HD3C12 – Grid Control

|  |  |
| --- | --- |
| **Product and Focus**  HANA Platform/SAPUI5 | **MOTIVATION**  The Grid control is one of the more useful for creating responsive layouts in SAPUI5 applications.  **PREREQUISITES**  HD3C06 – The Base Application |
| **Target Audience**  Undergrduate/Graduate Beginner to Intermediate |
| **Author**  Ross Hightower |
| https://bgoerke.files.wordpress.com/2013/05/section1.png | |

# The Grid Control

The Grid layout control organizes its content into 12 columns. Because the number of columns is configurable based on screen size, the Grid layout is ideal for designing responsive layouts that work on multiple screen sizes.

The template for the Grid control in XML is shown in the box below.

|  |
| --- |
| <Grid xmlns=*"sap.ui.layout"*  busy=*"false"*  busyIndicatorDelay=*"1000"*  visible=*"true"*  width=*"100%"*  vSpacing=*"1"*  hSpacing=*"1"*  position=*"Left"*  defaultSpan=*"L3 M6 S12"*  defaultIndent=*"L0 M0 S0"*  containerQuery=*"false"*>  <tooltip></tooltip> <!-- sap.ui.core.TooltipBase -->  <dependents></dependents> <!-- sap.ui.core.Control, since 1.19 -->  <content></content> <!-- sap.ui.core.Control -->  </Grid> |

The **defaultSpan** property sets the grid’s default span values for large, medium and small screens. The integers represent the number of columns and can take values from 1 to 12. The order must be L, M then S. The default value of L3 M6 S12 means the grid will take one-quarter the screen for large screens (3/12), one-half of the screen for medium screens (6/12) and the entire screen on small screens (12/12).

The **defaultIndent** attribute defines the number of empty columns before the grid content so it can be used to offset the grid from the left side of the screen.

Some of the other attributes are described in the table below.

|  |  |  |
| --- | --- | --- |
| Attribute | Description | Usage |
| vSpacing and hSpacing | The spacing between rows (columns) in rem | Allows values are 0, .5, 1 and 2 |
| position | Position of the grid in its container | Left, Center and Right |

Layout data can be used to configure the layout behavior of controls within the grid. Layout data is applied like the example shown below. The data is applied within the element tags to which it is embedded. For example, in the listing below the GridData attributes apply to the Image control.

|  |
| --- |
| <Grid defaultSpan="L6 M6 S10">  <content>  <Image src="{people>0/ContactPicUrl}" width="100%">  <layoutData>  <l:GridData span="L3 M3 S8" linebreakL="true"  linebreakM="true" linebreakS="true" />  </layoutData>  </Image>  </content>  </Grid> |

Some of the layout data options are shown in the table below.

|  |  |  |
| --- | --- | --- |
| Attribute | Description | Usage |
| span  spanL,spanM,spanS | The number of columns a control will span. | span=”L3 M6 S10” |
| Indent  indentL,indentM,indentS | The number of columns a control will be offset | Indent=”L3 M6 S10” |
| linebreak  linebreakL,linebreakM,linebreakS | If this is set to true then the control causes a linebreak and begins a new row in the grid | LinebreakS=”true” |
| width | The width taken within the grid. Default is 100%. | With=”50%” |
| visible visibleL,visibleM,visibleS | If this is set to false, the control won’t be visible for the specified screen size | visibleS=”false” |
| moveBackwards  moveForwards | Indicates the number of columns to move a cell backwards or forwards on indicated screen sizes. | moveBackwards = “L1 M2 S3” |

# Create the Grids View

### Add the View to the App Navigation

Add a new object for the Grids view to the views.json file. You can find an icon [here](https://openui5.hana.ondemand.com/iconExplorer.html). Also, add a route to the Component.js file.

### Grids.view.xml

Create a file called **Grids.view.xml** in the **view** package. Enter the code below into the file.

|  |
| --- |
| <mvc:View  controllerName="ui5.controller.Grids"  xmlns:html="http://www.w3.org/1999/xhtml"  xmlns:mvc="sap.ui.core.mvc"  xmlns="sap.m">  <Page title="Start Coding Grids!"  showNavButton = "true"  navButtonPress = "handleNavButtonPress">  <IconTabBar class="iconTabBarPaddingTop" id="idIconTabBarMulti">  <items>  <IconTabFilter icon="sap-icon://hint" text="Grid">  <Text text="Grid content goes here ..." />  </IconTabFilter>  <IconTabFilter icon="sap-icon://attachment" text="Complex Grid">  <Text text="Complex Grid content goes here ..." />  </IconTabFilter>  </items>  </IconTabBar>  </Page>  </mvc:View> |

Listing 1

This code uses the IconTabFilter control which is one of the more iconic looks for SAPUI5. There is a later case that describes this control in more detail. For this case, it is just used as a way to organize the layout content.

