

Spring Security Explicado

Professor Isidro

[isi]



CONTEÚDO PREVISTO



01

Primeiros passos

Configurando e entendendo nosso projeto com Spring Security

02

Liberando/Restringindo

Como configurar nossa aplicação para liberar e restringir endpoints

03

Tokens

Usando JWT para autenticar nossos usuários e criando perfis de acesso

04

Extras

Como configurar itens adicionais como o CORS e o acesso ao H2 Console



[isi]



Agora vai!

Vamos começar a proteger nossas APIs de forma rápida e eficiente

[isi]



BASICAMENTE

Pela própria definição disponível na documentação, o Spring Security é um framework altamente customizável para você garantir acesso autorizado e autenticado nas suas APIs



A collection of isometric icons in shades of teal and light blue. The icons include a large shield with a checkmark, a padlock, several speech bubbles, and multiple cloud shapes. These icons are arranged in a cluster in the upper center of the slide. The background is a dark teal color with decorative elements: a grid of white circles in the top left, a large teal circle in the top right, and a large teal ring in the bottom left.

01

Como funciona?

Ao ser inicialmente configurado como dependência, o Spring Security gera um mecanismo padrão de autenticação

[isi]



02

Como funciona?

Toda URL é redirecionada para uma página de Login, onde uma senha padrão é gerada. Você consegue acessar a URL com seu usuário "user" e a respectiva senha

[isi]



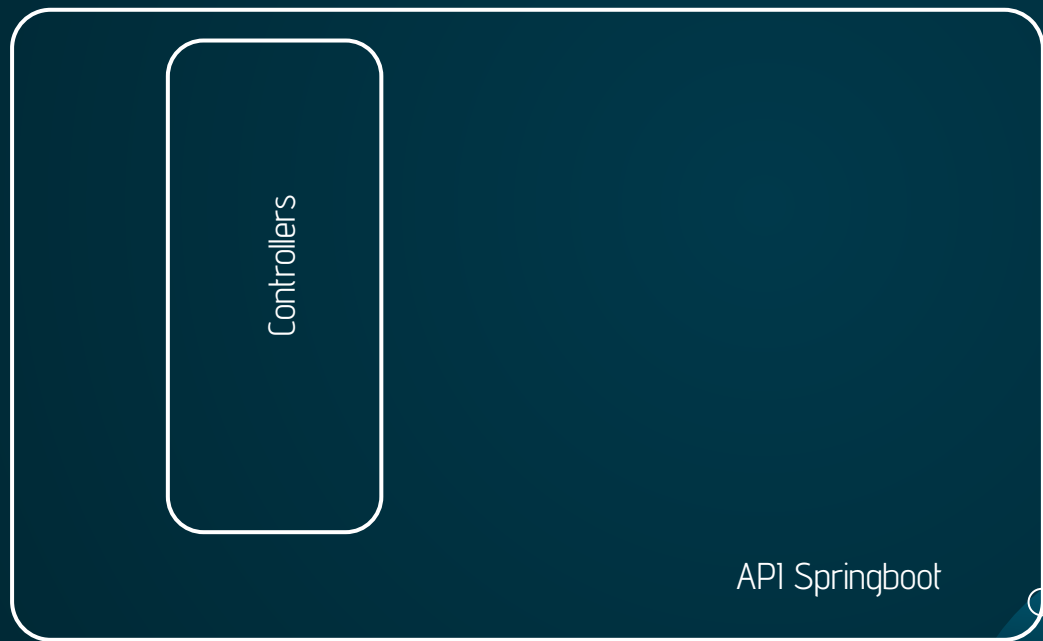
03

Como funciona?

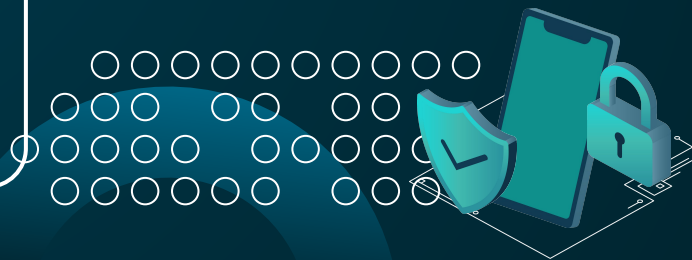
Entretanto essa abordagem é totalmente inviável para uma aplicação RestFul, pois a autenticação precisa manter minimamente uma credencial de identificação

[isi]

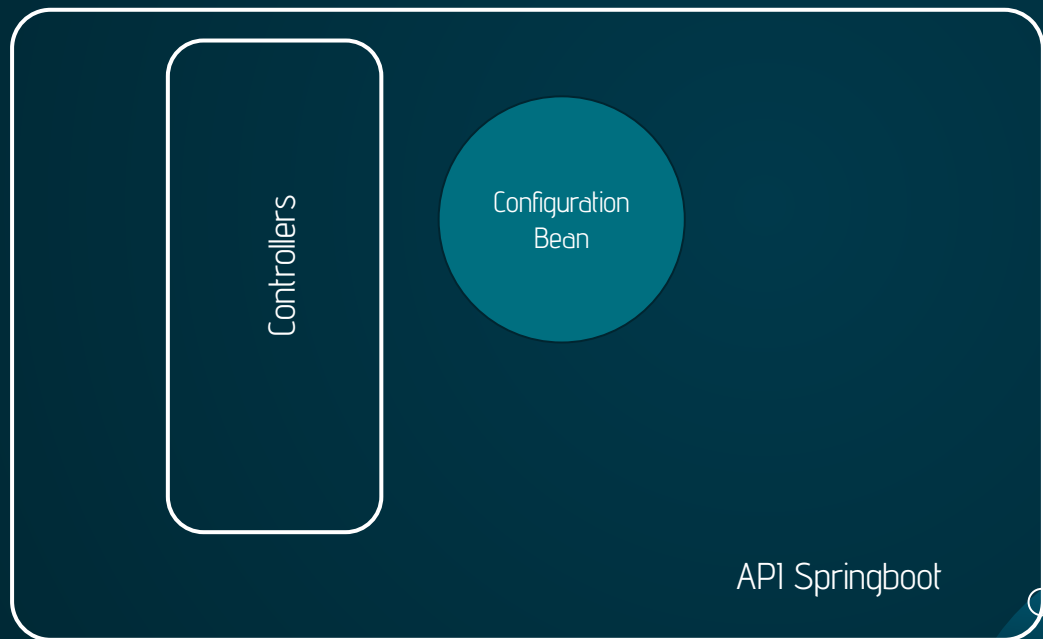
Arquitetura do Spring Security



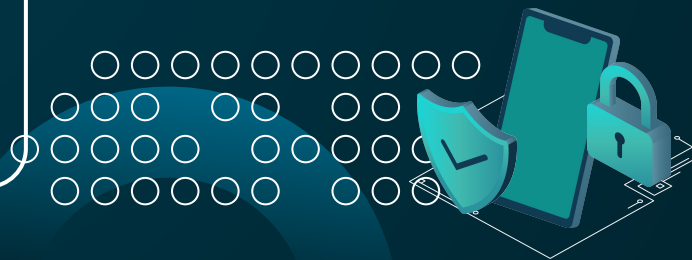
[isi]



