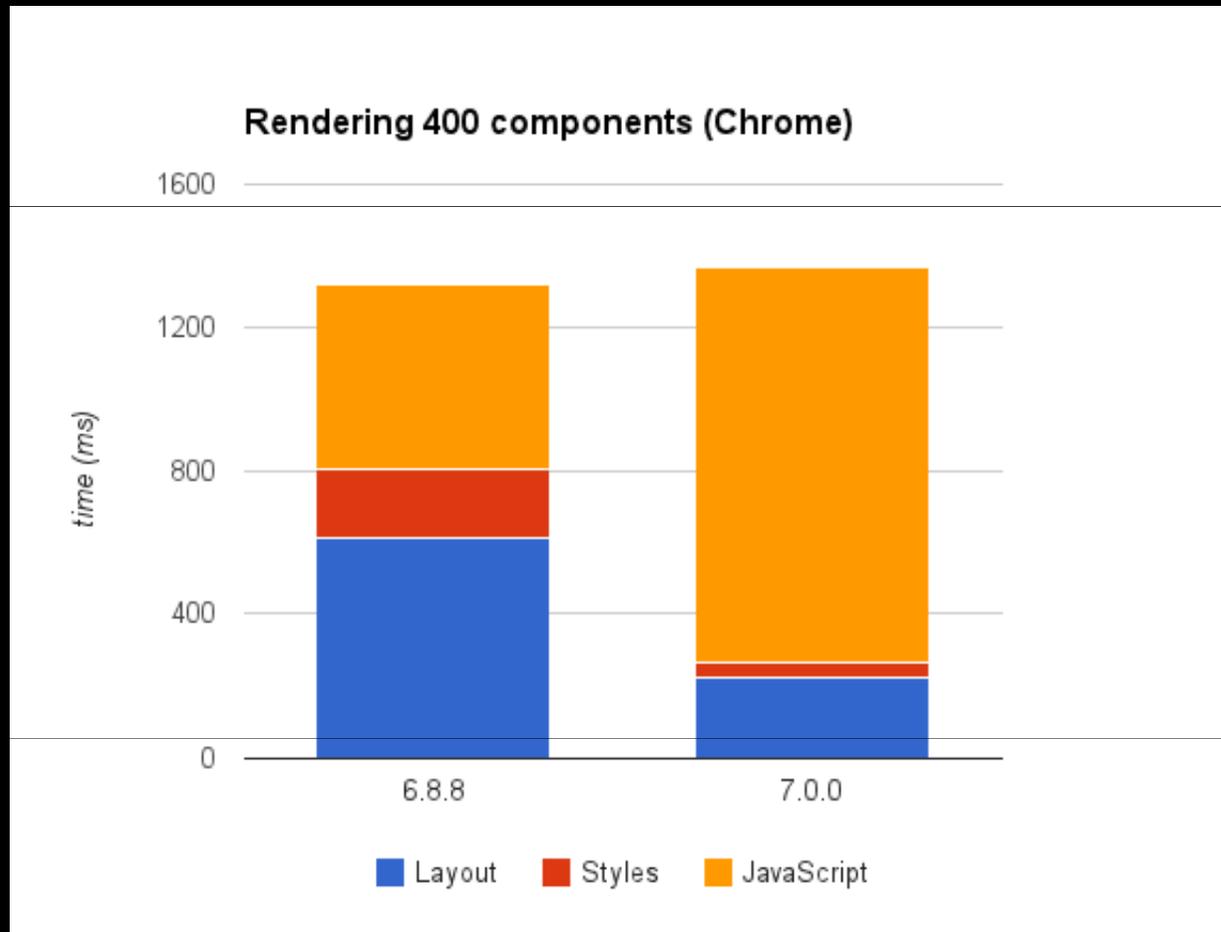
Layout responsibility moved to browser

Lighter DOM

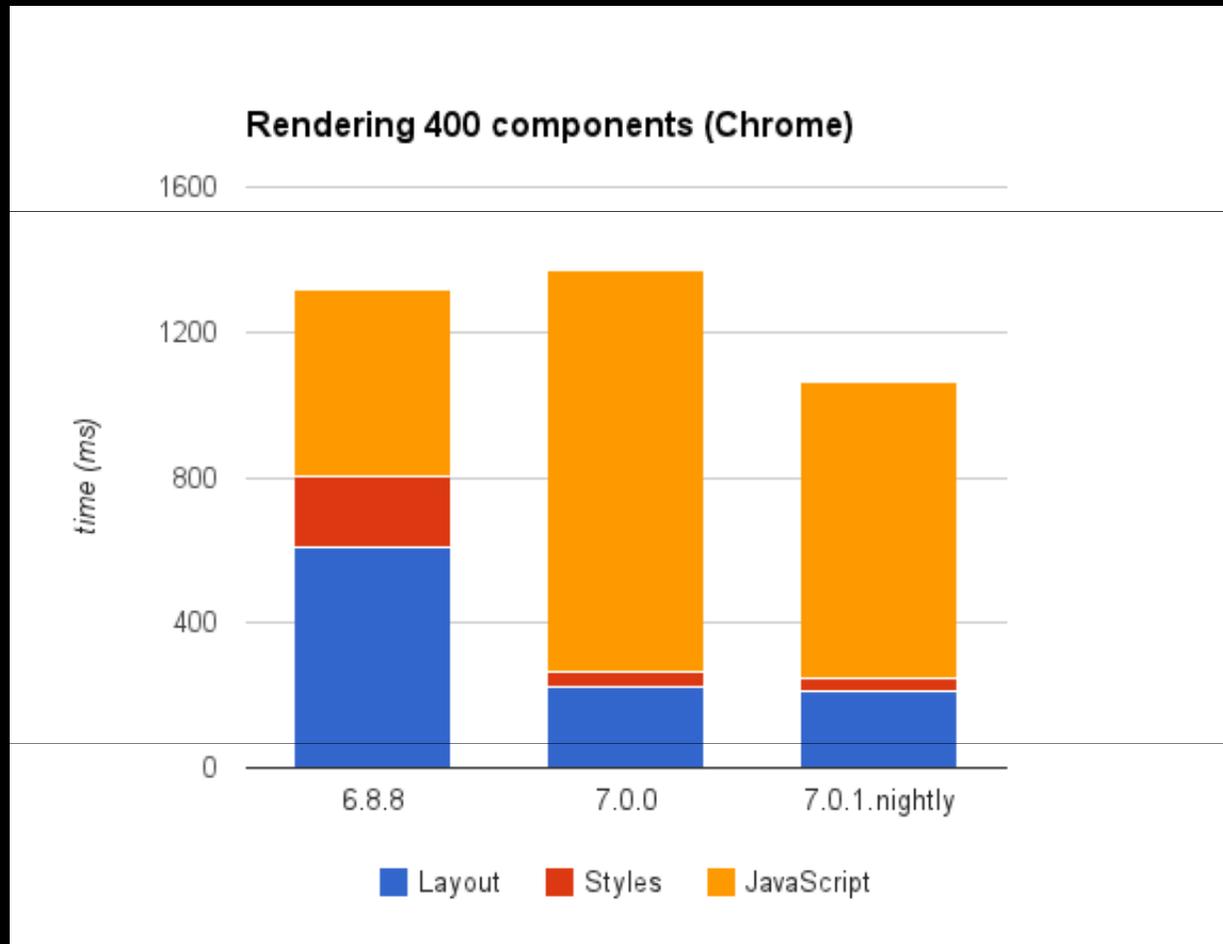
Minimized reflows



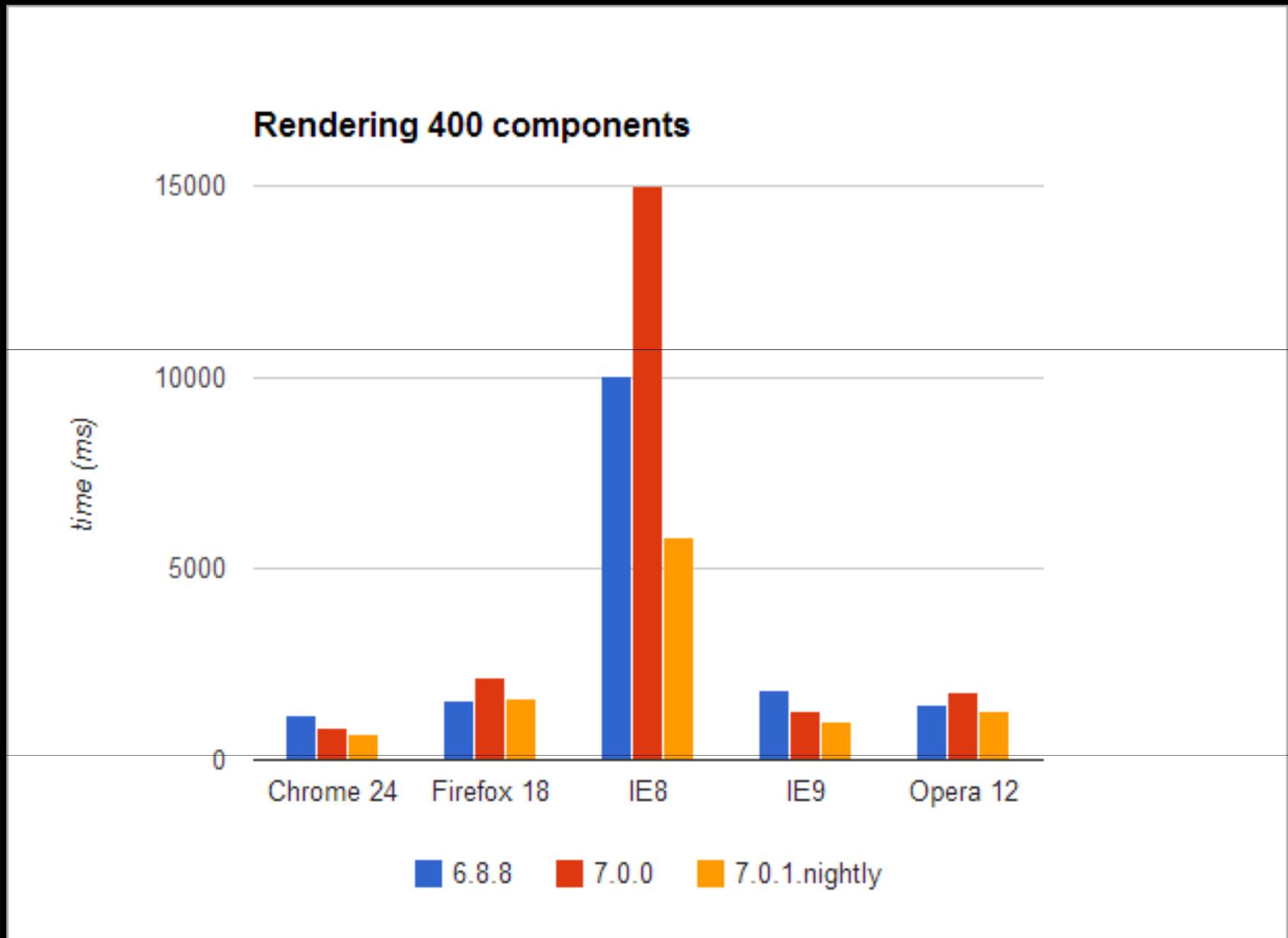
Did you compare the total times?



We still have some work to do...



.. but it is looking promising



... even for Internet Explorer

There are more features...

Optionally keeps session alive
Reliable browser window close monitoring
Redesigned layouts
Minimal or no layout calculations to maximize layout speed
Full control of layouts with CSS including borders and margins
Redesigned lighter DOM for vertical, horizontal and css layout
Split to seven jars to allow deploying only what you need
Hierarchical error handling
Delegate component errors to parent components
All component containers now support addComponents(Component...)
Support adding multiple components in constructors
UI replaces Application
New architecture makes writing Vaadin applications simpler
Supports multiple browser tabs by default