### Grids.controller.js

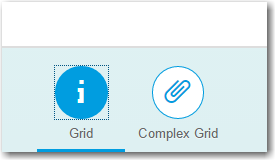
Create a file called **Grids.controller.js** in the controller package andcopy the code shown below into the file:

|  |
| --- |
| sap.ui.define([  "sap/ui/core/mvc/Controller",  "sap/m/MessageToast"  ], function(Controller) {  "use strict";  return Controller.extend("ui5.controller.Grids", {  onInit: function() {  this.router = sap.ui.core.UIComponent.getRouterFor(this);  },  handleNavButtonPress: function() {  this.router.navTo("Master", {  from: "Grids"  });  }  });  }); |

Listing 2

You should be familiar with this code by now. It handles the back navigation button press and navigates back to the Master view.

Now you can run the application. It doesn’t do much yet so let’s add some code to the tabs.



## Add Some Content

The Grid layout can be used to create grids of columns and rows. This first examples demonstrates some of the layout options.

### Grids.view.xml

Use the code below to replace the <Text> control in the first IconTabFilter.

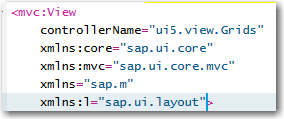
|  |
| --- |
| <l:Grid class="gridMarginTop" hSpacing="2" defaultSpan="L12 M6 S10" defaultIndent="L0 M1 S1">  <l:content>  <Text width="100%" text="Item 1" editable="false" class="item1">  </Text>  </l:content>  </l:Grid> |

Listing 3

Note some of the controls are prefixed with an l:. This indicates the namespace in which SAPUI5 can find the code. Add the reference to the namespce in the <mvc:View> tag at the top of the file.

|  |
| --- |
| xmlns:l="sap.ui.layout" |

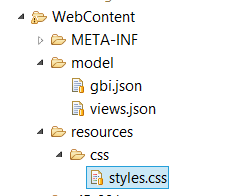
Listing 4



### styles.css

|  |
| --- |
| This step is only necessary if you did not complete the FlexBox case. |

We also need to add the following styles to the **styles.css** file. If you haven’t already created it, create a package called **css** in the project package and create a file called **styles.css**.



Add the styles shown below the styles.css file.

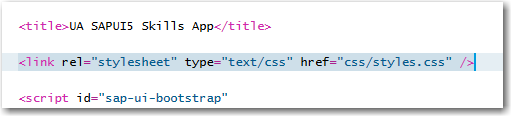
|  |
| --- |
| *.item1 {*  *padding: 1em;*  *background-color: #d1dbbd;*  *}*  *.item2 {*  *padding: 1em;*  *background-color: #7D8A2E;*  *}*  *.item3 {*  *padding: 1em;*  *background-color: #C9D787;*  *}*  *.item4 {*  *padding: 1em;*  *background-color: #F8F8FF;*  *}*  *.item5 {*  *padding: 1em;*  *background-color: #FFC0A9;*  *}*  *.item6 {*  *padding: 1em;*  *background-color: #FF8598;*  *}* |

Listing 5

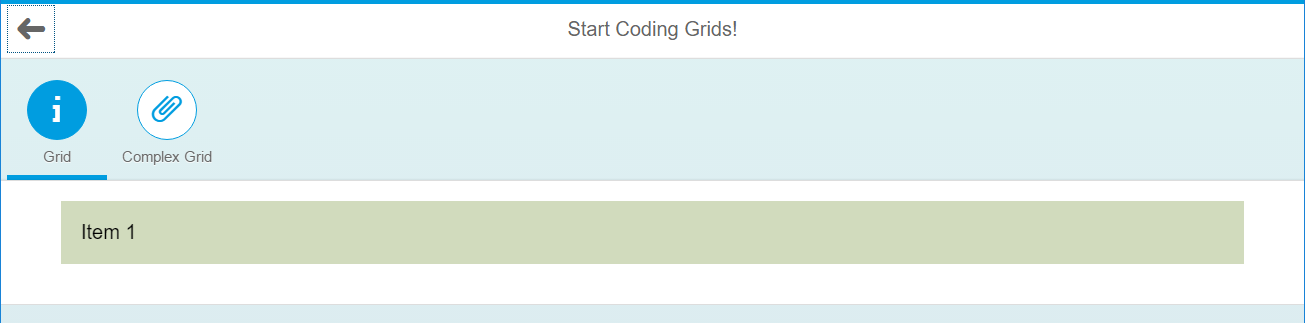
Make sure you have the reference to the styles.css file in your index.html file.

