

Para saber mais: registrando diretivas de outra maneira

Hoje, em nosso arquivo `Transform.js` registramos diretamente nossa diretiva através do *global view object*. No entanto, podemos ter maior controle do que utilizarmos em nossa aplicação explicitando a importação da diretiva no próprio componente.

Para isso, vamos alterar o arquivo `alurapic/src/directives/transform.js` para:

```
export default {  
  
  bind(el, binding, vnode) {  
  
    let current = 0;  
  
    el.addEventListener('dblclick', function() {  
  
      let incremento = binding.value || 90;  
      let efeito;  
  
      if(!binding.arg || binding.arg == 'rotate') {  
  
        if(binding.modifiers.reverse) {  
          current-=incremento;  
        } else {  
          current+=incremento;  
        }  
  
        efeito = `rotate(${current}deg)`;  
  
      } else if(binding.arg == "scale") {  
        efeito = `scale(${incremento})`;  
      }  
  
      el.style.transform = efeito;  
  
      if(binding.modifiers.animate) el.style.transition = 'transform 0.5s';  
    });  
  }  
  
};
```

O módulo `Transform` agora exporta um objeto que contém as configurações da nossa diretiva. Inclusive veja que nenhum nome foi definido para ela.

Por fim, precisamos alterar o componente `alurapic/src/components/home/Home.vue` para importar nossa diretiva e adicioná-la ao componente:

```
<template>  
<!-- código omitido -->  
</template>
```

```
<script>
import Pannel from '../shared/painel/Painel.vue';
import ImagemResponsiva from '../shared/imagem-responsiva/ImagemResponsiva.vue';
import Botao from '../shared/botao/Botao.vue';

// importou diretiva. Tem que adicionar na propriedade directives logo abaixo!
import transform from '../..../directives/Transform';

export default {

  components: {
    'meu-painel' : Pannel,
    'imagem-responsiva': ImagemResponsiva,
    'meu-botao' : Botao
  },

  directives: {
    'meu-transform': transform
  },

  // código posterior omitido
}
```

Essa forma permite darmos o apelido que desejarmos para a diretiva em nossos componentes.