Closing UI programmatically both from background thread and in ui event handlers
Reliable clean of UI when browser is closed
Multiple concurrent themes
Support for showing components with different themes on one page at once
Theme names are included in CSS selectors
Browser and request details available in init
Direct access to request and session on UI init
Access detected browser details on UI init
Call server-side methods from clients-side in Java
Call client-side methods from server
Static typing allows compile time checking
Supports: primitive types, boxed primitive types, String, beans, Connectors, enum, arrays, URLReference, lists, maps, sets
Delayed RPC allows piggybacking the next XHR call instead of calling to server immediately
Supports optional folding of RPC calls together preserving only last value
Connectors provide a flexible communication channel between client and server
Separating communication code from widgets promotes reusability
Allow calculating component state on the fly just before state is sent to client
Only parts of the state that are modified are sent over the wire
Only parts that differ from default values are sent in initial state synchronization
Nothing is sent for unchanged components
Java objects can be shared between client and server for easy component development
State is automatically mirrored from server to client
State supports both public fields and bean properties
High level view navigation
Support for URI fragment based view management
Support for registering both views instances as well as view classes
Programmatic navigation with navigateTo()
Supports saving bookmarks to views
Supports parameterized views
Views can block navigation
ColorPicker component, easy to use interface with clickable color gradients
Color history
Color preview
Write addFooListener() instead of generic addListener(FooEvent) - better code completion support
Listeners compatible with Java 8 Lambdas
Automated repaints
No more need to call requestRepaint() in your components
Renewed Vaadin Maven Plugin
Now includes features from GWT Maven Plugin