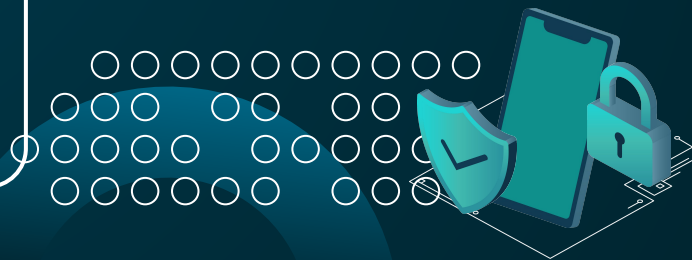
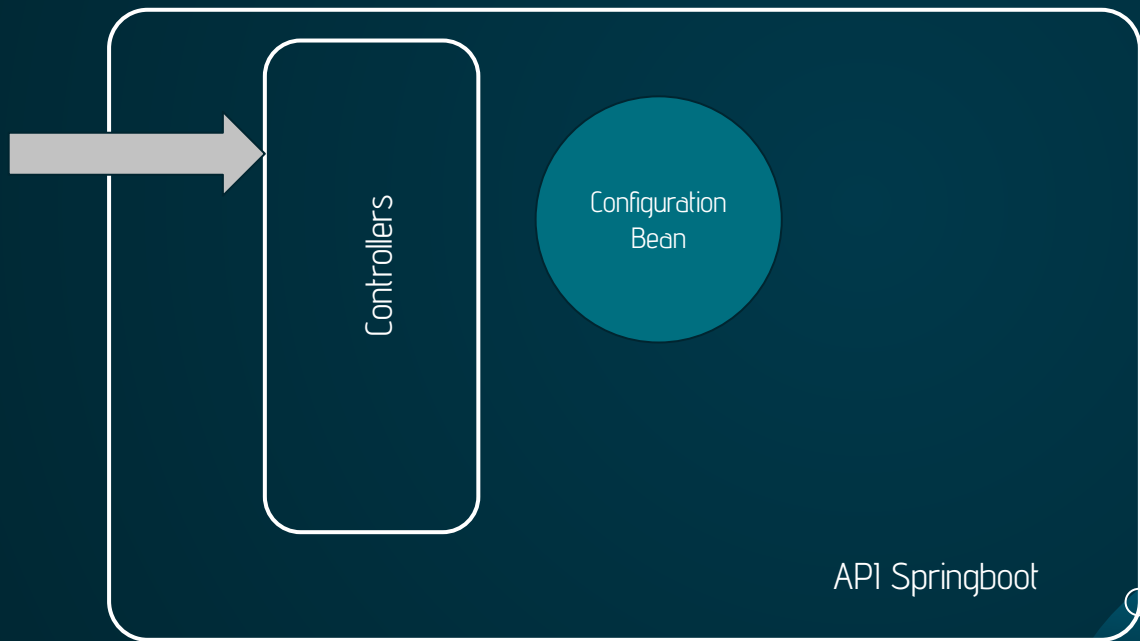
Arquitetura do Spring Security



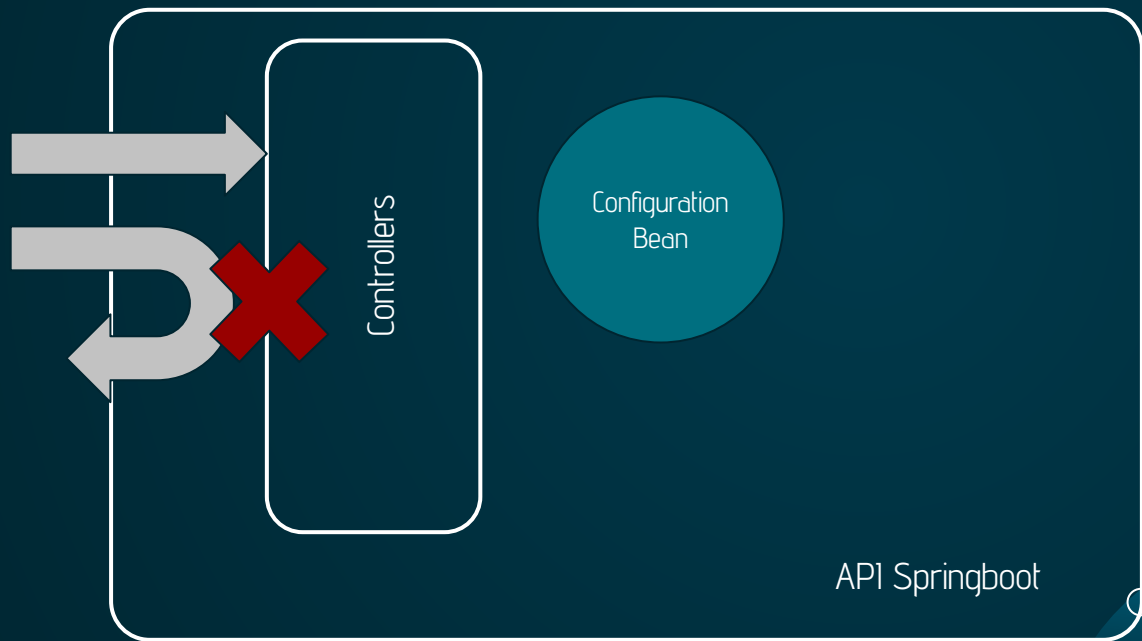
[isi]



