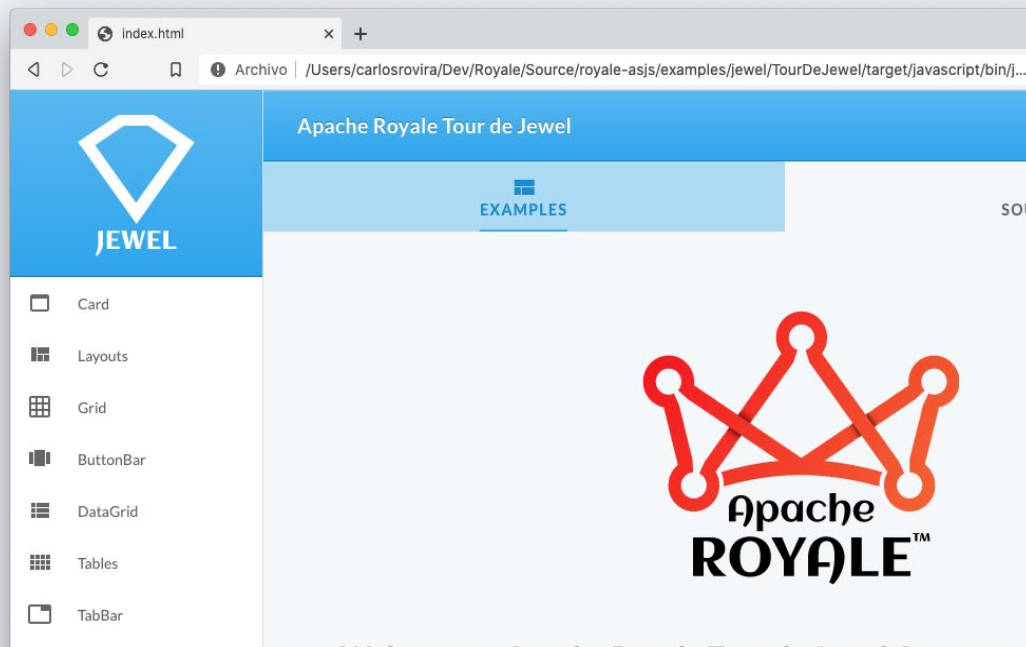


Apache Royale

Tour De Jewel

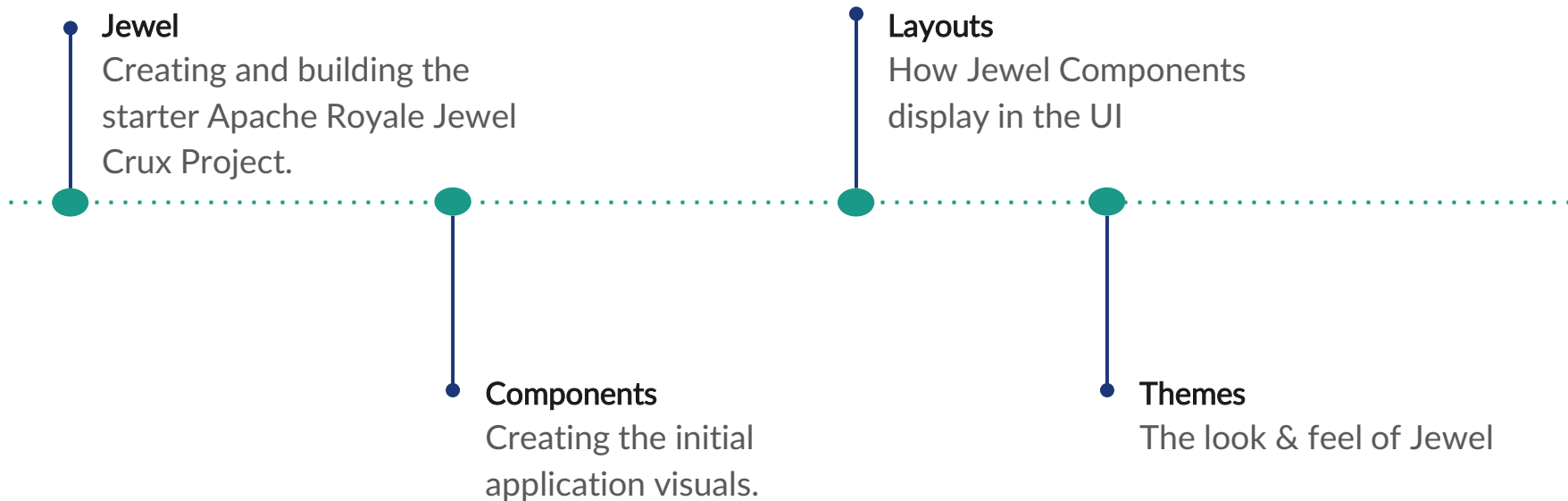
01.10.2020 18:55 CEST

Carlos Rovira & Andrew Wetmore



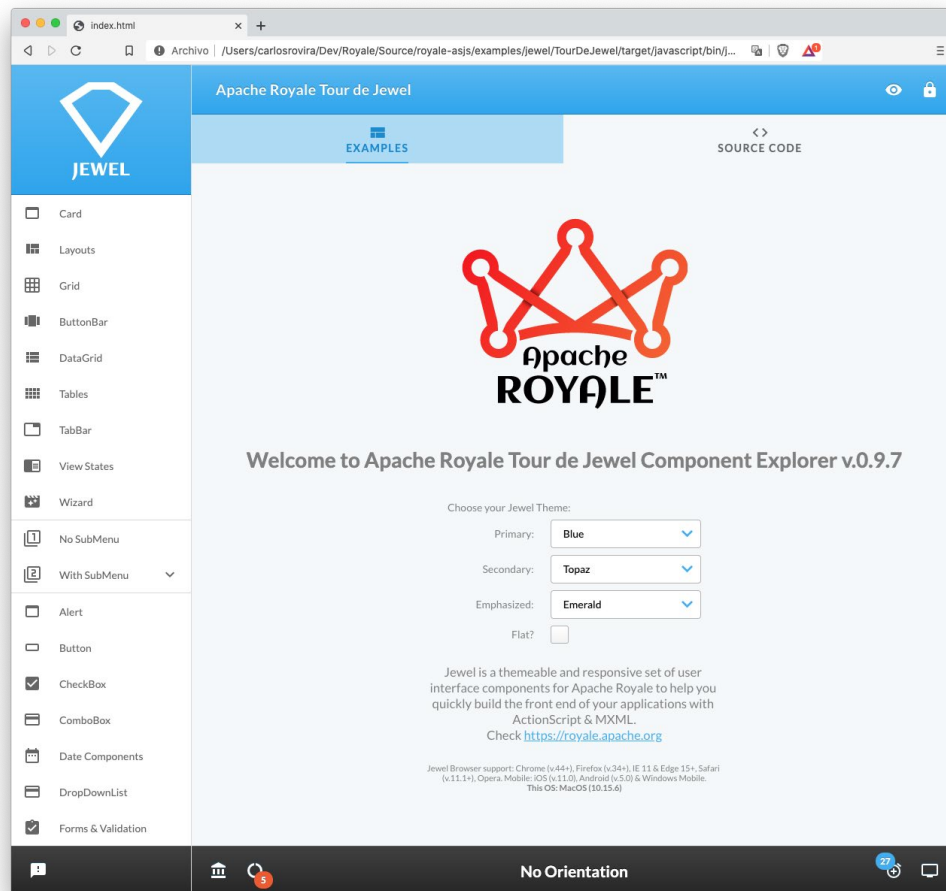


Journey

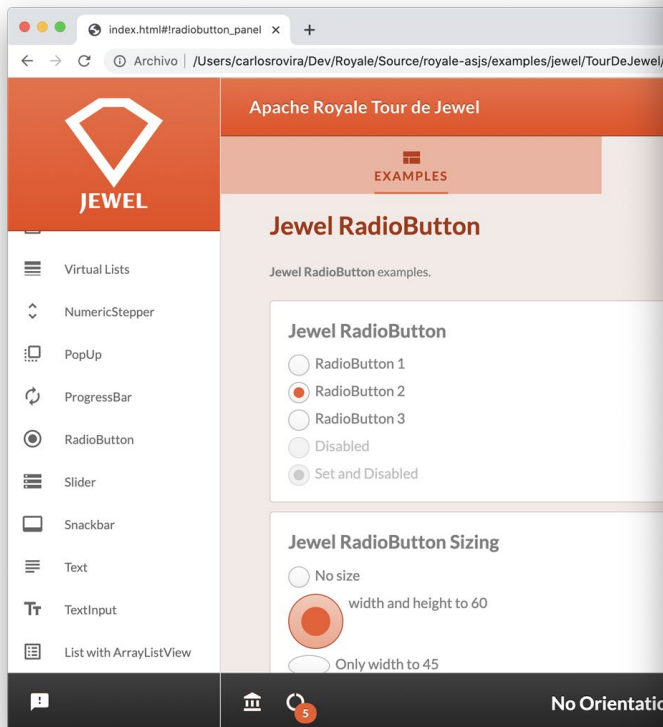


Jewel

Know what is Jewel, the UI Set behind Tour de Jewel



The screenshot shows a web browser displaying the Apache Royale Tour de Jewel Component Explorer v0.9.7. The browser's address bar shows the file path: `/Users/carlosrovira/Dev/Royale/Source/royale-asjs/examples/jewel/TourDeJewel/target/javascript/bin/j...`. The application has a blue header with the "JEWEL" logo on the left and "Apache Royale Tour de Jewel" on the right. Below the header, there are tabs for "EXAMPLES" (selected) and "SOURCE CODE". A left sidebar contains a list of component categories: Card, Layouts, Grid, ButtonBar, DataGrid, Tables, TabBar, View States, Wizard, No SubMenu, With SubMenu (expanded), Alert, Button, CheckBox, ComboBox, Date Components, DropDownList, and Forms & Validation. The main content area features the Apache Royale logo, the title "Welcome to Apache Royale Tour de Jewel Component Explorer v0.9.7", and a section titled "Choose your Jewel Theme:". This section includes three dropdown menus: "Primary:" set to "Blue", "Secondary:" set to "Topaz", and "Emphasized:" set to "Emerald". There is also a "Flat?" checkbox which is unchecked. Below the theme selection, a paragraph states: "Jewel is a themeable and responsive set of user interface components for Apache Royale to help you quickly build the front end of your applications with ActionScript & MXML. Check <https://royale.apache.org>". At the bottom, a footer provides browser support information: "Jewel Browser support: Chrome (v44+), Firefox (v34+), IE 11 & Edge 15+, Safari (v11.1+), Opera Mobile: KOS (v11.0), Android (v5.0) & Windows Mobile. Thin OS: MacOS (10.15.6)". The bottom status bar shows "No Orientation" and a battery icon.



index.html#tradiobutton_panel x +

Apache Royale Tour de Jewel

JEWEL

EXAMPLES

Jewel RadioButton

Jewel RadioButton examples.

Jewel RadioButton

- ☐ RadioButton 1
- ☒ RadioButton 2
- ☐ RadioButton 3
- ☐ Disabled
- ☐ Set and Disabled

