

Para saber mais: Forward vs redirect

Na aula, vimos o prefixo `redirect` na *action*:

```
@PostMapping("novo")
public String novo(@Valid RequisicaoNovoPedido requisicao,
BindingResult result) {
    if(result.hasErrors()) {
        return "pedido/formulario";
    }

    Pedido pedido = requisicao.toPedido();
    pedidoRepository.save(pedido);

    return "redirect:/home";
}
```

[COPIAR CÓDIGO](#)

Nesse caso, o Spring MVC executará o redirecionamento *client-side*. Isto é, ele devolve uma resposta HTTP para pedir uma nova requisição para a URL `/home`.

Também existe o redirecionamento *server-side*, mas devemos usar o prefixo `forward`. Veja o código:

```
@PostMapping("novo")
public String novo(@Valid RequisicaoNovoPedido requisicao,
BindingResult result) {
    if(result.hasErrors()) {
        return "pedido/formulario";
    }
}
```

```
Pedido pedido = requisicao.toPedido();  
pedidoRepository.save(pedido);  
  
return "forward:/home";  
}
```

[COPIAR CÓDIGO](#)

Nesse caso, o fluxo volta apenas para o *Front-Controller* do Spring MVC e ele chama a nova *action*. Enquanto o redirecionamento *client-side* causa uma nova requisição, o *server-side* continua dentro da mesma requisição HTTP.

Nesse exemplo, o melhor seria usar `redirect`, pois estamos trabalhando com uma requisição *POST*, seguindo a regra: *"always redirect after post"*.