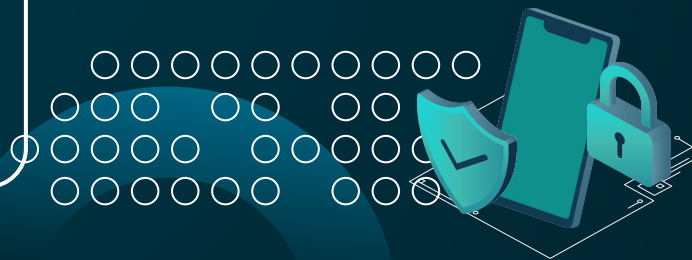
Arquitetura do Spring Security



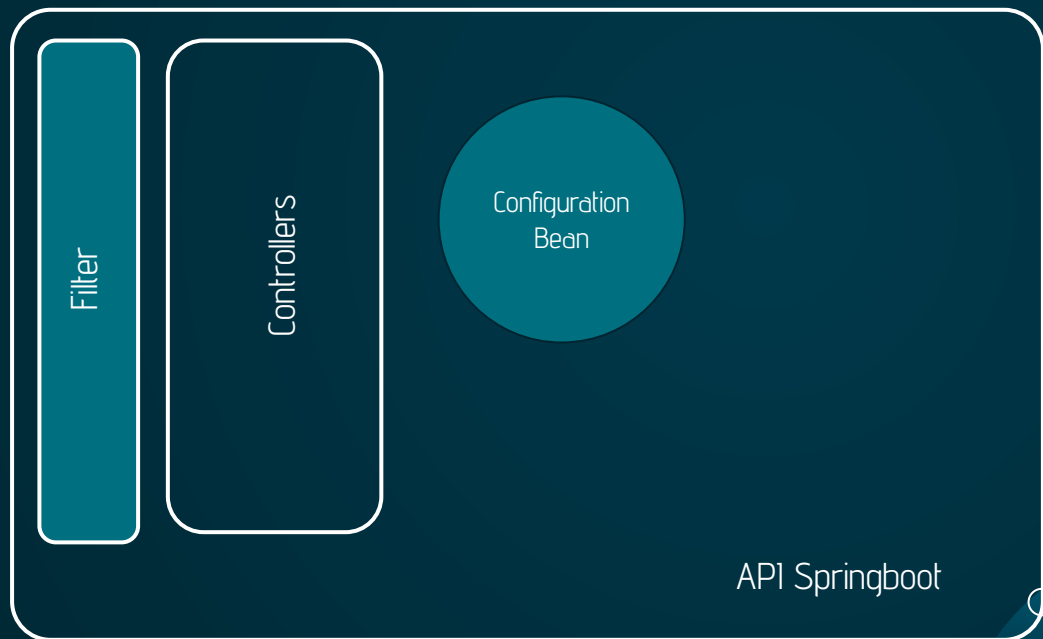
Arquitetura do Spring Security



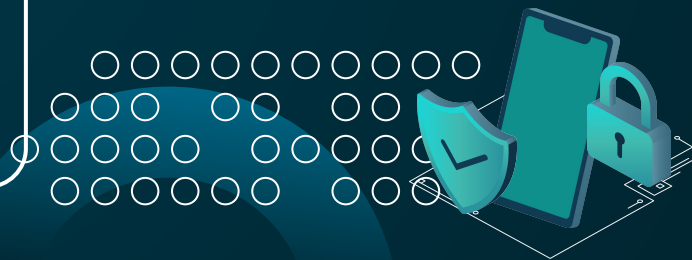
[isi]



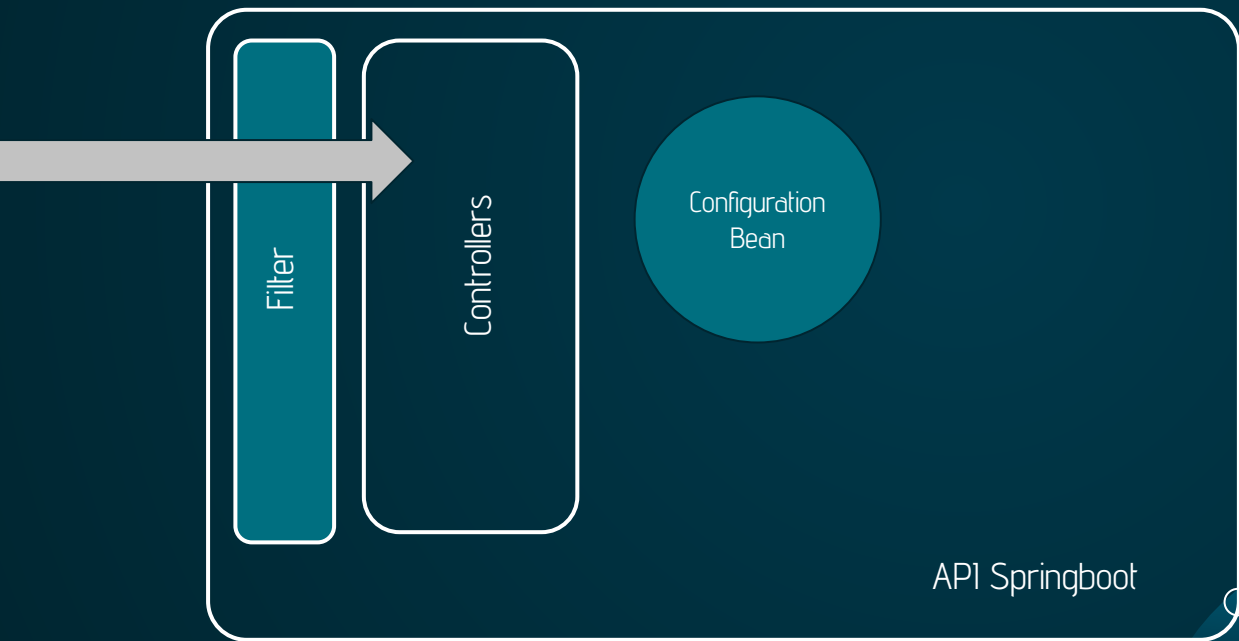
Arquitetura do Spring Security



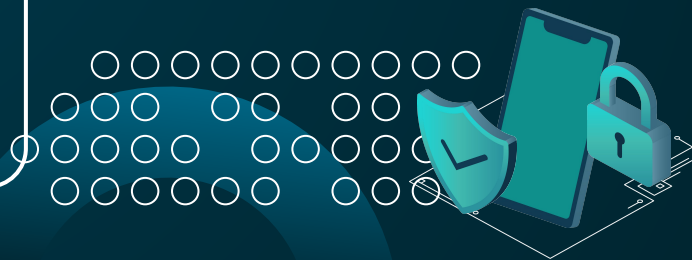
[isi]