Jewel RadioButton Sizing

- ☐ No size
- ☒ width and height to 60
- ☐ Only width to 45

Virtual Lists

NumericStepper

PopUp

ProgressBar

RadioButton

Slider

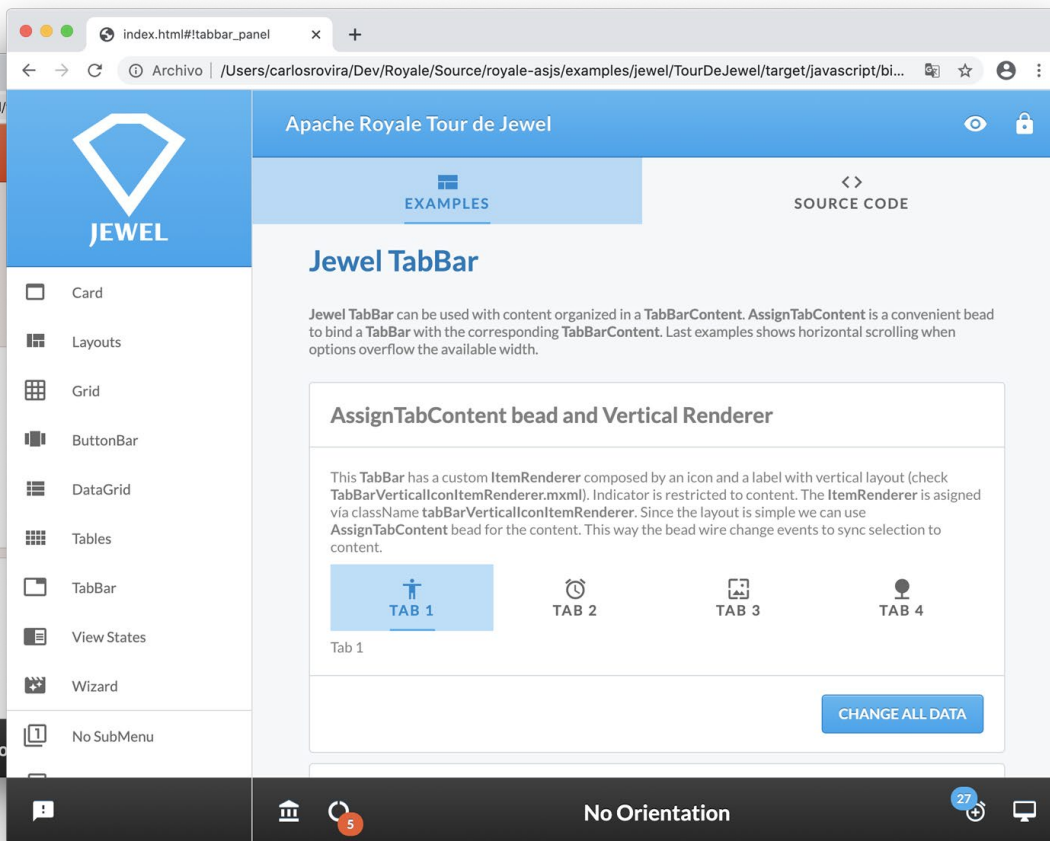
Snackbar

Text

TextInput

List with ArrayListView

No Orientation



index.html#tabbar_panel x +

Apache Royale Tour de Jewel

JEWEL

EXAMPLES

SOURCE CODE

Jewel TabBar

Jewel TabBar can be used with content organized in a `TabBarContent`. `AssignTabContent` is a convenient bead to bind a `TabBar` with the corresponding `TabBarContent`. Last examples shows horizontal scrolling when options overflow the available width.

AssignTabContent bead and Vertical Renderer

This `TabBar` has a custom `ItemRenderer` composed by an icon and a label with vertical layout (check `TabBarVerticalContentRenderer.mxml`). Indicator is restricted to content. The `ItemRenderer` is assigned via `className tabBarVerticalContentRenderer`. Since the layout is simple we can use `AssignTabContent` bead for the content. This way the bead wire change events to sync selection to content.

TAB 1 TAB 2 TAB 3 TAB 4

Tab 1

CHANGE ALL DATA

Card

Layouts

Grid

ButtonBar

DataGrid

Tables


TabBar

View States

Wizard

No SubMenu

No Orientation


JEWEL

Apache Royale Tour de Jewel

EXAMPLES

Jewel DateField & DateChooser


Jewel DateField & DateChooser examples.

Jewel DateChooser (Default)

OCT 2020						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Jewel DateField

L	M
5	6
12	13
19	20
26	27


JEWEL

Apache Royale Tour de Jewel

EXAMPLES

SOURCE CODE

Jewel TextInput

Jewel TextInput examples.

Jewel TextInput

CHECK IT!

A TextInput

With prompt...

CHANGE PROMPT

Disabled with text...

Disabled with prompt...

HSlider resize...

A label to check layout

textInput.width='156'

BEAD FOR TEXT UPPE!

TextInput Sizes

Size xsmall

Size small

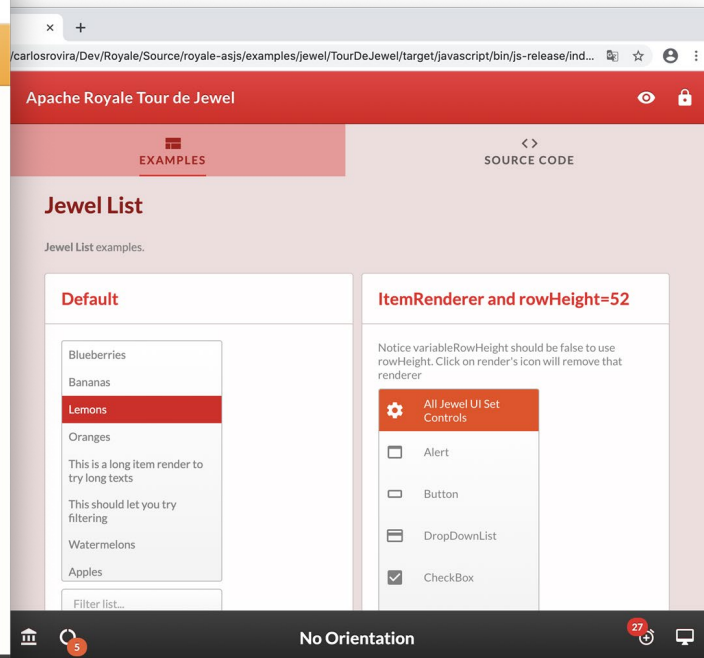
Size normal

Size large

Size xlarge


- Virtual Lists
- NumericStepper
- PopUp
- ProgressBar
- RadioButton
- Slider
- Snackbar
- Text
- TextInput
- List with ArrayListView

No Orientation



index.html#form_validation_p...

