

NGX Bootstrap

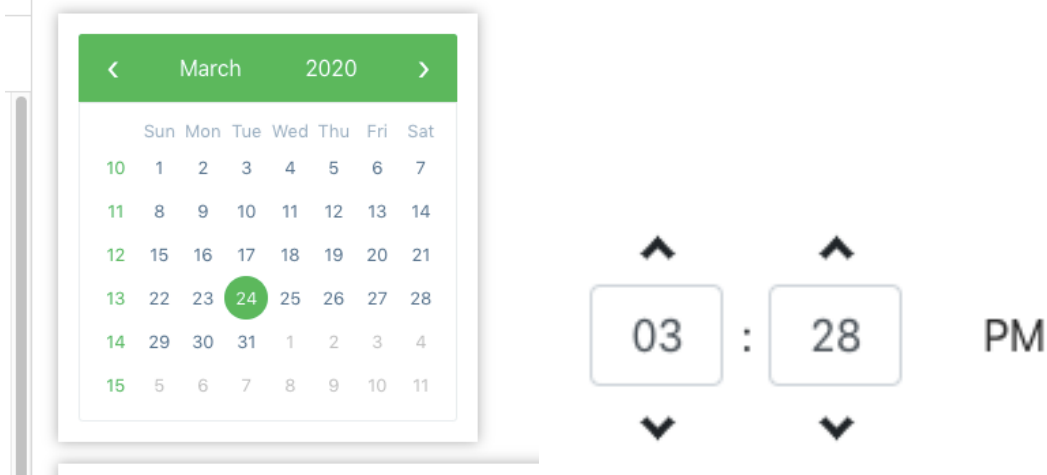
Так как проект «Город для всех» для верстки использует bootstrap, эта же библиотека имеет в себе собственный DatePicker.

Рассмотрим на примере bootstrap для angular, предоставляемый opensource

URL: <https://valor-software.com/ngx-bootstrap/>

API использования datepicker: <https://valor-software.com/ngx-bootstrap/#/datepicker>

Пример выбора даты и времени



Плюсы: не нужно использовать сторонние библиотеки, достаточно широкий API – кастомизации, формата вывода данных (форматов достаточно много, описывать здесь не вижу смысла, пример приведен ниже) – подробнее <https://valor-software.com/ngx-bootstrap/#/datepicker>.

Минусы: Возможность кастомизации timepicker, в исходном варианте не возможна, но есть возможность, о которой описано далее в исследовании.

Ревью: Вызов элемента достаточно прост:

```
1. <div class="row">
2.   <div class="col-xs-12 col-12 col-md-4 form-group">
3.     <input type="text"
4.       placeholder="Datepicker"
5.       class="form-control"
6.       bsDatepicker
7.       [bsConfig]="{ isAnimated: true }">
8.   </div>
```

Для отключения «ненужных дней» используется `[daysDisabled]="[6,0]"`, согласно номеру дня в неделе, либо минимальный и максимальный день

```
1. [minDate]="minDate"
2. [maxDate]="maxDate"
```

На выходе мы получаем данные в формате ISO (либо иной вариант, допустимый библиотекой (выбор зависит от разработчика проекта)), что позволяет получить данные

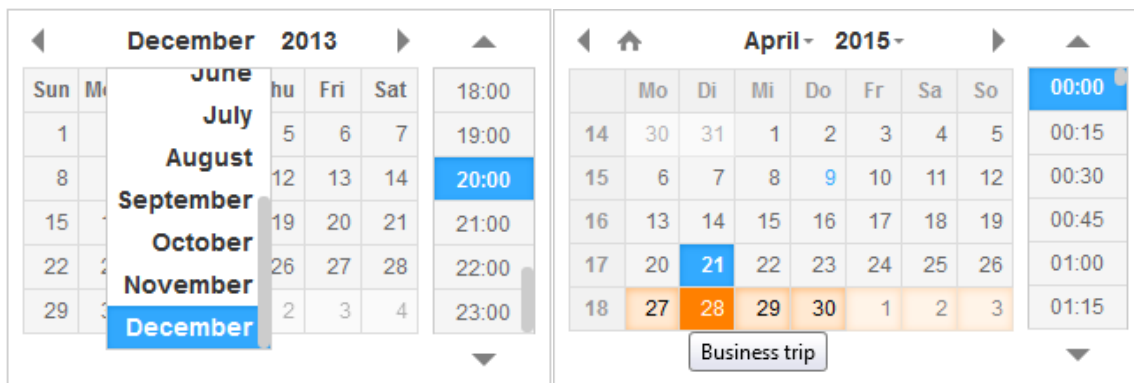
даты и, вывести время, доступное для выбора, что не составит труда написать модуль команде:

1. При выборе даты (использование datepicker «as is»), появляется интерфейс выбора времени (самописанный)
2. Пользователь выбирает нужное ему время – вероятно модальное окно с выбором времени (чекбоксы, к примеру)

Таким образом, эта библиотека позволит легко создать нужный интерфейс.

jQuery DateTimePicker

Данная библиотека «из коробки» умеет в datetimepicker, что позволяет одновременно выбирать и дату и время, в том числе закрывать недоступные варианты.



URL: <https://www.npmjs.com/package/jquery-datetimepicker>

Demo: <https://xdsoft.net/jqplugins/datetimepicker/>

Документация и исходный код: <https://github.com/xdan/datetimepicker>

В качестве ревью: Вызов модуля достаточно прост –

```
jQuery('#datetimepicker3').datetimepicker({
  format: 'd.m.Y H:i',
  inline: true,
  lang: 'ru'
});
```

На выходе мы получаем «сырые» данные с датой и временем, со стороны ревьюера, считаю, что в данном решении невозможно использовать «несколько временных промежутков» вернее выбор нескольких часов, к примеру, либо использовать самописанные решения (дополнить библиотеку нужным функционалом), хотя библиотека и так весьма перегружена лишним функционалом, который нам не понадобится в ходе разработки и, лишь, «утяжелит» исходный проект, что скажется на скорости работы решения. Допускаю вариант использования этой библиотеки, но настоятельно не рекомендую из-за лишнего функционала, который попросту использоваться не будет.

Примеры: Выбор разрешенного времени вводится через массив:

```
jQuery('#datetimepicker5').datetimepicker({
  allowTimes:[
    '12:00', '13:00', '15:00',
    '17:00', '17:05', '17:20', '19:00', '20:00'
  ]
});
```

```
});
```

Так же допустимо вводить конкретные промежутки времени. Вариантов сценария использования достаточно много.

Можно выводить «минимальное» время

```
{minTime:0,}// now  
{minTime:new Date()}  
{minTime:'12:00'}  
{minTime:'13:45:34',formatTime:'H:i:s'}
```

Аналогично для «максимального»

```
{maxTime:0,}  
{maxTime:'12:00'}  
{maxTime:'13:45:34',formatTime:'H:i:s'}
```

В качестве идеи, предлагаю два варианта использования:

1. Использовать оригинальный bootstrap datepicker – при выборе даты просто появляется массив времени (занятые соответственно нельзя выбрать), о идеи использования написано в исследовании.
2. Использовать jquery плагин – да, весьма емкое решение, но слишком перегружено по функционалу, о чем написано в исследовании, но удобное, что позволит команде максимально быстро создать нужный интерфейс.