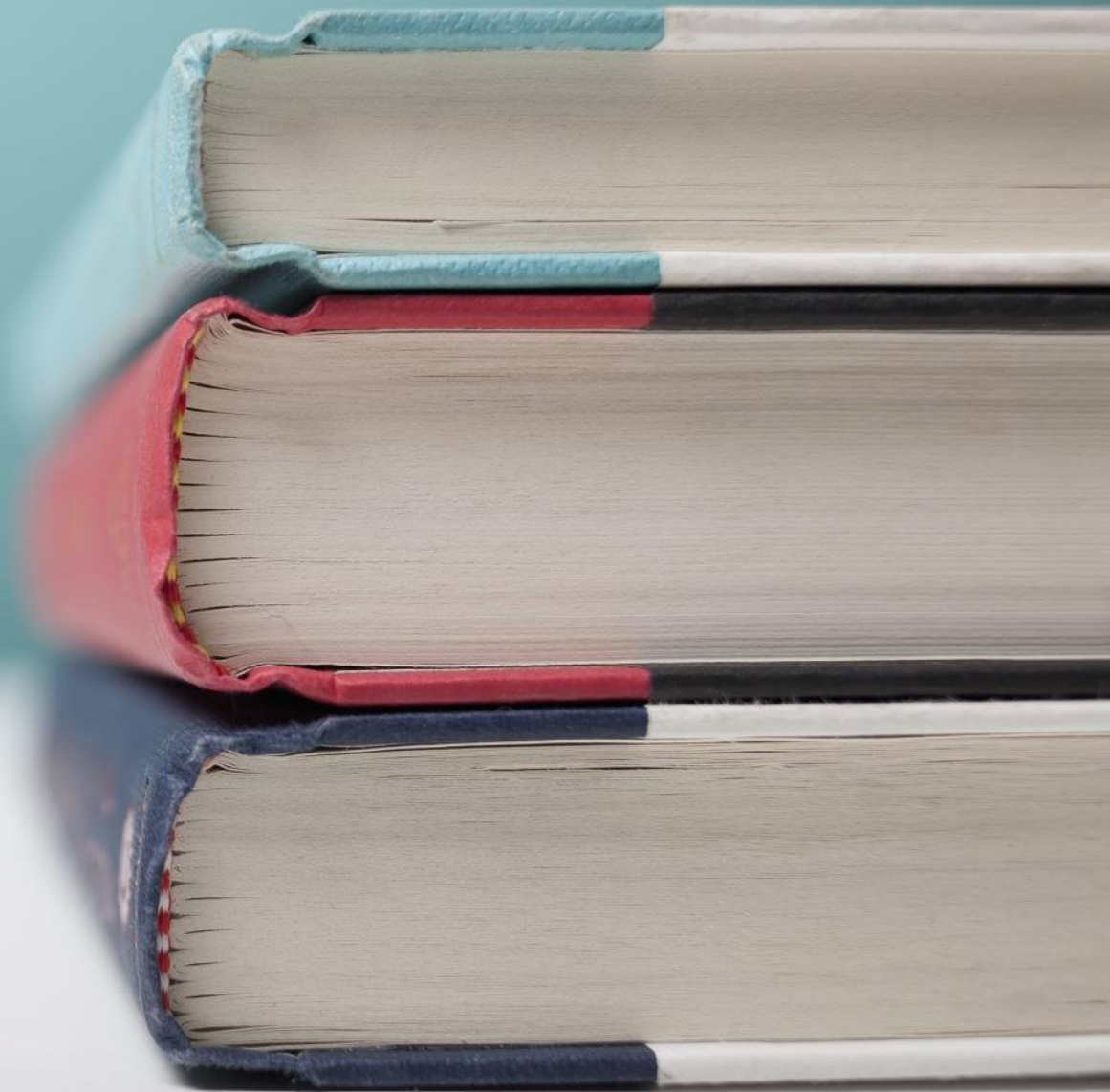


KỸ THUẬT WEB VỚI  
ỨNG DỤNG DI ĐỘNG ĐA  
NỀN TẢNG  
Bài 3: Tiện Ích





## III NỘI DUNG CHÍNH

I

V- TEMPLATE

II

V - VIEW

III

THÀNH PHẦN LAYOUTS

## 1. V-TEMPLATE

**<v-template>** là thành phần cho phép bạn đánh dấu việc định nghĩa cái mà bạn có thể sử dụng lại template.

### Props

Tên	Kiểu	Sự miêu tả
<code>if</code>	<code>String</code>	Điều kiện để sử dụng mẫu này.
<code>name</code>	<code>String</code>	Tên của mẫu, được tạo tự động nếu bị bỏ qua.

## 1. V-TEMPLATE

### Cách sử dụng cơ bản:

Thành phần **<v-template>** được sử dụng nội bộ bởi thành phần ListView để lặp qua các mục danh sách của nó.

### Cách sử dụng nâng cao:

Có thể sử dụng thành phần **v-template** để triển khai việc tùy chỉnh yêu cầu một template hoặc nhiều template.

**v-template** không hiển thị bất kỳ thứ gì khi được đặt trong template. Thay vào đó, nó thêm một thuộc tính **\$templates** vào phần tử cha. Thuộc tính \$templates này là một thực thể của TemplateBag .

Tiếp theo, v-template đăng ký chính nó như là một mẫu có sẵn trong thực thể TemplateBag tương ứng cho phần tử cha. Mọi thực thể TemplateBag tồn tại trên phần tử cha đều được sử dụng lại.

## 1. V-TEMPLATE

### **Class TemplateBag :**

Các lớp TemplateBag cho phép bạn đăng ký nhiều mẫu và chọn mẫu chính xác dựa trên các mục và các điều kiện quy định đối với từng mẫu.

Các mẫu được lưu trữ dưới dạng các đối tượng phù hợp với giao diện KeyedTemplate

### **Thuộc tính selectorFn:**

Các thuộc tính selectorFn trả về một hàm chấp nhận một tham số duy nhất. Tham số này là mục (item) có mẫu được chọn.

Hàm có tham số duy nhất đi qua tất cả các mẫu được đăng ký trong thực thể TemplateBag và trả về mẫu đầu tiên nếu thỏa điều kiện. Nếu không có mẫu nào phù hợp, trả về default.

# 1. V-TEMPLATE

## Các phương thức có sẵn:

Phương thức	Mô Tả
<code>registerTemplate(name: String, condition: String?, scopedFn: Function): void</code>	<ul style="list-style-type: none"><li>- Sử dụng cục bộ</li><li>- Registers templates trong thuộc tính <code>TemplateBag</code></li><li>- <code>scopedFn</code> phải là hàm kết xuất của <b>scoped slot</b></li></ul>
<code>getConditionFn(condition: String): Function</code>	<ul style="list-style-type: none"><li>- Sử dụng cục bộ</li><li>- Tạo một hàm trả về giá trị đã cho</li></ul>
<code>getAvailable(): Array&lt;String&gt;</code>	<ul style="list-style-type: none"><li>- Trả về giá trị của mảng <code>KeyedTemplates</code> (Trả về một mảng tên của Template)</li></ul>
<code>getKeyedTemplate(name: String): KeyedTemplate</code>	<ul style="list-style-type: none"><li>- Trả về <code>KeyedTemplate</code> với tên được chỉ định</li></ul>
<code>getKeyedTemplates(): Array&lt;KeyedTemplate&gt;</code>	<ul style="list-style-type: none"><li>- Trả về một mảng của tất cả những gì <code>KeyedTemplates</code> đã đăng ký trong <code>TemplateBag</code>.</li></ul>
<code>patchTemplate(name: String, context: any, oldVnode: VNode?): View</code>	<ul style="list-style-type: none"><li>- Path tồn tại <code>VNode</code> sử dụng cung cấp context. Nếu <code>oldVnode</code> không được cung cấp, hãy tạo một thực thể View mới cho mẫu đã chỉ định.</li></ul>

## 2. V-VIEW

**Directive v-view** cho phép bạn thiết lập các thành phần view hiện tại như thuộc tính cha.

```
<Child v-view:parentPropertyName />  
<!-- same as: -->  
<Child ~parentPropertyName />
```

```
<Child v-view:parentArrayPropertyName.array />  
<!-- same as: -->  
<Child ~parentArrayPropertyName.array />
```

## 2. V-VIEW

### Thí dụ: <RadSideDrawer>

Thành phần <RadSideDrawer> là một phần của gói giao diện người dùng Progress NativeScript .

Thành phần <RadSideDrawer> yêu cầu các thuộc tính drawerContent và mainContent được đặt thành các thực thể View. Sử dụng lệnh v-view này, bạn có thể thực hiện việc này với một vài dòng code:

```
<RadSideDrawer>  
  <StackLayout ~drawerContent />  
  <StackLayout ~mainContent />  
</RadSideDrawer>
```

Nếu không có directive v-view, bạn cần phải đi một con đường tế nhạ hơn và dễ xảy ra lỗi:



## 2. V-VIEW

```
<RadSideDrawer ref="drawer">  
  <StackLayout ref="drawerContent" />  
  <StackLayout ref="mainContent" />  
</RadSideDrawer>
```

```
{  
  mounted() {  
    this.$refs.drawer.nativeView.drawerContent = this.$refs.drawerContent.nativeView  
    this.$refs.drawer.nativeView.mainContent = this.$refs.mainContent.nativeView  
  }  
}
```

### 3. Thành phần: Layout

#### 1. Absolute Layout

Vùng chứa `<AbsoluteLayout>` là vùng chứa bố cục đơn giản nhất trong NativeScript.

**Các trạng thái của `<AbsoluteLayout>` như sau:**

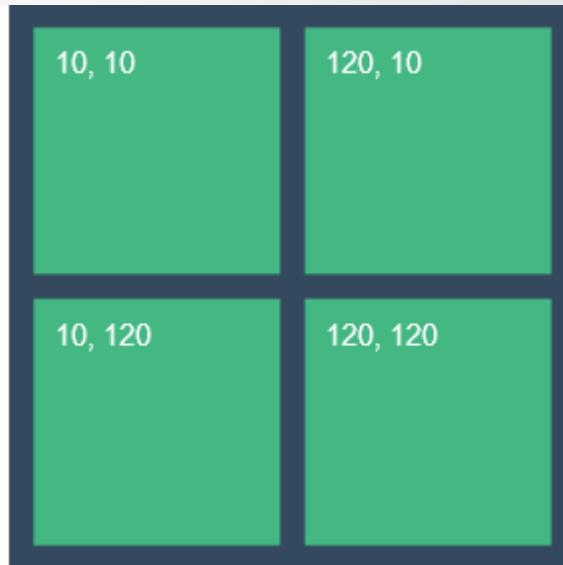
- Sử dụng một cặp tọa độ tuyệt đối bên trái / trên cùng để xác định vị trí con của nó.
- Không thực thi bất kỳ ràng buộc bố cục nào đối với con của nó.
- Không thay đổi kích thước con của nó trong thời gian chạy khi kích thước của nó thay đổi.

### 3. Thành phần: Layout

#### 1. Absolute Layout

##### Ví dụ: Bố cục dạng lưới (Grid Layout)

```
<AbsoluteLayout backgroundColor="#3c495e">  
  <Label text="10,10" left="10" top="10" width="100" height="100" backgroundColor="#43b883"/>  
  <Label text="120,10" left="120" top="10" width="100" height="100" backgroundColor="#43b883"/>  
  <Label text="10,120" left="10" top="120" width="100" height="100" backgroundColor="#43b883"/>  
  <Label text="120,120" left="120" top="120" width="100" height="100" backgroundColor="#43b883"/>  
</AbsoluteLayout>
```

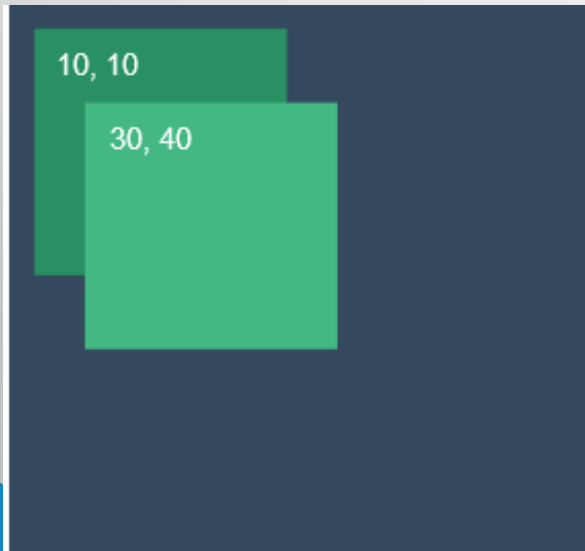


### 3. Thành phần: Layout

#### 1. Absolute Layout

**Các thành phần chồng lên nhau:** Cho phép tạo nhóm mục chồng lên nhau

```
<AbsoluteLayout backgroundColor="#3c495e">  
  <Label text="10,10" left="10" top="10" width="100" height="100" backgroundColor="#289062"/>  
  <Label text="30,40" left="30" top="40" width="100" height="100" backgroundColor="#43b883"/>  
</AbsoluteLayout>
```



### 3. Thành phần: Layout

#### 1. Absolute Layout

**Props:** Thêm Prop con

Khi một phần tử là con trực tiếp của `<AbsoluteLayout>`, bạn có thể làm việc với các thuộc tính bổ sung sau.

Tên	Kiểu	Sự miêu tả
<code>top</code>	Number	Lấy hoặc đặt khoảng cách, tính bằng pixel, giữa cạnh trên của phần con và cạnh trên của phần cha của nó.
<code>left</code>	Number	Lấy hoặc đặt khoảng cách, tính bằng pixel, giữa cạnh trái của phần con và cạnh trái của phần cha của nó.

### 3. Thành phần: Layout

#### 2. Dock Layout

<DockLayout> là một vùng chứa bố cục cho phép bạn gắn các phần tử con vào hai bên hoặc trung tâm của bố cục.

**Các trạng thái của <DockLayout> như sau:**

- Sử dụng thuộc tính dock để dock nó đến left, right, top, bottom hay center của cách bố trí.
- Để gắn một phần tử con vào trung tâm, nó phải là phần tử con cuối cùng của vùng chứa và bạn phải đặt thuộc tính stretchLastChild của phần tử gốc thành true.
- Thực thi các ràng buộc về bố cục đối với con của nó.
- Thay đổi kích thước con của nó trong thời gian chạy khi kích thước của nó thay đổi.

### 3. Thành phần: Layout

#### 2. Dock Layout

Mọi cạnh trong Dock không bị kéo căng phần cuối cùng

Ví dụ: tạo bố cục dạng khung bao gồm 4 phần tử, vị trí ở 4 cạnh của màn hình.

```
<DockLayout stretchLastChild="false" backgroundColor="#3c495e">  
  <Label text="left" dock="left" width="40" backgroundColor="#43b883"/>  
  <Label text="top" dock="top" height="40" backgroundColor="#289062"/>  
  <Label text="right" dock="right" width="40" backgroundColor="#43b883"/>  
  <Label text="bottom" dock="bottom" height="40" backgroundColor="#289062"/>  
</DockLayout>
```



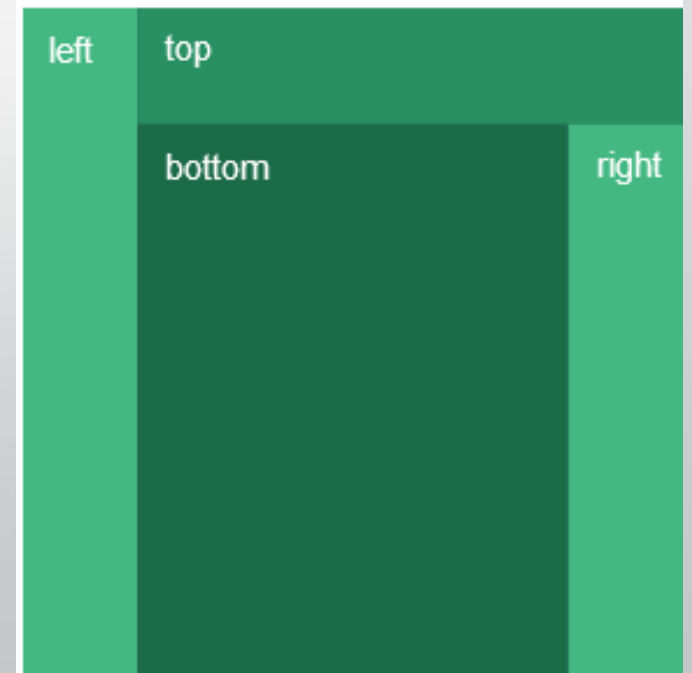
### 3. Thành phần: Layout

## 2. Dock Layout

Mọi cạnh trong Dock và kéo căng phần cuối cùng

Ví dụ: sau đây cho thấy `stretchLastChild` ảnh hưởng như thế nào đến việc định vị các phần tử con trong vùng chứa `DockLayout`. Con cuối cùng ( `bottom`) được kéo dài để chiếm tất cả không gian còn lại sau khi định vị ba phần tử đầu tiên.

```
<DockLayout stretchLastChild="true" backgroundColor="#3c495e">  
  <Label text="left" dock="left" width="40" backgroundColor="#43b883"/>  
  <Label text="top" dock="top" height="40" backgroundColor="#289062"/>  
  <Label text="right" dock="right" width="40" backgroundColor="#43b883"/>  
  <Label text="bottom" dock="bottom" backgroundColor="#1c6b48"/>  
</DockLayout>
```





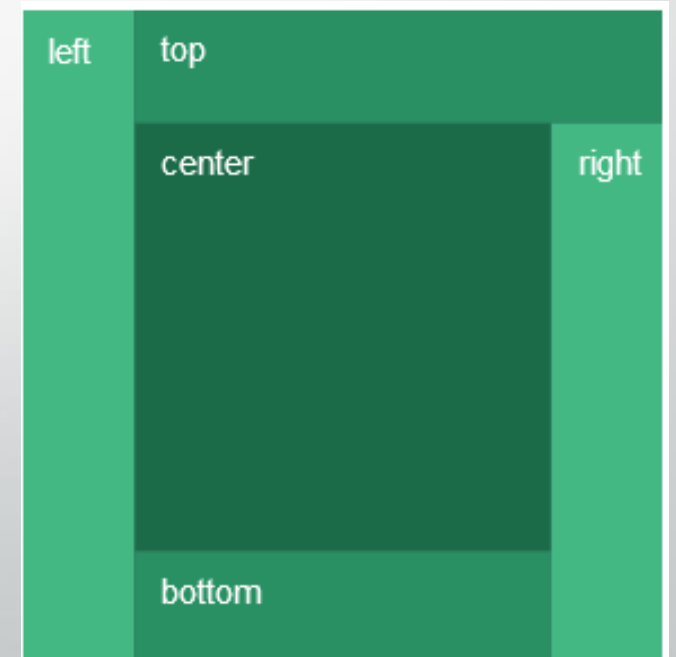
### 3. Thành phần: Layout

## 2. Dock Layout

Mọi cạnh Dock và vị trí trung tâm

Ví dụ: tạo ra một <DockLayout> trong 5 phần tử. Bốn phần đầu tiên bao bọc phần tử trung tâm trong một khung.

```
<DockLayout stretchLastChild="true" backgroundColor="#3c495e">  
  <Label text="left" dock="left" width="40" backgroundColor="#43b883"/>  
  <Label text="top" dock="top" height="40" backgroundColor="#289062"/>  
  <Label text="right" dock="right" width="40" backgroundColor="#43b883"/>  
  <Label text="bottom" dock="bottom" height="40" backgroundColor="#289062"/>  
  <Label text="center" backgroundColor="#1c6b48" />  
</DockLayout>
```



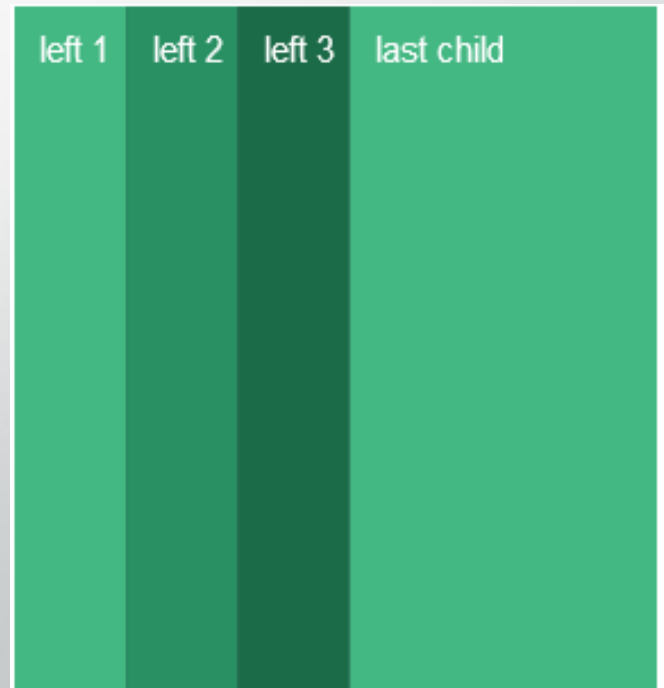
### 3. Thành phần: Layout

## 2. Dock Layout

Gắn nhiều trẻ vào cùng một phía

Ví dụ: Tạo một dòng gồm 4 phần tử trải dài trên toàn bộ chiều cao và chiều rộng của màn hình.

```
<DockLayout stretchLastChild="true" backgroundColor="#3c495e">  
  <Label text="left 1" dock="left" width="40" backgroundColor="#43b883"/>  
  <Label text="left 2" dock="left" width="40" backgroundColor="#289062"/>  
  <Label text="left 3" dock="left" width="40" backgroundColor="#1c6b48"/>  
  <Label text="last child" backgroundColor="#43b883"/>  
</DockLayout>
```



### 3. Thành phần: Layout

## 2. Dock Layout

### Props

Tên	Kiểu	Sự miêu tả
<code>stretchLastChild</code>	Boolean	Bật hoặc tắt kéo giãn con cuối cùng để vừa với không gian còn lại.

### Additional children props

Khi một phần tử là con trực tiếp của `<DockLayout>`, bạn có thể làm việc với các thuộc tính bổ sung sau.

Tên	Kiểu	Sự miêu tả
Dock	String	Chỉ định cạnh gắn vào thành phần nào Giá trị hợp lệ: top, right, bottom, hoặc left.