|  |
| --- |
| <link rel="stylesheet" type="text/css" href="css/styles.css" /> |

Listing 6



The result is shown below.



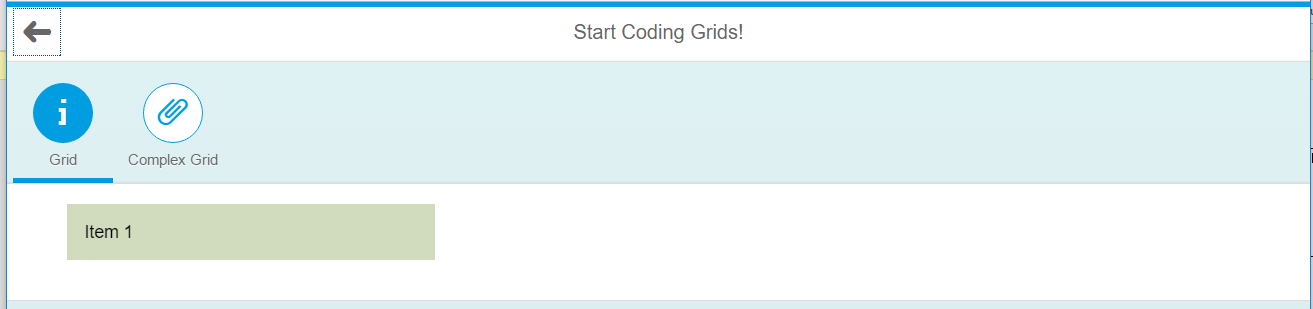
Pretty simple so far. The next few additions will add some complexity so you can see how the Grid control works. Replace the code between the <l:content> and </l:content> controls.

Step 1:

Set the width of the Text control.

|  |
| --- |
| <Text width="100%" text="Item 1" editable="false" class="item1">  <layoutData>  <l:GridData span="L4"/>  </layoutData>  </Text> |

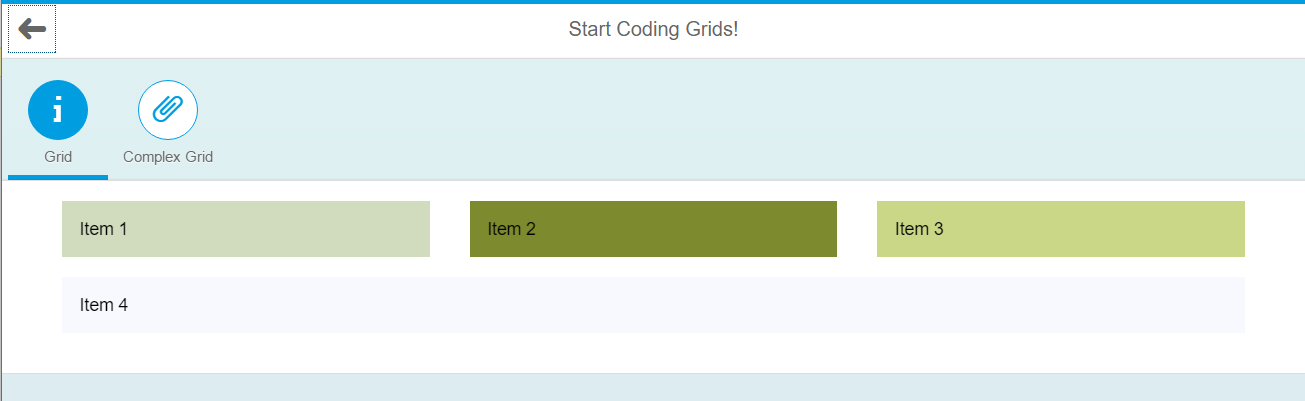
This sets the width of the Text control to one third (4/12) of the screen on large screens.