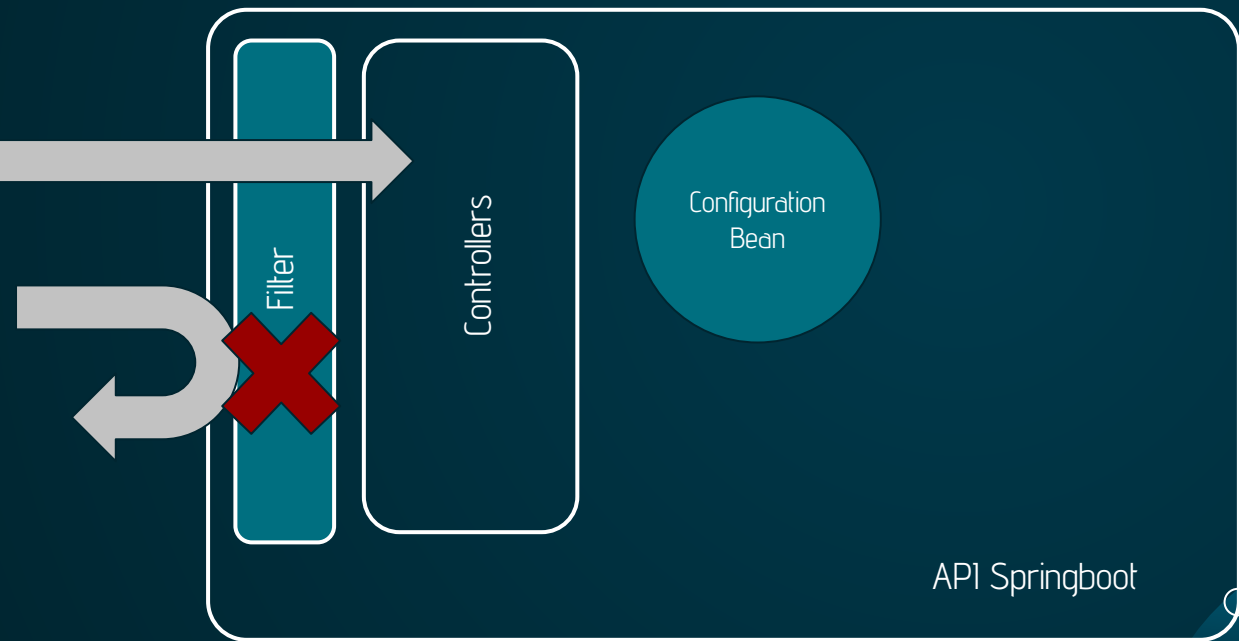
Arquitetura do Spring Security



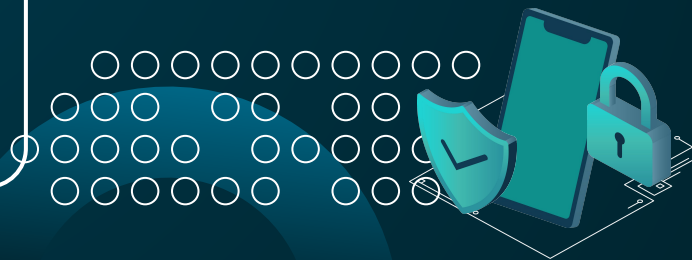
[isi]



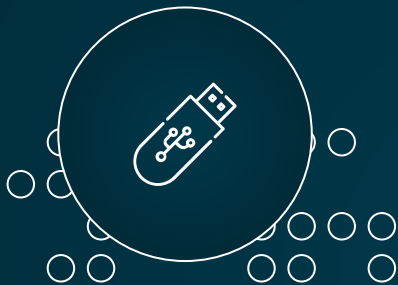
Arquitetura do Spring Security



[isi]

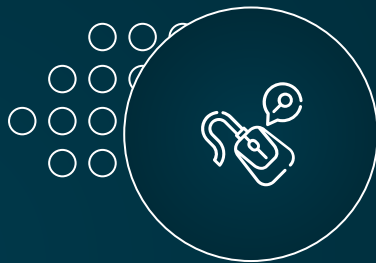


Vamos entender os cenários



1. Requisição de Login

Precisamos liberar o acesso a um endpoint de login para que nossos usuários possam minimamente gerar um token de acesso

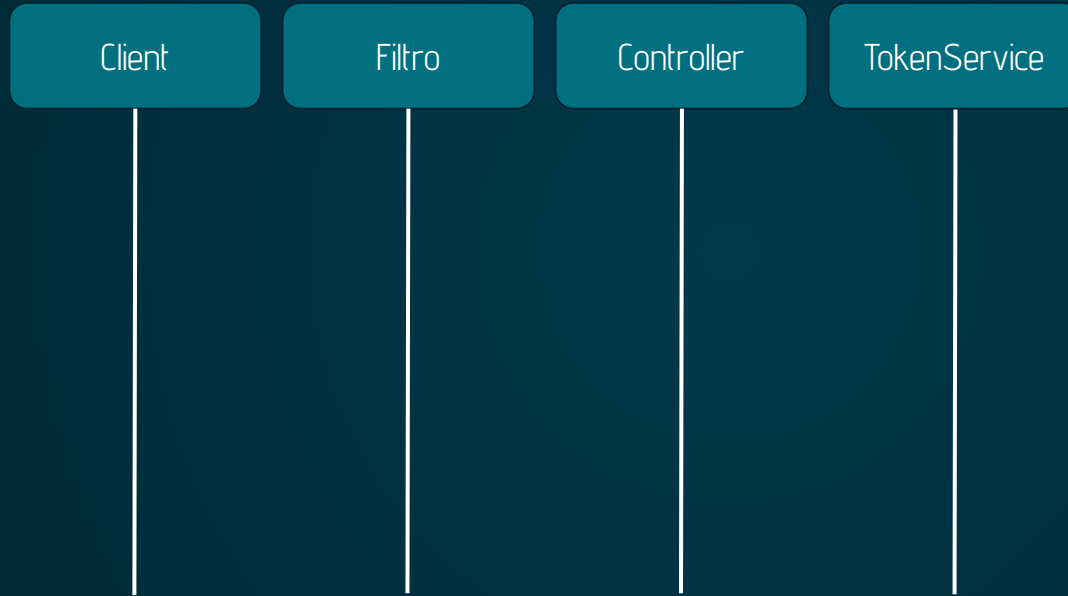


2. Outras Requisições

Uma vez de posse do token, vamos passar esse token como cabeçalho da nossa requisição. Isso precisa ser validado para a requisição ser encaminhada



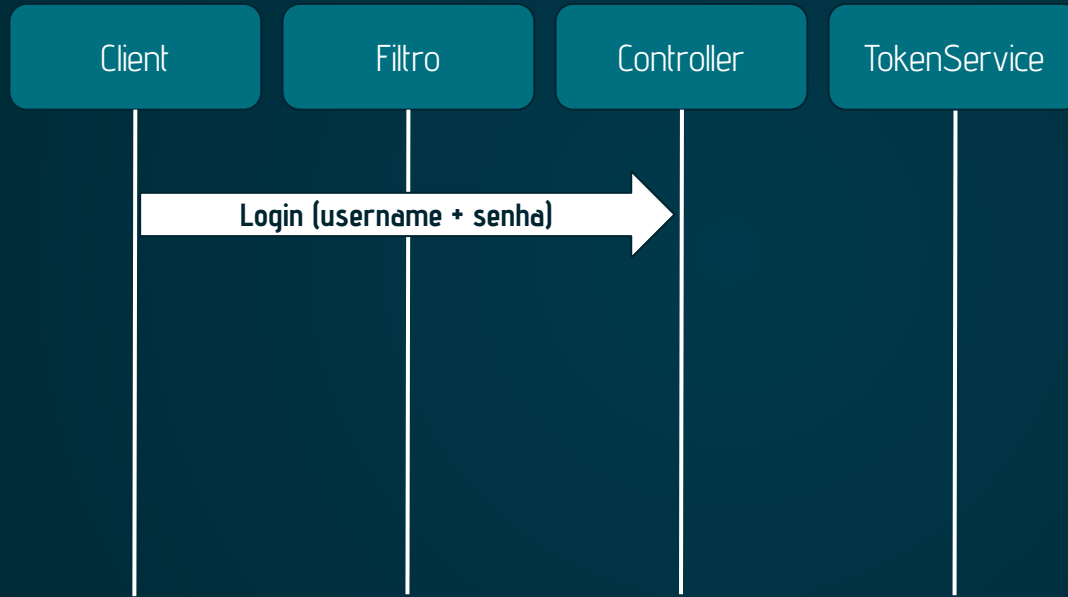
Mas como gerar todo esse fluxo?