## 3. Thành phần: Layout

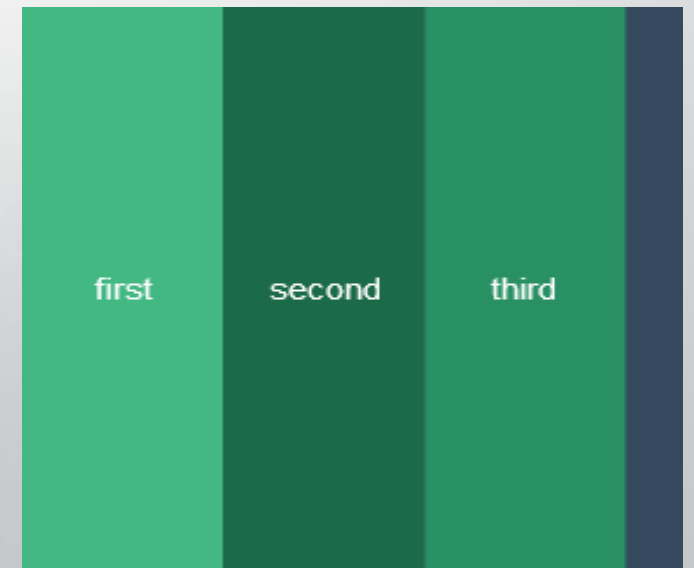
### 3. FlexboxLayout

<FlexboxLayout> là một vùng chứa bố cục cung cấp thành phần không chính xác của bố cục CSS Flexbox. Bố cục này cho phép bạn sắp xếp các thành phần con theo cả chiều ngang và chiều dọc.

#### Default flex layout

Ví dụ sau tạo một hàng gồm ba phần tử có kích thước bằng nhau kéo dài trên toàn bộ chiều cao của màn hình.

```
<FlexboxLayout backgroundColor="#3c495e">  
  <Label text="first" width="70" backgroundColor="#43b883"/>  
  <Label text="second" width="70" backgroundColor="#1c6b48"/>  
  <Label text="third" width="70" backgroundColor="#289062"/>  
</FlexboxLayout>
```



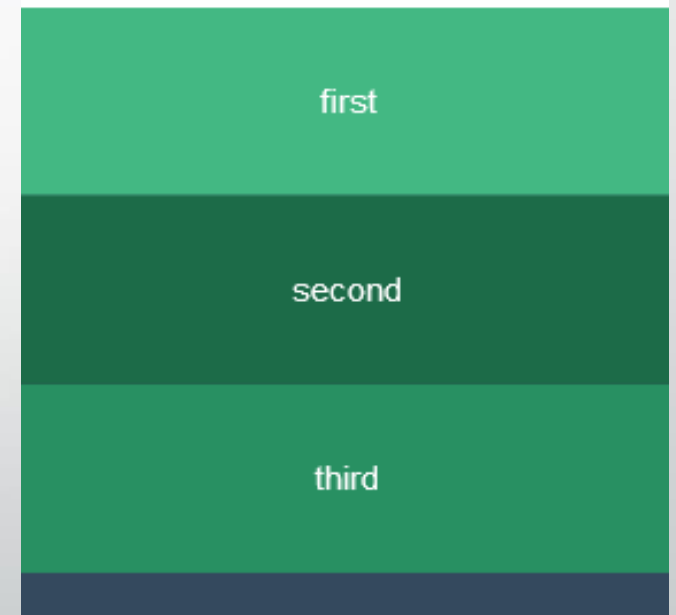
## 3. Thành phần: Layout

### 3. FlexboxLayout

#### Column flex layout

Ví dụ sau tạo một cột gồm ba phần tử có kích thước bằng nhau trải dài trên toàn bộ chiều rộng của màn hình.

```
<FlexboxLayout flexDirection="column" backgroundColor="#3c495e">  
  <Label text="first" height="70" backgroundColor="#43b883"/>  
  <Label text="second" height="70" backgroundColor="#1c6b48"/>  
  <Label text="third" height="70" backgroundColor="#289062"/>  
</FlexboxLayout>
```



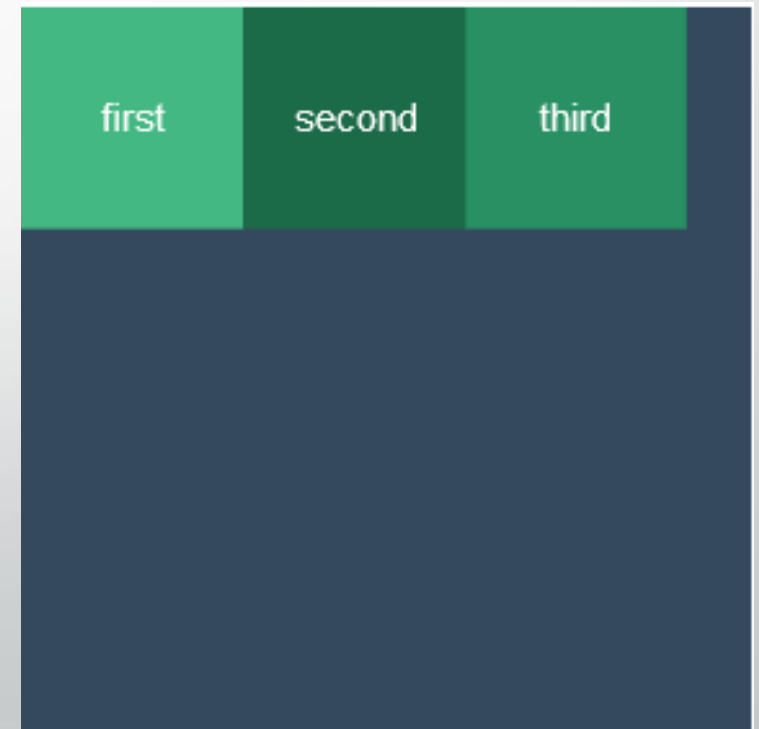
## 3. Thành phần: Layout

### 3. FlexboxLayout

#### Row flex layout với các mục được canh chỉnh theo flex-start

Ví dụ sau tạo một hàng gồm ba mục được đặt ở đầu màn hình. Các mục được đặt theo thứ tự mà chúng đã được khai báo.

```
<FlexboxLayout alignItems="flex-start" backgroundColor="#3c495e">  
  <Label text="first" width="70" height="70" backgroundColor="#43b883"/>  
  <Label text="second" width="70" height="70" backgroundColor="#1c6b48"/>  
  <Label text="third" width="70" height="70" backgroundColor="#289062"/>  
</FlexboxLayout>
```



## 3. Thành phần: Layout

### 3. FlexboxLayout

#### Row flex layout với custom order

Ví dụ sau tạo một hàng gồm ba mục được đặt ở đầu màn hình. Các mục được đặt theo thứ tự tùy chỉnh.

```
<FlexboxLayout alignItems="flex-start" backgroundColor="#3c495e">  
  <Label text="first" order="2" width="70" height="70" backgroundColor="#43b883"/>  
  <Label text="second" order="3" width="70" height="70" backgroundColor="#1c6b48"/>  
  <Label text="third" order="1" width="70" height="70" backgroundColor="#289062"/>  
</FlexboxLayout>
```



## 3. Thành phần: Layout

### 3. FlexboxLayout

#### Row flex layout với wrapping

Ví dụ sau đây tạo ra bốn mục có kích hoạt wrapping line. Khi hàng hết không gian, vùng chứa wrap sẽ tạo dòng mới chứa mục cuối cùng

```
<FlexboxLayout flexWrap="wrap" backgroundColor="#3c495e">  
  <Label text="first" width="30%" backgroundColor="#43b883"/>  
  <Label text="second" width="30%" backgroundColor="#1c6b48"/>  
  <Label text="third" width="30%" backgroundColor="#289062"/>  
  <Label text="fourth" width="30%" backgroundColor="#289062"/>  
</FlexboxLayout>
```





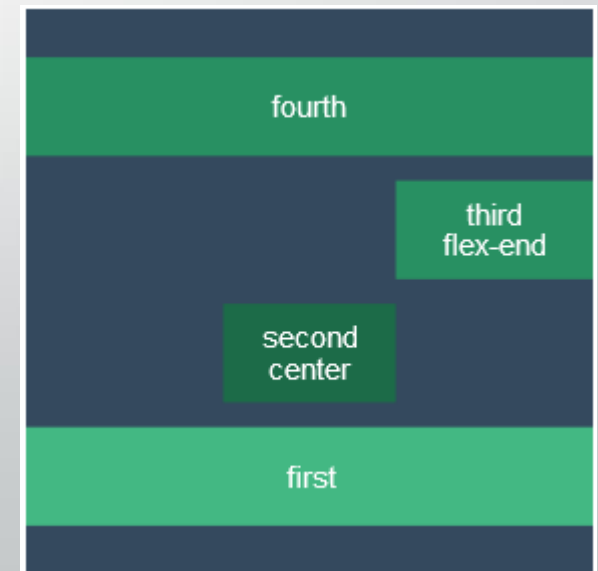
## 3. Thành phần: Layout

### 3. FlexboxLayout

**Column flex layout với thứ tự ngược lại và các mục khác với alignSelf**

- **flexDirection** để đặt các mục trong một cột, bắt đầu từ dưới cùng.
- **justifyContent** để tạo khoảng cách bằng nhau giữa các mục được đặt theo chiều dọc.
- **alignSelf** để sửa đổi vị trí của các mục trên trục chính.

```
<FlexboxLayout flexDirection="column-reverse"
    justifyContent="space-around" backgroundColor="#3c495e">
  <Label text="first" height="70" backgroundColor="#43b883"/>
  <Label text="second" alignSelf="center" width="70" height="70" backgroundColor="#1c6b48"/>
  <Label text="third\nflex-end" alignSelf="flex-end" width="70" height="70" backgroundColor="#289062"/>
  <Label text="fourth" height="70" backgroundColor="#289062"/>
</FlexboxLayout>
```



## 3. Thành phần: Layout

### 3. FlexboxLayout

#### Props

Tên	Kiểu	Mô tả
flexDirection	String	Thiết lập hướng để đặt các phần tử con trong vùng chứa flexbox. Giá trị hợp lệ <ul style="list-style-type: none"><li>row (horizontal, left to right),</li><li>row-reverse (horizontal, right to left),</li><li>column (vertical, top to bottom),</li><li>column-reverse (vertical, bottom to top).</li></ul> Giá trị mặc định: row.
flexWrap	String	Thiết lập các phần tử con bị ép trong một dòng hay có thể chảy thành nhiều dòng. Nếu được đặt thành nhiều dòng, cũng xác định trục chéo, xác định hướng các dòng mới được xếp chồng lên nhau. Các giá trị hợp lệ: <ul style="list-style-type: none"><li>nowrap(một dòng có thể khiến vùng chứa bị tràn),</li><li>wrap(nhiều dòng, hướng được xác định bởi flexDirection) và</li><li>wrap-reverse(nhiều dòng , ngược với hướng được xác định bởi flexDirection).</li></ul> Giá trị mặc định: nowrap.

## 3. Thành phần: Layout

### 3. FlexboxLayout

#### Props

Tên	Kiểu	Mô tả
justifyContent	String	<p>Căn chỉnh các phần tử con dọc theo trục chính. Bạn có thể sử dụng nó để phân bổ không gian còn lại khi tất cả các phần tử con trên một dòng là không linh hoạt hoặc linh hoạt nhưng đã đạt đến kích thước tối đa. Bạn cũng có thể sử dụng nó để kiểm soát sự căn chỉnh của các mục khi chúng tràn dòng.</p> <p>Giá trị hợp lệ:</p> <ul style="list-style-type: none"><li>• flex-start(các mục được đóng gói về phía dòng bắt đầu),</li><li>• flex-end(các mục được đóng gói về phía dòng cuối),</li><li>• center(các mục được căn giữa dọc theo dòng),</li><li>• space-between(các mục được phân bổ đều trên dòng; mục đầu tiên ở dòng bắt đầu, mục cuối cùng trên dòng cuối),</li><li>• space-around(các mục được phân bổ đều trên dòng với khoảng cách bằng nhau xung quanh chúng).</li></ul> <p>Giá trị mặc định: flex-start.</p>

## 3. Thành phần: Layout

### 3. FlexboxLayout

#### Props

Tên	Kiểu	Mô tả
alignItems	String	<p>(Chỉ dành cho Android) căn chỉnh của các phần tử con dọc theo trục chéo trên dòng hiện tại. Hoạt động như justifyContent đối với trục chéo.</p> <p>Các giá trị hợp lệ:</p> <p>flex-start(cross-start cạnh lề chéo của các mục được đặt trên dòng cross-start),</p> <p>flex-end(cross-end cạnh lề cuối của các mục được đặt trên dòng cross-end),</p> <p>center(các mục được căn giữa trên trục chéo) ,</p> <p>baseline(các đường cơ sở của item được căn chỉnh) và</p> <p>stretch(các item được kéo dài để lấp đầy vùng chứa nhưng vẫn liên quan đến min-width và max-width).</p> <p>Giá trị mặc định: stretch.</p>

## 3. Thành phần: Layout

### 3. FlexboxLayout

#### Props

Tên	Kiểu	Mô tả
alignContent	String	<p>Các dòng được căn chỉnh trong vùng chứa flex trên trục chéo, tương tự như cách justifyContent căn chỉnh các mục riêng lẻ trong trục chính.</p> <p>Thuộc tính này không có hiệu lực khi vùng chứa flex chỉ có một dòng.</p> <p>Giá trị hợp lệ:</p> <ul style="list-style-type: none"><li>flex-start(các dòng được đóng gói đến đầu vùng chứa),</li><li>flex-end(các dòng được đóng gói đến cuối vùng chứa),</li><li>center(các dòng được đóng gói ở giữa vùng chứa),</li><li>space-between(các dòng được phân bố đều; dòng đầu tiên ở đầu của vùng chứa trong khi vùng cuối cùng ở cuối),</li><li>space-around(các dòng được phân bố đều với khoảng cách bằng nhau giữa chúng)</li><li>stretch(các dòng được kéo dài để chiếm không gian còn lại).</li></ul> <p>Giá trị mặc định: stretch.</p>

## 3. Thành phần: Layout

### 3. FlexboxLayout

- **Additional children props**

Tên	Kiểu	Mô tả
order	Number	Thiết lập thứ tự mà phần tử con xuất hiện trong mối quan hệ với nhau.
flexGrow	Number	Thiết lập về mặt phát triển kích thước. .
flexShrink	Number	Thu nhỏ khi hàng hết chỗ. Đặt mức độ mà mục flex sẽ co lại tương ứng với phần còn lại của các phần tử con trong vùng chứa flex. Khi không được chỉ định, giá trị của nó được đặt thành 1.
alignSelf	String	(Chỉ dành cho Android) Các giá trị hợp lệ: flex-start flex-end center(mục được căn giữa trên trục chéo) , baseline(các đường cơ sở của item được căn chỉnh) stretch(các item được kéo căng để lấp đầy vùng chứa nhưng vẫn quan tâm đến min-width và max-width). Giá trị mặc định: stretch.
flexWrapBefore	Boolean	Khi nào true, buộc mục phải wrap vào một dòng mới. Thuộc tính này không phải là một phần của đặc điểm kỹ thuật Flexbox chính thức.

### 3. Thành phần: Layout

#### 4. GridLayout

<GridLayout> là một vùng chứa bố cục cho phép bạn sắp xếp các phần tử con của nó theo cách giống như bảng.

Lưới bao gồm các hàng, cột và ô. Một ô có thể kéo dài một hoặc nhiều hàng và một hoặc nhiều cột. Nó có thể chứa nhiều phần tử con có thể kéo dài trên nhiều hàng và cột, thậm chí chồng chéo lên nhau.

Theo mặc định, <GridLayout> có một cột và một hàng. Bạn có thể thêm các cột và các hàng bằng cách thiết lập thuộc tính của columns và rows. Trong các thuộc tính này, bạn cần đặt số cột và hàng cũng như chiều rộng và chiều cao của chúng. Bạn đặt số lượng cột bằng cách liệt kê chiều rộng của chúng, phân tách bằng dấu phẩy. Bạn đặt số hàng bằng cách liệt kê chiều cao của chúng, phân tách bằng dấu phẩy.

Bạn có thể đặt kích thước cố định cho chiều rộng cột và chiều cao hàng hoặc bạn có thể tạo chúng theo cách sau:

### 3. Thành phần: Layout

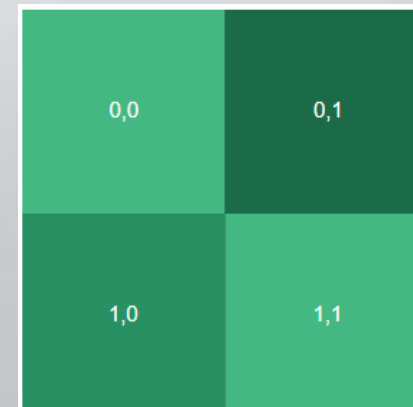
#### 4. GridLayout

- **Một số tuyệt đối:** Cho biết một kích thước cố định.
- **auto:** Làm cho cột rộng bằng với con rộng nhất của nó hoặc làm cho hàng cao bằng với con cao nhất của nó.
- **\***: Dành nhiều không gian nhất có thể sau khi điền vào tất cả các cột hoặc hàng có kích thước tự động và cố định.

#### Bố cục lưới với kích thước cố định

Ví dụ sau đây tạo một lưới đơn giản 2 x 2 với chiều rộng cột và chiều cao hàng cố định.

```
<GridLayout columns="115, 115" rows="115, 115">  
  <Label text="0,0" row="0" col="0" backgroundColor="#43b883"/>  
  <Label text="0,1" row="0" col="1" backgroundColor="#1c6b48"/>  
  <Label text="1,0" row="1" col="0" backgroundColor="#289062"/>  
  <Label text="1,1" row="1" col="1" backgroundColor="#43b883"/>  
</GridLayout>
```





### 3. Thành phần: Layout

## 4. GridLayout

### Bố cục lưới với kích thước \*

Ví dụ sau tạo một lưới với thiết kế responsive, trong đó không gian được phân bổ tương ứng với các phần tử con.

```
<GridLayout columns="*, 2*" rows="2*, 3*" backgroundColor="#3c495e">  
  <Label text="0,0" row="0" col="0" backgroundColor="#43b883"/>  
  <Label text="0,1" row="0" col="1" backgroundColor="#1c6b48"/>  
  <Label text="1,0" row="1" col="0" backgroundColor="#289062"/>  
  <Label text="1,1" row="1" col="1" backgroundColor="#43b883"/>  
</GridLayout>
```

0,0	0,1
1,0	1,1

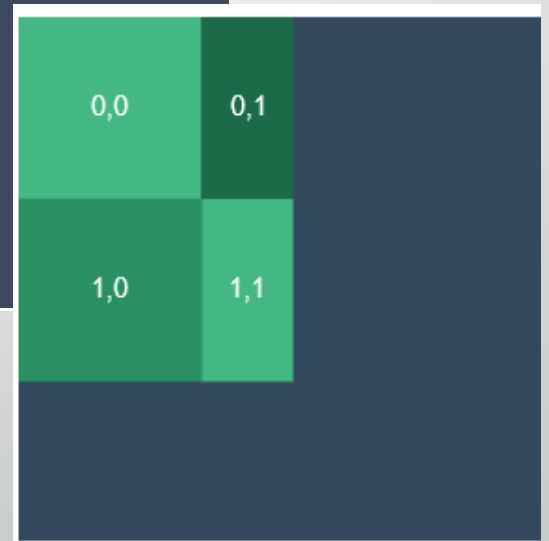
### 3. Thành phần: Layout

## 4. GridLayout

### Bố cục lưới với kích thước cố định và tự động

Ví dụ sau tạo lưới với một cột có kích thước tự động và một cột có kích thước cố định. Hàng có chiều cao cố định.

```
<GridLayout columns="80, auto" rows="80, 80" backgroundColor="#3c495e">  
  <Label text="0,0" row="0" col="0" backgroundColor="#43b883"/>  
  <Label text="0,1" row="0" col="1" backgroundColor="#1c6b48"/>  
  <Label text="1,0" row="1" col="0" backgroundColor="#289062"/>  
  <Label text="1,1" row="1" col="1" backgroundColor="#43b883"/>  
</GridLayout>
```



### 3. Thành phần: Layout

## 4. GridLayout

**Bố cục lưới với định cỡ hỗn hợp và các ô hợp nhất**

Ví dụ sau tạo một lưới phức tạp với thiết kế responsive, thiết lập chiều rộng và chiều cao hỗn hợp (mix) và một số ô được hợp nhất.(merged)

```
<GridLayout columns="40, auto, *" rows="40, auto, *" backgroundColor="#3c495e">
  <Label text="0,0" row="0" col="0" backgroundColor="#43b883"/>
  <Label text="0,1" row="0" col="1" colSpan="2" backgroundColor="#1c6b48"/>
  <Label text="1,0" row="1" col="0" rowSpan="2" backgroundColor="#289062"/>
  <Label text="1,1" row="1" col="1" backgroundColor="#43b883"/>
  <Label text="1,2" row="1" col="2" backgroundColor="#289062"/>
  <Label text="2,1" row="2" col="1" backgroundColor="#1c6b48"/>
  <Label text="2,2" row="2" col="2" backgroundColor="#43b883"/>
</GridLayout>
```

0,0	0,1 colSpan=2	
1,0 rowSpan 2	1,1	1,2
	2,1	2,2

### 3. Thành phần: Layout

## 4. GridLayout

### Props

Tên	Kiểu	Sự miêu tả
columns	String	<p>Giá trị chuỗi biểu thị độ rộng cột được phân tách bằng dấu phẩy.</p> <p>Giá trị hợp lệ: số tuyệt đối auto, hoặc *.</p> <p>Một số cho biết chiều rộng cột tuyệt đối. <b>Auto</b> làm cho cột rộng bằng con rộng nhất của nó. * làm cho cột chiếm tất cả không gian ngang có sẵn. Không gian được phân chia theo tỷ lệ trên tất cả các cột có kích thước hình sao. Bạn có thể đặt các giá trị như 3* và 5* để chỉ ra tỷ lệ 3: 5 về kích thước.</p>
rows	String	<p>Giá trị chuỗi biểu thị độ cao của hàng được phân tách bằng dấu phẩy.</p> <p>Giá trị hợp lệ: số tuyệt đối auto, hoặc *.</p> <p>Một số cho biết chiều cao hàng tuyệt đối. <b>Auto</b> làm cho hàng cao bằng con cao nhất của nó. * làm cho hàng chiếm tất cả không gian dọc có sẵn. Không gian được chia theo tỷ lệ trên tất cả các hàng có kích thước hình sao. Bạn có thể đặt các giá trị như 3* và 5* để chỉ ra tỷ lệ 3: 5 về kích thước.</p>

### 3. Thành phần: Layout

## 4. GridLayout

- Additional children props

Tên	Kiểu	Sự miêu tả
row	Number	Chỉ định hàng cho phần tử này. Kết hợp với một thuộc tính col chỉ định tọa độ ô của phần tử. Hàng đầu tiên được chỉ định bởi 0.
col	Number	Chỉ định cột cho phần tử. Kết hợp với một thuộc tính row, chỉ định tọa độ ô của phần tử. Cột đầu tiên được biểu thị bằng 0.
rowSpan	Number	Chỉ định số hàng mà phần tử này kéo dài qua.
colSpan	Number	Chỉ định số cột mà phần tử này kéo dài qua.