Renewed Eclipse Plugin adds Apache Ivy based dependency management
Page bootstrapping renewed
Simpler inclusion of Vaadin UIs to custom web pages
Default UI provider chooses UI based on servlet parameters
Allows providing a different UI based on request parameters, URI or browser
Full control over session lifecycle
Abstract away session from servlets and portlets
Access deployment information and HTTP requests easily
Abstract away from servlets and portlets
Generics in data model
Property getValue/setValue() use expected type
Two phase commit support
Transactional properties support commit/rollback
Component extension API
Add functionality and customizations to any component
Modify DOM and hook event listeners
JavaScript callbacks
Declare client-side JavaScript API from server
Eases integration with parts of the page not controlled with Vaadin
Relative paths on host page gives more flexible deployment
Supports Apache ProxyPass and other similar proxies
New Maven archetype eases creation of Vaadin 7 applications
Supports widgetset compilation and precompiled widgetsets
Uses HTML5 doctype
Use any parts of HTML5 in your application
State change listeners
Client-side can listen to shared state changes to update just the changed parts of the widget
Listen to page resizes
Control navigation
Loading custom JavaScript
Annotate server-side classes with @JavaScript to request loading of JavaScript files
Automated control of loading order and ensuring that files are loaded only once
Cleaned API by removing API deprecated in Vaadin 6 or before
Provide enums everywhere where possible instead of constants
Specialized UI component for showing images
Specialized UI component for embedding web pages with iframes
Flash component
Supports the latest Firefox extended support release in addition to latest stable Firefox release
Sass Compiler implemented in Java
Allows modularization of themes for better reuse and easier maintenance
Support the most important features of Scss
On the fly conversion of Scss to css in servlet
Built in themes are now based on Sass
Can be used in client-side projects as well
State can be annotated to automatically delegate state changes to corresponding widgets
Removed need to implement state change listeners and calling widget methods
A full copy of Google Web Toolkit is included in Vaadin Framework
Vaadin team maintains a branch of GWT that brings newest features and bug fixes early on
All functionality of GWT is included to enable writing of client side UIs, stateless applications, offline functionality and custom widgets
Included Elemental library gives direct access to all cutting edge browser features
Both browser plugin based dev mode debugging as well as super dev mode

are supported
Direct access to VaadinService, VaadinRequest, VaadinResponse, VaadinSession and UI from anywhere during HTTP request
Service and UI are accessible also from outside HTTP request
Component ID replaces debug ids to allow wider use possibilities for identifying corresponding widget elements in DOM
API for retrieving a range of Item ids in Container.Indexed to optimize performance
Native support for percent sizes
Letting browser do the percent to pixel calculation speeds up rendering
Allow specifying custom class loaders to better support Java EE, CDI and Spring
Supports both servlets and portlets
Wrap around any existing JavaScript based widget to adapt it for use in Vaadin
No widgetset compilation needed
Write client-side connectors in JavaScript
Support for shared state and rpc
Let add-ons modify the bootstrap page to remove need for complicated custom servlets
Annotate beans with JSR-303 standard annotations to automatically create validators for the fields
Allow data binding of multiple fields together to item data source
Fieldgroup supports buffering
Supports two phase commit
Annotation based and field name based property mapping
All fields support explicit bidirectional conversion from presentation format to data source format
Allow setting converters to components directly in components or for the whole application at once
Built-in default converters
Automated conversions between String, Boolean, Long, Date, Double, Float, Integer and Number
Conversion direction reversion
Built in converters support internationalization
Custom field component for building new fields as composition of existing components
Simplified validation API. No need to implement isValid() in validators anymore
Really old browsers are detected and customizable instructions given on updating the browser
BeanItem supports nested properties to allow flattening complex datatypes
Optional preservation of UI state on refresh
A set of helper classes are provided to ease migration from Vaadin 6
Unification of API for component parents
Window and Panel components now require setting layout explicitly
Converters can be set per Table column to customize column formatting
Allows building custom layout calculations for widgets when browser based layouts are not powerful enough
Optimizes number of reflows by batching layout calculations from multiple widgets together
Client-side component hierarchy
Access the components hierarchy enables optimizations
List connector hierarchy in debug console
Automatic injection of CSS files
Inject custom CSS with @StyleSheet annotation
Open popups and start downloads in a way not stopped by popup blockers
Mapping between server-side and client-side component annotations to avoid server-side classpath issues
Access computed CSS styles on the client-side



