07

Para saber mais: ModelAndView

Aprendemos que, para a *view* funcionar, é preciso receber o modelo através do `Model` que recebemos como parâmetro e é preciso definir o *nome da view* no retorno do método. Veja a estrutura básica da nossa *action*:

```
@GetMapping("/home")
public String home(Model model) {
    List<Pedido> pedidos = repository.findAll();
    model.addAttribute("pedidos", pedidos);
    return "home";
}
```

COPIAR CÓDIGO

Ou seja, para trabalhar com uma página, é preciso definir um ***ModeloEPagina*** e para tal existe uma classe justamente com esse nome: `ModelAndView` . Veja a mesma *action*, mas usando `ModelAndView` do Spring MVC:

```
@GetMapping("/home")
public ModelAndView home() {
    List<Pedido> pedidos = repository.findAll();
    ModelAndView mv = new ModelAndView("home");
    mv.addObject("pedidos", pedidos);
    return mv;
}
```

COPIAR CÓDIGO

Desse forma, os parâmetros do método ficam reservados para os dados da requisição (o *input*) e o retorno do método fica reservado para os dados da *view* (o

output).

Você vai encontrar ambos os exemplos na literatura e nos projetos com Spring MVC!