## 3. Thành phần: Layout

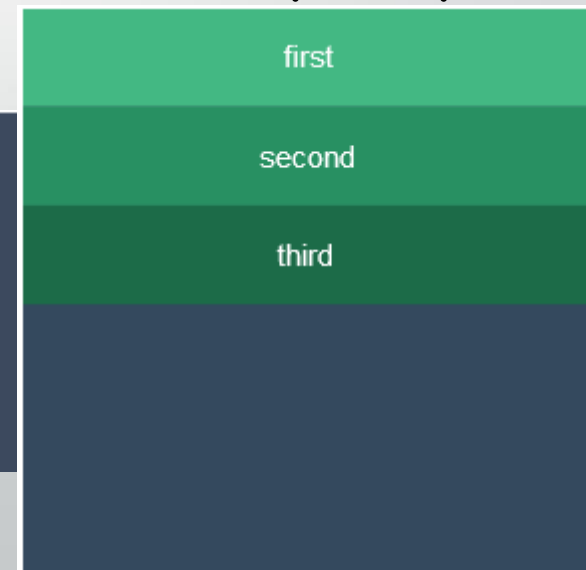
### 5. StackLayout

<StackLayout> là một vùng chứa bố cục cho phép bạn xếp chồng các phần tử con theo chiều dọc (mặc định) hoặc theo chiều ngang.

#### Xếp chồng mặc định

Ví dụ sau tạo một ngăn xếp dọc gồm 3 phần tử có kích thước bằng nhau. Các mục được kéo dài để bao phủ toàn bộ chiều rộng của màn hình. Các mục được đặt theo thứ tự mà chúng đã được khai báo.

```
<StackLayout backgroundColor="#3c495e">  
  <Label text="first" height="70" backgroundColor="#43b883"/>  
  <Label text="second" height="70" backgroundColor="#289062"/>  
  <Label text="third" height="70" backgroundColor="#1c6b48"/>  
</StackLayout>
```



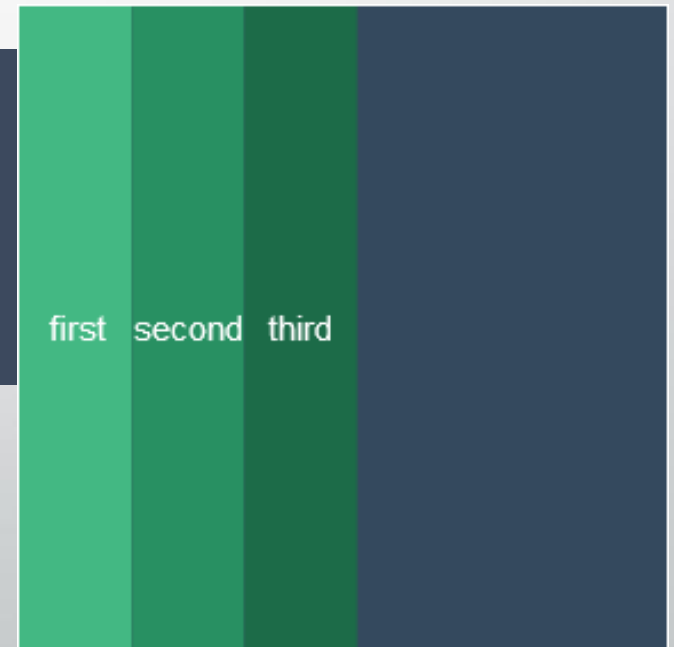
### 3. Thành phần: Layout

#### 5. StackLayout

##### Xếp chồng theo chiều ngang

Ví dụ sau tạo một ngăn xếp ngang gồm 3 phần tử có kích thước bằng nhau. Các mục được kéo dài để bao phủ toàn bộ chiều cao của màn hình. Các mục được đặt theo thứ tự mà chúng đã được khai báo.

```
<StackLayout orientation="horizontal" backgroundColor="#3c495e">  
  <Label text="first" width="70" backgroundColor="#43b883"/>  
  <Label text="second" width="70" backgroundColor="#289062"/>  
  <Label text="third" width="70" backgroundColor="#1c6b48"/>  
</StackLayout>
```



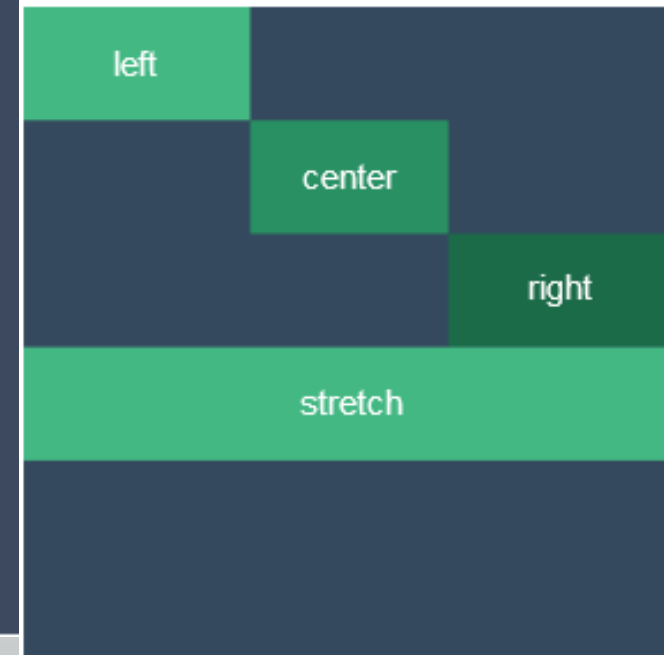
### 3. Thành phần: Layout

#### 5. StackLayout

**Bố cục ngăn xếp với các con được căn chỉnh theo chiều ngang**

Ví dụ sau tạo một stack các mục chồng lên nhau với kích thước responsive. Các mục được xếp chồng lên nhau theo chiều dọc.

```
<StackLayout backgroundColor="#3c495e">
  <Label text="left" horizontalAlignment="left"
    width="33%" height="70" backgroundColor="#43b883"/>
  <Label text="center" horizontalAlignment="center"
    width="33%" height="70" backgroundColor="#289062"/>
  <Label text="right" horizontalAlignment="right"
    width="33%" height="70" backgroundColor="#1c6b48"/>
  <Label text="stretch" horizontalAlignment="stretch"
    height="70" backgroundColor="#43b883"/>
</StackLayout>
```





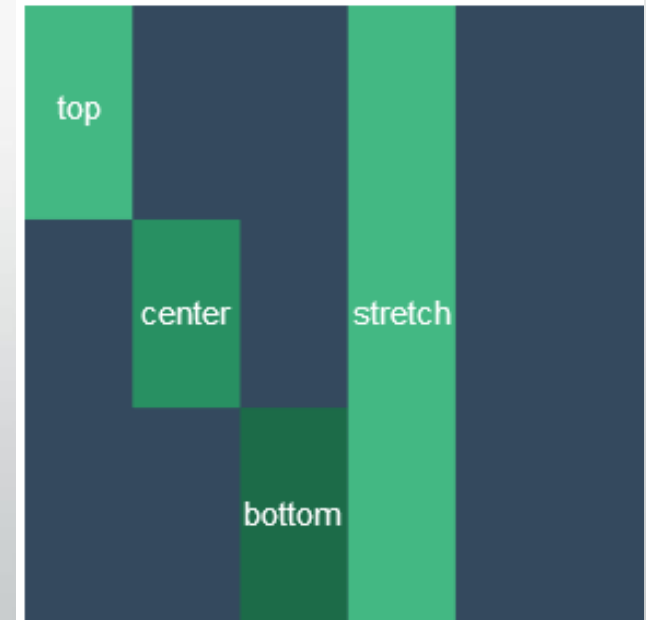
### 3. Thành phần: Layout

#### 5. StackLayout

**Bố cục ngăn xếp theo chiều ngang với các con được căn chỉnh theo chiều dọc**

Ví dụ sau tạo một stack các mục chồng lên nhau với kích thước responsive. Các mục được xếp chồng lên nhau theo chiều ngang.

```
<StackLayout orientation="horizontal" backgroundColor="#3c495e">  
  <Label text="top" verticalAlignment="top"  
    width="70" height="33%" backgroundColor="#43b883"/>  
  <Label text="center" verticalAlignment="center"  
    width="70" height="33%" backgroundColor="#289062"/>  
  <Label text="bottom" verticalAlignment="bottom"  
    width="70" height="33%" backgroundColor="#1c6b48"/>  
  <Label text="stretch" verticalAlignment="stretch"  
    width="70" backgroundColor="#43b883"/>  
</StackLayout>
```



### 3. Thành phần: Layout

#### 5. StackLayout

##### Props

Tên	Kiểu	Sự miêu tả
<code>orientation</code>	<code>String</code>	Chỉ định hướng xếp chồng. Giá trị hợp lệ: <code>vertical</code> và <code>horizontal</code> . Giá trị mặc định: <code>vertical</code> .

### 3. Thành phần: Layout

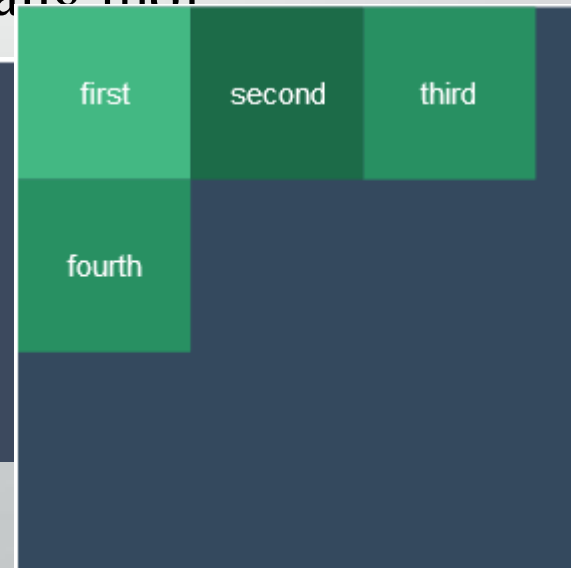
#### 6. WrapLayout

<WrapLayout> là một vùng chứa bố cục cho phép bạn định vị trí các mục trong hàng hoặc cột, dựa trên thuộc tính orientation. Khi không gian được lấp đầy, vùng chứa sẽ tự động wrap các mục vào một hàng hoặc cột mới.

##### Default wrap layout

Ví dụ sau đây tạo một hàng các mục có kích thước bằng nhau. Khi hàng hết không gian chứa, vùng chứa sẽ wrap mục cuối cùng trên một hàng mới.

```
<WrapLayout backgroundColor="#3c495e">
  <Label text="first" width="30%" height="30%" backgroundColor="#43b883"/>
  <Label text="second" width="30%" height="30%" backgroundColor="#1c6b48"/>
  <Label text="third" width="30%" height="30%" backgroundColor="#289062"/>
  <Label text="fourth" width="30%" height="30%" backgroundColor="#289062"/>
</WrapLayout>
```



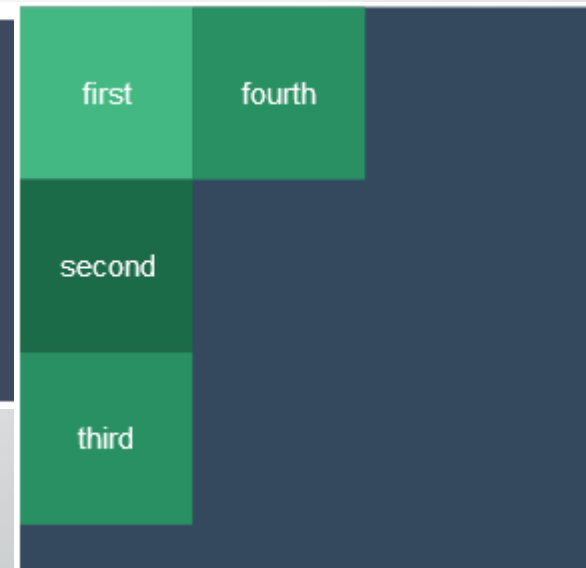
### 3. Thành phần: Layout

## 6. WrapLayout

### Vertical wrap layout

Ví dụ sau tạo một cột gồm các mục có kích thước bằng nhau. Khi hàng hết không gian chứa, vùng chứa sẽ wrap mục cuối cùng trên một cột mới.

```
<WrapLayout orientation="vertical" backgroundColor="#3c495e">  
  <Label text="first" width="30%" height="30%" backgroundColor="#43b883"/>  
  <Label text="second" width="30%" height="30%" backgroundColor="#1c6b48"/>  
  <Label text="third" width="30%" height="30%" backgroundColor="#289062"/>  
  <Label text="fourth" width="30%" height="30%" backgroundColor="#289062"/>  
</WrapLayout>
```



### 3. Thành phần: Layout

## 6. WrapLayout

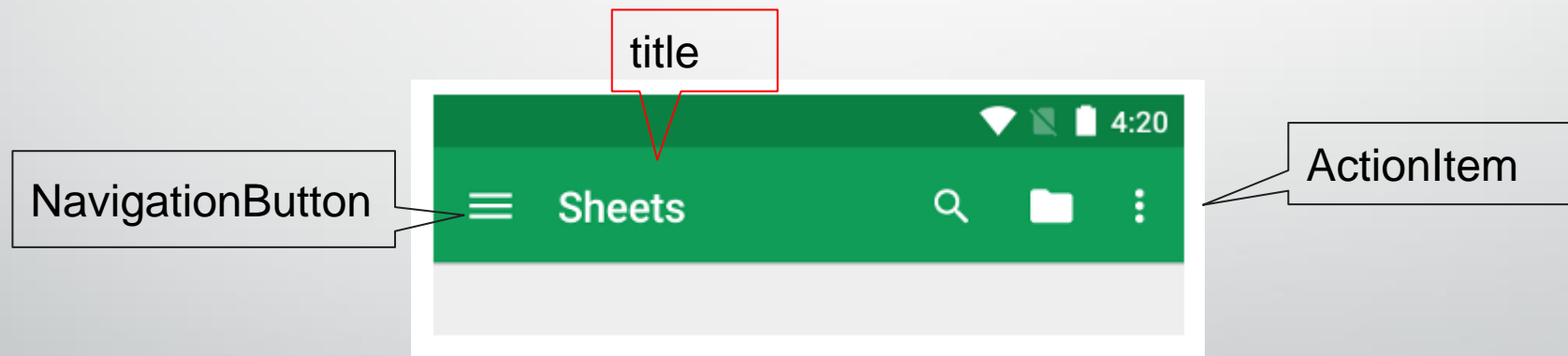
### Props

Tên	Kiểu	Sự miêu tả
orientation	String	Chỉ định hướng xếp chồng. Giá trị hợp lệ: horizontal(sắp xếp các mục theo hàng) và vertical(sắp xếp các mục theo cột). Giá trị mặc định: horizontal.
itemWidth	Number	Thiết lập chiều rộng được sử dụng để đo và bố trí thành phần con Giá trị mặc định: , Number.NaN, không giới hạn thành phần con
itemHeight	Number	Thiết lập chiều cao được sử dụng để đo và bố trí thành phần con. Giá trị mặc định là Number.NaN, không giới hạn thành phần con.

## 4. Thành phần: action-bar

### 1. ActionBar

<ActionBar> là một thành phần giao diện người dùng cung cấp một thanh công cụ ở đầu màn hình của bạn.



## 4. Thành phần: action-bar

### 1. ActionBar

Sử dụng với title

```
<ActionBar title="MyApp" />
```

Sử dụng chế độ xem tiêu đề tùy chỉnh (custom title view)

```
<ActionBar>  
  <StackLayout orientation="horizontal">  
    <Image src="res://icon" width="40" height="40" verticalAlignment="center" />  
    <Label text="NativeScript" fontSize="24" verticalAlignment="center" />  
  </StackLayout>  
</ActionBar>
```

Đặt biểu tượng ứng dụng cho Android

```
<ActionBar title="My App" android.icon="res://icon" android.iconVisibility="always" />
```

## 4. Thành phần: action-bar

### 1. ActionBar

#### Xóa đường viền

Theo mặc định, một đường viền được vẽ ở dưới cùng của <ActionBar>. Ngoài đường viền, trên thiết bị iOS, một bộ lọc trong suốt cũng được áp dụng trên <ActionBar>.

Để xóa kiểu này khỏi ứng dụng của bạn, bạn có thể đặt thuộc tính **flat** thành **true**.

```
<ActionBar title="My App" flat="true" />
```



## 4. Thành phần: action-bar

### 1. ActionBar

#### Props

Tên	Kiểu	Sự miêu tả
title	String	Lấy hoặc gán tiêu đề hiển thị trên thanh.
android.icon	String	Lấy hoặc gán biểu tượng để hiển thị trên thiết bị Android.
android.iconVisibility	String	Lấy hoặc gán biểu tượng hiển thị trên thiết bị Android.
flat	boolean	Loại bỏ viền trên Android và trong suốt trên iOS. Giá trị mặc định là false.

#### Thành phần Native

Android	iOS
<code>android.widget.Toolbar</code>	<code>UINavigationController</code>

## 4. Thành phần: action-bar

### 2. ActionItem

<ActionItem> là một thành phần giao diện người dùng cho phép bạn thêm các nút hành động vào thành phần <ActionBar>

#### Dạng cơ bản

```
<ActionBar title="My App">
  <ActionItem @tap="onTapShare"
    ios.systemIcon="9" ios.position="left"
    android.systemIcon="ic_menu_share" android.position="actionBar" />
  <ActionItem @tap="onTapDelete"
    ios.systemIcon="16" ios.position="right"
    text="delete" android.position="popup" />
</ActionBar>
```

## 4. Thành phần: action-bar

### 2. ActionItem

#### Hiển thị các mục hành động có điều kiện

Có thể sử dụng v-show để hiển thị các thành phần <ActionItem> dựa trên một điều kiện.

```
<ActionBar title="My App">
  <ActionItem @tap="onTapEdit"
    v-show="!isEditing"
    ios.systemIcon="2" ios.position="right"
    android.systemIcon="ic_menu_edit" />
  <ActionItem @tap="onTapSave"
    v-show="isEditing"
    ios.systemIcon="3" ios.position="right"
    android.systemIcon="ic_menu_save" />
  <ActionItem @tap="onTapCancel"
    v-show="isEditing"
    ios.systemIcon="1"
    android.systemIcon="ic_menu_close_clear_cancel" />
</ActionBar>
```

## 4. Thành phần: action-bar

### 2. ActionItem

#### Props

Tên	Kiểu	Sự miêu tả
ios.systemIcon	Number	Lấy hoặc gán biểu tượng của ActionItemiOS. Giá trị phải là một số từ kiểu liệt kê UIBarButtonSystemItem.
android.systemIcon	String	Lấy hoặc gán biểu tượng của ActionItem cho Android. Giá trị phải là tên của drawable resource.
ios.position	String	Lấy hoặc gán biểu tượng của ActionItem trong ActionBar cho iOS. Giá trị hợp lệ: left hoặc right. Giá trị mặc định là left.
android.position	String	Lấy hoặc gán biểu tượng của ActionItem bên trong ActionBar dành cho Android. Giá trị hợp lệ: actionBar(nơi các mục trong ActionBar) popup(nơi các mục trong menu tùy chọn; làm cho các mục dưới dạng văn bản) actionBarIfRoom(nơi các mục trong ActionBar nếu có đủ chỗ cho nó ở đó, nếu không đặt nó trong menu tùy chọn)

## 4. Thành phần: action-bar

### 2. ActionItem

#### Event

Tên	Sự miêu tả
<code>tap</code>	Phát ra khi <code>ActionItem</code> được khai thác.

#### Thành phần Native

Android	iOS
<code>android.widget.Toolbar</code>	<code>UINavigationController</code>

## 4. Thành phần: action-bar

### 3. NavigationButton

<NavigationButton> là một thành phần giao diện người dùng cung cấp sự trừu tượng cho nút điều hướng Android và nút quay lại iOS.

Các phần mở rộng <ActionItem>.

```
<ActionBar title="My App">
  <NavigationButton text="Go back" android.systemIcon="ic_menu_back" @tap="goBack" />
</ActionBar>
```

### Props

Tên	Kiểu	Sự miêu tả
text	String	(Chỉ dành cho iOS) Gán văn bản của nút.
android.systemIcon	String	(Chỉ dành cho Android) Biểu tượng được hiển thị trong nút. Bạn có thể chỉ định bất kỳ biểu tượng hệ thống nào có tên bắt đầu bằng tiền tố ic_. Đối với một danh sách có sẵn đầy đủ các biểu tượng, xem các lớp <b>R.drawable Android</b> .

## 4. Thành phần: action-bar

### 3. NavigationButton

#### Event

Tên	Sự miêu tả
tap	Phát ra khi <code>&lt;NavigationButton&gt;</code> được khai thác.

#### Thành phần Native

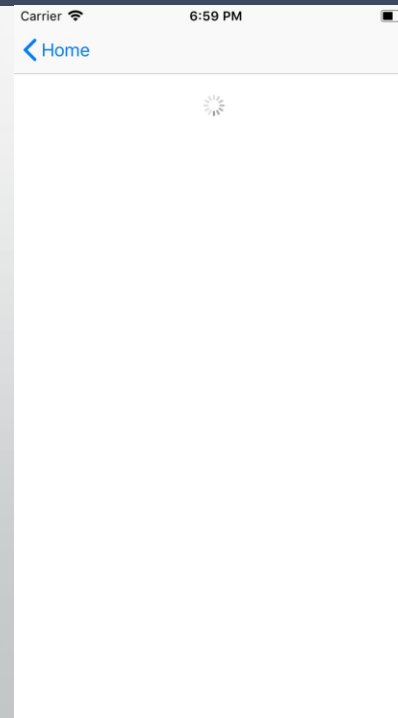
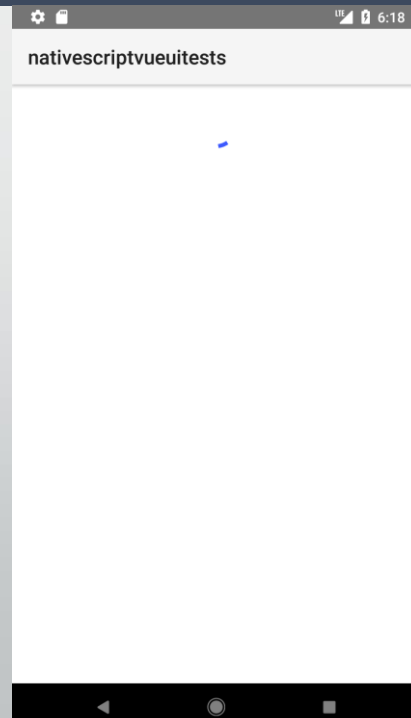
Android	iOS
<code>android.widget.Toolbar</code>	<code>UINavigationController</code>

## 5. Thành phần: Components

### 1. ActivityIndicator

<ActivityIndicator> là một thành phần giao diện người dùng hiển thị một chỉ thị tiến trình, báo hiệu cho người dùng về một hoạt động đang chạy trong background.

```
<ActivityIndicator busy="true" @busyChange="onBusyChanged" />
```





## 5. Thành phần: Components

### 1. ActivityIndicator

#### Props

Tên	Kiểu	Sự miêu tả
busy	Boolean	Lấy hoặc gán xem indicator (chỉ thị) có hoạt động hay không. Khi true, indicator (chỉ báo) hoạt động.