### Step 2:

Add additional Text controls.

|  |
| --- |
| <Text width="100%" text="Item 1" editable="false" class="item1">  <layoutData>  <l:GridData span="L4"/>  </layoutData>  </Text>  <Text width="100%" text="Item 2" editable="false" class="item2">  <layoutData>  <l:GridData span="L4"/>  </layoutData>  </Text>  <Text width="100%" text="Item 3" editable="false" class="item3">  <layoutData>  <l:GridData span="L4"/>  </layoutData>  </Text>  <Text width="100%" text="Item 4" editable="false" class="item4">  <layoutData>  <l:GridData span="L12"/>  </layoutData>  </Text> |

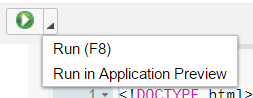


### Step 3:

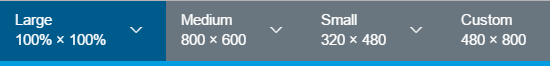
Add responsiveness.

|  |
| --- |
| <Text width="100%" text="Item 1" editable="false" class="item1">  <layoutData>  <l:GridData span="L6 M3 S12"/>  </layoutData>  </Text>  <Text width="100%" text="Item 2" editable="false" class="item2">  <layoutData>  <l:GridData span="L6 M3 S12"/>  </layoutData>  </Text>  <Text width="100%" text="Item 3" editable="false" class="item3">  <layoutData>  <l:GridData span="L8 M3 S12"/>  </layoutData>  </Text>  <Text width="100%" text="Item 4" editable="false" class="item4">  <layoutData>  <l:GridData span="L4 M3 S12"/>  </layoutData>  </Text> |

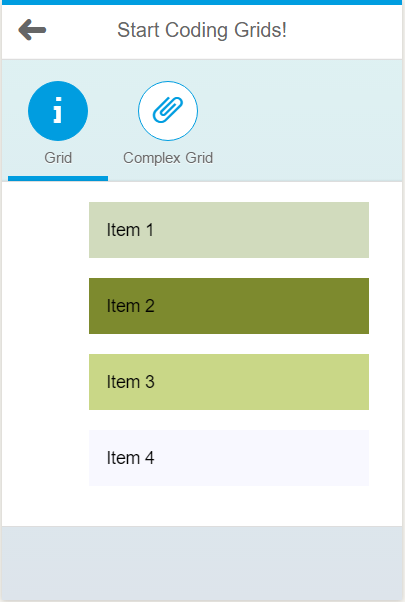
This code adds widths (spans) for large, medium and small screens. To see it work, run the application using the Application Preview using the drop down list on the Run icon.



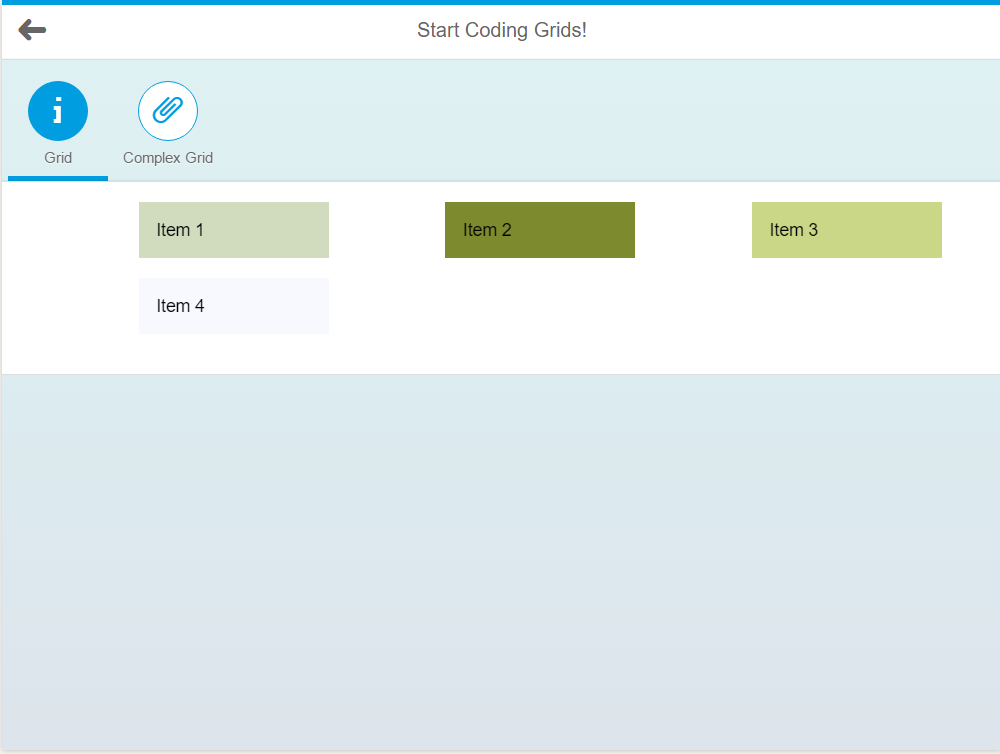
You can change the screen size using the buttons on the top left.