What next?

Maintenance releases of
Vaadin 7 **every other week**

Maintenance releases of
Vaadin 6 **every four weeks**

Server Push

**SASS
arithmetics**

**Debug Console
Improvements**

**Native IE10
support**

Vaadin 7.1

**Vaadin
Calendar**
(Apache 2)

**Dynamic CSS
Injection**

April 2013

Contexts and Dependency Injection

Vaadin CDI

Annotate Vaadin UI and Views

```
@VaadinUI  
public class MyUI extends UI
```

<http://mydomain.com/myapp/>

```
@VaadinView  
public MainView implements View
```

<http://mydomain.com/myapp/#!/mainview>

Dependency Injection

```
@VaadinUI  
public class MyUI extends UI {  
  
    @EJB  
    private MyEJB ejb;  
  
    @Inject  
    private Event<MyEvent> event;
```

Vaadin CDI beta

Apache 2 license

March 2013

Vaadin Plug-in for Spring Roo

Vaadin WSRP

are discontinued

Vaadin XS

will be ported to Vaadin 7 at a
later date

Vaadin Control Panel for Liferay

Vaadin 7 version is being developed by Arcusys

Vaadin JPAContainer 3.0

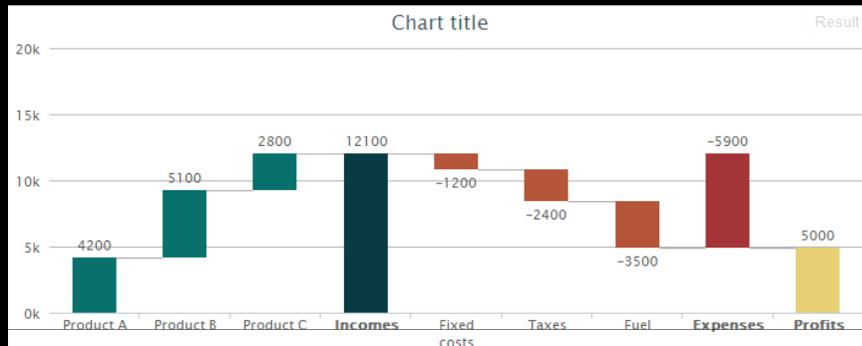
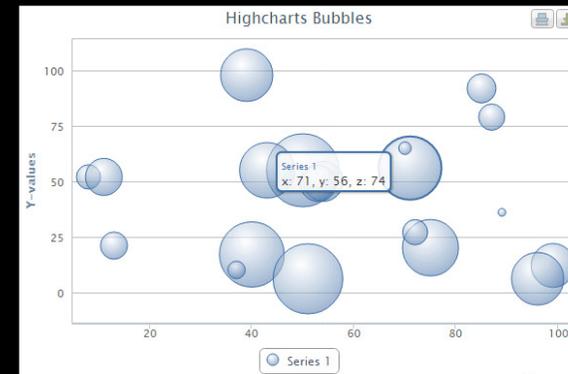
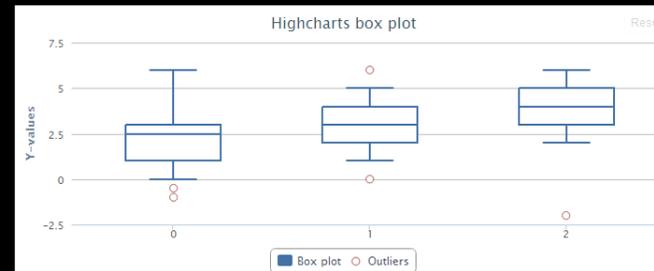
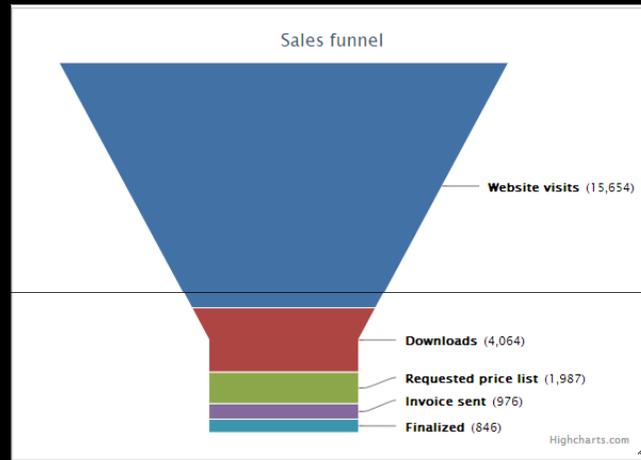
Apache 2 license

