

Projeto de Interfaces WEB

Introdução ao Angular
Datepicker
Passagem de Parâmetro (Output)
Exercício - IBetMyMoney

Objetivos

- Usar a classe Datepicker (widget do Bootstrap)
 - <https://ng-bootstrap.github.io/#/components/datepicker/overview>
- Passar parâmetros de um componente filho para um componente pai.
 - <https://angular.io/guide/component-interaction>

Instalação

- Entre no site do ng-bootstrap
 - <https://ng-bootstrap.github.io/#/getting-started>
- A instalação é feita via npm:
 - `npm install --save @ng-bootstrap/ng-bootstrap`
- Importe o NgbModule em app.module.ts

```
import {NgbModule} from '@ng-bootstrap/ng-bootstrap';
@NgModule({
  ...
  imports: [NgbModule, ...],
  ...
})
```

Componente datepicker-popup

- Vamos usar o exemplo do datepicker-popup
 - <https://ng-bootstrap.github.io/#/components/datepicker/examples#datepicker-popup>
- Crie um novo componente no seu projeto, para que não se misture com o componente raiz, o app.component:
 - `ng g c datepicker-popup --spec=false`

Obs.: o nome datepicker-popup é algo escolhido pelo autor, você pode escolher qualquer nome.

Componente datepicker-popup

- Em datepicker-popup.component.css, você irá colar o código do CSS para o calendário.

```
button.calendar, button.calendar:active {  
  width: 2.75rem;  
  background-image:  
url('  
AAAlwSFizAAAOxAAADsQBISsOGwAAABI0RVh0U29mdHdhcmUAAd3d3Lmlua3NjYXBILm9yZ5vuPBoAAAEUSURBVEiJ  
7ZQxToVAEIY/YCHGxN6XGOlpnpaEsBSeQC9ArZbm9TZ6ADyBNzAhQGGI8Riv4BLAWAgmkpBYkH1b8FWT2WK/  
zJ8ZJ4qil6XUI3ANnGKWBnht2/ZBDRK3hgVGNsCd7/ui+JkElrKtqurLpEWaphd933+lyl3LEldpCYCiKD6HcuOa/  
nwOa0ScJEnk0BJg0UTUWJRI6RxCYEzEmomsIIPU3IPW+grIAbquy+q6fluy/28RIBeRMwDXdXMgXLj/  
B2uimRXpui4D9sBeRLKI+1N+L+t6RwbWrZliTTTr1oxYtzVWiTQAcRxvTX+eJMnlUDaO1vpZRO5NS0x48slwfPc87xg4B04  
MCzQi8hlEwe4bl1DnFMCN2zsAAAAASUVORK5CYII=') !important;  
  background-repeat: no-repeat;  
  background-size: 23px;  
  background-position: center;  
}
```

Componente datepicker-popup

- Em datepicker-popup.component.ts

```
import { Component, Output, EventEmitter } from '@angular/core';
```

```
@Component({  
  selector: 'app-datepicker-popup',  
  templateUrl: './datepicker-popup.component.html',  
  styleUrls: ['./datepicker-popup.component.css']  
})
```

```
export class DatepickerPopupComponent{
```

```
  model;
```

```
  @Output() minhaData = new EventEmitter<Object>();
```

```
  selecionouData(){  
    this.minhaData.emit(this.model);  
  }
```

```
}
```

Importe o EventEmitter

A variável “model” é a data selecionada em json.

A variável “minhaData” será usada como um EventBiding personalizado do Componente DatepickerPopupComponent. Usa-se o @Output para torná-la visível do lado de fora
Da tag `<app-datepicker-popup>`

Esse método será chamado pelo evento de selecionar uma data no Popup (veja próximo slide!)

Componente datepicker-popup

- Em datepicker-popup.component.html

```
<div>
  <form class="form-inline">
    <div class="form-group">
      <div class="input-group">
        <input class="form-control" placeholder="yyyy-mm-dd" name="dp" [(ngModel)]="model" ngbDatepicker
          #d="ngbDatepicker" (dateSelect)="selecionouData()">
        <div class="input-group-append">
          <button class="btn btn-outline-secondary calendar" (click)="d.toggle()" type="button"></button>
        </div>
      </div>
    </div>
  </form>
  <pre>Model: {{ model | json }}</pre>
  <b>{{model!=undefined?model.year:""}}</b>
</div>
```

O evento (dateSelect) é específico do datepicker. No nosso caso, ele irá disparar a chamada do método “selecionouData()” que por sua vez irá disparar o evento “minhaData” (ver slide anterior).

No componente Raiz

- Em app.componente.ts
 - Crie o método mostrar Data selecionada, apenas para testes:

```
mostrarData(data:Object){  
  console.log("Ano:" + data['year']);  
  console.log("Mês:" + data['month']);  
  console.log("Dia:" + data['day']);  
}
```

Vamos checar se esse método está realmente pegando a data do componente filho
Do slide anterior.

No componente Raiz

- Em app.component.html

```
<app-datepicker-popup (minhaData)="mostrarData($event)">
```

```
</app-datepicker-popup>
```

“minhaData” é o evento que você criou no componente DatePickerPopup, usando o @Output. Ele é disparado indiretamente pela evento (dateSelect) – slide 7, ou seja, o ato de selecionar um data no popup. Quando disparado, ele chama o método selecionouData o qual irá emitir um evento dentro de “minhaData” (`this.minhaData.emit(this.model);`) Slide 6. Note que esse “emit” passa como parâmetro a variável “model”, que é a data selecionada como um objeto json. Esse mesmo “model” é passado como \$event para o método MostrarData – slide 8, o qual mostra a data no console.