[isi]



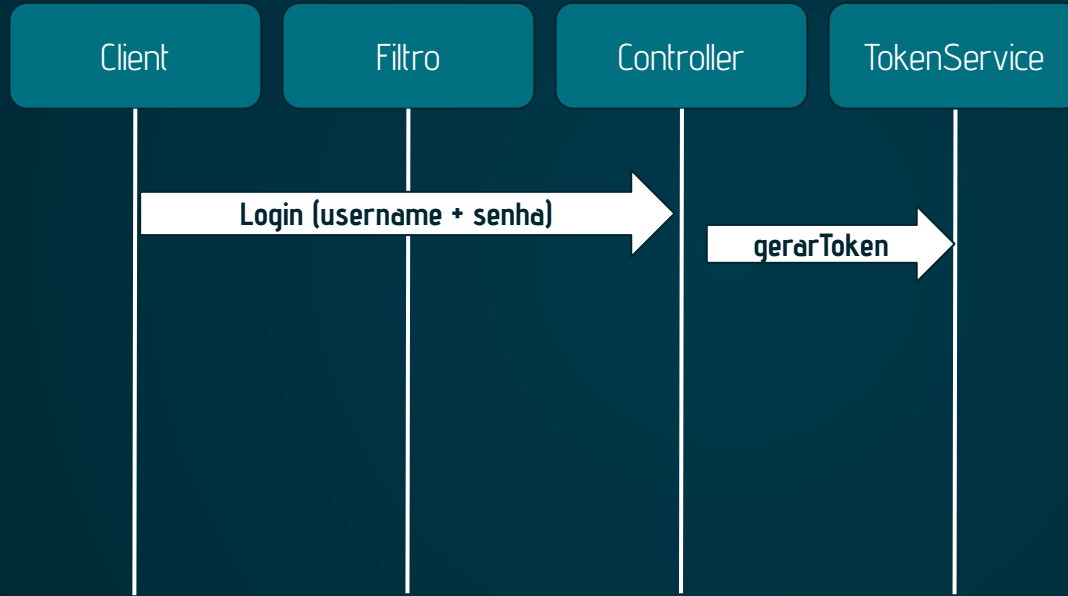
Mas como gerar todo esse fluxo?



[isi]



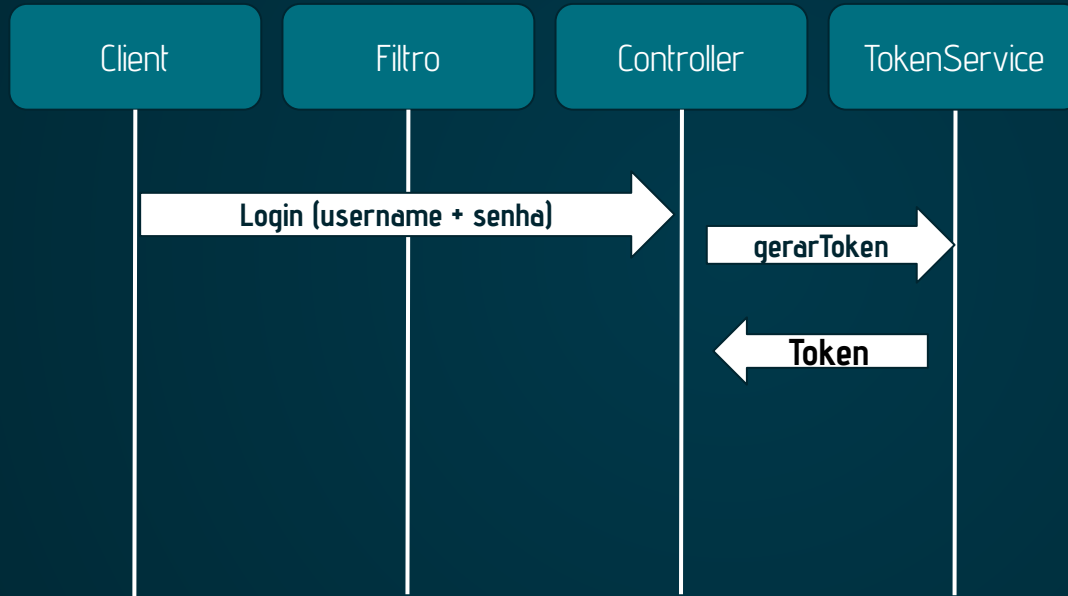
Mas como gerar todo esse fluxo?



[isi]



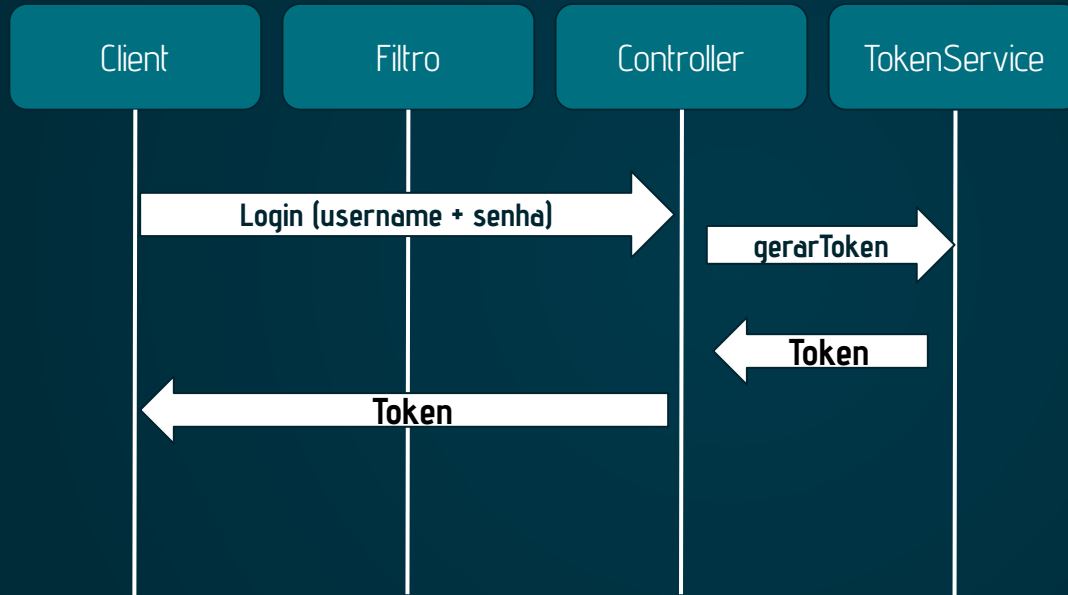
Mas como gerar todo esse fluxo?