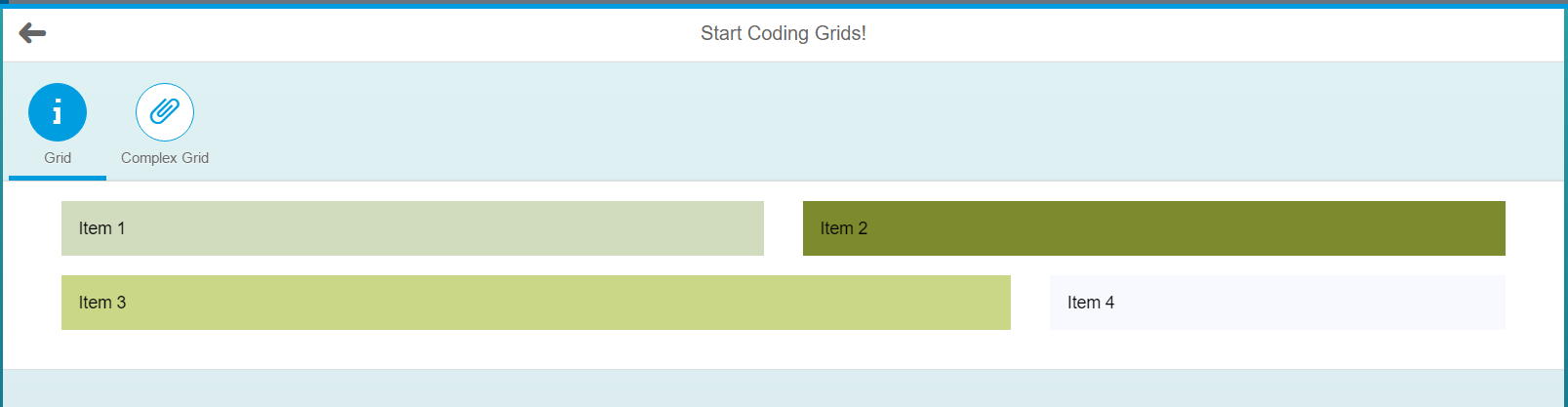
**Small (span = 12)**



**Medium (span = 3)**



**Large (span = various)**



## Add a Bound Grid

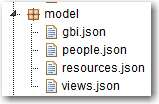
The Grid control also has an aggregation called binding to which you can bind data. In this example, you will create a data file in JSON format, then bind it to a grid control.

### people.json

Create a new file in the **model** folder called **people.json** and insert the code shown below:

|  |
| --- |
| {  "PersonCollection": [  {  "FirstName": "George",  "LastName": "Washington",  "Title": "1st U.S. President",  "ContactPicUrl": "http://upload.wikimedia.org/wikipedia/commons/2/25/George\_Washington\_as\_CIC\_of\_the\_Continental\_Army\_bust.jpg",  "Description": "George Washington was the first President of the United States, the commander-in-chief of the Continental Army during the American Revolutionary War, and one of the Founding Fathers of the United States."  },  {  "FirstName": "Alexandrina",  "LastName": "Victoria",  "Title": "Former Queen regnant",  "ContactPicUrl": "http://upload.wikimedia.org/wikipedia/commons/a/aa/Dronning\_victoria.jpg",  "Description": "Queen Victoria was the monarch of the United Kingdom of Great Britain and Ireland from 20 June 1837 until her death. From 1 May 1876, she used the additional 'Title' of Empress of India."  },  {  "FirstName": "Friedrich",  "LastName": "Der GroÃŸe",  "Title": "King of Prussia 1740-1786",  "ContactPicUrl": "http://upload.wikimedia.org/wikipedia/commons/f/fc/Frederic\_II\_de\_prusse.jpg",  "Description": "Frederick II was King in Prussia of the Hohenzollern dynasty. He is best known for his brilliance in military campaigning and organization of Prussian armies. He became known as Frederick the Great and was nicknamed Der Alte Fritz."  }  ]  } |