Archivo | /Users/carlosrovira/Dev/Royale/Source/royale-asjs/examples/jewel/TourDeJewel/target/javascript/bin/js-release/ind...


JEWEL

Apache Royale Tour de Jewel

EXAMPLES


Jewel Forms & Validation


Jewel Forms & Validation examples.

Jewel Form And Validators

Form with validation example

Your name *

Your born date * 

Your Favorite Avenger * 

☐ The Godfather

☐ Scent of a woman


☐ Star Wars

☐ The Shawshank Redemption

Favorite movies (at least 2) *


Verification code: * 08589

- Button
- ☒ CheckBox
- ComboBox
- Date Components
- DropDownList
- ☒ Forms & Validation
- Icon
- Images
- Label
- List
- Virtual Lists


No Orientation

index.html#card_panel

Archivo | file:///Users/carlosrovira/Dev/Royale/Source/royale-asjs/examples/jewel/TourDeJewel/target/javascript/bin/js-release/


JEWEL

Apache Royale Tour de Jewel

EXAMPLES

Jewel Card

This page shows how to use **Jewel Card** alone or with other companion components like **CardHeader**, **CardPrimaryContent** and **CardActions**. You can see how content is aligned in each section and how to add variations in order to get several configurations. Special attention to different button styles and text colors.

Notice that unlike other pages, this one doesn't use **ItemsVerticalAlign** set to "itemsSameHeight" in order to show how Card height works depending on content.

Jewel Simple Card

Apache Royale Card simplest version uses just a Card component that layout content elements vertically with some default gap between them. It applies some default padding. Card width is 100%.

Jewel Simple Card


Below this text the element is an HGroup with **ItemsExpand** that makes internal content expand all available space.

Jewel Card

This Card use **CardPrimaryContent** to add the main

This Card with **CardPrimaryContent** and **CardActions** align action buttons to the right with **ItemsHorizontalAlign** property set to "itemsRight". This label is colored with "primary-normal" color.

- ☐ Card
- ☒ Layouts
- Grid
- ButtonBar
- DataGrid
- Tables
- TabBar
- View States
- Wizard
- No SubMenu


No Orientation

Why Royale?

First why choose Royale instead other mainstream tech?

- **Supported by Apache.** No Big Corporations driving decisions for its interest.
- **Best of Flex and web dev.** Brings the best of latest 20 years in web development.
- **Multi-Target.** Right now: JS,SWF. In the future: WASM? iOS? Android?.
- **Compiled / Transpiled Apps.** Processing of source code
- **Powerful features (not available any other technology):**
 - **MXML.** Declarative language is the real power behind Royale.
 - **AMF** protocol. Best performance for RPC calls (over JSON). **RemoteObject** to interact with Java, .NET, PHP, Python,...
 - **Crux + Metadata.** **IoC**(Inversion of Control), **DI** (Dependency Injection).
 - **Data Binding.** Change model values and update views and use in conjunction with MXML
 - **Modules.** Separate your app in pieces and load independently.
 - **Performance:** Is amazing if we compared with other current mainstream tech (i.e React)
 - (*) See comparsion image for the same component Royale vs React.
- **OOP, Strongly typing, Error Checking...**
- **Composition:** PAYG & Strand & Beads.

DevTools -
unhurdle.github.io/spectrum-royale/

Range: 2.32 s – 3.56 s



35 ms Loading
138 ms Scripting
12 ms Rendering
2 ms Painting
120 ms System
929 ms Idle
1238 ms Total

DevTools -
opensource.adobe.com/spectrum-css/components/accordion/

Range: 2.55 s – 4.36 s



24 ms Loading
646 ms Scripting
326 ms Rendering
30 ms Painting
187 ms System
599 ms Idle
1813 ms Total



Royale UI Sets

- Different UI Sets
 - Basic
 - Express
 - Jewel
 - Emulation (for Flex MX/Spark)

So...



Why Jewel?

- Modern browser web development
 - Modern components (i.e: Drawer, Grid, Card...)
 - Support Browser **Responsiveness**
- Style customization
 - **Themeable** (libraries to change all UI component styles at once)
 - Set of predefined themes (144 different combinations)
- Components adapt to different screens (i.e: DateField)





Jewel Considerations

- Get the best of what we had with Flex and modern web development
 - **Best of Flex** (i.e: MXML, RemoteObject and AMF,...)
 - **Best of Browsers** (i.e: CSS, No plugins,...)
- **PAYG**
 - But with a bit of balance to keep in mind real world needs (Jewel is a bit more permissive than Basic).
- **Styling**
 - Promote **SASS** over CSS (but don't enforce for end users)



Browser Support

	Browser	Minimum Version	Release Date
Desktop	Google Chrome	44+	July, 21, 2015
	Mozilla Firefox	34+	Dec, 1, 2014
	Apple Safari	11.1+(High Sierra)	Feb, 22 2018
	Microsoft Edge	15+	
	Microsoft IE	11+	
	Opera		
Mobile	iOS SafariMobile	11.0+	Sep, 19, 2017
	Android	5.0	2014
	Windows Mobile		

<https://apache.github.io/royale-docs/component-sets/jewel#browser-support>

Basic vs Jewel

- **Basic is:**
 - Foundation of Royale components and beads.
 - 100% PAYG.
 - The most basic representation of components.
 - No additional structure considerations for visuals.
 - Visuals are very raw (no fonts, margins, colors,...mostly black/white)
 - SWF and JS considered.
- **Jewel is:**
 - Build on top of Basic Foundation.
 - PAYG (but not 100%).
 - Advanced representation of components.
 - Internal structures to build good visuals.
 - Separation of styles between structure (jewel.swc) and styling (themes.swc).
 - Only JS considered (for now, but could be extended in the future).