[isi]



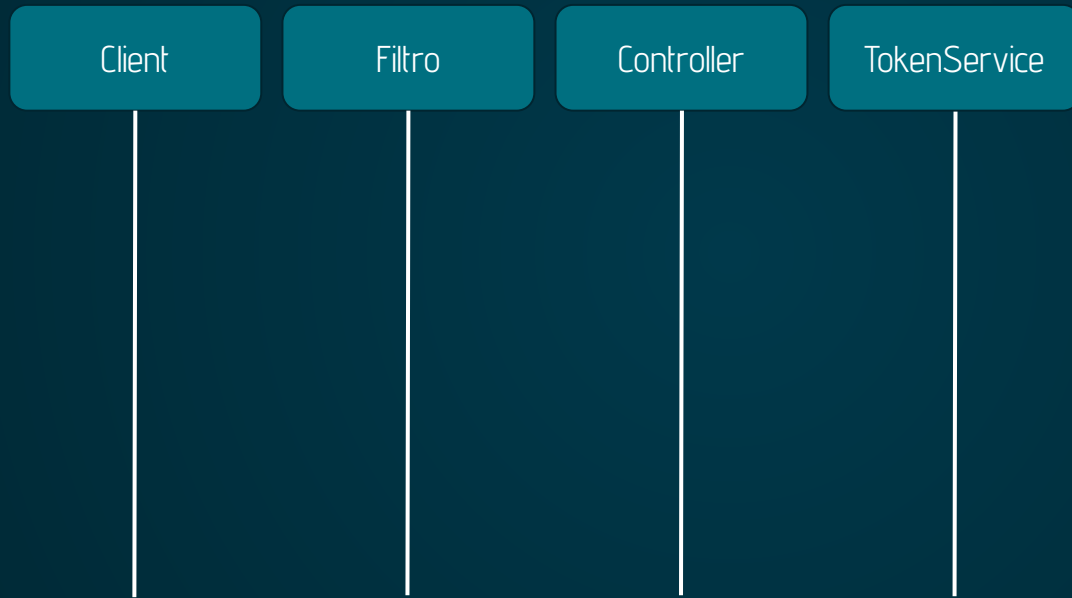
Mas como gerar todo esse fluxo?



[isi]



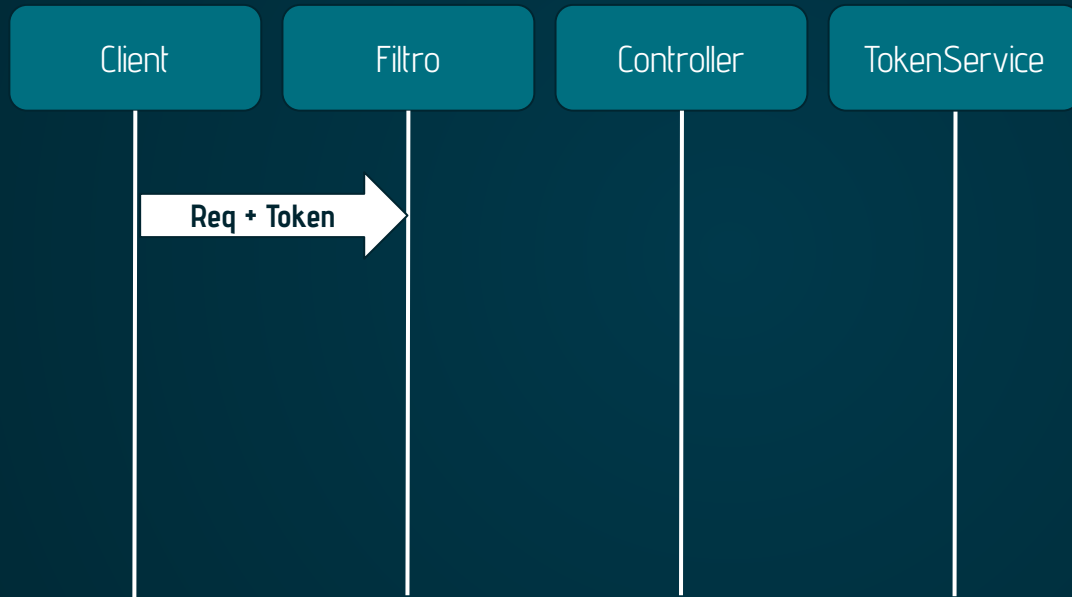
E as requisições autorizadas?



[isi]



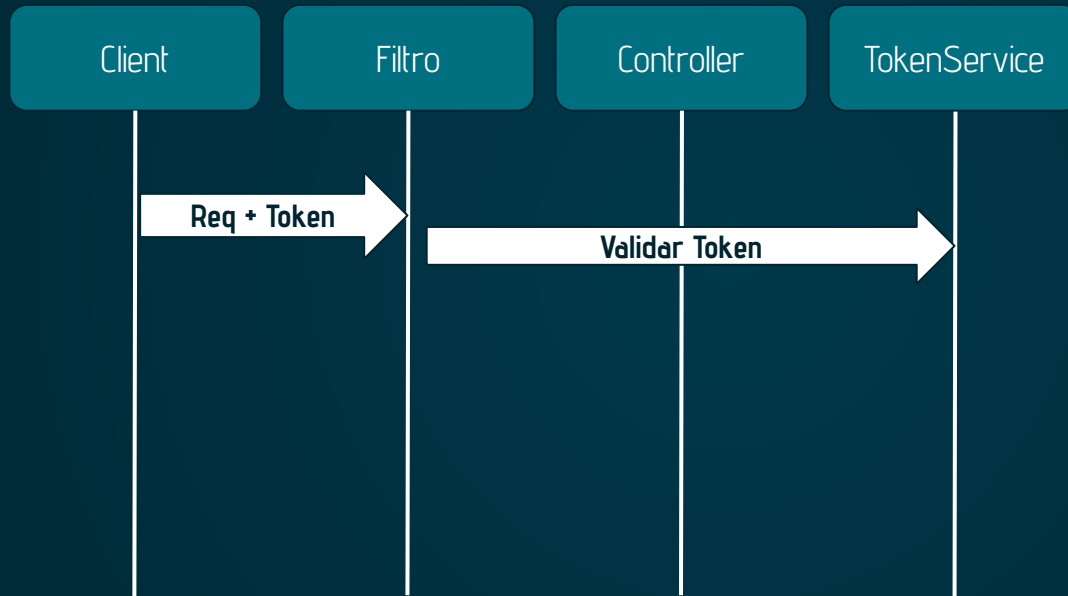
E as requisições autorizadas?



[isi]



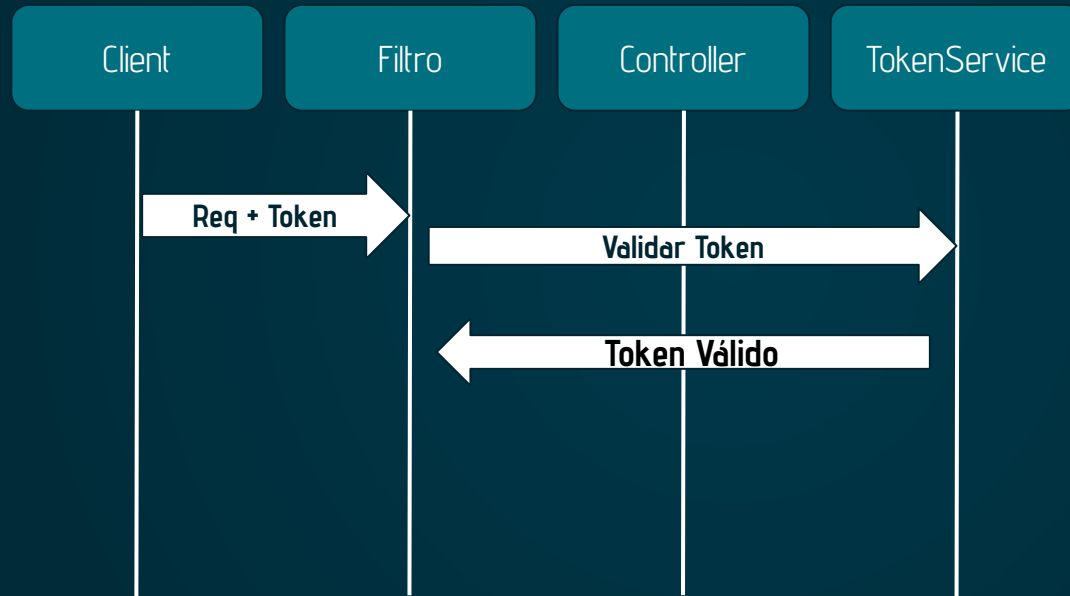
E as requisições autorizadas?