#### Thành phần Native

Android	iOS
<code>android.widget.ProgressBar</code> ( <code>indeterminate = true</code> )	<code>UIActivityIndicatorView</code>

## 5. Thành phần: Components

### 2. Button

`<Button>` là một thành phần giao diện người dùng hiển thị một nút phản ứng với cử chỉ của người dùng.

```
<Button text="Button" @tap="onButtonTap" />
```

nativescriptvueitests

BUTTON

Carrier 6:59 PM

[< Home](#)

Button

## 5. Thành phần: Components

### 2. Button

#### Tạo kiểu cho nút

Nếu bạn cần tạo kiểu cho các phần của văn bản, bạn có thể sử dụng kết hợp các thành phần `FormattedString` và `Span`.

```
<Button>
  <FormattedString>
    <Span text="This text has a " />
    <Span text="red " style="color: red" />
    <Span text="piece of text. " />
    <Span text="Also, this bit is italic, " fontStyle="italic" />
    <Span text="and this bit is bold." fontWeight="bold" />
  </FormattedString>
</Button>
```

## 5. Thành phần: Components

### 2. Button

#### Props

Tên	Kiểu	Sự miêu tả
text	String	Đặt nhãn của nút.
textWrap	Boolean	Lấy hoặc gán tiện ích có wraps văn bản của nhãn hay không. Hữu ích cho các nhãn dài hơn. Giá trị mặc định là false.
isEnabled	Boolean	Tắt hoặc bật nút. Một nút bị vô hiệu hóa không sử dụng được và không thể nhấp vào. Giá trị mặc định là true.

#### Event

Tên	Sự miêu tả
tap	Phát ra khi nhấn vào nút.

#### Thành phần Native

Android	iOS
<code>android.widget.Button</code>	<code>UIButton</code>

## 5. Thành phần: Components

### 3. DatePicker

`<DatePicker>` là một thành phần giao diện người dùng cho phép người dùng chọn ngày từ phạm vi được cấu hình trước.

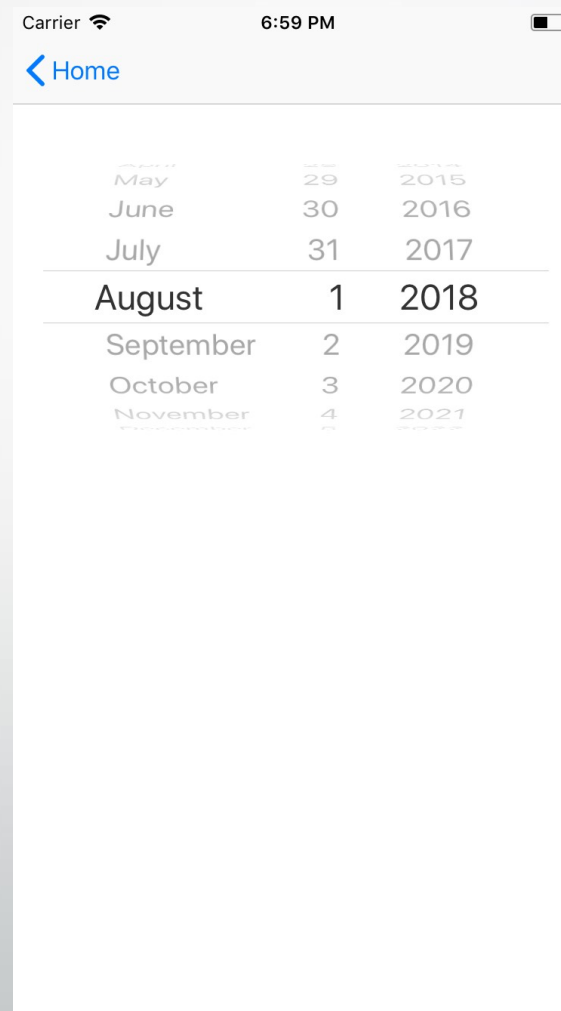
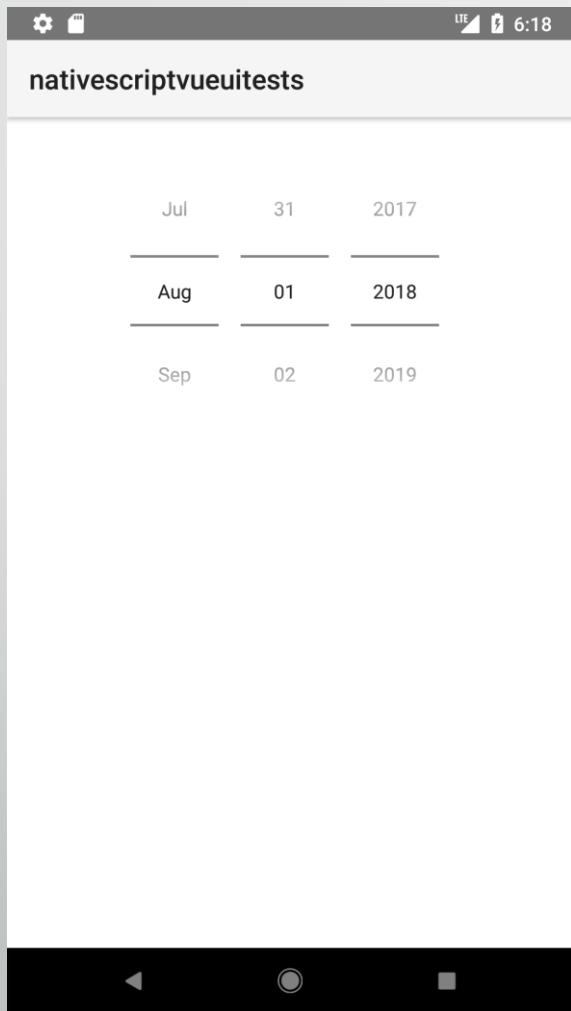
```
<DatePicker :date="someDate" />
```

`<DatePicker>` cung cấp liên kết dữ liệu hai chiều bằng cách sử dụng **v-model**.

```
<DatePicker v-model="selectedDate" />
```

## 5. Thành phần: Components

### 3. DatePicker



## 5. Thành phần: Components

### 3. DatePicker

#### Props

Tên	Kiểu	Sự miêu tả
date	Date	Lấy hoặc gán ngày hoàn chỉnh.
minDate	Date	Lấy hoặc gán ngày sớm nhất có thể để chọn.
maxDate	Date	Lấy hoặc gán ngày mới nhất có thể để chọn.
day	Number	Lấy hoặc gán ngày.
month	Number	Lấy hoặc gán tháng.
year	Number	Lấy hoặc gán năm.

#### Event

Tên	Sự miêu tả
dateChange	Xảy ra khi chọn thay đổi ngày

#### Thành phần Native

Android	iOS
<code>android.widget.DatePicker</code>	<code>UIDatePicker</code>

## 5. Thành phần: Components

### 4. Frame

<Frame> là một thành phần giao diện người dùng được sử dụng để hiển thị các thành phần <Page>. Mỗi ứng dụng cần ít nhất một thành phần single <Frame>, thường được đặt làm thành phần gốc.

#### A single root Frame

Nếu bạn chuyển từ nativescript 3.x và muốn giữ nguyên trạng thái cũ, đoạn mã sau sẽ tạo khung gốc và hiển thị trang mặc định của bạn.

```
new Vue({  
  render: h => h('Frame', [ h(HomePageComponent) ])  
})
```



## 5. Thành phần: Components

### 4. Frame

#### Multiple Frame

Nếu bạn cần tạo nhiều khung, bạn có thể wrap chúng trong một bố cục, chẳng hạn như nếu bạn muốn có 2 khung cạnh nhau

```
<GridLayout columns="*, *">  
  <Frame col="0"/>  
  <Frame col="1"/>  
</GridLayout>
```

#### A frame with a default page

```
<Frame>  
  <Page>  
    <ActionBar title="Default Page Title" />  
    <GridLayout>  
      <Label text="Default Page Content" />  
    </GridLayout>  
  </Page>  
</Frame>
```

## 5. Thành phần: Components

### 4. Frame

A frame with a default page from an external component

```
<Frame>  
  <Page>  
    <Home />  
  </Page>  
</Frame>
```

```
import Home from './Home'  
  
export default {  
  components: {  
    Home  
  }  
}
```

## 5. Thành phần: Components

### 4. Frame

#### Thành phần Native

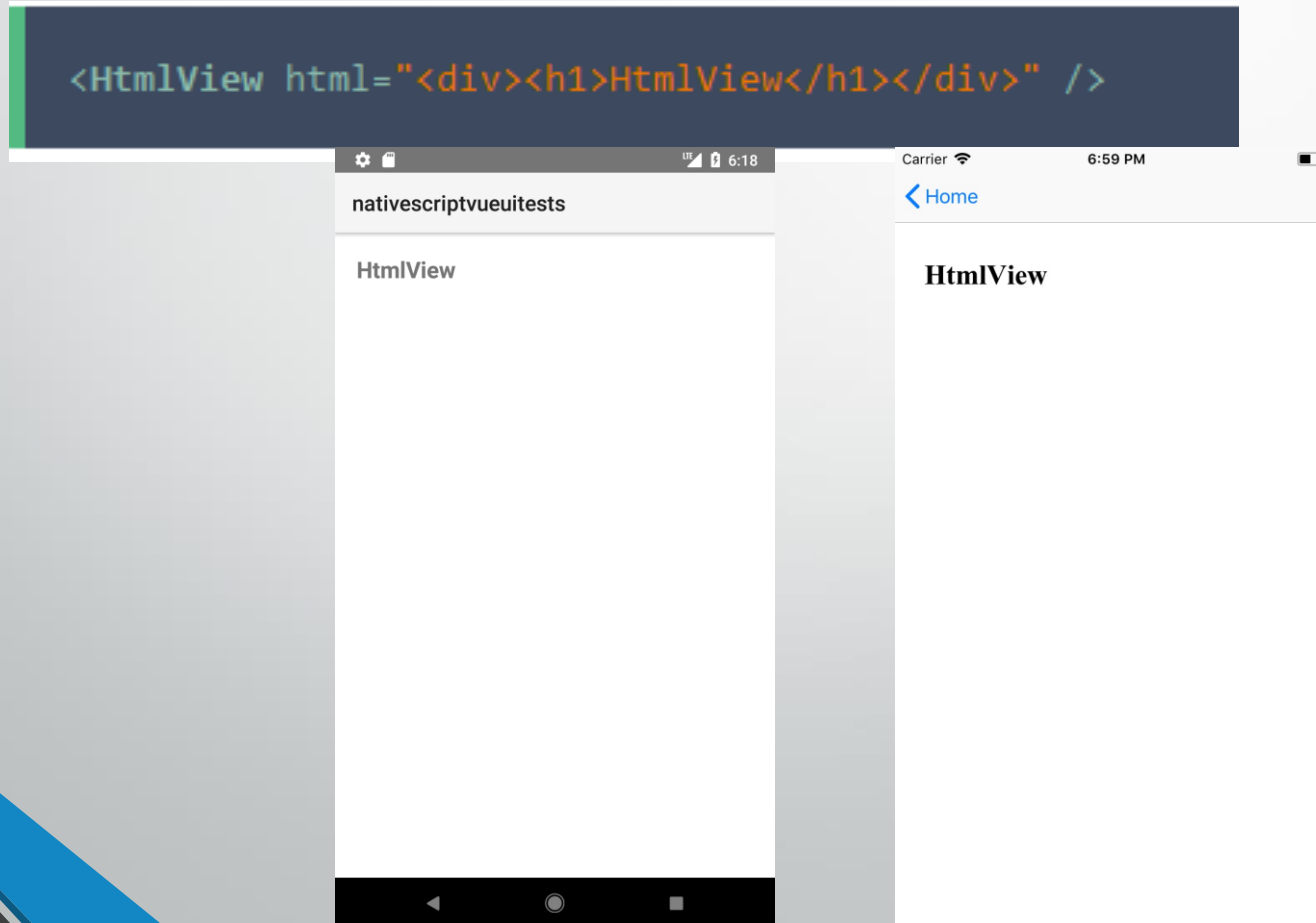
Android	iOS
<code>org.nativescript.widgets.ContentLayout</code>	<code>UINavigationController</code>

## 5. Thành phần: Components

### 5. HtmlView

<HtmlView> là một thành phần giao diện người dùng cho phép bạn hiển thị nội dung HTML tĩnh.

```
<HtmlView html="<div><h1>HtmlView</h1></div>" />
```



## 5. Thành phần: Components

### 5. HtmlView

<HtmlView> là một thành phần giao diện người dùng cho phép bạn hiển thị nội dung HTML tĩnh.

#### Props

Tên	Kiểu	Sự miêu tả
<code>html</code>	<code>String</code>	Nội dung HTML sẽ được hiển thị.

#### Thành phần Native

Android	iOS
<code>android.widget.TextView</code>	<code>UITextView</code>

## 5. Thành phần: Components

### 6. Image

`<Image>` là một thành phần giao diện người dùng hiển thị hình ảnh từ nguồn ảnh hoặc từ một URL.

**Hiển thị một hình ảnh liên quan đến thư mục app**

```
<Image src="~/logo.png" stretch="none" />
```

**Hiển thị hình ảnh từ một URL**

```
<Image src="https://art.nativescript-vue.org/NativeScript-Vue-White-Green.png" stretch="none" />
```

**Hiển thị hình ảnh từ App\_Resources**

```
<Image src="res://icon" stretch="none" />
```

## 5. Thành phần: Components

### 6. Image

#### Hiển thị image base64-encoded

```
<Image src="data:image/png;base64,iVBORw..." stretch="none" />
```

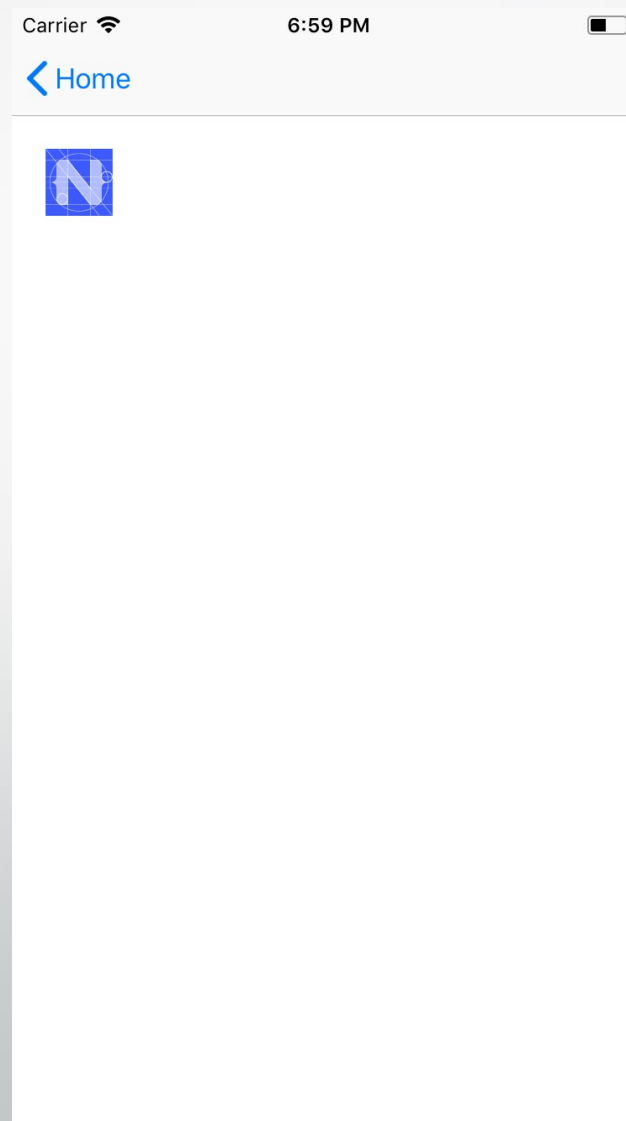
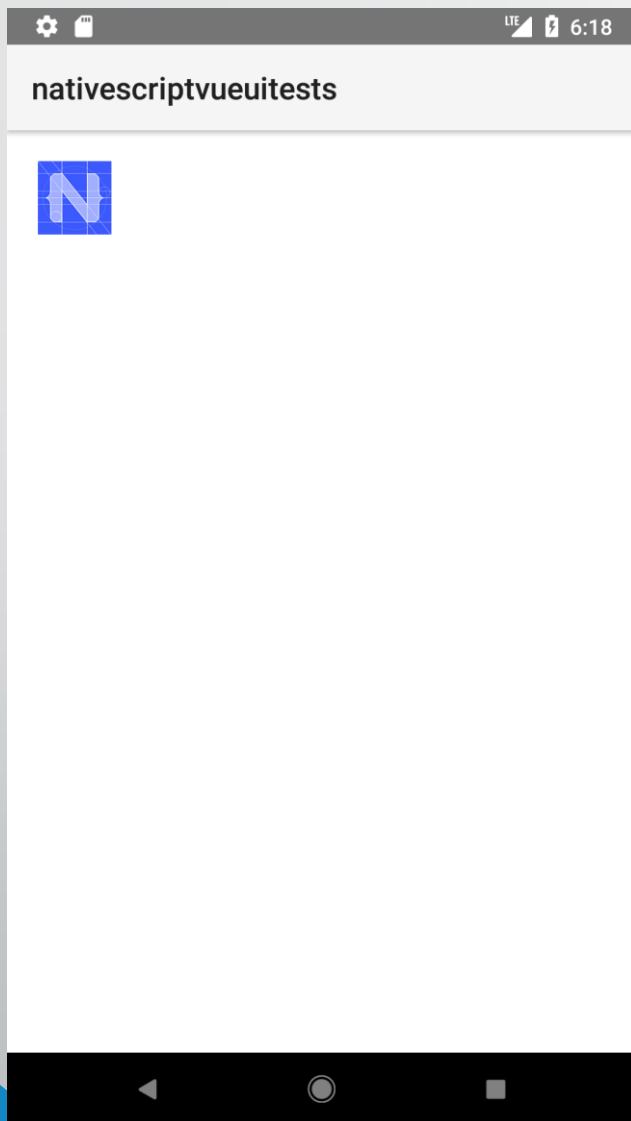
#### Hiển thị hình ảnh có biểu tượng phông chữ trong {N} 6.2+

Trong NativeScript-Vue, **.decode** bắt buộc phân tích cú pháp các thuộc tính có các thực thể HTML trong đó.

```
<Image src.decode="font://&#xf004;" class="fas" />
```

## 5. Thành phần: Components

### 6. Image





## 5. Thành phần: Components

### 6. Image

Tên	Kiểu	Sự miêu tả
src	String hoặc là <b>ImageSource</b>	Gán hoặc lấy nguồn của hình ảnh dưới dạng URL hoặc nguồn hình ảnh. Nếu bạn sử dụng giao thức font: // icon mới trong {N} 6.2, hãy đảm bảo bạn thêm .decode vào tên của thuộc tính - ví dụ:src.decode="font://&#xf004;"
imageSource	<b>ImageSource</b>	Gán và lấy nguồn của hình ảnh
tintColor	Color	(Thuộc tính kiểu) Đặt màu sắc cho ảnh mẫu.
stretch	Stretch	(Thuộc tính kiểu) Nhận hoặc đặt cách hình ảnh được thay đổi kích thước để lấp đầy không gian được phân bổ của nó. Giá trị hợp lệ: none, aspectFill, aspectFit, hoặc fill.
loadMode		Lấy hoặc gán tải hình ảnh lên trên hệ thống file cục bộ. Giá trị hợp lệ: sync hoặc async. Giá trị mặc định: async.

### Thành phần Native

Android	iOS
<code>android.widget.ImageView</code>	<code>UIImageView</code>

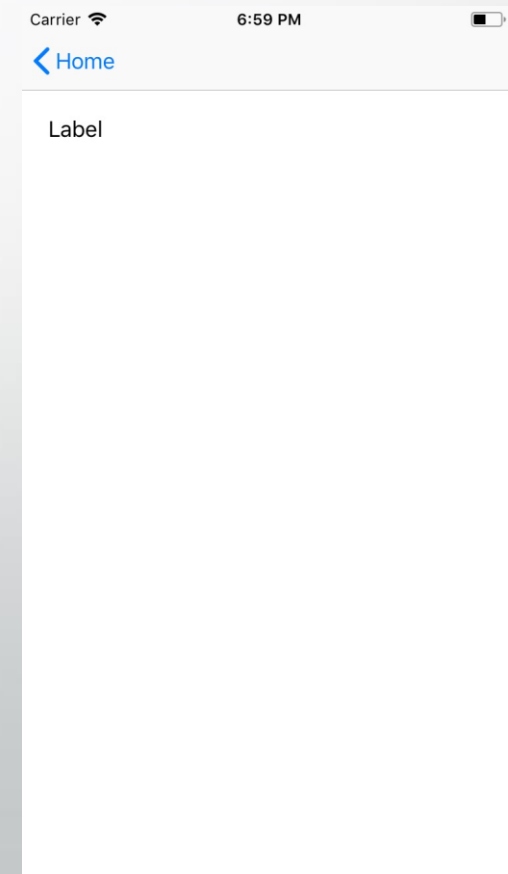
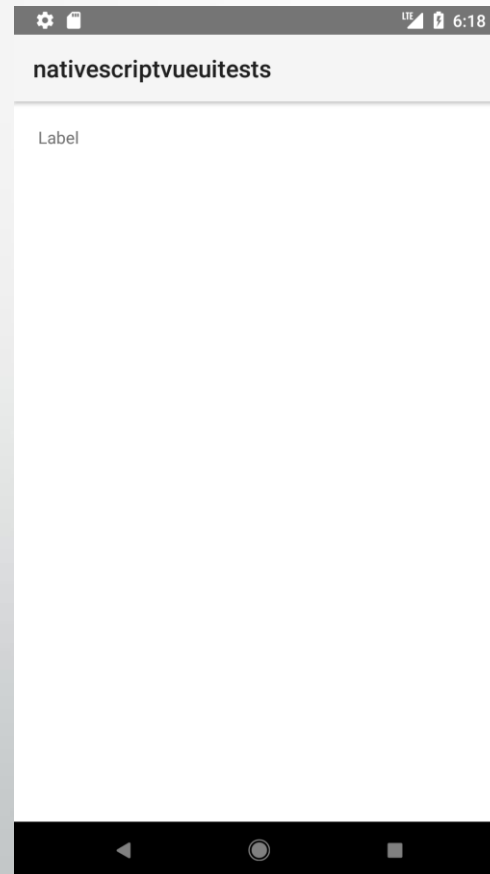
## 5. Thành phần: Components

### 7. Label

<Label> là một thành phần giao diện người dùng hiển thị văn bản chỉ đọc.