February 2013



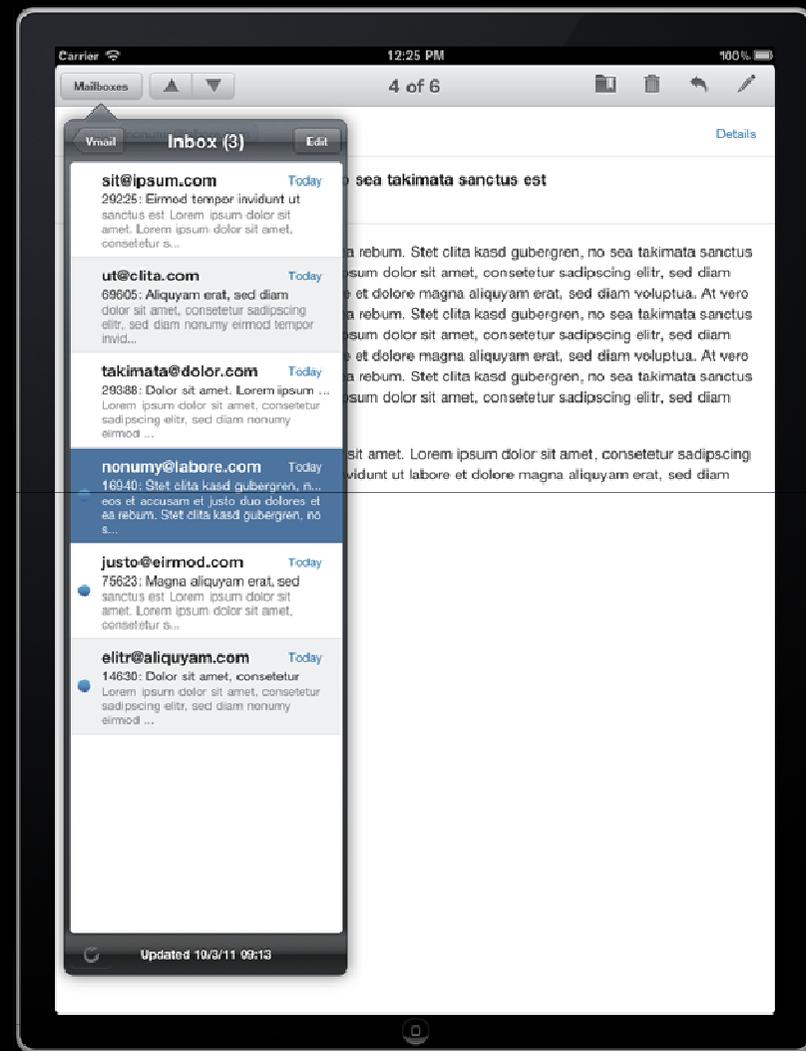
Vaadin Charts 1.0

February 2013



Vaadin Charts 1.1

March 2013



Vaadin TouchKit 3.0

March 2013

The New Table

Work start during Q1

Not expecting results before Q3

Vaadin 7 is on GitHub

Fork it at

<https://github.com/vaadin/vaadin>

<https://github.com/vaadin/gwt>

Thank you for your attention

**Vaadin 7.0.0
available right
now**

Questions?



Try it out

Make some cool stuff