Components

Jewel set of available components and containers ready to use





Components

- **Containers and Components**
 - **Containers** can hold an arbitrary set of other containers and/or components. Can be populated via code or data.
 - **Components** has a concrete functionality
 - **60+** containers and components and growing.
 - For a full list see:
 - <https://apache.github.io/royale-docs/component-sets/jewel#components>
- **Beads**
 - **Common** (i.e: **Disabled**) and **Specialized** (PasswordInput for TextInput)



Containers (Code Based Population)

- View / ResponsiveView
- Group / Container
 - HGroup, VGroup / HContainer, VContainer
- Card
- Grid & GridCell
- SectionContent / ScrollableSectionContent /
TabBarContent
- Form / FormItem
- Drawer
- Module
- Others (Wizard, TopAppBar, ...)

Containers (Code Based Population)

```
<j:Container width="100%">
  <j:beads>
    <j:TileHorizontalLayout localId="thl" verticalGap="6"
      horizontalGap="6" requestedColumnCount="3"/>
  </j:beads>

  <html:Div className="box" text="1"/>
  <html:Div className="box" text="2"/>
  <html:Div className="box" text="3"/>
  <html:Div className="box" text="4"/>
  <html:Div className="box" text="5"/>
  <html:Div className="box" text="6"/>
  <html:Div className="box" text="7"/>
  <html:Div className="box" text="8"/>
  <html:Div className="box" text="9"/>
  <html:Div className="box" text="10"/>
  <html:Div className="box" text="11"/>
  <html:Div className="box" text="12"/>
</j:Container>
```



Containers (Data Based Population)

- DataContainer
- List
 - TabBar
 - ButtonBar, IconButtonBar,...
 - Navigator
- ComboBox
- DropDownList
- Table, SimpleTable
- DataGrid

Considerations:

- Data populate through **dataProvider**.
 - **ArrayList**
 - **ArrayListView** (for sort and filtering)
- **ItemRenderer** (data item visualization)

Containers (Data Based Population)

```
<j:DataContainer width="100%" height="250"
  itemRenderer="itemRenderers.VerticalIconListItemRenderer">
  <j:beads>
    <j:ScrollingViewport/>
    <j:TileHorizontalLayout localId="thll" verticalGap="6"
      horizontalGap="6" columnWidth="130" rowHeight="70"/>
  </j:beads>

  <js:ArrayList>
    <fx:Array>
      <vos:IconListVO label="Ank" icon="{FontAwesome5IconType.ANKH}"/>
      <vos:IconListVO label="Atom" icon="{FontAwesome5IconType.ATOM}"/>
      <vos:IconListVO label="Burn" icon="{FontAwesome5IconType.BURN}"/>
      <vos:IconListVO label="Candy Cane" icon="{FontAwesome5IconType.CANDY_CANE}"/>
      <vos:IconListVO label="Fire" icon="{FontAwesome5IconType.FIRE_ALT}"/>
      <vos:IconListVO label="Duck" icon="{FontAwesome5IconType.DUCK}"/>
      <vos:IconListVO label="Cloud And Moon" icon="{FontAwesome5IconType.CLOUD_MOON}"/>
      <vos:IconListVO label="Europe" icon="{FontAwesome5IconType.GLOBE_EUROPE}"/>
      <vos:IconListVO label="Electric Guitar" icon="{FontAwesome5IconType.GUITAR_ELECTRIC}"/>
      <vos:IconListVO label="Mask" icon="{FontAwesome5IconType.MASK}"/>
      <vos:IconListVO label="Skull" icon="{FontAwesome5IconType.SKULL}"/>
      <vos:IconListVO label="Spider" icon="{FontAwesome5IconType.SPIDER}"/>
    </fx:Array>
  </js:ArrayList>
</j:DataContainer>
```



Components

- Button, IconButton, ToggleButton
- TextInput, IconTextInput, TextArea
- CheckBox, RadioButton
- DateField, DateChooser
- Image, Icon
- Label
- NumericStepper
- Slider
- Others (PopUp, SimpleLoader,...)
- Based on **StyledUIBase** (that extends **Basic UIBase**)
- Composed by Beads (Model, View, Controller, Layout,...)
- Beads are configured in CSS

Components

```
<j:Card visible="true" visible.loggedIn="false">
  <html:H1 text="View States with dot notation"/>
  <j:TextInput text="someuser"/>
  <j:TextInput text="somepass">
    <j:beads>
      <j:PasswordInput/>
    </j:beads>
  </j:TextInput>
  <j:Button text="Login" emphasis="primary" click="currentState = 'loggedIn'"/>
</j:Card>
```



Beads

- **Common**
 - Disabled
 - SizeControl
 - ToolTip
 - ResponsivSize
- **Specialized**
 - TabBar -> AssignTabContent
 - TextInput and TextArea -> TextPrompt

Default Beads

- Defined through CSS (SASS):
- i.e: List is composed of the following beads:

```
j|List
  IBeadView: ClassReference("org.apache.royale.jewel.beads.views.ListView")
  IBeadController: ClassReference("org.apache.royale.jewel.beads.controllers.ListSingleSelectionMouseController")
  IBeadKeyController: ClassReference("org.apache.royale.jewel.beads.controllers.ListKeyDownController")
  IItemRendererClassFactory: ClassReference("org.apache.royale.core.SelectableItemRendererClassFactory")
  IItemRenderer: ClassReference("org.apache.royale.jewel.itemRenderers.ListItemRenderer")
  IItemRendererInitializer: ClassReference("org.apache.royale.jewel.beads.itemRenderers.ListItemRendererInitializer")
  ISelectableItemRenderer: ClassReference("org.apache.royale.jewel.beads.itemRenderers.ClassSelectorListSelectableItemRendererBead")
  IViewport: ClassReference("org.apache.royale.jewel.supportClasses.scrollbar.ScrollingViewport")
  IBeadModel: ClassReference("org.apache.royale.jewel.beads.models.ArrayListSelectionModel")
  IDataProviderItemRendererMapper: ClassReference("org.apache.royale.html.beads.SelectionDataItemRendererFactoryForCollectionView")
```


Layouts

How Jewel components are displayed in screen

Layout Beads

- Based on CSS “flex”
- Tries to delegate layout to the browser when possible
- BasicLayout is for absolute positioning
- HorizontalLayout / VerticalLayout
- HorizontalCenteredLayout / VerticalCenteredLayout
- much more...