**Chú ý:** Label này không giống như Lable trong HTML

```
<Label text="Label" />
```



## 5. Thành phần: Components

### 7. Label

#### Tạo kiểu cho nhãn

Nếu bạn cần tạo kiểu cho các phần của văn bản, bạn có thể sử dụng kết hợp các phần tử `FormattedString` và `Span`.

```
<Label textWrap="true">
  <FormattedString>
    <Span text="This text has a " />
    <Span text="red " style="color: red" />
    <Span text="piece of text. " />
    <Span text="Also, this bit is italic, " fontStyle="italic" />
    <Span text="and this bit is bold." fontWeight="bold" />
  </FormattedString>
</Label>
```

## 5. Thành phần: Components

### 7. Label

#### Props

Tên	Kiểu	Sự miêu tả
text	String	Gán và lấy text của nhãn
textWrap	Boolean	Giá trị mặc định: false.

#### Thành phần Native

Android	iOS
<code>android.widget.TextView</code>	<code>UILabel</code>

## 5. Thành phần: Components

### 8. ListPicker

<ListPicker> là một thành phần giao diện người dùng cho phép người dùng chọn một giá trị từ danh sách được thiết lập trước.

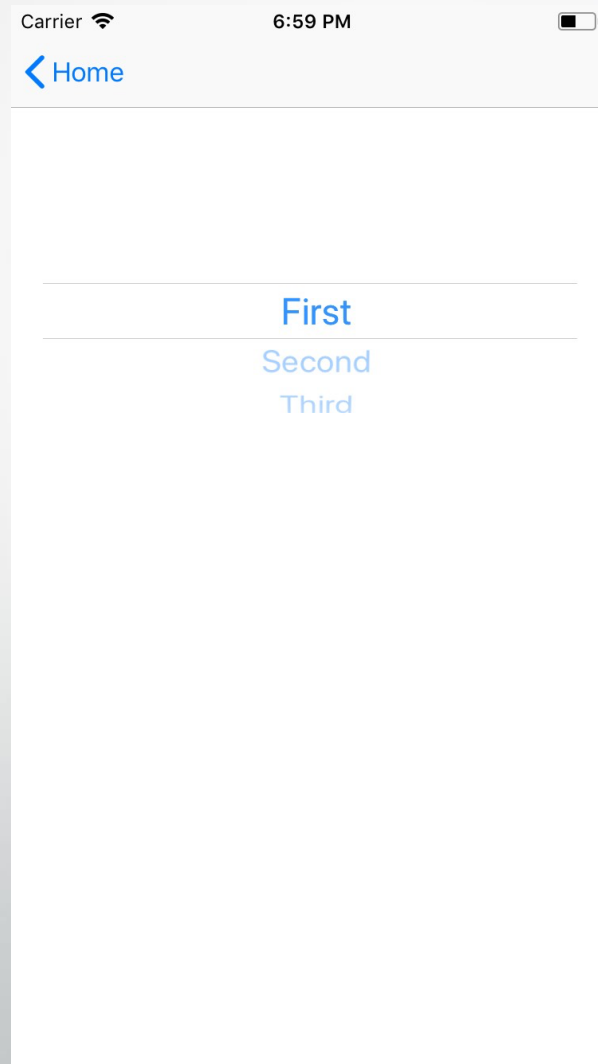
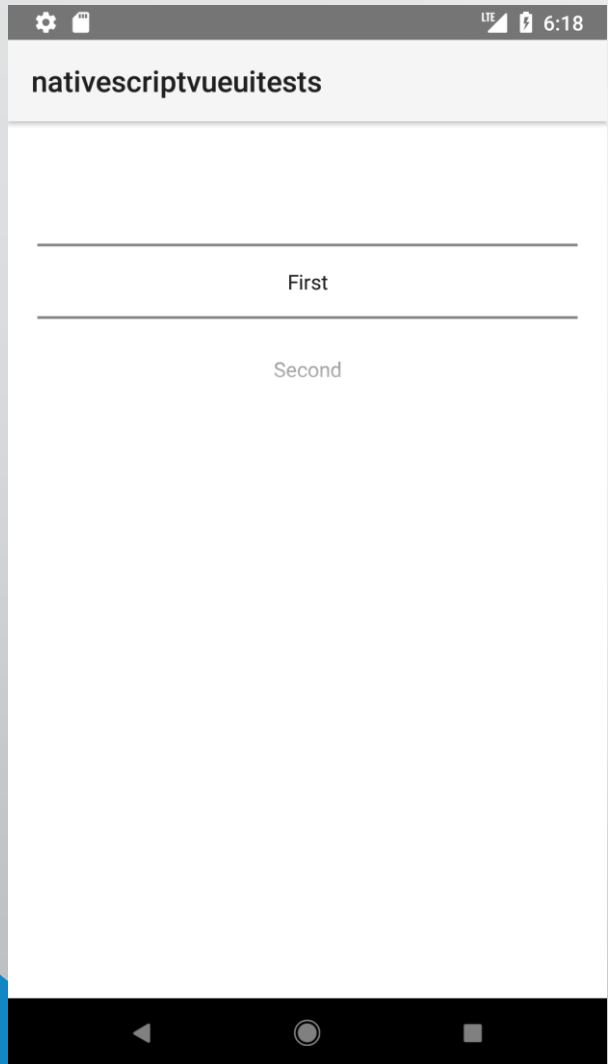
```
<ListPicker :items="listOfItems" selectedIndex="0"  
  @selectedIndexChange="selectedIndexChanged" />
```

<ListPicker> cung cấp liên kết dữ liệu hai chiều bằng cách sử dụng **v-model**.

```
<ListPicker :items="listOfItems" v-model="selectedItem" />
```

## 5. Thành phần: Components

### 8. ListPicker



## 5. Thành phần: Components

### 8. ListPicker

Tên	Kiểu	Sự miêu tả
items	Array<String>	Nhận hoặc đặt các mục được hiển thị dưới dạng tùy chọn trong bộ chọn danh sách.
selectedIndex	Number	Nhận hoặc đặt chỉ mục của mục hiện được chọn.

### Event

Tên	Sự miêu tả
selectedIndexChange	Xảy ra khi chỉ mục được thay đổi. Chỉ mục mới có thể được truy xuất thông qua \$event.value.

### Thành phần Native

Android	iOS
<code>android.widget.NumberPicker</code>	<code>UIPickerView</code>

## 5. Thành phần: Components

### 9. ListView

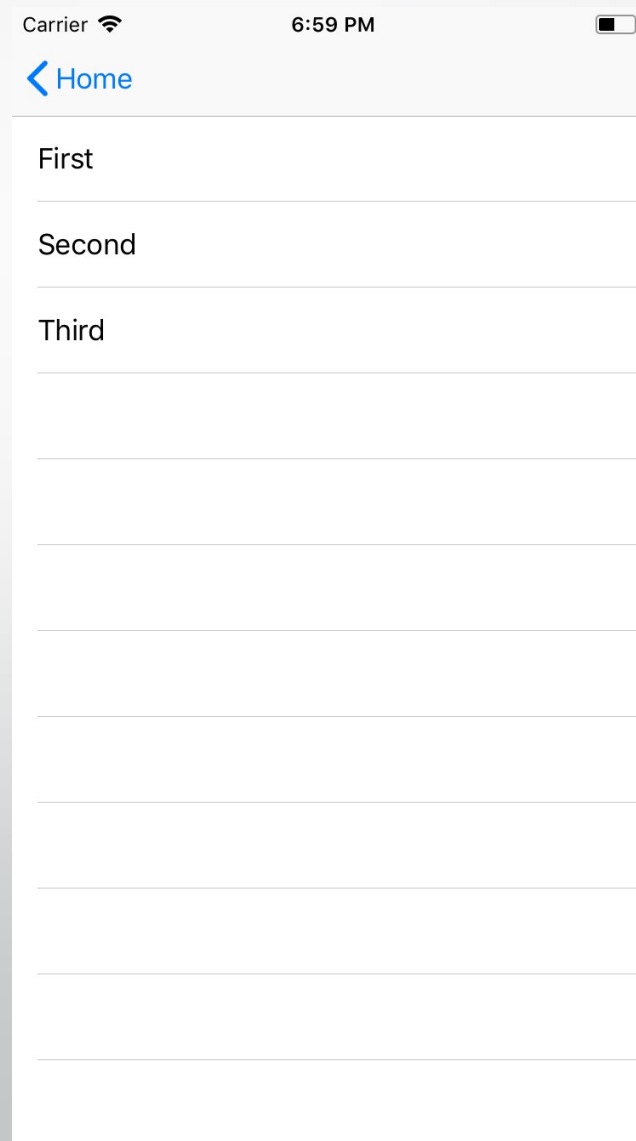
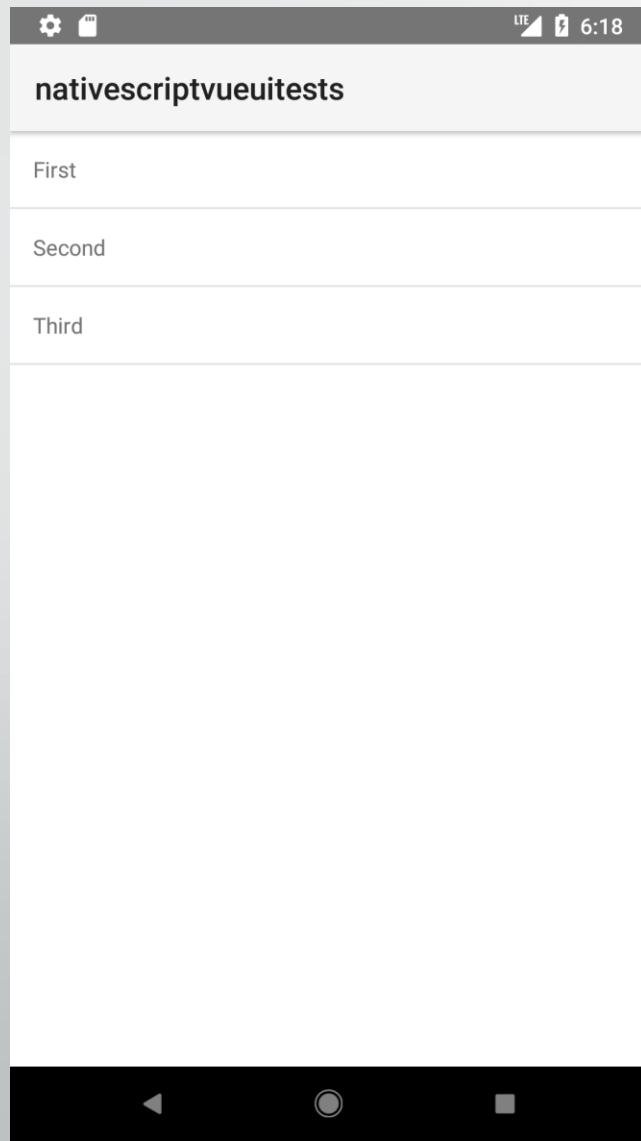
<ListView> là một thành phần giao diện người dùng hiển thị các mục trong danh sách cuộn theo chiều dọc. Để danh sách hiển thị các mục riêng lẻ, bạn có thể sử dụng thành phần <v-template>.

```
<ListView for="item in listOfItems" @itemTap="onItemTap">
  <v-template>
    <!-- Shows the list item label in the default color and style. -->
    <Label :text="item.text" />
  </v-template>
</ListView>
```



## 5. Thành phần: Components

### 9. ListView



## 5. Thành phần: Components

### 9. ListView

#### Sử dụng <ListView> với nhiều khối <v-template>

Thành phần v-template được sử dụng để xác định cách mỗi mục danh sách được hiển thị trên màn hình.

Nếu bạn cần trực quan hóa một hoặc nhiều mục danh sách khác nhau, bạn có thể thêm khối <v-template> và điều kiện sử dụng. Bạn có thể có nhiều khối <v-template> nếu cần trong một khối <ListView>.

```
<ListView for="item in listOfItems" @itemTap="onItemTap">
  <v-template>
    <Label :text="item.text" />
  </v-template>

  <v-template if="$odd">
    <!-- For items with an odd index, shows the label in red. -->
    <Label :text="item.text" color="red" />
  </v-template>
</ListView>
```

## 5. Thành phần: Components

### 9. ListView

Khi bạn tạo điều kiện cho `<v-template>`, bạn có thể sử dụng biểu thức JavaScript hợp lệ với các biến sau:

**\$index-** chỉ mục của mục hiện tại

**\$even- true** nếu chỉ mục của mục hiện tại là chẵn

**\$odd- true** nếu chỉ mục của mục hiện tại là lẻ

**item-** mục của danh sách (tên tương ứng với trình lặp trong thuộc tính `for`). Ví dụ `if="item.text == 'danger'"`

#### Một lưu ý quan trọng về v-for:

`<ListView>` không lặp lại các mục trong danh sách như khi sử dụng một vòng lặp `v-for`. Thay vào đó, `<ListView>` chỉ tạo các chế độ xem cần thiết để hiển thị các mục đang hiển thị trên màn hình và sử dụng lại các chế độ xem đã ở ngoài màn hình khi cuộn. Khái niệm này được gọi là tái chế lượt xem và thường được sử dụng trong các ứng dụng dành cho thiết bị di động để cải thiện hiệu suất.

## 5. Thành phần: Components

### 9. ListView

#### Một lưu ý quan trọng về v-for:

Điều này rất quan trọng, vì bạn không nên sử dụng key các thuộc tính trong v-template của mình, vì chúng sẽ buộc ListView phải tạo lại các khung nhìn và ngăn chế độ xem tái chế lại từ hoạt động bình thường.

Để sử dụng nhiều sự kiện lắng nghe trong ListView, bạn có thể chuyển mục hiện tại từ sự kiện nghe với @tap="onTap(item, \$event)".

Nếu bạn chỉ cần xử lý các lần nhấn trên toàn bộ ô, bạn có thể sử dụng sự kiện onTap chứa chỉ mục của mục được nhấn và mục thực tế từ danh sách.

```
onItemTap(event) {  
  console.log(event.index)  
  console.log(event.item)  
}
```

## 5. Thành phần: Components

### 9. ListView

**LƯU Ý:** Nếu v-for được sử dụng trên một <ListView> một cảnh báo sẽ được in ra trên bảng điều khiển và nó sẽ được chuyển đổi thành thuộc tính for

Tên	Kiểu	Sự miêu tả
for	String	<ul style="list-style-type: none"><li>•Cung cấp biểu thức để lặp qua các mục. Ví dụ:item in listOfItems</li><li>•(item, index) in listOfItems</li><li>•item in [1, 2, 3, 4, 5]</li></ul>
items	Array<any>	Một mảng các mục được hiển thị trong <ListView>. <b>Thuộc tính này chỉ dành cho mục đích sử dụng nâng cao. Sử dụng thuộc tính for thay thế.</b>
separatorColor	Color	Đặt màu đường phân cách. Gán transparent để xóa nó.

## 5. Thành phần: Components

### 9. ListView

#### Event

Tên	Sự miêu tả
itemTap	Xảy ra khi một mục trong mục <ListView> được khai thác. Để truy cập vào mục đã khai thác, hãy sử dụng event.item.

#### Method

Tên	Sự miêu tả
<code>refresh()</code>	Buộc <ListView> tải lại tất cả các mục của nó.

#### Thành phần Native

Android	iOS
<code>android.widget.ListView</code>	<code>UITableView</code>

## 5. Thành phần: Components

### 10. Page

<Page> là một thành phần giao diện người dùng đại diện cho màn hình ứng dụng. Ứng dụng NativeScript thường bao gồm một hoặc nhiều <Page> nội dung bao bọc như một <ActionBar> và các tiện ích giao diện người dùng khác.

#### A single Page

```
<Page>
  <ActionBar title="My App" />
  <GridLayout>
    <Label text="My Content"/>
  </GridLayout>
</Page>
```

## 5. Thành phần: Components

### 10. Page

Sử dụng sự kiện loaded để kích hoạt các thay đổi giao diện người dùng

Ví dụ để thực hiện các thay đổi giao diện người dùng sau khi trang được tải. Cách đề xuất để làm điều đó là sử dụng sự kiện loaded, được kích hoạt bởi NativeScript khi trang được tải đầy đủ:

```
<Page @loaded="greet">
  <ActionBar title="My App" />
  <GridLayout>
    <Label text="My Content"/>
  </GridLayout>
</Page>
```

```
export default {
  methods: {
    greet() {
      alert('Hello!').then(() => {
        console.log('Dialog closed')
      })
    }
  }
}
```



## 5. Thành phần: Components

### 10. Page

**Chú ý:** Các nhà phát triển đến từ background thường sẽ tiếp cận với vòng đời mounted hook mà Vue cung cấp, tuy nhiên trong NativeScript, ứng dụng và các phần tử nhất định có thể chưa được tải khi mounted hook được thực thi, do đó, một số hành động nhất định như cảnh báo, hộp thoại, điều hướng, v.v. không thể bên trong mountedhook. Để khắc phục hạn chế này, sự kiện loaded có thể được sử dụng, sự kiện này chỉ kích hoạt sau khi ứng dụng sẵn sàng. Trong trường hợp này, đang sử dụng sự kiện loaded của phần tử <Page>, nhưng sự kiện này có sẵn cho tất cả các phần tử NativeScript.

## 5. Thành phần: Components

### 10. Page

#### Props

Tên	Kiểu	Sự miêu tả
actionBarHidden	Boolean	Hiển thị hoặc ẩn <ActionBar>cho trang. Giá trị mặc định: false.
backgroundSpanUnderStatusBar	Boolean	Lấy và nhận giá trị nền của trang có trải dài dưới thanh trạng thái hay không. Giá trị mặc định: false.
androidStatusBarBackground	Color	(Chỉ dành cho Android) Lấy hoặc gián màu của thanh trạng thái trên thiết bị Android.
enableSwipeBackNavigation	Boolean	(Chỉ dành cho iOS) Lấy hoặc gán trang có thể được vuốt lại trên iOS hay không. Giá trị mặc định: true.
statusBarStyle	String	Lấy hoặc gán kiểu của thanh trạng thái. Giá trị hợp lệ: light, dark.

## 5. Thành phần: Components

### 10. Page

#### Event

Tên	Sự miêu tả
loaded	Xảy ra sau khi trang đã được tải.
navigatedFrom	Xảy ra sau khi ứng dụng đã điều hướng khỏi trang hiện tại.
navigatedTo	Xảy ra sau khi ứng dụng đã điều hướng đến trang hiện tại.
navigatingFrom	Xảy ra trước khi ứng dụng điều hướng khỏi trang hiện tại.
navigatingTo	Được kích hoạt trước khi ứng dụng điều hướng đến trang hiện tại.

#### Thành phần Native

Android	iOS
<code>org.nativescript.widgets.GridLayout</code>	<code>UIViewController</code>

## 5. Thành phần: Components

### 11. Placeholder

<Placeholder> cho phép bạn thêm bất kỳ widget native nào vào ứng dụng của mình. Để làm điều đó, bạn cần đặt placeholder ở đâu đó trong phân cấp giao diện người dùng, sau đó tạo và định cấu hình widget native mà bạn muốn xuất hiện ở đó. Cuối cùng, chuyển widget native của bạn vào các đối số sự kiện của sự kiện createView.

```
<Placeholder @creatingView="creatingView" />
```

### Ví dụ với TextView trong Android

```
methods: {  
  creatingView: function(args) {  
    const nativeView = new android.widget.TextView(args.context);  
    nativeView.setSingleLine(true);  
    nativeView.setEllipsize(android.text.TextUtils.TruncateAt.END);  
    nativeView.setText("Native View - Android");  
    args.view = nativeView;  
  }  
}
```

## 005. Thành phần: Components

### 11. Placeholder

Ví dụ với UILabel trong iOS

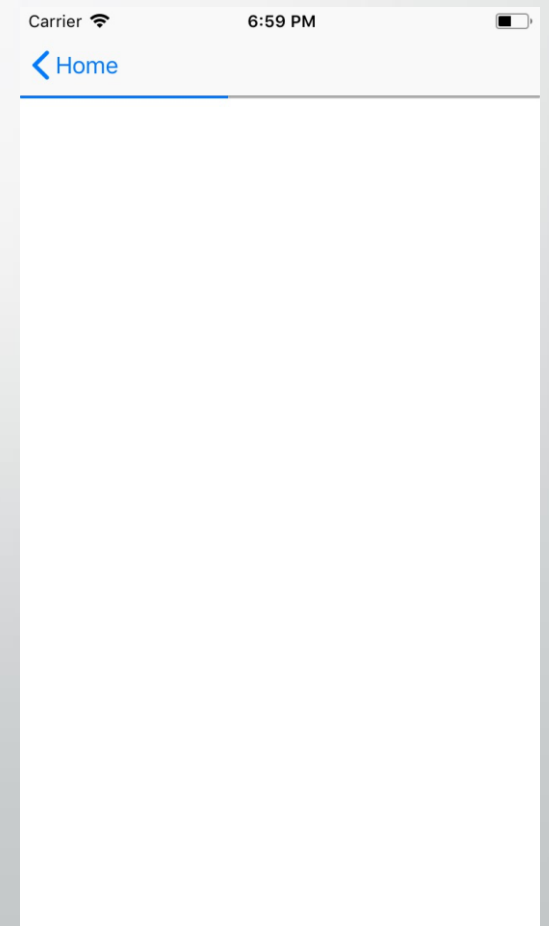
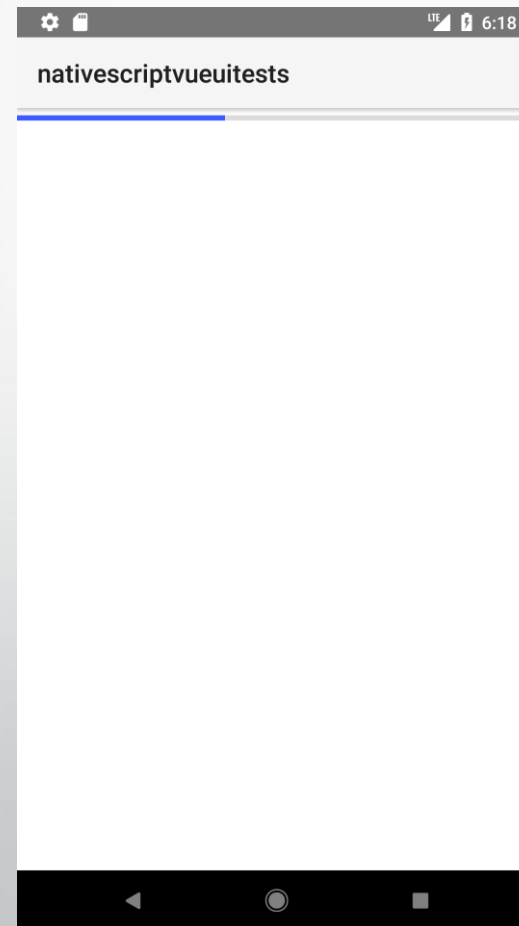
```
methods: {  
  creatingView: function(args) {  
    const nativeView = new UILabel();  
    nativeView.text = "Native View - iOS";  
    args.view = nativeView;  
  }  
}
```

## 5. Thành phần: Components

### 12. Progress

<Progress> là một thành phần giao diện người dùng hiển thị một thanh để chỉ ra tiến trình của một tác vụ.

```
<Progress :value="currentProgress" />
```



## 5. Thành phần: Components

### 12. Progress

#### Props

Tên	Kiểu	Sự miêu tả
value	Number	Lấy hoặc gán giá trị hiện tại của thanh tiến trình. Phải nằm trong phạm vi từ 0 đến max <b>Value</b> .
maxValue	Number	Lấy và gán giá trị lớn nhất của thanh tiến trình. Giá trị mặc định: 100.

