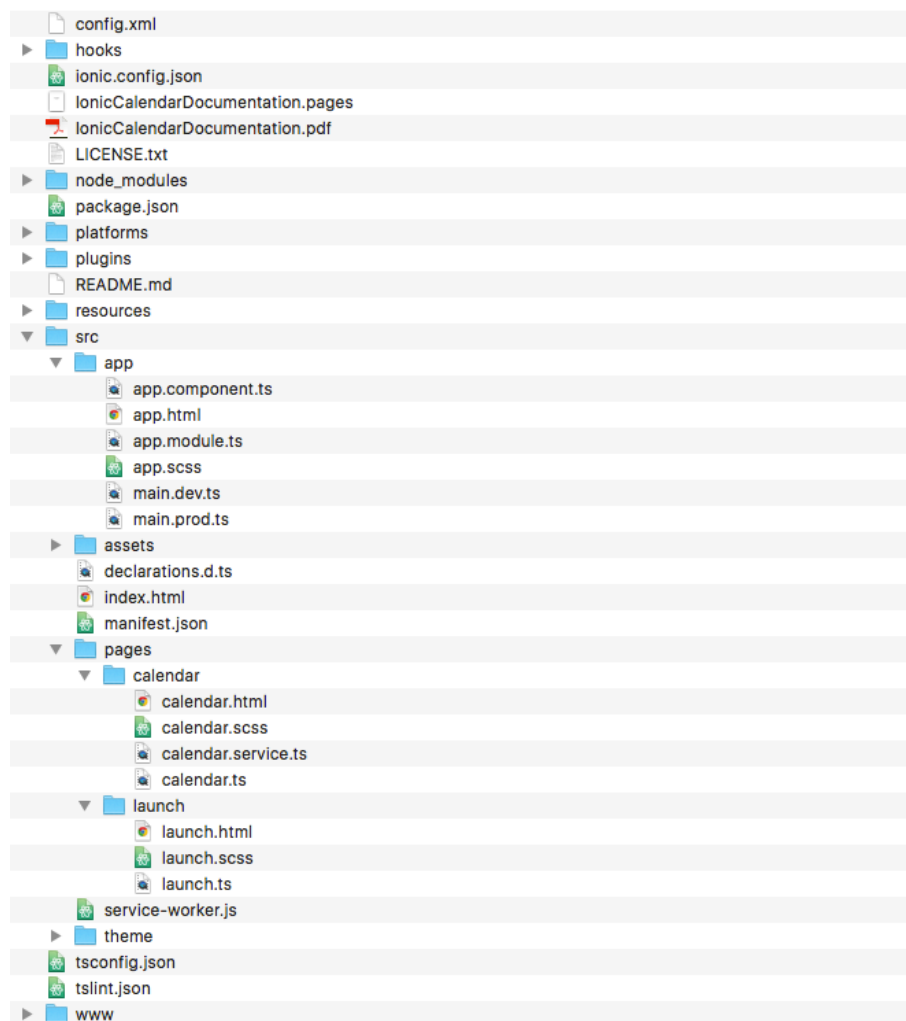


Ionic Calendar

Documentation

Project Structure



Folders	Remarks
src/app	Root folder for the component code base. Entry point component of the app is defined in app.component.ts and its templates is defined in app.html
src/pages/launch	The root pages of the app is set to launch component. This will be set during the app initialisation.
src/pages/calendar	Contains the component logic for calendar functionality
hooks, platforms, plugins, resources, etc.	These are ionic and cordova related folder. Please refer to ionic or cordova documentation

Calendar

Launch Template

The selector placeholder `<calendar></calendar>` is defined in `src/pages/launch/launch.html`. This is where Calendar component will attach itself on initialisation.

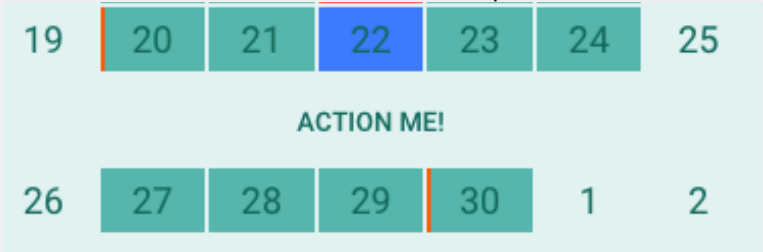
Calendar Service

This is defined in `src/pages/calendar/calendar.service.ts`. It is a utility class to fetch any specific data related to Events Information which is displayed in the Calendar

Calendar Component

This is defined in `src/pages/calendar/calendar.ts`. This component encapsulates the core logic for the calendar functionality.