[isi]

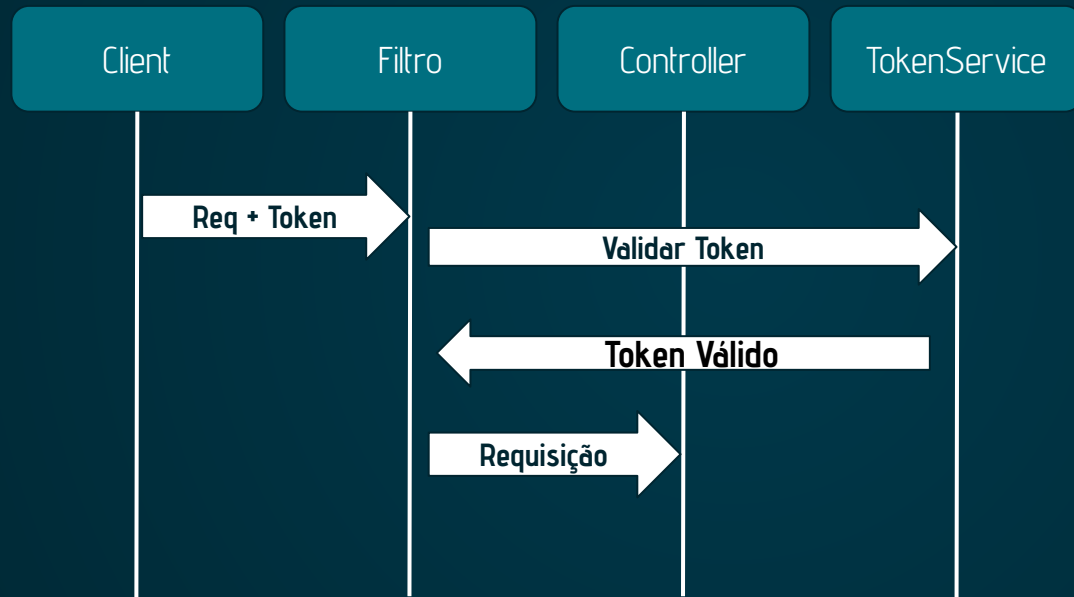


E as requisições autorizadas?



[isi]

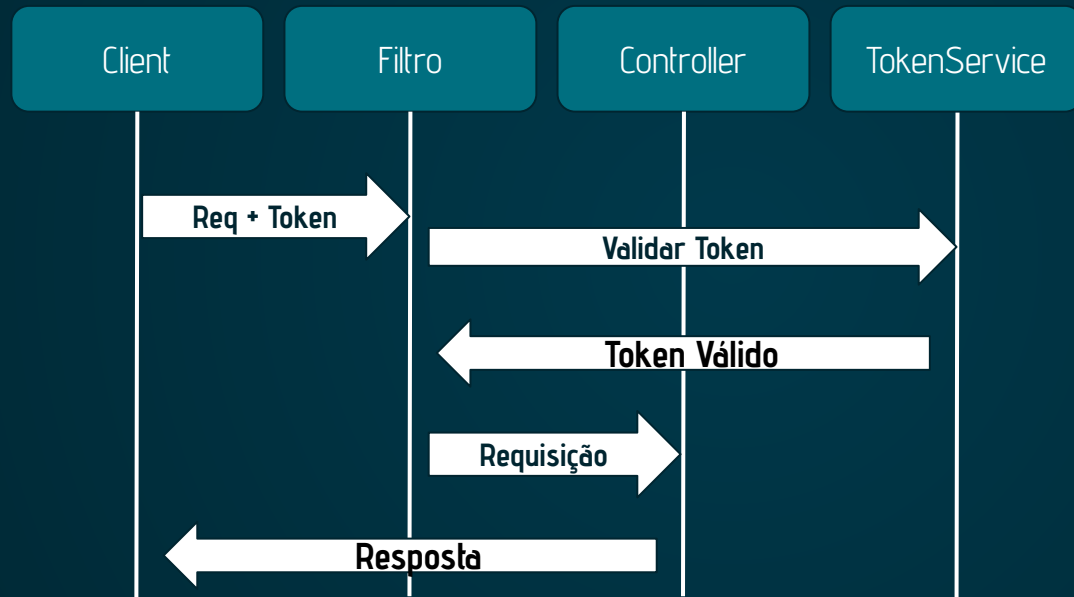
E as requisições autorizadas?



[isi]

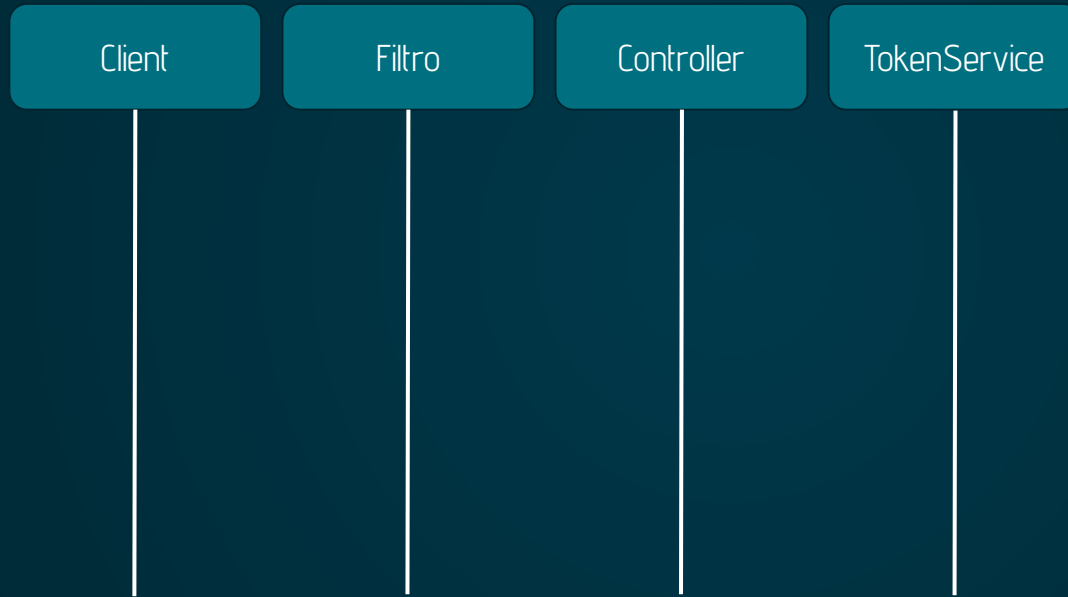


E as requisições autorizadas?