#### Event

Tên	Sự miêu tả
valueChange	Xảy ra khi giá trị thuộc tính thay đổi

#### Thành phần Native

Android	iOS
<code>android.widget.ProgressBar</code> (indeterminate = false)	<code>UIProgressView</code>

## 5. Thành phần: Components

### 13. ScrollView

<ScrollView> là một thành phần giao diện người dùng hiển thị một vùng nội dung có thể cuộn. Nội dung có thể được cuộn theo chiều dọc hoặc chiều ngang.

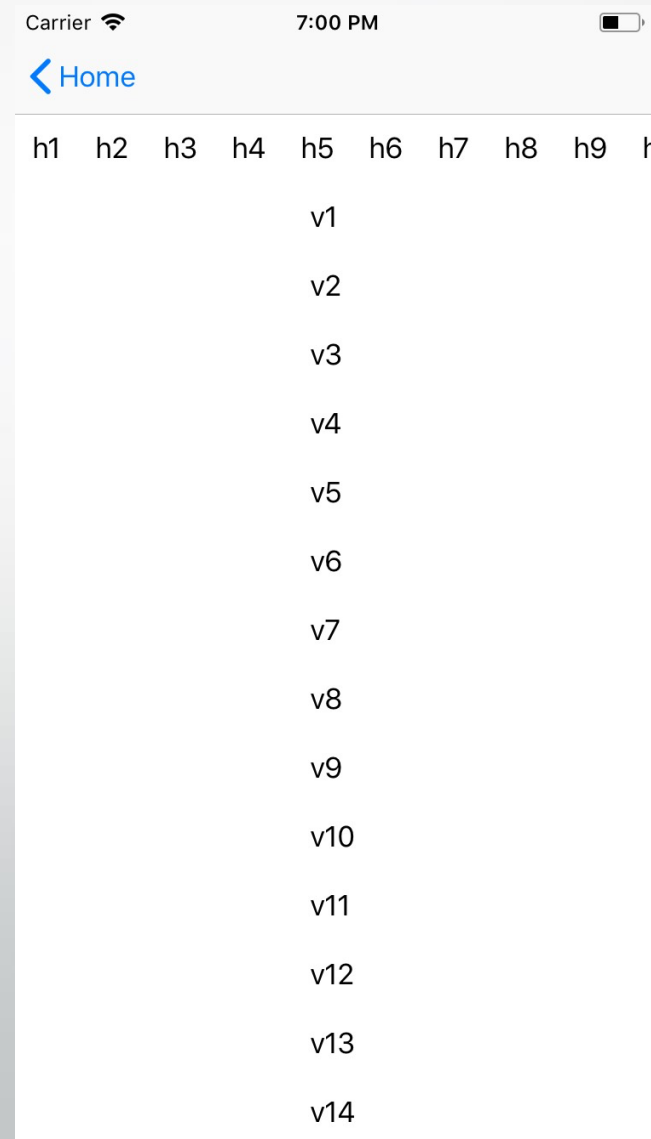
Điều quan trọng cần lưu ý là <ScrollView> là mở rộng **ContentView**, vì vậy nó chỉ có thể có một phần tử con duy nhất.

```
<ScrollView orientation="horizontal">
  <StackLayout orientation="horizontal">
    <Label text="this" />
    <Label text="text" />
    <Label text="scrolls" />
    <Label text="horizontally" />
    <Label text="if necessary" />
  </StackLayout>
</ScrollView>
```



## 5. Thành phần: Components

### 13. ScrollView



## 5. Thành phần: Components

### 13. ScrollView

Tên	kiểu	sự miêu tả
orientation	String	Lấy hoặc gán hướng nội dung có thể được cuộn: horizontal hoặc vertical. Giá trị mặc định: vertical.
scrollBarIndicatorVisible	Boolean	Chỉ định xem thanh cuộn có hiển thị hay không. Giá trị mặc định: true.

### Event

Tên	Sự miêu tả
scroll	Sự kiện xảy ra khi cuộn

### Thành phần Native

Android	iOS
<code>android.view</code>	<code>UIScrollView</code>

## 5. Thành phần: Components

### 14. SearchBar

**<SearchBar>** là một thành phần giao diện người dùng cung cấp giao diện người dùng để nhập các truy vấn tìm kiếm và gửi yêu cầu đến nhà cung cấp dịch vụ tìm kiếm.

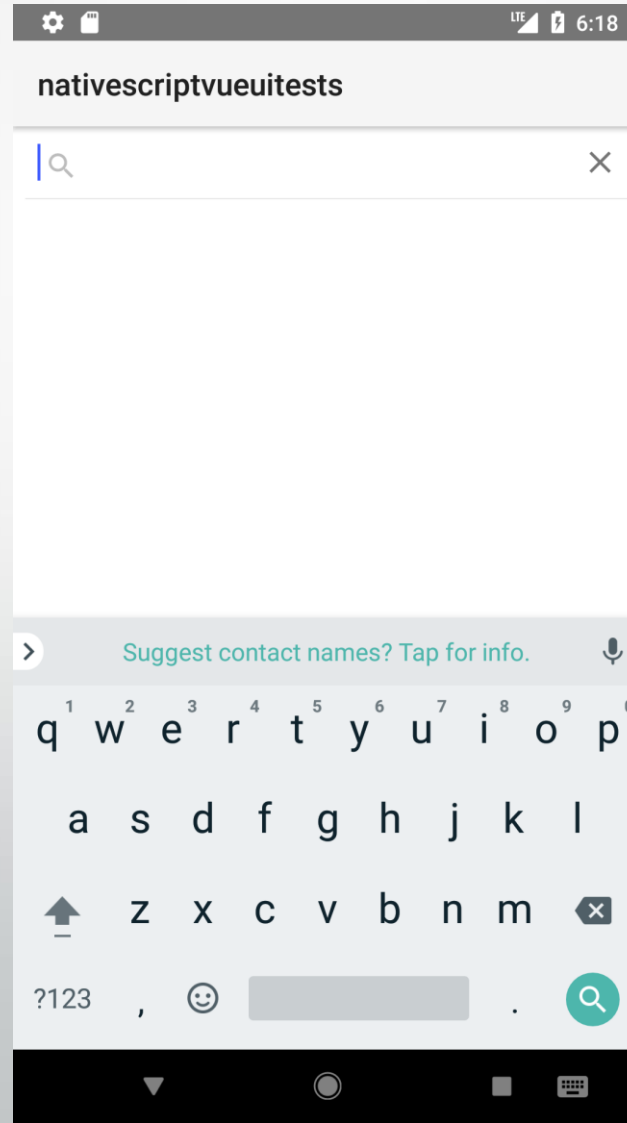
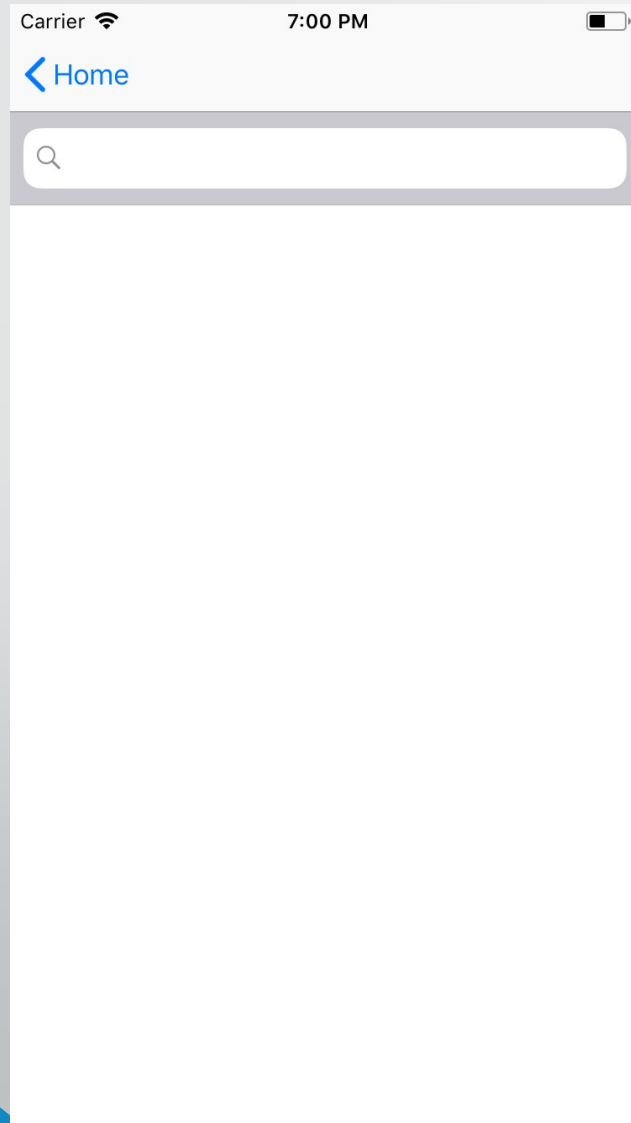
```
<SearchBar hint="Search hint" :text="searchPhrase" @textChange="onTextChanged" @submit="onSubmit" />
```

**<SearchBar>** cung cấp liên kết dữ liệu hai chiều bằng cách sử dụng v-model.

```
<SearchBar v-model="searchQuery" />
```

## 5. Thành phần: Components

### 14. SearchBar



## 5. Thành phần: Components

### 14. SearchBar

Tên	Kiểu	Sự miêu tả
hint	String	Gán và lấy text placeholder cho vùng nhập liệu.
text	String	Gán và lấy giá trị của truy vấn tìm kiếm.
textFieldBackgroundColor	Color	Gán hoặc lấy giá trị màu nền của vùng đầu vào.
textFieldHintColor	Color	Gán hoặc lấy màu của text placeholder

### Event

Tên	sự miêu tả
textChange	Xảy ra khi văn bản được thay đổi.
submit	Xảy ra khi đầu vào tìm kiếm được gửi.
clear	Xảy ra khi đầu vào tìm kiếm hiện tại được xóa thông qua nút <b>X</b> trong vùng nhập.

### Thành phần Native

Android	iOS
<code>android.widget.SearchView</code>	<code>UISearchBar</code>

## 5. Thành phần: Components

### 15. SegmentedBar

<SegmentedBar> là một thành phần thanh giao diện người dùng hiển thị một tập hợp các nút rời rạc để lựa chọn. Có thể hiển thị văn bản hoặc hình ảnh.

Trái ngược với <TabView>:

- Vị trí của <SegmentedBar> không cố định.
- Bạn có thể đặt và tạo kiểu cho nó nếu cần trên trang hoặc thêm bên trong các phần tử ứng dụng, chẳng hạn như menu bánh hamburger.
- Bạn cần xử lý riêng nội dung hiển thị sau khi chọn.

```
<SegmentedBar>  
  <SegmentedBarItem title="First" />  
  <SegmentedBarItem title="Second" />  
  <SegmentedBarItem title="Third" />  
</SegmentedBar>
```

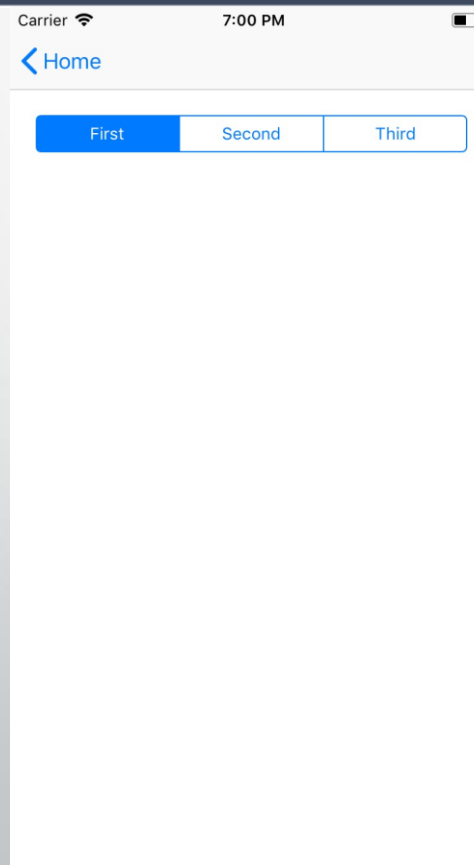
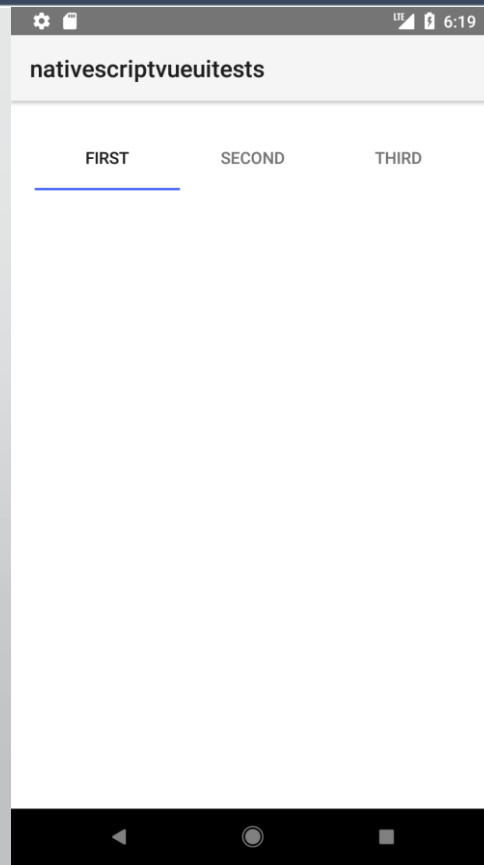
```
<SegmentedBar :items="listOfItems" selectedIndex="0"  
  @selectedIndexChange="onSelectedIndexChange" />
```

## 5. Thành phần: Components

### 15. SegmentedBar

<SegmentedBar> cung cấp liên kết dữ liệu hai chiều bằng cách sử dụng v-model.

```
<SegmentedBar :items="listOfItems" v-model="selectedItem" />
```



## 5. Thành phần: Components

### 15. SegmentedBar

#### Props

Tên	Kiểu	Sự miêu tả
items	Array<SegmentedBarItem>	Một mảng các mục sẽ được hiển thị trong thanh được phân đoạn. Đại diện cho các nhãn nút hoặc biểu tượng của thanh được phân đoạn. Mảng phải được tạo trước.
selectedIndex	Number	Gán hoặc lấy chỉ mục được chọn
selectedBackgroundColor	Color	(Thuộc tính kiểu) Gán hoặc lấy màu nền của mục đã chọn. Để đặt màu nền của toàn bộ thanh, hãy sử dụng backgroundColor.

#### Event

Tên	Sự miêu tả
selectedIndexChange	Xảy ra khi một mục trên thanh được nhấn

#### Thành phần Native

Android	iOS
<code>android.widget.TabHost</code>	<code>UISegmentedControl</code>



## 5. Thành phần: Components

### 16. Slider

<Slider> là một thành phần giao diện người dùng cung cấp điều khiển thanh trượt để chọn các giá trị trong một phạm vi số được chỉ định.

```
<Slider value="88" @valueChange="onValueChanged" />
```

<Slider> cung cấp liên kết dữ liệu hai chiều bằng cách sử dụng v-model:

```
<Slider v-model="value" />
```



## 5. Thành phần: Components

### 16. Slider

#### Props

Tên	Kiểu	Sự miêu tả
value	Number	Lấy hoặc gán giá trị được chọn của thanh trượt. Giá trị mặc định: 0.
minValue	Number	Lấy hoặc gán giá trị nhỏ nhất của thanh trượt. Giá trị mặc định: 0.
maxValue	Number	Lấy hoặc gán giá trị lớn nhất của thanh trượt. Giá trị mặc định: 100.

#### Event:

Tên	Sự miêu tả
valueChange	Xảy Ra khi thay đổi giá trị của thanh trượt

#### Thành phần Native

Android	iOS
<code>android.widget.SeekBar</code>	<code>UISlider</code>

## 5. Thành phần: Components

### 17. Switch

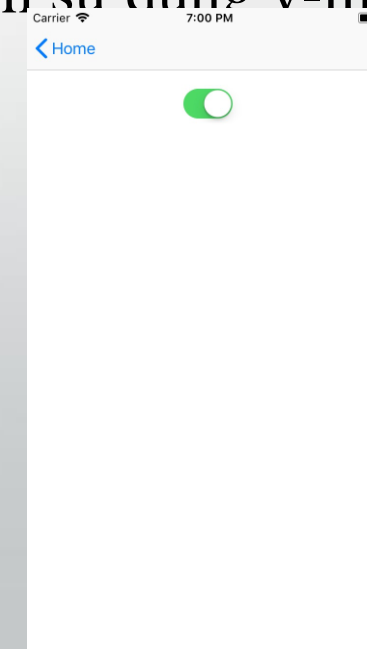
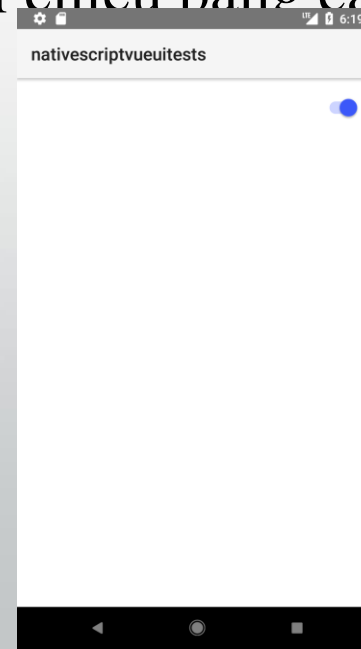
<Switch> là một thành phần giao diện người dùng cho phép người dùng chuyển đổi giữa hai trạng thái.

Trạng thái mặc định là false hoặc OFF.

```
<Switch checked="true" />
```

<Switch> cung cấp liên kết dữ liệu hai chiều bằng cách sử dụng v-model.

```
<Switch v-model="itemEnabled" />
```



## 5. Thành phần: Components

### 17. Switch

#### Props:

Tên	Kiểu	Sự miêu tả
checked	Boolean	Gán hoặc lấy giá trị khi chọn Switch. Giá trị mặc định: false.

#### Event:

Tên	Sự miêu tả
checkedChange	Xảy ra khi switch thay đổi

#### Thành phần Native:

Android	iOS
<code>android.widget.Switch</code>	<code>UISwitch</code>

## 5. Thành phần: Components

### 18. TabView

**Lưu ý :** NativeScript 6 đã giới thiệu hai thành phần điều hướng tab mới <BottomNavigation> và <Tabs>. Đây là những lựa chọn thay thế mới và tốt hơn cho thành phần hiện có <TabView>.

<TabView> là một thành phần điều hướng hiển thị nội dung được nhóm thành các tab và cho phép người dùng chuyển đổi giữa các tab.

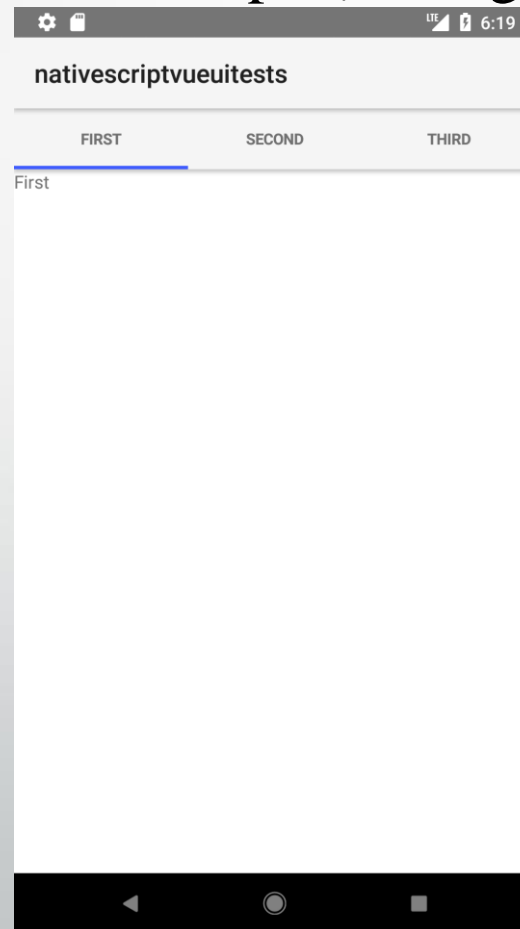
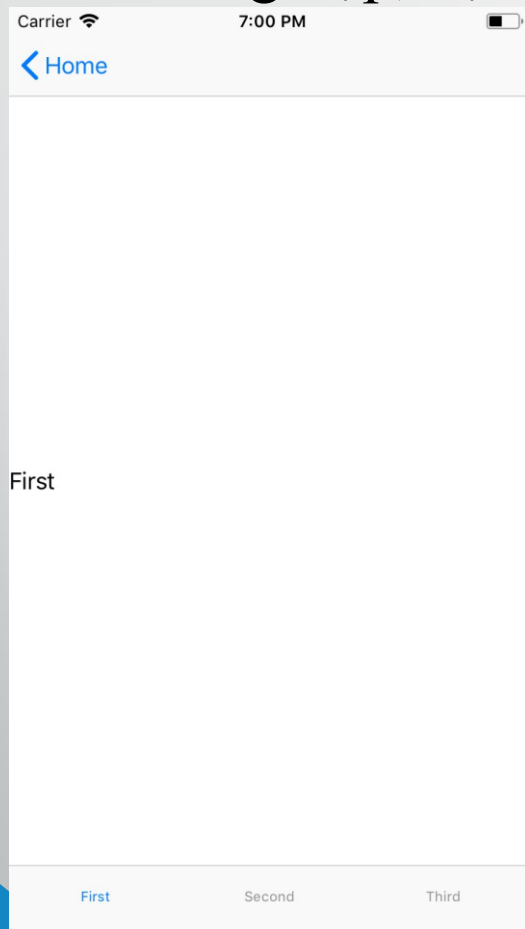
```
<TabView :selectedIndex="selectedIndex" @selectedIndexChange="indexChange">
  <TabViewItem title="Tab 1">
    <Label text="Content for Tab 1" />
  </TabViewItem>
  <TabViewItem title="Tab 2">
    <Label text="Content for Tab 2" />
  </TabViewItem>
</TabView>
```

```
methods: {
  indexChange: function(args) {
    let newIndex = args.value
    console.log('Current tab index: ' + newIndex)
  }
}
```

## 5. Thành phần: Components

### 18. TabView

**LƯU Ý:** Hiện tại, TabViewItem mong đợi một phần tử con duy nhất. Trong hầu hết các trường hợp, bạn có thể muốn wrap nội dung của mình trong một bố cục.



