



[REST-API]



Testes automatizados

@WebMvcTest

O **Spring** também tem uma anotação específica para testar **camadas de MVC (model, view e controller)**, isso faz com que o **Spring carregue apenas módulos dessa camada**, mas pode ser que a classe **Controller a ser testada precise de mais coisas do que apenas o MVC**, então dependendo não vale a pena usar essa anotação em específico.

Testando controller

Uma vez que eu tenho a **URI** e o **Json** do controller que eu quero testar eu **preciso dispara-lo**, assim sendo eu posso **injetar o MockMvc** que é uma classe utilitária do próprio **Spring Web**, como todo mock ele serve para simular algo, nesse caso simula chamadas. Para fazer o uso do **MockMvc** sem usar a anotação de **WebMvcTest**, precisamos adicionar a anotação de **@AutoConfigureMockMvc**:

```
@SpringBootTest
@AutoConfigureMockMvc
@ActiveProfiles("test")
public class AutenticacaoControllerTest {

    @Autowired
    private MockMvc mockMvc;
```

O **mock** pode ser construído passando o **verbo HTTP e a URI** que **ele** deve utilizar, o **conteúdo da requisição e o tipo dele**. No próprio **mock** também dá pra **fazer a assertiva do teste**:

```
@Test
public void deveriaLancarExceptionDevolver400CasoDadosAutenticacaoIncorretos() throws Exception {
    URI uri = new URI("str: "/auth");
    String json = "{ \"email\": \"invalido@hotmail.com\", \"senha:123456\"}";

    mockMvc.perform(MockMvcRequestBuilders.post(uri)
        .content(json)
        .contentType(MediaType.APPLICATION_JSON))
        .andExpect(MockMvcResultMatchers.status().is(status: 400));
}
```