Listing 7



### Grids.controller.js

Insert the code shown below into the onInit function of the **Grids.controller.js** file.

|  |
| --- |
| sap.ui.define([  "sap/ui/core/mvc/Controller",  **"sap/ui/model/json/JSONModel"**  ], function(Controller, **JSONModel**) {  "use strict";  return Controller.extend("ui5.controller.Grids", {  onInit: function() {  this.router = sap.ui.core.UIComponent.getRouterFor(this);    var oModel = new JSONModel("model/people.json");  this.getView().setModel(oModel,'people');  },  handleNavButtonPress: function() {  this.router.navTo("Master", {  from: "Grids"  });  }  });  }); |

Listing 8

This code creates a model named people based on a JSON file in the model folder and assigns it to the view.

### Grids.view.xml

Replace the Text control in the second IconTabFilter section in the **Grids.view.xml** file with the code shown below:

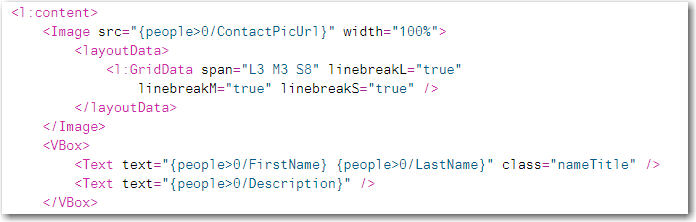
|  |
| --- |
| <l:Grid binding="{people>/PersonCollection}" class="gridMarginTop"  hSpacing="2" defaultSpan="L6 M6 S10" defaultIndent="L1 M3 S1">  <l:content>  <Image src="{people>0/ContactPicUrl}" width="100%">  <layoutData>  <l:GridData span="L3 M3 S8" linebreakL="true"  linebreakM="true" linebreakS="true" />  </layoutData>  </Image>  <VBox>  <Text text="{people>0/FirstName} {people>0/LastName}" class="nameTitle" />  <Text text="{people>0/Description}" />  </VBox>  <Image src="{people>1/ContactPicUrl}" width="100%">  <layoutData>  <l:GridData span="L3 M3 S8" linebreakL="true"  linebreakM="true" linebreakS="true" />  </layoutData>  </Image>  <VBox>  <Text text="{people>1/FirstName} {people>1/LastName}" class="nameTitle" />  <Text text="{people>1/Description}" />  </VBox>  <Image src="{people>2/ContactPicUrl}" width="100%">  <layoutData>  <l:GridData span="L3 M3 S8" linebreakL="true"  linebreakM="true" linebreakS="true" />  </layoutData>  </Image>  <VBox>  <Text text="{people>2/FirstName} {people>2/LastName}" class="nameTitle" />  <Text text="{people>2/Description}" />  </VBox>  </l:content>  </l:Grid> |

Listing 9

The first line of this code creates the Grid control and binds it to the PersonCollection array in the people model.



The first row of the Grid is defined by this code:



The Image control is bound to the ContactPicUrl property of the first object (0/ContactPicUrl) in the model. The 0 indicates the index of the object in the PersonCollection array that this attribute is bound to. Since the Grid control is bound to /PersonCollection, the url is bound to /PersonCollection/0/ContactPicUrl.

The layoutData specifies that the image should take up 3, 3 and 8 columns on large, medium and small screens respectively. The linebreak attributes indicate that the control will cause a linebreak on large, medium and small screens and the image control will be at the beginning of the next line. This insures that the image will appear above the accompanying text when the grid must adjust for screen size.