BasicLayout

Jewel BasicLayout (width:200, height:200px)




Diagram illustrating BasicLayout (width:200, height:200px). The layout shows four beads (X, Y, Z, O) positioned relative to the origin (0,0). X is at (30,30), Y is at (60,60), and Z is at (60,60). The origin is marked as ORIGIN. The bottom/right boundary is marked as BOTTOM/RIGHT = 0.

HorizontalLayout




Diagram illustrating HorizontalLayout. The layout shows four beads (X, Y, Z, O) arranged horizontally. A slider is present below the diagram.

VerticalLayout




Diagram illustrating VerticalLayout. The layout shows four beads (X, Y, Z, O) arranged vertically. A slider is present below the diagram.

HorizontalCenteredLayout




Diagram illustrating HorizontalCenteredLayout. The layout shows four beads (X, Y, Z, O) arranged horizontally and centered. A slider is present below the diagram.

VerticalCenteredLayout






Diagram illustrating VerticalCenteredLayout. The layout shows four beads (X, Y, Z, O) arranged vertically and centered. A slider is present below the diagram.

Layouts Beads


- TileHorizontalLayout uses a combination of css flex and AS3


Using TileHorizontalLayout


 **TILE CONTAINER**


 **TILE LIST**


1	2	3	4
5	6	7	8
9	10	11	12

columnWidth


horizontalGap


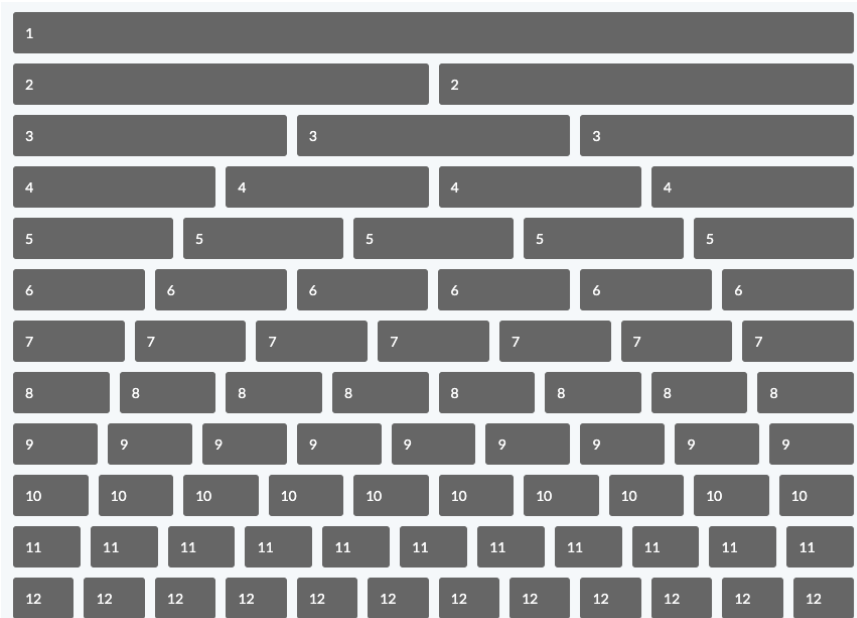
requestedColumnCount


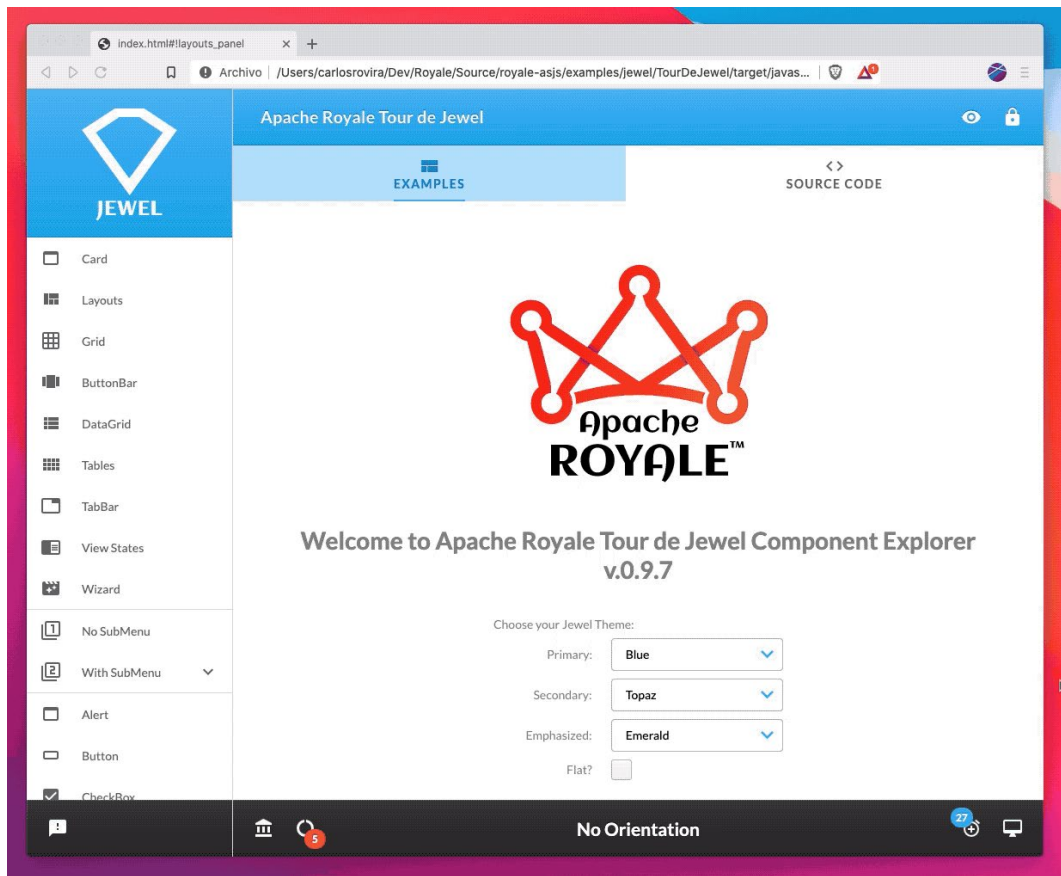
rowHeight


verticalGap


Responsive Layout

- Grid / GridCell
 - GridLayout / GridCellLayout
- Sizes can adapt to screen width
 - Fonts size change between mobile / tablet / desktop
- ResponsiveView (instead of View)
- Beads:
 - ResponsiveVisibility
 - ResponsiveSize
 - ...



