Sim — **você deve explicitamente permitir o novo endpoint /fiberguardian/sessao/valida na sua SecurityFilterChain**, pois atualmente ele está implícito dentro de .anyRequest().authenticated().

**✅ Objetivo**

Permitir requisições GET ao endpoint /fiberguardian/sessao/valida apenas **para usuários autenticados**, sem exigir token CSRF.

**🔐 O que precisa ser ajustado**

**1. Permitir o acesso sem CSRF**

Como GET /fiberguardian/sessao/valida **não altera estado**, não precisa de proteção CSRF. Então, adicione na configuração:

java

CopiarEditar

.ignoringRequestMatchers("/fiberguardian/csrf-token", "/fiberguardian/sessao/valida")

➡️ Atualizando a parte do .csrf(...):

java

CopiarEditar

.csrf(csrf -> csrf

.csrfTokenRepository(csrfTokenRepository())

.csrfTokenRequestHandler(new XorCsrfTokenRequestAttributeHandler())

.ignoringRequestMatchers(

"/fiberguardian/csrf-token",

"/fiberguardian/sessao/valida"

)

)

**2. Não precisa adicionar em .authorizeHttpRequests(...)**

O endpoint /fiberguardian/sessao/valida **deve continuar protegido** por autenticação, então ele **não precisa de .permitAll()**.

Seu trecho atual:

java

CopiarEditar

.authorizeHttpRequests(authz -> authz

.requestMatchers(HttpMethod.POST, "/fiberguardian/login").permitAll()

.requestMatchers(HttpMethod.GET, "/fiberguardian/csrf-token").permitAll()

.requestMatchers(HttpMethod.GET, "/public/\*\*").permitAll()

.anyRequest().authenticated()

)

Está correto — qualquer coisa que não tenha .permitAll() cairá em .authenticated().

**✅ Resultado final**

| **Endpoint** | **CSRF** | **Autenticação** |
| --- | --- | --- |
| GET /fiberguardian/sessao/valida | Ignorado | ✅ Sim, exige autenticação |
| GET /fiberguardian/csrf-token | Ignorado | ✅ Sim, exige autenticação para criar sessão |
| POST /fiberguardian/login | Ignorado por não usar CSRF (autenticador customizado) | ❌ Não exige autenticação |

**✅ Conclusão**

Sim, você deve **adicionar /fiberguardian/sessao/valida em .ignoringRequestMatchers(...) dentro de .csrf(...)**, mas **não deve permitir explicitamente em .authorizeHttpRequests()**, pois queremos que apenas usuários autenticados possam acessá-lo.