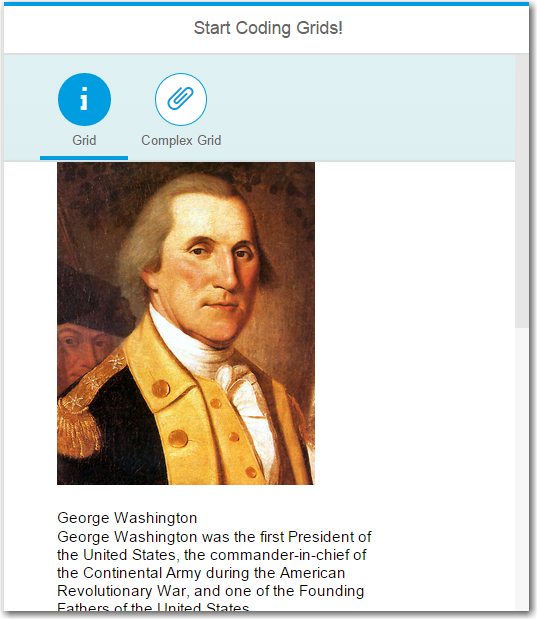
When you refresh the application and in the last IconTab the screen should look like this:



In the editor, select the index.html file then use the drop-down list on the execute button to select **Run in Application Preview.** This gives you the option to view the application on different resolutions. On a tablet the screen looks like this:



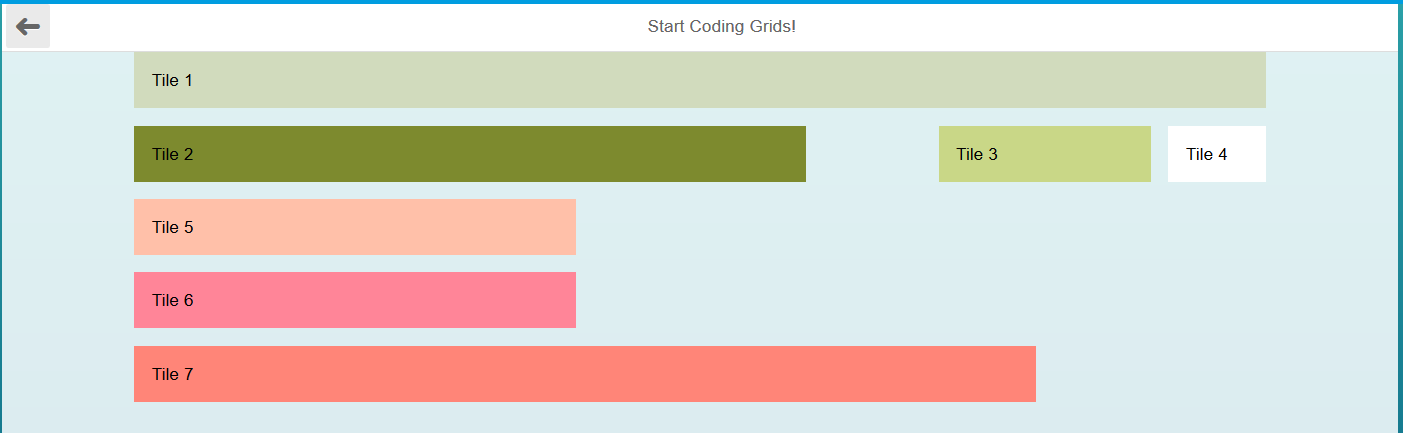
On a phone the image takes up most of the space so the text is wrapped to the next line.



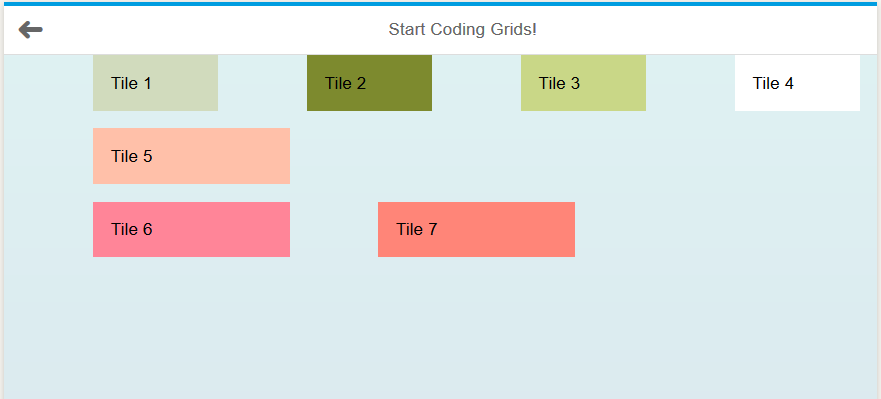
# Exercise

Add a view that implements a layout like the image below.

### Large Screens



### Medium Screens



### Small Screens