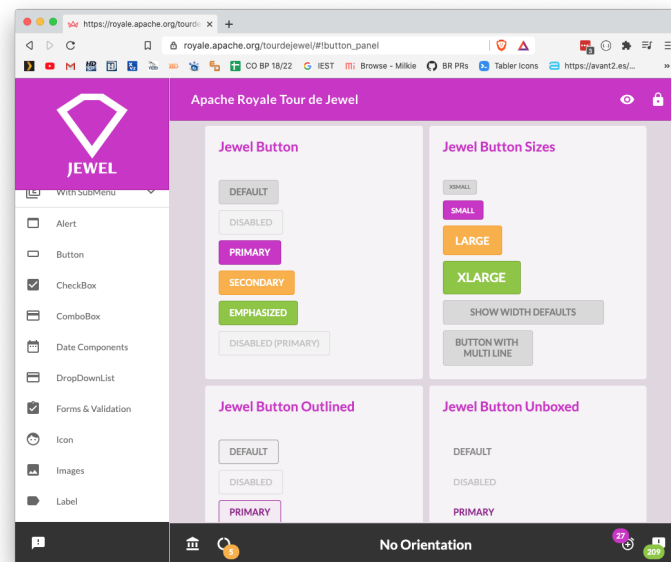
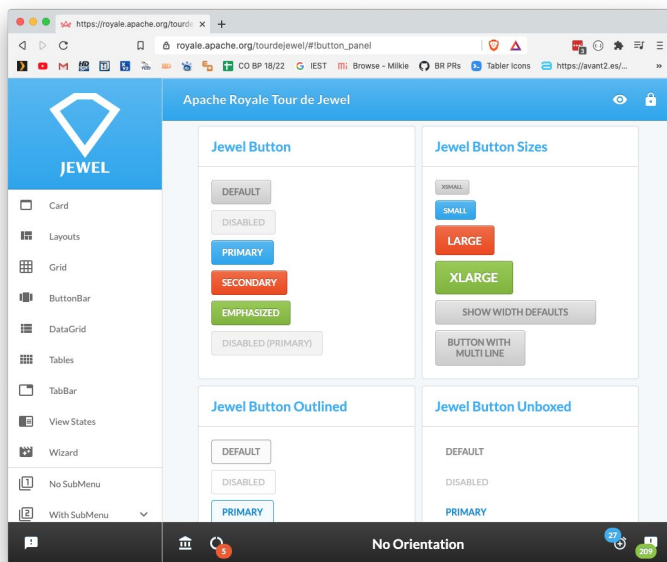


The screenshot shows a web browser displaying the Apache Royale Tour de Jewel Component Explorer. The browser's address bar shows the URL: `index.html#layouts_panel`. The page has a blue header with the Apache Royale logo and the text "Apache Royale Tour de Jewel". Below the header, there are two tabs: "EXAMPLES" (selected) and "SOURCE CODE". The main content area features the Apache Royale logo and the text "Welcome to Apache Royale Tour de Jewel Component Explorer v.0.9.7". Below this, there is a section titled "Choose your Jewel Theme:" with three dropdown menus: "Primary" (set to "Blue"), "Secondary" (set to "Topaz"), and "Emphasized" (set to "Emerald"). There is also a "Flat?" checkbox which is currently unchecked. On the left side of the page, there is a sidebar with a list of components: Card, Layouts, Grid, ButtonBar, DataGrid, Tables, TabBar, View States, Wizard, No SubMenu, With SubMenu (expanded), Alert, Button, and CheckBox. The bottom of the page has a dark blue footer with the text "No Orientation" and a small icon of a person.