Dependencies	<p>Moment.js - For all date manipulations, the library available at http://momentjs.com/ is used</p> <p>CalendarService - Service Class is injected into the component. This class can fetch any details that are required to render Event Information in the calendar template</p>							
constructor	<p>Initialises the attributes used in the calendar template.</p> <p>weekNames - array which contains the week names. (e.g S, M, T, W, T, F, S)</p> <table><tr><td>S</td><td>M</td><td>T</td><td>W</td><td>T</td><td>F</td><td>S</td></tr></table> <p>slideOptions - this is used to initialise <ion-slides> component defined in ionic framework. <ion-slides> is used in the calendar template</p>	S	M	T	W	T	F	S
S	M	T	W	T	F	S		
ngOnInit()	<p>Lifecycle method of Ionic/Angular Component. Is called during initialisation of the component. Initialises data for 3 months (currentMonth, prevMonth, nextMonth). This initialisation is required in order to provide a better performance when user swipes through the calendar months.</p> <p>There are 3 slides always in memory. Slide 1 (Prev Month) , Slide 2 (Current Month) , Slide 3 (Next Month)</p> <p>E.g Slide 1 (May), Slide 2 (June), Slide 3 (July)</p> <p>In the above example, slide for June will be active slide</p> <p>If the user swipes to the next month, then July Month Slide will be active and a new Slide data for August needs to be built.</p> <p>If the use swipes to the previous month, then May slide will be active and a new Slide Data for April Month needs to be built.</p> <p>The above process ensures that data for 3 slides (months) are always readily available in memory so that it give better user experience.</p>							

<pre>init(), initMonth(), createMonth(), createWeek()</pre>	<p>ngOnInit() -> init() -> initMonth() -> createMonth() -> createWeek() This sequence will build a MonthObject Map. The structure of MonthObject will result as below. The below example is for Jun,2016</p> <pre>{ weeks: [{ days:[29,30,31,1,2,3,4] }, //week1 {}, //empty { days:[5,6,7,8,9,10,11] }, //week 2 {}, //empty { days:[12,13,14,15,16,17,18] }, //week 3 {}, //empty { days:[19,20,21,22,23,24,25]}, //week 4 {}, //empty { days:[26,27,28,29,30,1,2] } //week 5] }</pre> <p>Note: The empty rows in the weeks Array above is to introduce the “Action ME” row as defined in the template.</p> 
<pre>onSlideChanged(), handleSlideView()</pre>	<p>onSlideChange() -> handleSlideView() onSlideChange() is triggered when user swipes the calendar view. handleSlideView() is called to create a MonthObject based on the direction of the slide swipe. The requirement to build a MonthObject is detailed in ngOnInit() section above.</p>
<pre>select()</pre>	<p>This is triggered when the user taps on a specific date. The function is a placeholder to respond to the user action.</p>

Calendar Template

This is defined in src/pages/calendar/calendar.html. High Level Structure of the template is defined below

```
<ion-slides>
  </ion-slide>
  .... // dots indicate that all the information is populated as per the structure below
</ion-slide> // for May



<ion-slide> // for Jun
  <ion-row>
    Month , Year
  </ion-row>
</ion-row>
```

```

        S, M, T, W, T, F, S
    </ion-row>
    <ion-row>
        29,30,31,1,2,3,4
    </ion-row>
    <ion-row> //Displayed only when user taps on the previous rows date
        "Action ME"
    </ion-row>
    <ion-row>
        5,6,7,8,9,10,11
    </ion-row>
    <ion-row> //Displayed only when user taps on the previous rows date
        "Action ME"
    </ion-row>
    <ion-row>
        12,13,14,15,16,17,18
    </ion-row>
    <ion-row> //Displayed only when user taps on the previous rows date
        "Action ME"
    </ion-row>
    <ion-row>
        19,20,21,22,23,24,25
    </ion-row>
    <ion-row> //Displayed only when user taps on the previous rows date
        "Action ME"
    </ion-row>
    <ion-row>
        26,27,28,29,30,1,2
    </ion-row>
    <ion-row> //Displayed only when user taps on the previous rows date
        "Action ME"
    </ion-row>
</ion-slide>

<ion-slide> // for Jul
    ...
</ion-slide>
</ion-slides>

```

<pre> <ion-slides (change)="onSlideChanged(\$event)" [options]="slideOptions"> </pre>	<p>Ionic Component/Directive to hold the slides. When the user swipes the slides, onSlideChanged() defined in CalendarComponent Class will be triggered.</p>
<pre> <ion-slide *ngFor="#monthObj of months"> </pre>	<p>As detailed earlier for CalendarComponent->ngOnInit(), there will be 3 slides rendered at any given point of time</p>
<pre> <ion-row class="text-center" *ngFor="#week of monthObj.weeks:#rowIndex = index" > <template [ngIf]="week.days" > </pre>	<p>For each week of the month, render a row with the dates of the week</p> 
<pre> <template [ngIf]="((monthObj.selectedDate.month() === monthObj.selectedMonth.month()) && (!week.days) && (monthObj.selectedRowIndex === (rowIndex - 1)))"> </pre>	<p>This is for the row "Action ME" that appears, when the user taps on the date</p> 

Calendar SASS

This is defined in `src/pages/calendar/calendar.scss`

It uses 3 important variables to control the theme colours. **favorite**, **favoriteMedium**, **favoriteDark**

These three variables are defined in `src/theme/variables.scss`. The values can be changed in this file to reflect the change in theme colors for the template