vaadin }> Cheat Sheet

For Vaadin 6.2

Server-Side Components

Legend

Interface — Inner
Class — Inner
Abstract Class — Inner

Inherits or implements

method()
abstractMethod() (must or may implement)

Data Model

Ordered   Indexed
Filterable   Sortable
Hierarchical

Viewer / Editor — **Container** — *addItem()*   n
Viewer / Editor — **Item** — *addItemProperty()*   n
Viewer / Editor — **Property** — setValue() getValue()

ValueChangeEvent
ValueChangeListener — *valueChange()*

IndexedContainer
HierarchicalContainer

VariableOwner — *changeVariables()*

Paintable — *paint()* setDebugId()

Sizeable
setWidth()
setHeight()
setFullSize()
setUndefinedSize()

focus()
setTabIndex()
Focusable

Component
Event / Listener
addStyleName/Listener()
setCaption/Icon()
setEnabled/Visible()
getParent(), attach/detach()

Field
setRequired()
setRequiredError()

commit/discard()
setWriteThrough()
setReadThrough()
Buffered

BufferedValidatable   **Validatable**
addValidator()
isValid()
validate()

Embedded v-container
Link v-link
Label v-label
v-menubar
MenuBar
MenuItem   Command
addItem()

AbstractComponent
paintContent()
setDescription()
setImmediate()

Upload v-upload

AbstractField

buttonClick()
ClickListener
ClickEvent
Button v-button
CheckBox v-checkbox

ProgressIndicator v-progressindicator
setPollingInterval/Indeterminate()

TextField v-textfield
RichTextArea v-richtextarea
setRows/Columns/Secret/InputPrompt/MaxLength()

LoginForm v-loginform
Slider v-slider
setOrientation/Resolution()
setMin/Max()

DateField v-datefield
InlineDateField v-inlinedatefield
setResolution()

CustomComponent v-customcomponent
setCompositionRoot()

Buffered Validatable Item Editor

Form v-form
setLayout()
setFooter()
PopupDateField v-popupdatefield

addComponent()

ComponentContainer

setContent()   1

v-panel
Panel

AbstractComponentContainer

AbstractSelect
Container

Select v-select
ComboBox v-filterselect
setFormFieldFactory()

setMultiSelect()

v-window
Window

PopupView v-popupview

v-tabsheet
TabSheet   addTab()   n   **Tab**
setVisible/Enabled()
setCaption/Icon()
setDescription()

com.vaadin.
Application *init()*
Your Application
setMainWindow()   1
setTheme/Locale/LogoutURL()

Accordion v-accordion

ListSelect v-listselect
NativeSelect v-nativeselect
TwinColSelect v-twincolselect
OptionGroup v-optiongroup

FormFieldFactory   *createField()*   1
DefaultFieldFactory
TableFieldFactory *createField()*

Layout
setMargin()

AbstractLayout

ComponentPosition
AbsoluteLayout v-absolutelayout
getPosition()

CustomLayout v-customlayout
CustomLayout("template")

SplitPanel v-splitpanel-...
setSplitPosition/Locked()

GridLayout v-gridlayout
addComponent(c)/(c, col, row)/(c, c1, r1, c2, r2)