Themes

The look & feel of Jewel

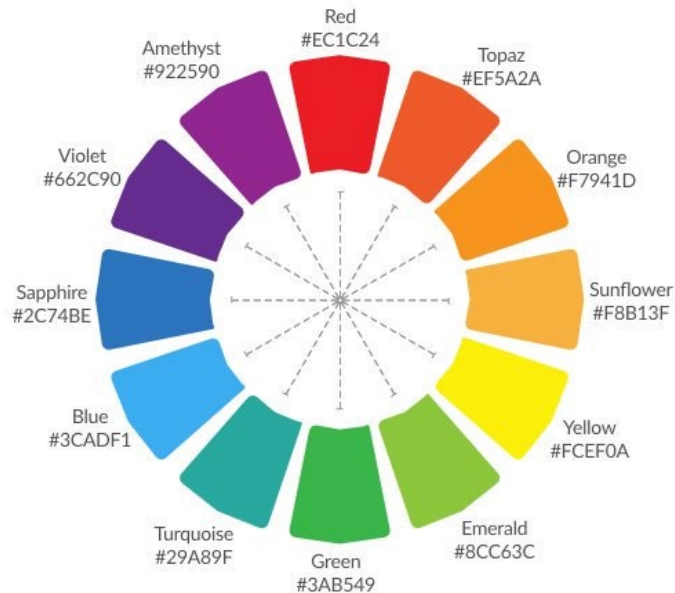
Themes



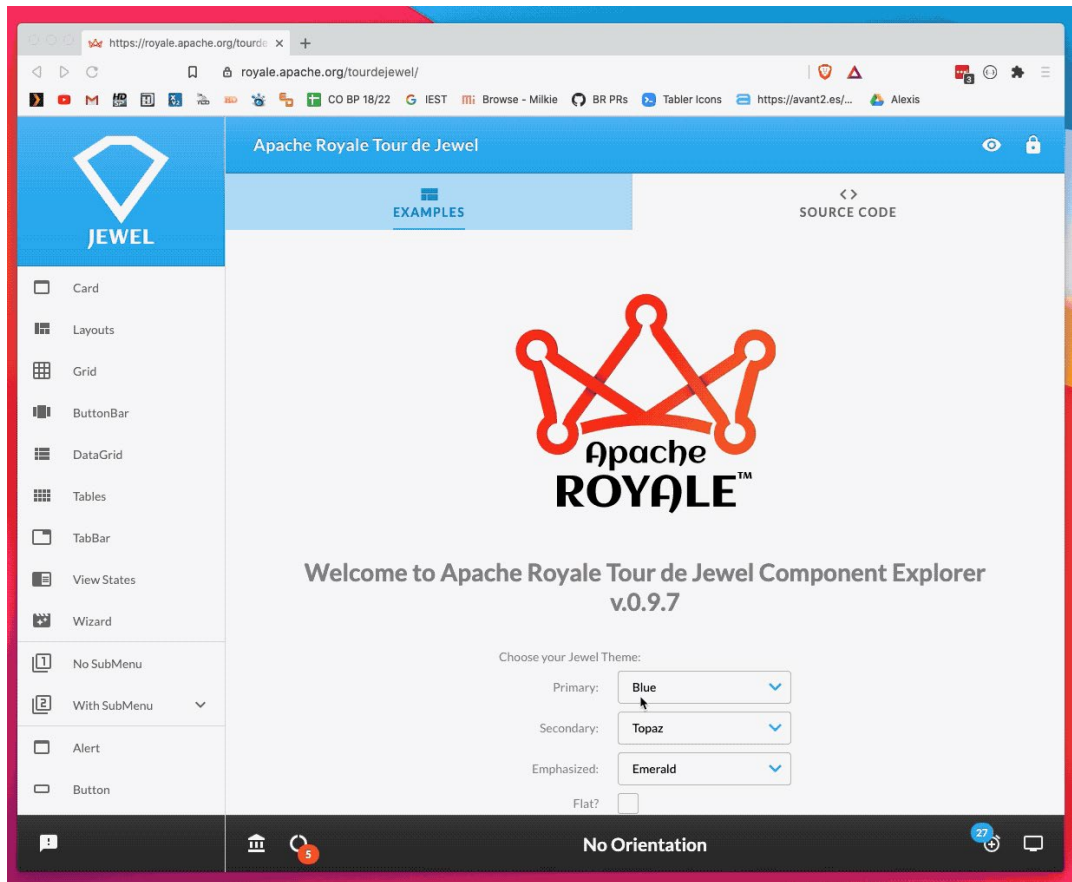
About themes

- **JewelTheme** is the master theme
- Use **SASS**
- Light/Dark and Flat/No Flat variations
- 12 Colors
- primary, secondary and emphasized
- 144 variations
- We can use
 - Jewel Theme (if you can compile SASS)
 - A combination of three themes (primary, secondary and emphasized)
- Dark themes still WIP

Apache Royale Jewel 12 Color Wheel



Themes Runtime Switching





Jewel Links

- Tour de Jewel
 - <https://royale.apache.org/tourdejewel>
- Jewel Blog Examples
 - <https://royale.apache.org/category/jewel-ui-set/>
- Jewel Documentation
 - <https://apache.github.io/royale-docs/component-sets/jewel>

Future

A look into the Jewel future



Jewel's Future

- Continue improving the UI Set
 - Continue improving actual components
 - Provide missing functionality
 - Bug fixing
 - Develop new features
 - Components still missing
 - new beads
 - Dark Themes

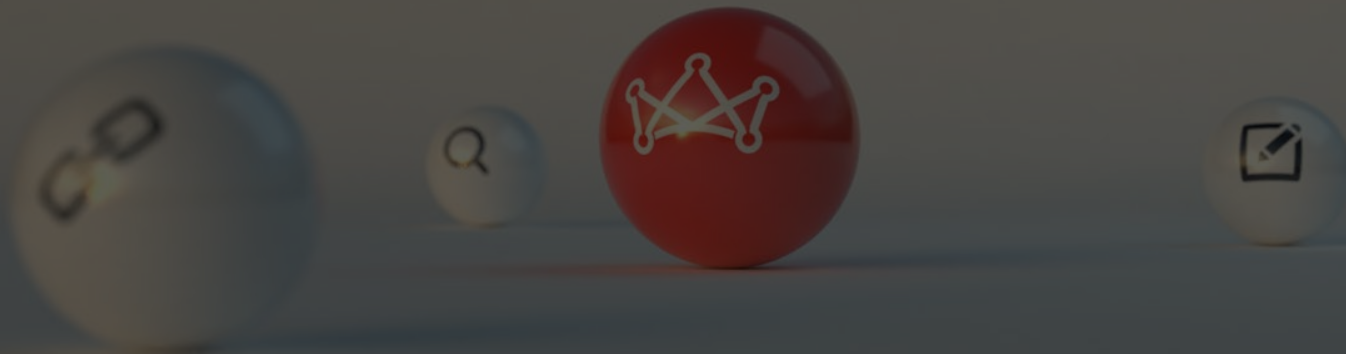


Where to go from here

- Join our mailing list (“users” or “dev”)
 - <https://royale.apache.org/mailling-lists/>
- Do you want to join us?
 - <https://royale.apache.org/get-involved/>
- StackOverFlow
 - <https://stackoverflow.com/questions/tagged/apache-royale>
- If you care this project -> Follow us on Social Networks!! :)
 - **Twitter:** <https://twitter.com/apacheroyale>
 - **LinkedIn:** <https://www.linkedin.com/groups/12118437>
 - **FaceBook:** <https://www.facebook.com/ApacheRoyaleSDK>

Thank you for your attention!

Questions?



royale.apache.org