## 5. Thành phần: Components

### 18. TabView

#### Thêm biểu tượng vào tab

```
<TabView :selectedIndex="selectedIndex" iosIconRenderingMode="alwaysOriginal">
  <TabViewItem title="Tab 1" iconSource="~/images/icon.png">
    <Label text="Content for Tab 1" />
  </TabViewItem>
  <TabViewItem title="Tab 2" iconSource="~/images/icon.png">
    <Label text="Content for Tab 2" />
  </TabViewItem>
</TabView>
```

**LƯU Ý:** Bạn có thể sử dụng hình ảnh cho biểu tượng tab thay vì biểu tượng phong chữ.

Tên	Kiểu	Sự miêu tả
selectedIndex	Number	Lấy hoặc đặt tab hiện được chọn. Mặc định là 0.
tabTextColor	Color	(Thuộc tính kiểu) Lấy hoặc đặt màu văn bản của tiêu đề tab.
tabBackgroundColor	Color	(Thuộc tính kiểu) Lấy hoặc đặt màu nền của các tab.
selectedTabTextColor	Color	(Thuộc tính kiểu) Nhận hoặc đặt màu văn bản của tiêu đề tab đã chọn.

## 5. Thành phần: Components

### 18. TabView

#### Props

Tên	Kiểu	Sự miêu tả
selectedIndex	Number	Lấy hoặc gán tab được chọn. Mặc định là 0.
tabTextColor	Color	(Thuộc tính kiểu) Lấy hoặc gán màu văn bản của tiêu đề tab.
tabBackgroundColor	Color	(Thuộc tính kiểu) Lấy hoặc gán màu nền của các tab.
selectedTabTextColor	Color	(Thuộc tính kiểu) Lấy hoặc gán màu văn bản của tiêu đề tab đã chọn.
androidTabsPosition	String	Đặt vị trí của TabView trong nền tảng Android Giá trị hợp lệ: top hoặc bottom.

#### Event

Tên	Sự miêu tả
selectedIndexChange	Xảy ra <a href="#">một đối tượng sự kiện</a> có chứa một thuộc tính newIndex với chỉ mục của tab được khai thác <TabViewItem> (và một thuộc oldIndex tính có chỉ mục

#### Thành phần Native

Android	iOS
<code>android.support.v4.view.ViewPager</code>	<code>UITabBarController</code>



## 5. Thành phần: Components

### 19. TextField

<TextField> là một thành phần đầu vào tạo ra một hộp, một dòng có thể chỉnh sửa.

<TextField> mở rộng TextBase và EditableTextBase cung cấp các thuộc tính và thêm sự kiện.

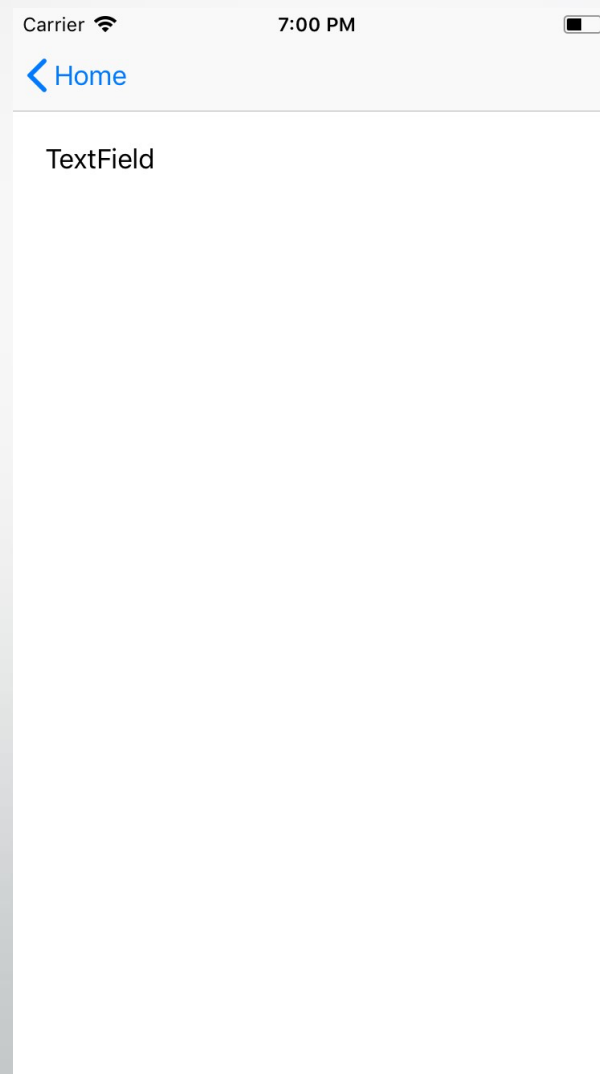
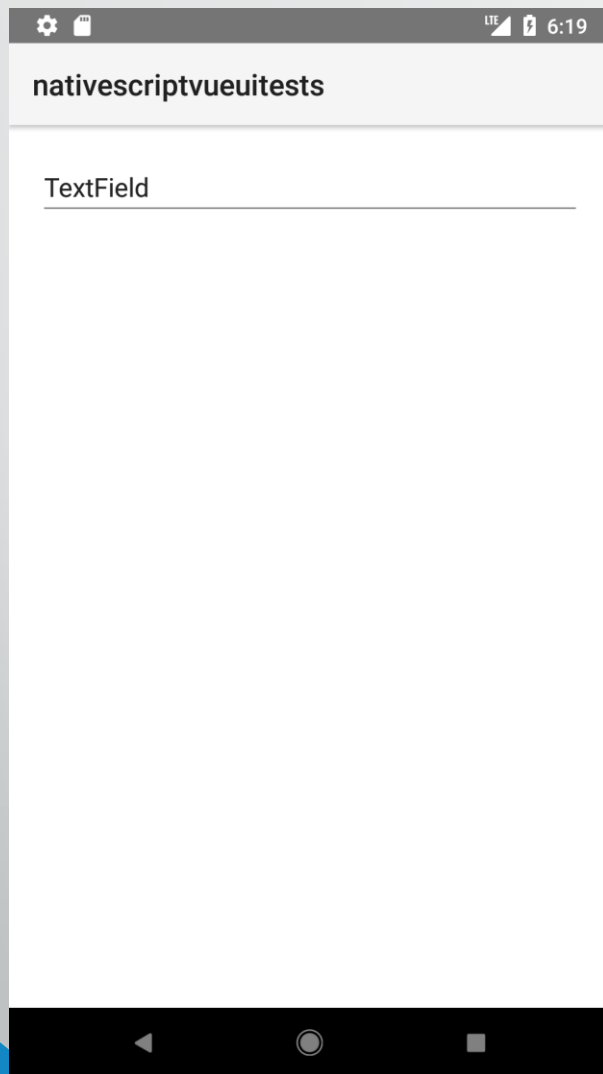
```
<TextField :text="textFieldValue" hint="Enter text..." />
```

<TextField> cung cấp liên kết dữ liệu hai chiều bằng cách sử dụng v-model.

```
<TextField v-model="textFieldValue" />
```

## 5. Thành phần: Components

### 19. TextField



## 5. Thành phần: Components

### 19. TextField

#### Props

Tên	Kiểu	Sự miêu tả
text	String	Lấy hoặc gán giá trị cho field (trường)
hint	String	Lấy hoặc gán placeholder text
isEnabled	Boolean	Tắt hoặc bật trường. Giá trị mặc định là true.
editable	Boolean	Khi nào true, cho biết rằng người dùng có thể chỉnh sửa giá trị của trường.
maxLength	Number	Giới hạn đầu vào cho số lượng ký tự được chỉ định.
secure	Boolean	Ẩn văn bản đã nhập khi true. Sử dụng thuộc tính này để tạo các trường nhập mật khẩu. Giá trị mặc định: false.
keyboardType	KeyboardType	Hiển thị bàn phím tùy chỉnh để nhập văn bản dễ dàng hơn. Giá trị hợp lệ: datetime, phone, number, url, hoặc email.
returnKeyType	ReturnKeyType	Nhận hoặc đặt nhãn của khóa quay lại. Giá trị hợp lệ: done, next, go, search, hoặc send.
autocorrect	Boolean	Bật hoặc tắt tính năng tự động sửa

## 5. Thành phần: Components

### 19. TextField

#### Event

Tên	Sự miêu tả
textChange	Xảy ra khi văn bản thay đổi.
returnPress	Xảy ra khi nhấn phím quay lại.
focus	Xảy ra khi trường được focus
blur	Xảy ra khi trường mất focus

#### Thành phần Native

Android	iOS
<code>android.widget.EditText</code>	<code>UITextField</code>

## 5. Thành phần: Components

### 20. TextView

<TextView> là một thành phần giao diện người dùng hiển thị một vùng chứa văn bản nhiều dòng có thể chỉnh sửa hoặc chỉ đọc. Bạn có thể sử dụng nó để cho phép người dùng nhập văn bản lớn trong ứng dụng của mình hoặc để hiển thị văn bản dài hơn, nhiều dòng trên màn hình.

<TextView> mở rộng TextBase và EditableTextBase cung cấp các thuộc tính và sự kiện bổ sung.

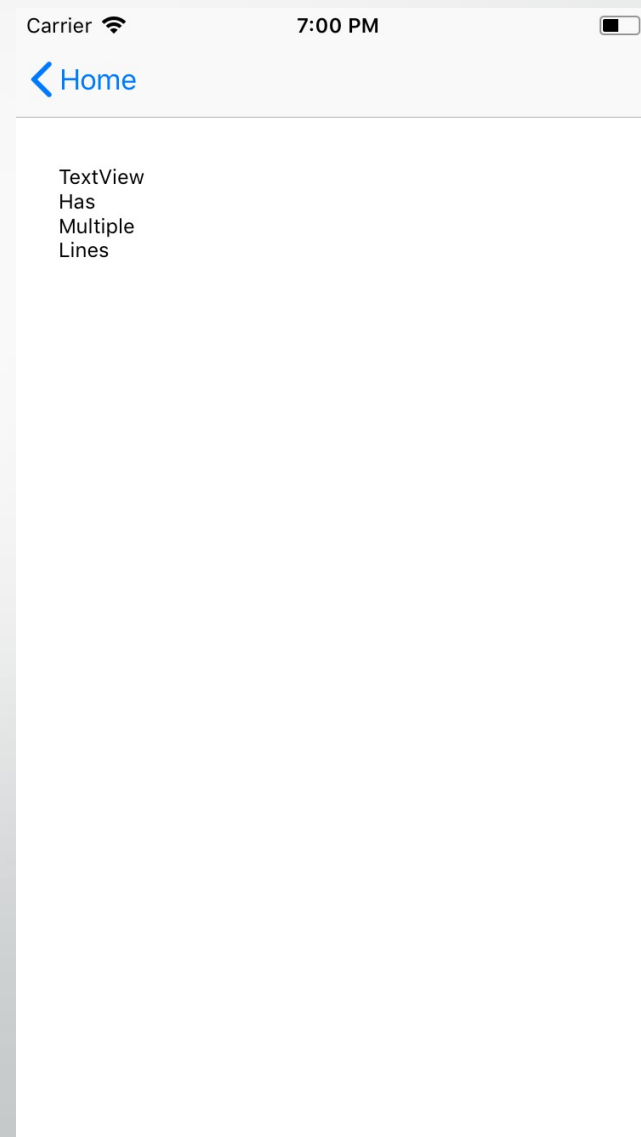
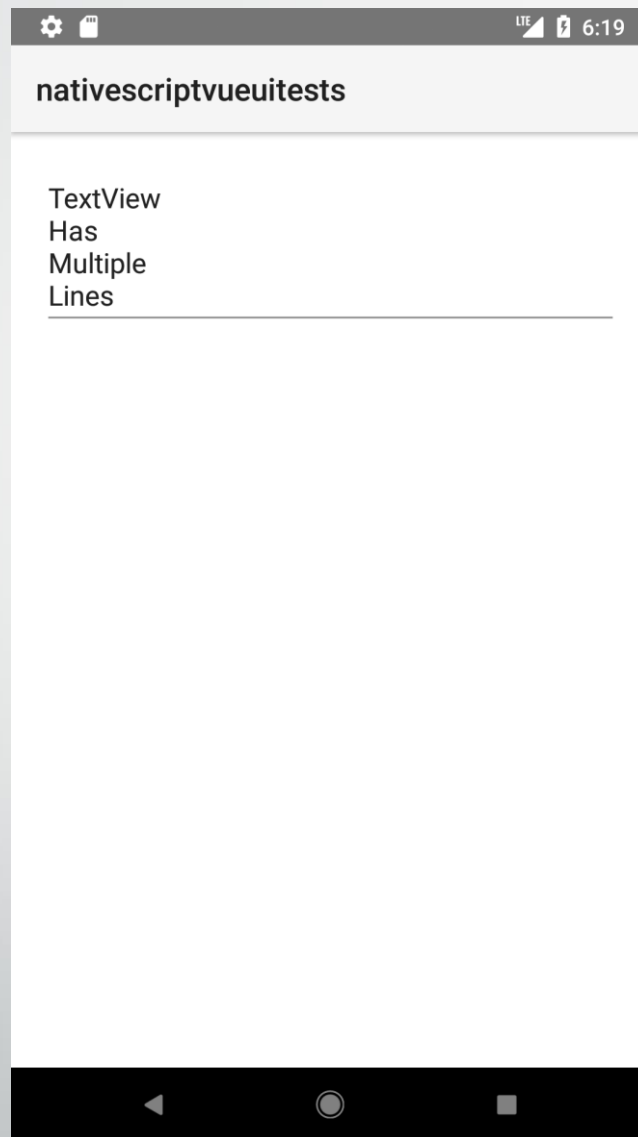
```
<TextView text="Multi\nLine\nText" />
```

<TextView> cung cấp liên kết dữ liệu hai chiều bằng cách sử dụng v-model.

```
<TextView v-model="textViewValue" />
```

## 5. Thành phần: Components

### 20. TextView



## 5. Thành phần: Components

### 20. TextView

Hiển thị Multi-style text: Dùng <FormattedString>

```
<TextView editable="false">
  <FormattedString>
    <Span text="You can use text attributes such as " />
    <Span text="bold, " fontWeight="Bold" />
    <Span text="italic " fontStyle="Italic" />
    <Span text="and " />
    <Span text="underline." textDecoration="Underline" />
  </FormattedString>
</TextView>
```

## 5. Thành phần: Components

### 20. TextView

#### Props

Tên	Kiểu	Mô tả
text	String	Gán hoặc lấy giá trị của thành phần.
hint	String	Gán hoặc lấy placeholder text khi thành phần có thể chỉnh sửa.
editable	Boolean	Khi bằng true, cho biết rằng người dùng có thể chỉnh sửa nội dung của vùng chứa.
maxLength	Number	Đặt số ký tự tối đa có thể được nhập vào vùng chứa.
keyboardType	KeyboardType	Hiển thị bàn phím tùy chỉnh để nhập văn bản dễ dàng hơn. Giá trị hợp lệ: datetime, phone, number, url, hoặc email.
returnKeyType		Gán hoặc lấy nhãn của khóa quay lại. Hiện chỉ hỗ trợ trên iOS. Giá trị hợp lệ: done, next, go, search, hoặc send.
autocorrect	Boolean	Bật hoặc tắt tính năng tự động sửa.



## 5. Thành phần: Components

### 21. TimePicker

#### Props

Tên	Kiểu	Sự miêu tả
hour	Number	Lấy hoặc gán giờ đã chọn.
minute	Number	Lấy hoặc gán phút đã chọn.
time	Date	Lấy hoặc gán thời gian đã chọn.
minHour	Number	Lấy hoặc gán giờ tối thiểu có thể chọn.
maxHour	Number	Lấy hoặc gán giờ tối đa có thể chọn.
minMinute	Number	Lấy hoặc gán phút tối thiểu có thể chọn.
maxMinute	Number	Lấy hoặc gán phút tối đa có thể chọn.
minuteInterval	Number	Lấy hoặc gán khoảng thời gian phút có thể chọn. Ví dụ: 5 hoặc 15 phút. Giá trị mặc định: 1.

## 5. Thành phần: Components

### 20. TextView

#### Event

Tên	Sự miêu tả
textChange	Xảy ra khi văn bản thay đổi.
returnPress	Xảy ra khi nhấn phím quay lại.
focus	Xảy ra khi vùng chứa được focus
blur	Xảy ra khi vùng chứa mất focus.

#### Thành phần Native

Android	iOS
<code>android.widget.EditText</code>	<code>UITextField</code>

## 5. Thành phần: Components

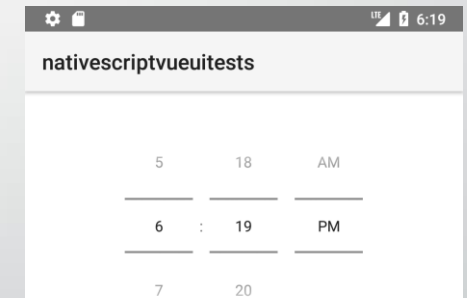
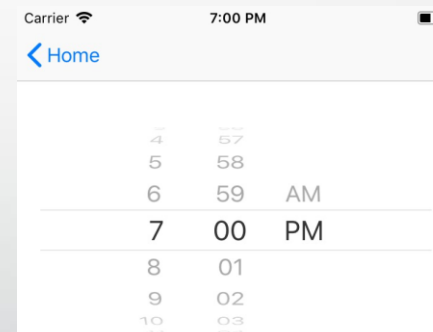
### 21. TimePicker

<TimePicker> là một thành phần giao diện người dùng cho phép người dùng chọn thời gian.

```
<TimePicker :hour="selectedHour" :minute="selectedMinute" />
```

<TimePicker> cung cấp liên kết dữ liệu hai chiều bằng cách sử dụng v-model.

```
<TimePicker v-model="selectedTime" />
```



## 5. Thành phần: Components

### 21. TimePicker

#### Event

Tên	Sự miêu tả
timeChange	Xảy ra khi thời gian thay đổi

#### Thành phần Native

Android	iOS
<code>android.widget.TimePicker</code>	<code>UIDatePicker</code>

## 5. Thành phần: Components

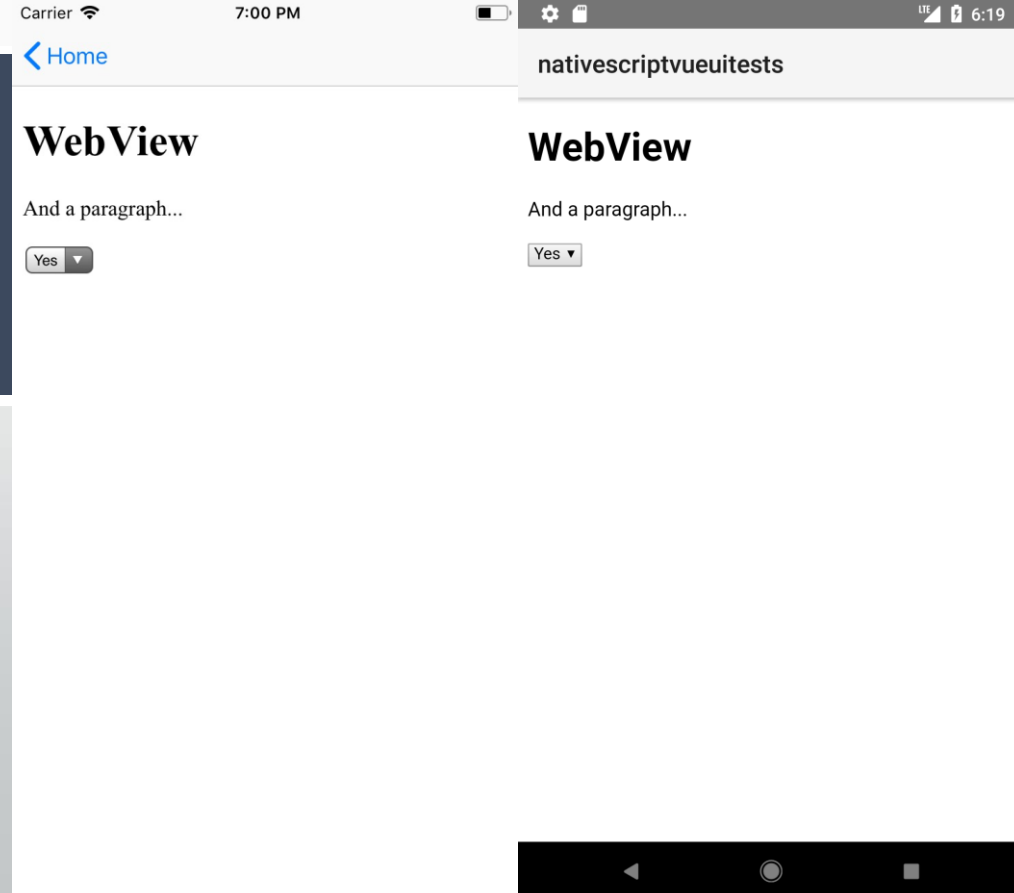
### 22. WebView

<WebView> là một thành phần giao diện người dùng cho phép bạn hiển thị nội dung web trong ứng dụng của mình. Bạn có thể kéo và hiển thị nội dung từ URL hoặc tệp HTML cục bộ hoặc bạn có thể hiển thị nội dung HTML tĩnh.

```
<WebView src="http://nativescript-vue.org/" />

<WebView src="~/html" />

<WebView src="<div><h1>Some static HTML</h1></div>" />
```



## 5. Thành phần: Components

### 22. WebView

#### Props

Tên	Kiểu	Sự miêu tả
src	String	Gán hoặc lấy nội dung web được hiển thị. Giá trị hợp lệ: URL tuyệt đối, đường dẫn đến tệp HTML cục bộ hoặc HTML tĩnh.