ColumnGenerator
CellStyleGenerator

Table v-table
setTableFieldFactory()   1
Container.Ordered
Container.Sortable

FormLayout v-formlayout
AbstractOrderedLayout

HorizontalLayout v-horizontallayout
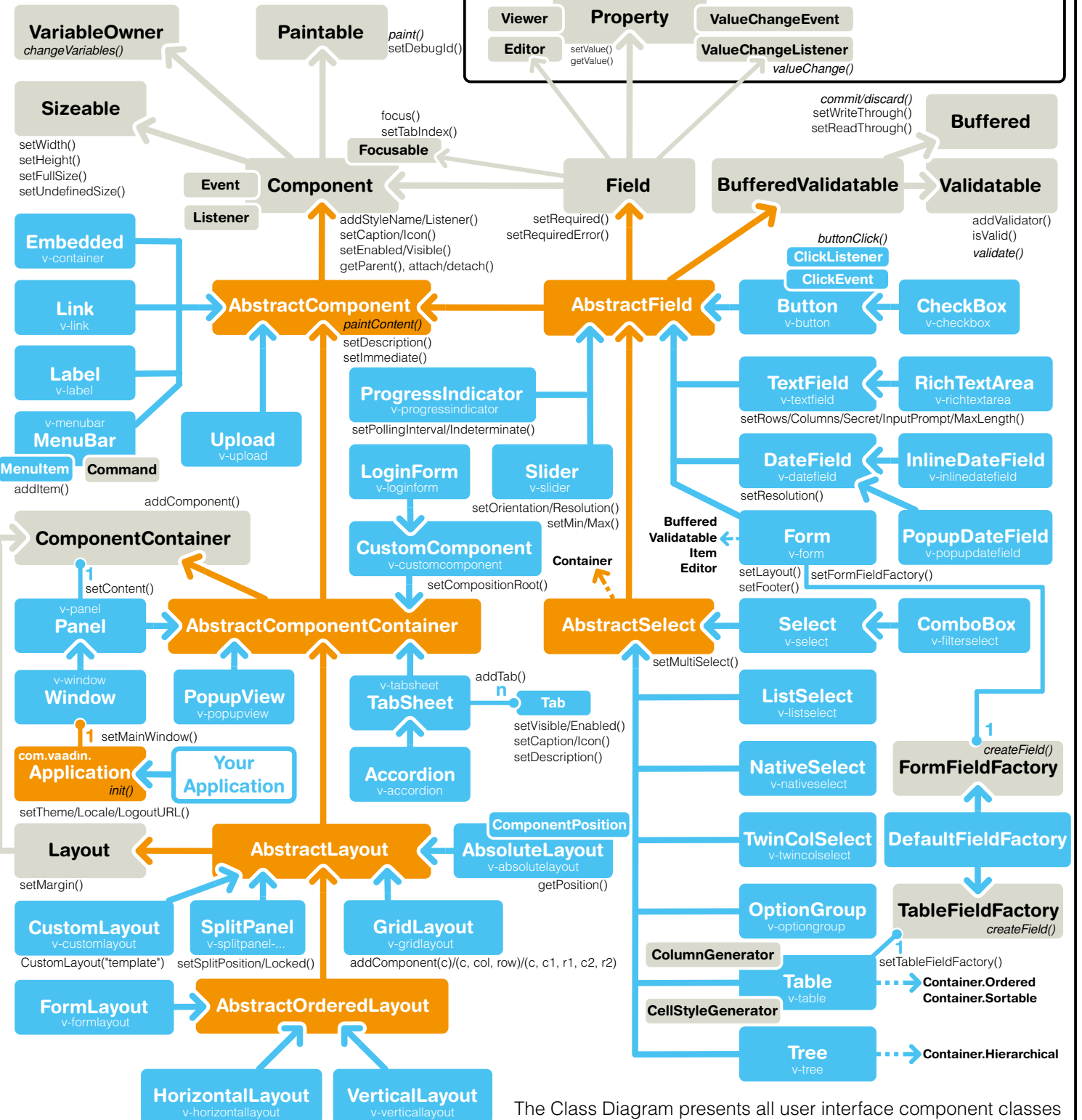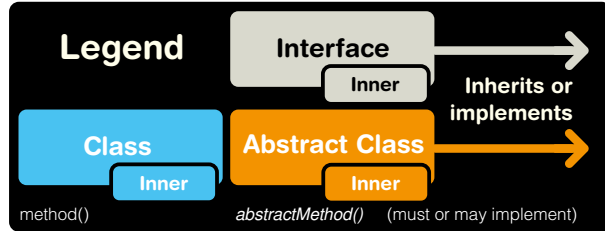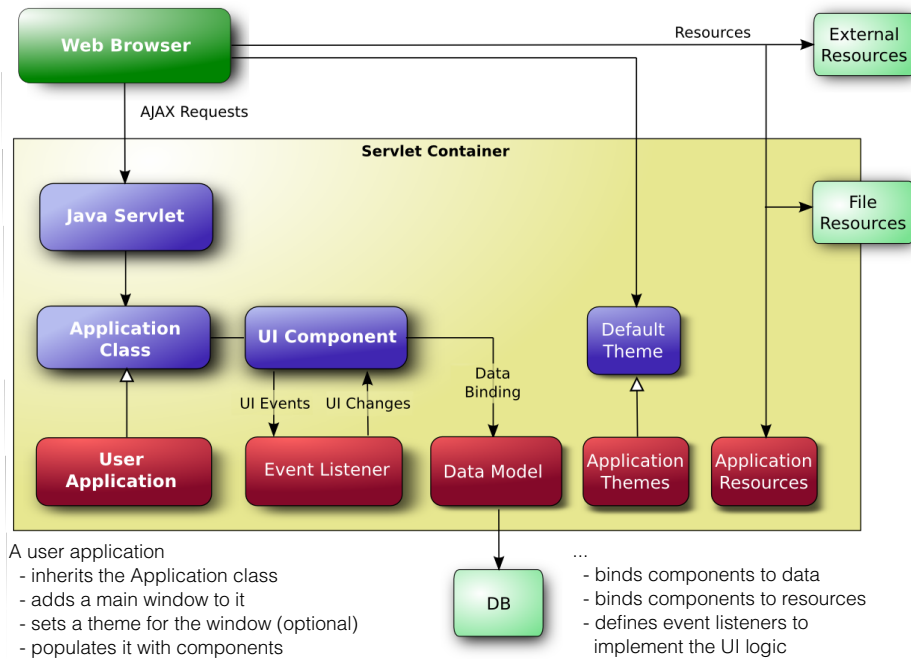**VerticalLayout** v-verticallayout
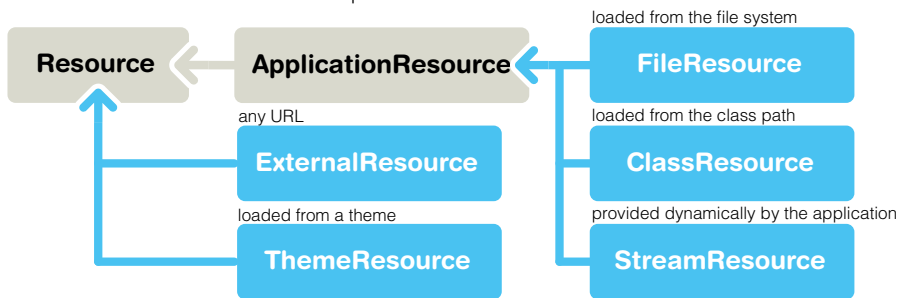
Tree v-tree
Container.Hierarchical

