04

Declarando o componente no módulo da aplicação

Transcrição

O Angular nos dá uma pista sobre o erro que houve anteriormente: "'ap-photo' is not a known element", ou seja, "'ap-photo' é um elemento desconhecido". O primeiro ponto que o programa pede para verificar é se ele faz parte de um módulo, caso seja um componente do Angular. **Isto porque um componente obrigatoriamente precisa fazer parte de um módulo**.

Em se tratando de componentes, para o Angular um módulo é uma maneira de agruparmos componentes afins. Vamos supor que iremos criar um *Data table*, formada por vários componentes — cabeçalho da tabela, rodapé, coluna, e por aí vai. Isso tudo se relaciona a um único *Data table*, sendo assim, poderemos ter **um** módulo na aplicação, no qual agregaremos todos estes componentes.

Se houver outra parte da aplicação que queira usar o *Data table*, basta ela importar este módulo. Isso fará com que todos os outros componentes necessários para o *Data table* também sejam importados. Se não tivermos um módulo, caso ele queira utilizar o *Data table*, terá que importar todos eles individualmente.

O módulo, então, é uma forma de **organizarmos** a nossa aplicação. Por ora, ela só possui um módulo, `app.module.ts`, o primeiro a ser carregado, lembram? Ao abrirmos o arquivo, perceberemos que ele importa `AppComponent`. No template correspondente, queremos utilizar o componente `PhotoComponent`.

Sendo assim, é necessário declararmos o componente neste módulo:

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { AppComponent } from './app.component';

import { PhotoComponent } from './photo/photo.component';

@NgModule({
    declarations: [
        AppComponent,
        PhotoComponent
    ],
    imports: [
        BrowserModule
    ],
    providers: [],
    bootstrap: [AppComponent]
})
export class AppModule { }
```

COPIAR CÓDIGO

Portanto, primeiro vêm os *imports* de tudo que foi criado pelo Angular, e por último os nossos *imports*. Trata-se de uma convenção proveniente do site do Angular. Assim, estando o `PhotoComponent` em `declarations`, o componente será acessível para `AppComponent`, isto é, se eles fazem parte do mesmo grupo, um é "enxergado" pelo outro.

Salvaremos o arquivo, e a imagem será exibida no navegador, como esperado. Inclusive, poderemos voltar a `app.component.html` e incluir outra imagem:

```
<ap-photo></ap-photo>
<ap-photo></ap-photo>
```

COPIAR CÓDIGO

Se voltarmos ao navegador, teremos duas imagens, uma ao lado da outra, ambas responsivas. No entanto, isso não é muito útil, uma vez que queremos

utilizar títulos e informações diferentes para cada imagem. Caso contrário, como poderemos reutilizar este componente?