#### Event

Tên	Sự miêu tả
loadStarted	Xảy ra khi trang đã bắt đầu tải trong <WebView>.
loadFinished	Xảy ra khi trang đã tải xong trong <WebView>.

#### Thành phần Native

Android	iOS
<code>android.webkit.WebView</code>	<code>WKWebView</code>

## 6. Thành phần: Dialogs

### 1. AlertDialog

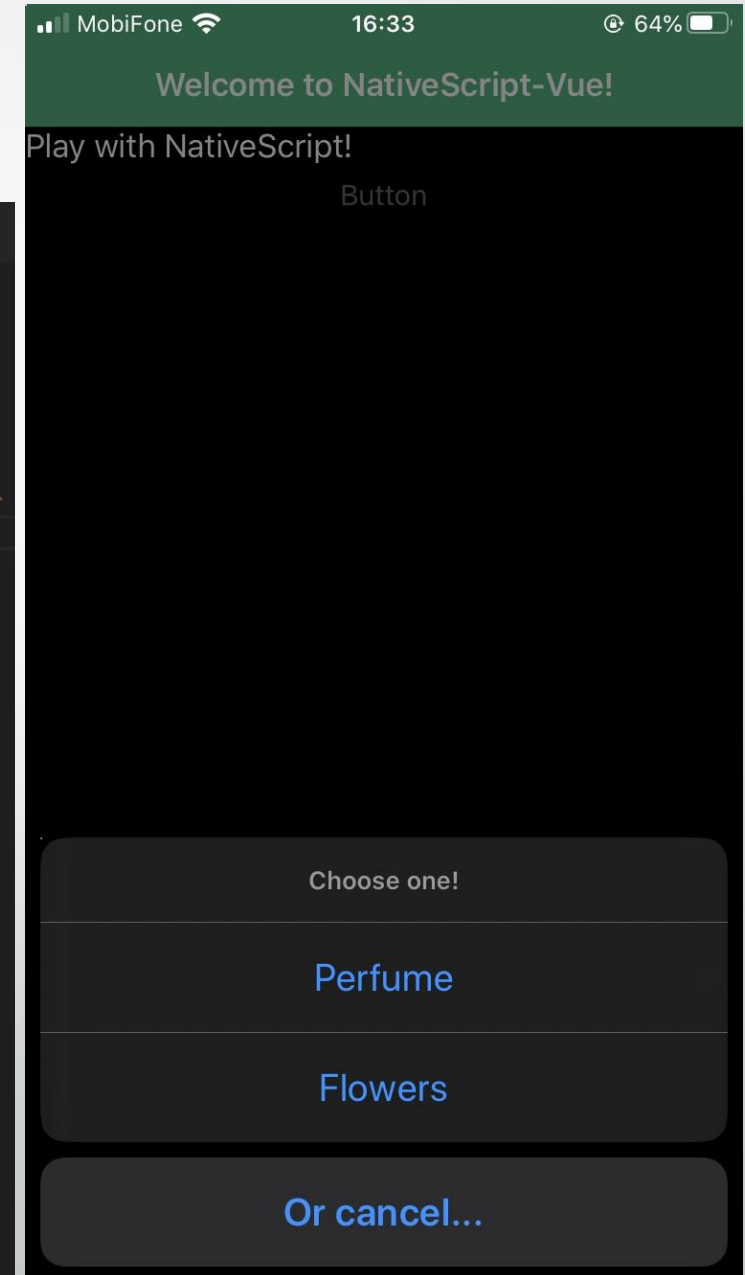
- Phương thức *action()* hiển thị danh sách các tùy chọn có thể chọn và nút hủy. Sử dụng nó để cho phép người dùng chọn giữa các tùy chọn hoặc loại bỏ lựa chọn.
- Phương thức này thuộc module [“ui/dialogs”](#)
- Cách sử dụng cơ bản:

```
action("Your message", "Cancel button text", ["Option1", "Option2"])  
  .then(result => {  
    console.log(result);  
  });
```

# 6. Thành phần: Dialogs

## 1. AlertDialog (t.t)

```
JS main.js x App.vue x TS nativescript.config.ts JS store.js app.scss
src > components > App.vue > {} "App.vue" > template > Page > StackLayout.home-panel > Button
1 <template>
2   <Page>
3     <ActionBar title="Welcome to NativeScript-Vue!" />
4     <StackLayout class="home-panel">
5       <!--Add your page content here-->
6       <Label textWrap="true" text="Play with NativeScript!" class="h2 description-label" />
7       <Button text="Button" @tap="showAlertDialog" />
8     </StackLayout>
9   </Page>
10 </template>
11
12 <script>
13   export default {
14     data() {
15       return {
16         msg: 'Hello World!'
17       }
18     },
19     methods: {
20       showAlertDialog() {
21         var dialogs = require("tns-core-modules/ui/dialogs");
22         dialogs.action("Choose one!", "Or cancel...", ["Perfume", "Flowers"])
23           .then(result => {
24             console.log(result);
25           });
26       }
27     }
28   }
29 </script>
```





## 6. Thành phần: Dialogs

### 2. AlertDialog

- Phương thức `alert()` hiển thị thông báo và nút OK. Sử dụng nó để hiển thị thông tin và thông báo mà không cần người dùng thực hiện.
- Phương thức này thuộc module [“ui/dialogs”](#)
- Cách sử dụng cơ bản:

```
alert('Your message')
  .then(() => {
    console.log("Alert dialog closed.");
  });
```

- Cấu hình Dialog

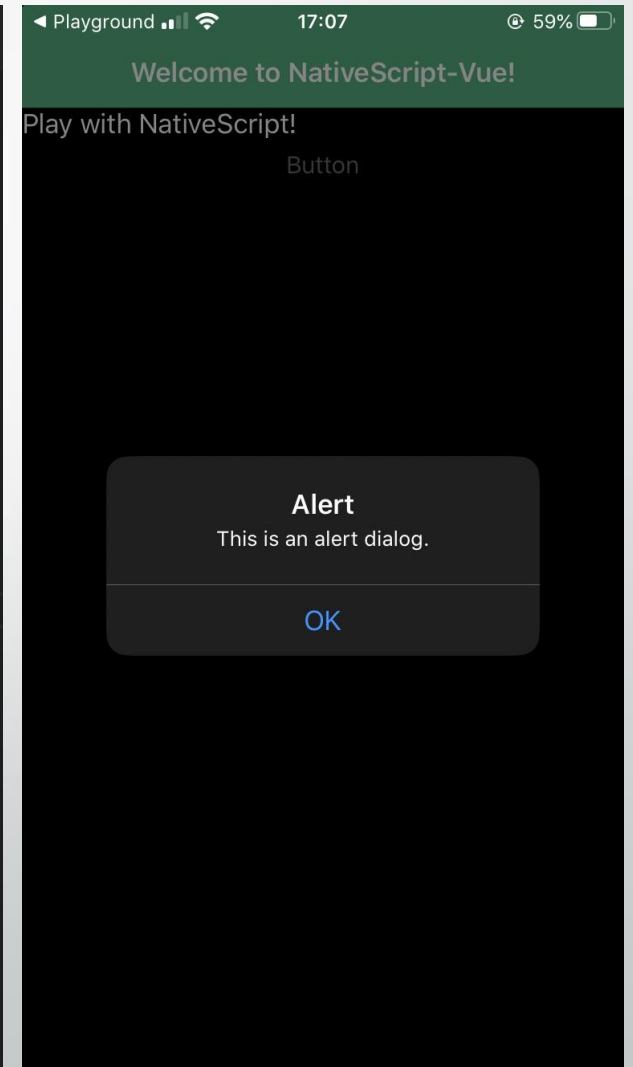
```
alert({
  title: "Your title",
  message: "Your message",
  okButtonText: "Your OK button text"
}).then(() => {
  console.log("Alert dialog closed");
});
```

## 6. Thành phần: Dialogs

### 2. AlertDialog (t.t)

```
JS main.js x App.vue x TS nativescript.config.ts JS store.js app.scss
src > components > App.vue > {} "App.vue" > script > default > data
1 <template>
2   <Page>
3     <ActionBar title="Welcome to NativeScript-Vue!" />
4     <StackLayout class="home-panel">
5       <!--Add your page content here-->
6       <Label textWrap="true" text="Play with NativeScript!" class="h2 description-label" />
7       <Button text="Button" @tap="showAlertDialog" />
8     </StackLayout>
9   </Page>
10 </template>
11
12 <script>
13   export default {
14     data() {
15       return {
16         msg: 'Hello!'
17       }
18     },
19     methods: {
20       showAlertDialog() {
21         var alerts = require("tns-core-modules/ui/dialogs");
22         alerts.alert('This is an alert dialog.')
23           .then(() => {
24             console.log("Alert dialog closed.");
25           });
26       },

```



## 6. Thành phần: Dialogs

### 3. ConfirmDialog

- Phương thức *confirm()* hiển thị thông báo xác nhận và các nút Hủy và Đồng ý.
- Phương thức này thuộc module [“ui/dialogs”](#)
- Cách sử dụng cơ bản:

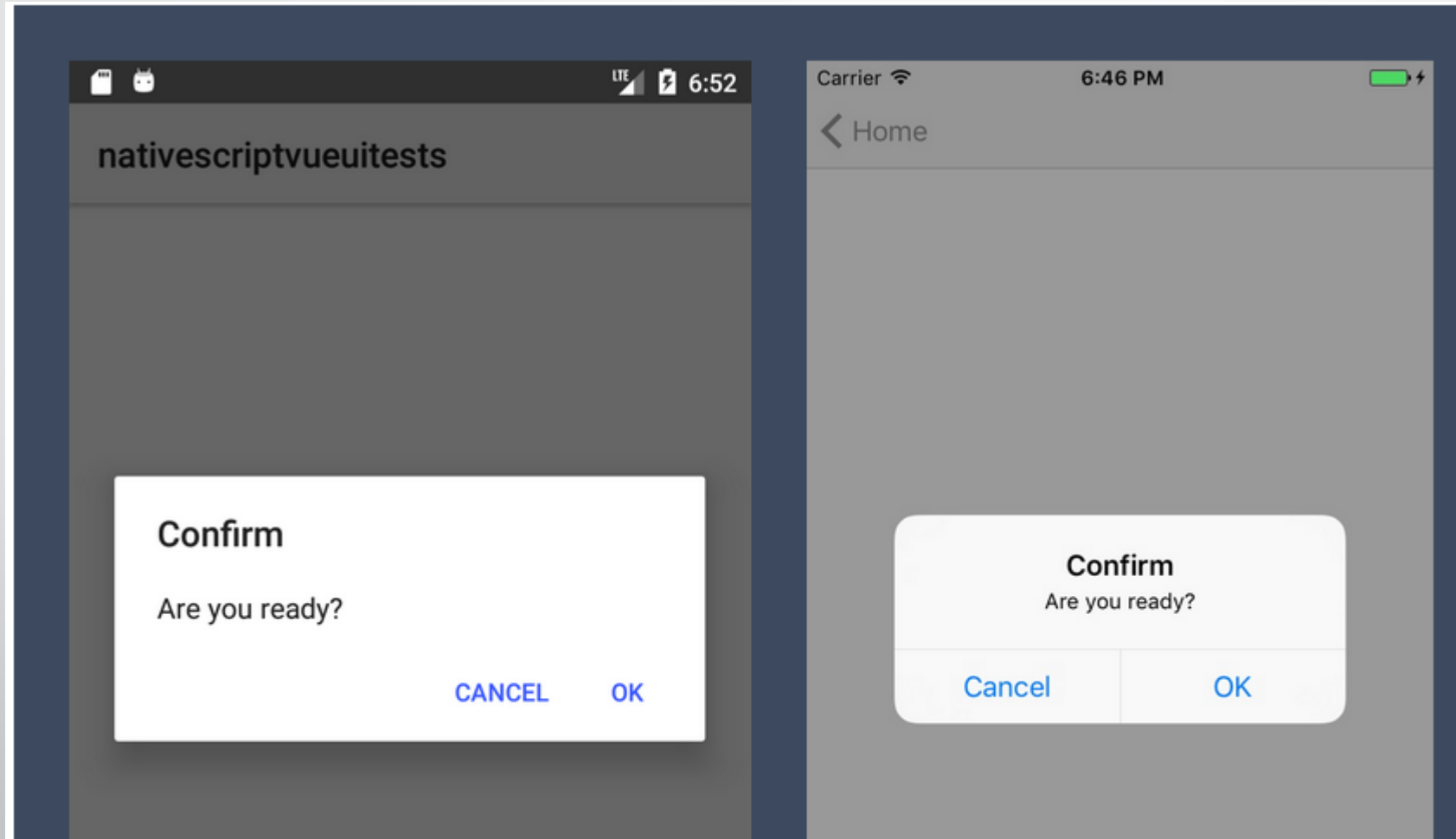
```
confirm('Your message')  
  .then(result => {  
    console.log(result);  
  });
```

- Cấu hình Dialog

```
confirm({  
  title: "Your title",  
  message: "Your message",  
  okButtonText: "Your OK button text",  
  cancelButtonText: "Your Cancel text"  
}).then(result => {  
  console.log(result);  
});
```

## 6. Thành phần: Dialogs

### 3. ConfirmDialog (t.t.)



## 6. Thành phần: Dialogs

### 4. LoginDialog

- Phương thức login() hiển thị hộp thoại nơi người dùng có thể cung cấp thông tin đăng nhập.
- Phương thức này thuộc module [“ui/dialogs”](#)
- Cách sử dụng cơ bản:

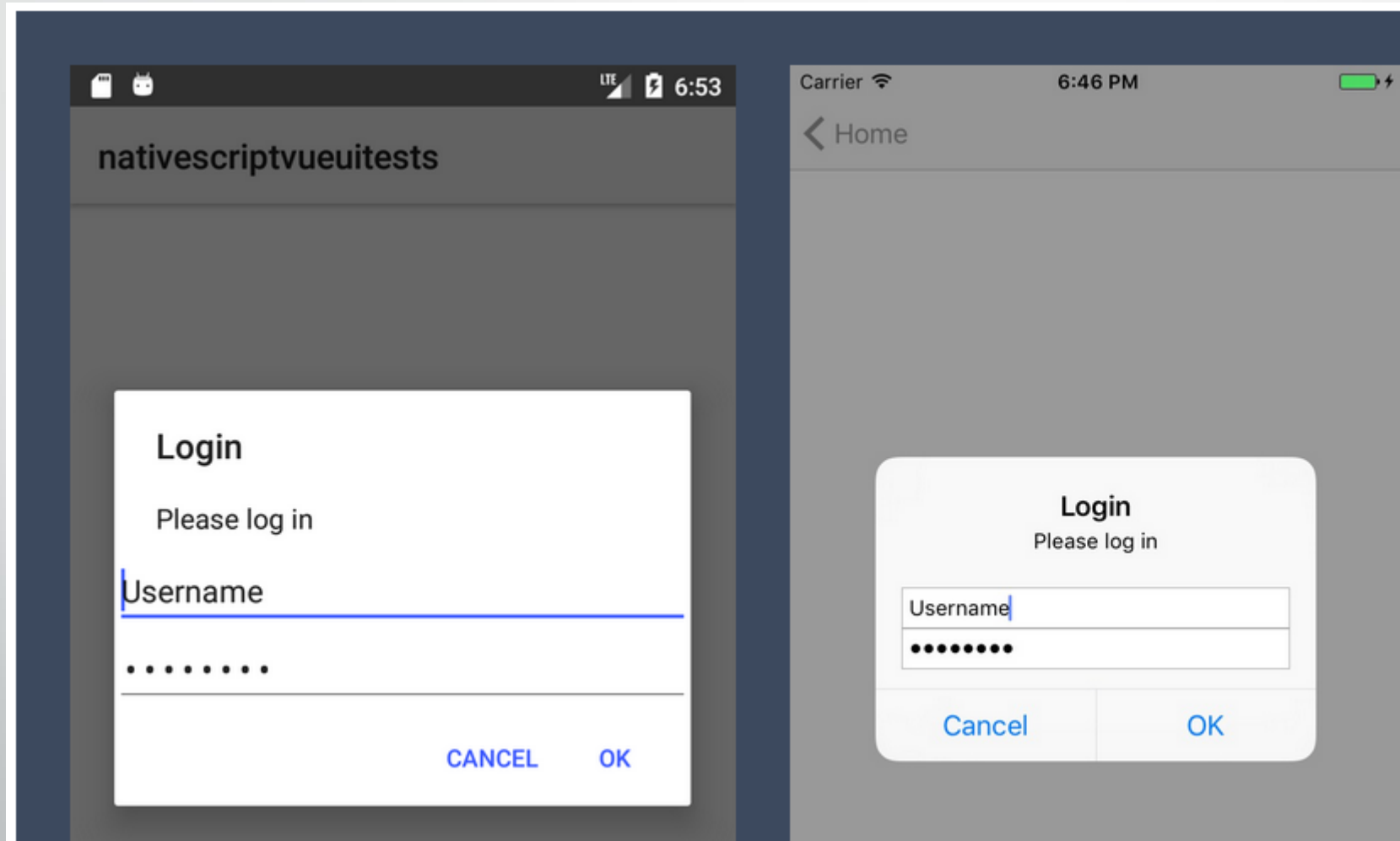
```
login("Your message", "Username field value", "Password field value").then(result => {  
    console.log(`Dialog result: ${result.result}, user: ${result.userName}, pwd: ${result.password}`);  
});
```

- Cấu hình Dialog

```
login({  
    title: "Your login title",  
    message: "Your login message",  
    okButtonText: "Your OK button text",  
    cancelButtonText: "Your Cancel button text",  
    userName: "Username field value",  
    password: "Password field value"  
}).then(result => {  
    console.log(`Dialog result: ${result.result}, user: ${result.userName}, pwd: ${result.password}`);  
});
```

## 6. Thành phần: Dialogs

### 4. LoginDialog (t.t.)



## 6. Thành phần: Dialogs

### 5. PromptDialog

- Phương thức `prompt()` hiển thị hộp thoại có trường một dòng để người dùng nhập.
- Phương thức này thuộc module [“ui/dialogs”](#)
- Cách sử dụng cơ bản:

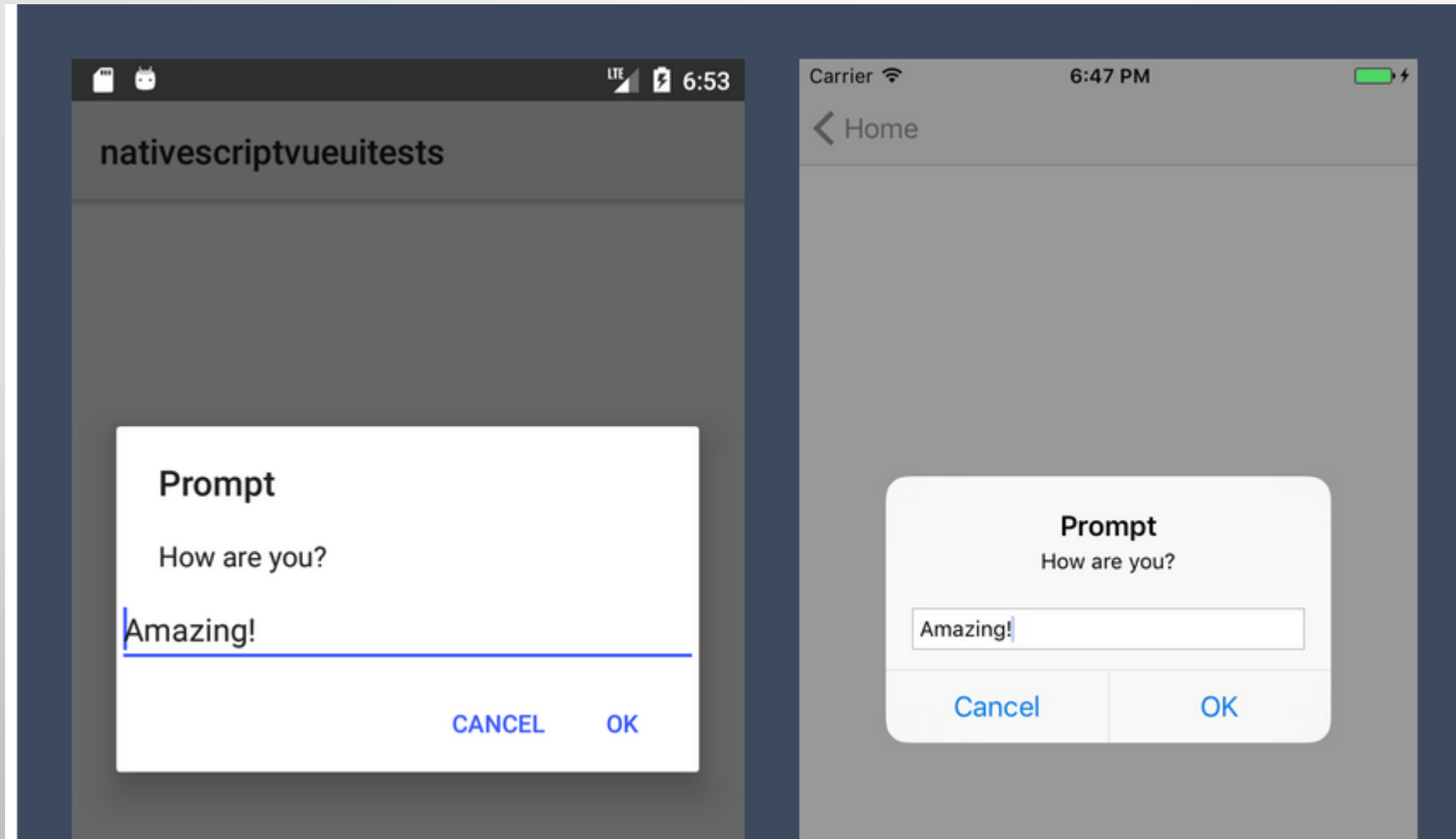
```
prompt('Your message to the user', 'Suggested user input')
  .then(result => {
    console.log(`Dialog result: ${result.result}, text: ${result.text}`)
  })
```

- Cấu hình Dialog

```
prompt({
  title: "Your dialog title",
  message: "Your message",
  okButtonText: "Your OK button text",
  cancelButtonText: "Your Cancel button text",
  defaultText: "Suggested user input",
}).then(result => {
  console.log(`Dialog result: ${result.result}, text: ${result.text}`)
});
```

## 6. Thành phần: Dialogs

### 5. PromptDialog (t.t.)







The  
end