The Class Diagram presents all user interface component classes and the most important interfaces, relationships, and methods.

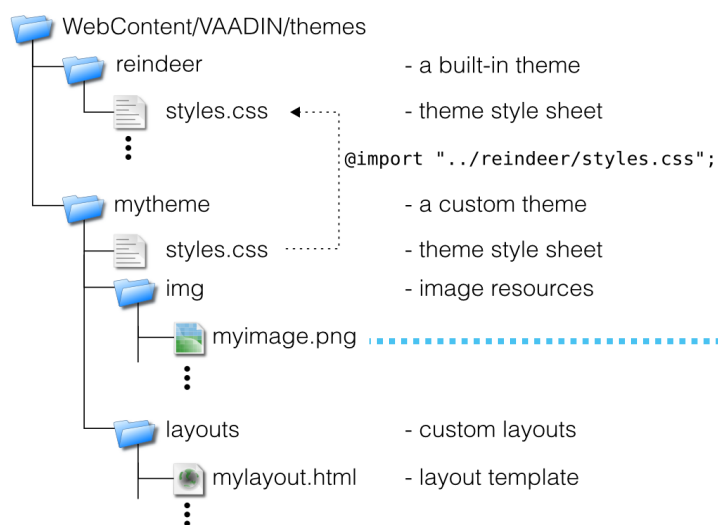## Application Architecture for Vaadin Applications

A user application
- inherits the Application class
- adds a main window to it
- sets a theme for the window (optional)
- populates it with components

...
- binds components to data
- binds components to resources
- defines event listeners to implement the UI logic

Resources

All images, embedded objects, and downloadable files are represented as resources.

loaded from the file system
any URL
loaded from the class path
loaded from a theme
provided dynamically by the application

Client-Side Widget Integration

Google Web Toolkit

Any GWT widget - an existing or your own

Widget ... **MyWidget**

Vaadin Client-Side Integration

Must implement updateFromUIDL() to deserialize state from server

com.vaadin.terminal.gwt.client.
Paintable
updateFromUIDL()

VMyWidget

n

com.vaadin.terminal.gwt.client.
ApplicationConnection
updateVariable()

1

Needs to call updateVariable() to to serialize state to server

Makes XMLHttpRequest

Server connection: UIDL / JSON / HTTP(S)

Server-Side Integration

com.vaadin.terminal.server.
CommunicationManager

n

com.vaadin.terminal.
Paintable
paint()

com.vaadin.terminal.
VariableOwner
changeVariables()

com.vaadin.ui.
Component

com.vaadin.terminal.
PaintTarget

(Implements paint())

AbstractComponent
getTag()   *paintContent()*

addAttribute()
addVariable()

...

MyComponent

Must implement changeVariables() for deserialization and paintContent() for serialization using the PaintTarget interface.

Theme Contents

WebContent/VAADIN/themes
- reindeer — a built-in theme
  - styles.css — theme style sheet
  - `@import "../reindeer/styles.css";`
- mytheme — a custom theme
  - styles.css — theme style sheet
  - img — image resources
    - myimage.png
  - layouts — custom layouts
    - mylayout.html — layout template

Use a custom theme by setting it in the main window object:
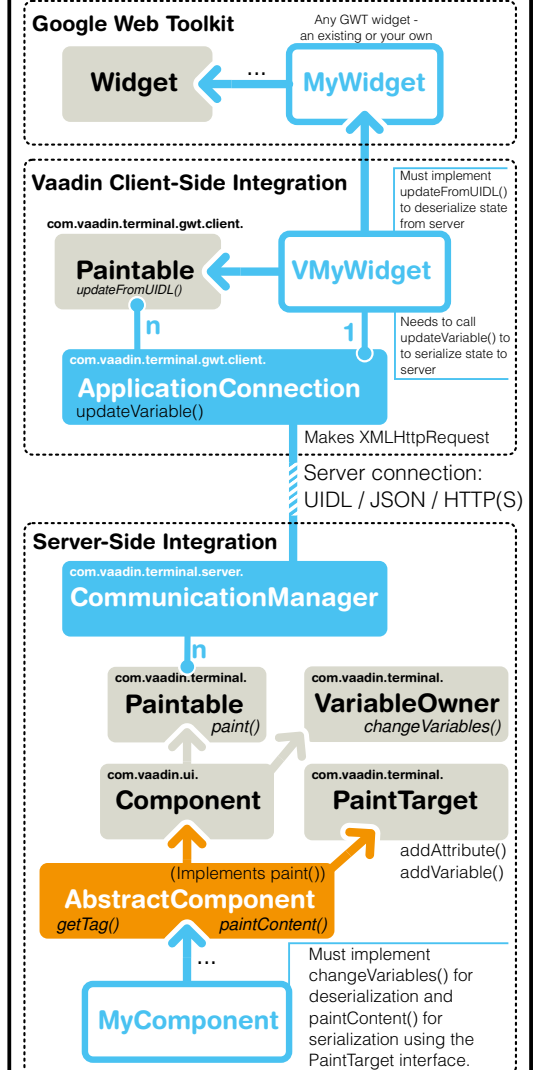
```
mainWindow.setTheme("mytheme");
```

Use images included in a team as a ThemeResource:

```
Button button = new Button("Button with Icon");
button.setIcon(new ThemeResource("img/myimage.png"));
```

Use a HTML template having `<div location="hello"/>` location tags with the CustomLayout component. Add components by their location tag:
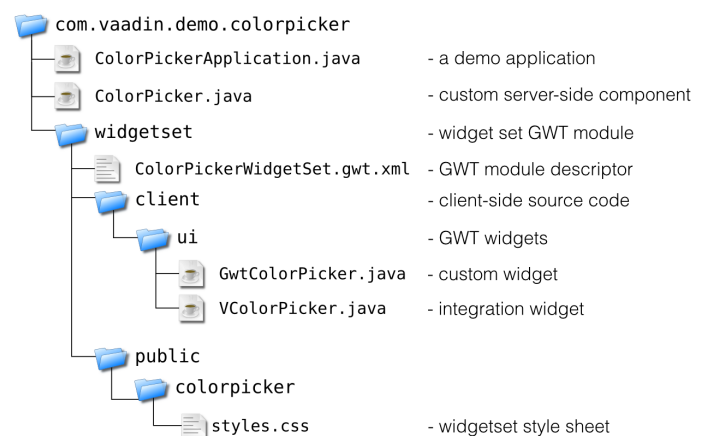
```
CustomLayout layout = new CustomLayout("mylayout");
layout.addComponent(new Button("Hello"), "hello");
```

Widget Project

(See the Color Picker demo)

com.vaadin.demo.colorpicker
- ColorPickerApplication.java — a demo application
- ColorPicker.java — custom server-side component
- widgetset — widget set GWT module
  - ColorPickerWidgetSet.gwt.xml — GWT module descriptor
  - client — client-side source code
    - ui — GWT widgets
      - GwtColorPicker.java — custom widget
      - VColorPicker.java — integration widget
  - public
    - colorpicker
      - styles.css — widgetset style sheet

Hello, World!

```java
public class MyApp extends com.vaadin.Application {
    public void init() {
        Window main = new Window("Hello Window");
        setMainWindow(main);

        main.setTheme("mytheme"); // Optional

        Label label = new Label("Hello, World!");
        main.addComponent(label);
    }
}
```