[isi]

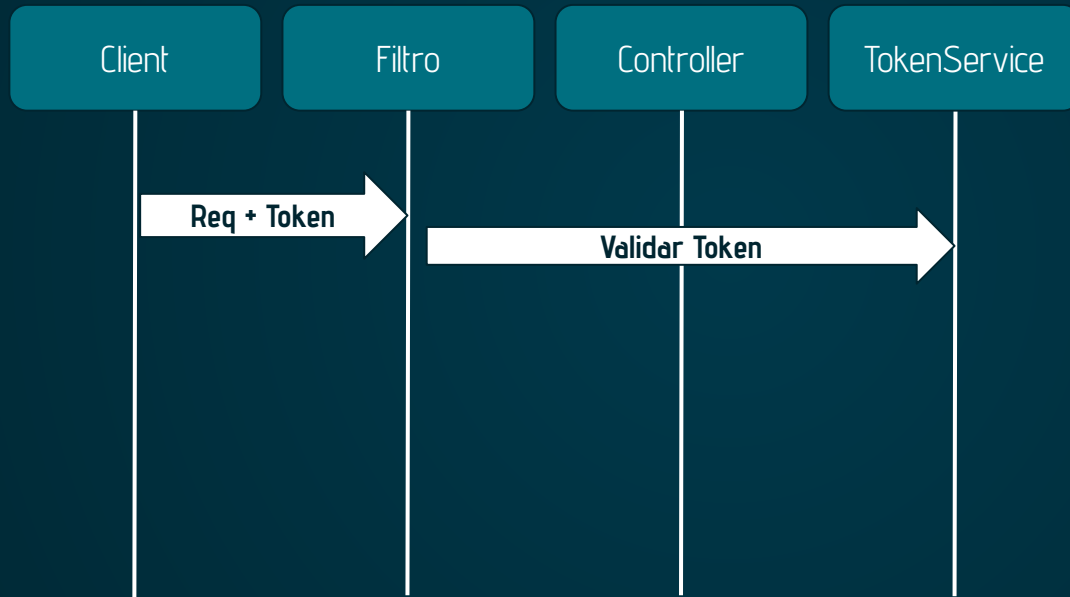
“Barrando” Requisições



[isi]

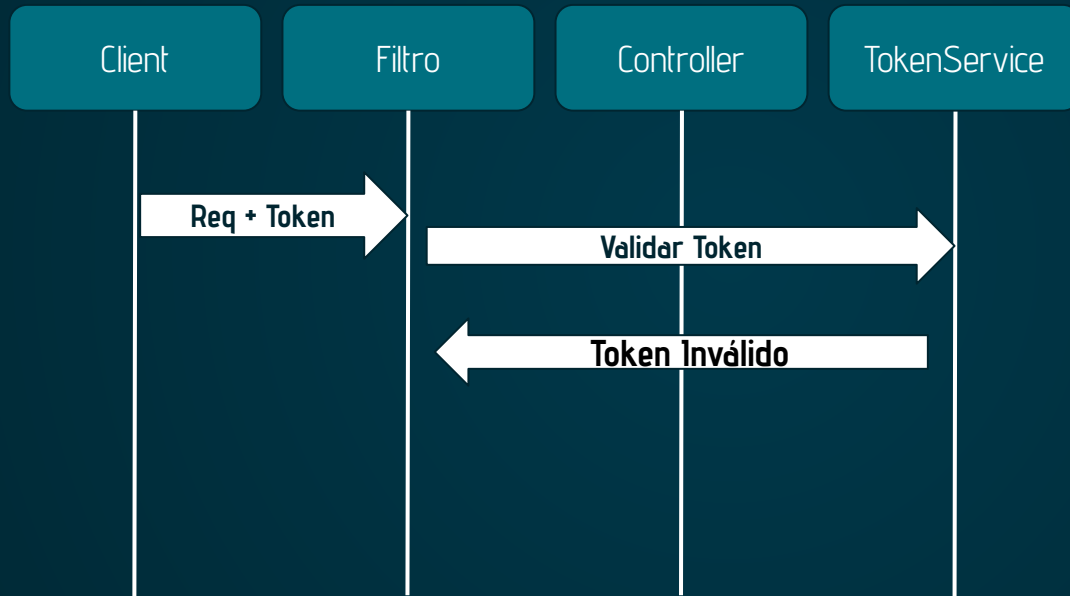


“Barrando” Requisições



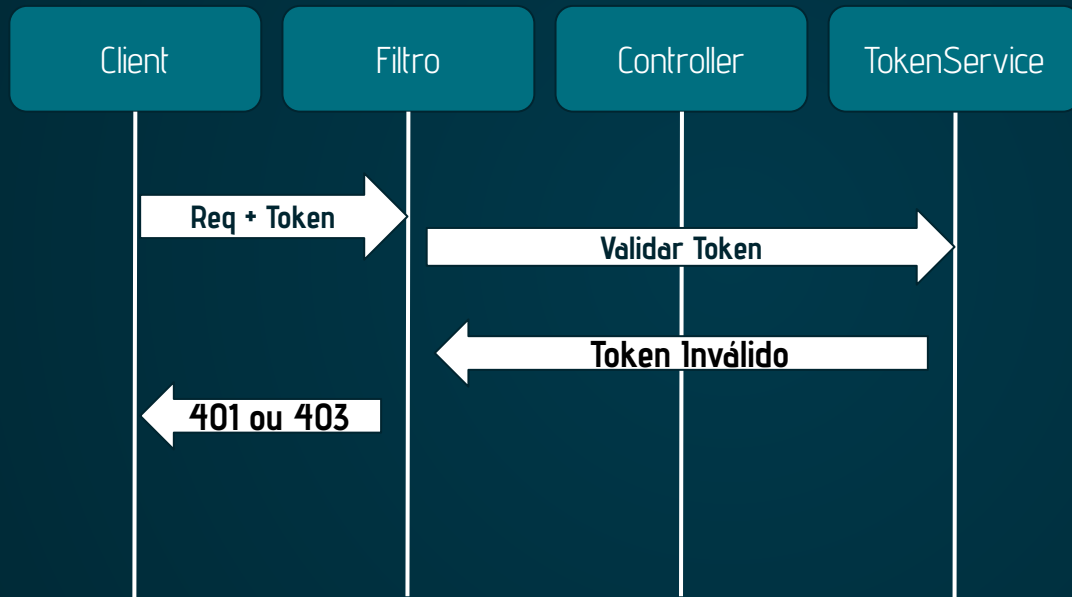
[isi]

“Barrando” Requisições



[isi]

“Barrando” Requisições



[isi]



[isi]

01

Protegendo nossa API

Vamos minimamente entender
como liberar acesso e barrar
acesso aos Endpoints



[isi]

02

Validando nosso Usuário

Uma vez nosso usuário autenticado,
vamos identificá-lo através de um token
para que ele possa ser autorizado a
acessar os Endpoints



[isi]

03

Criando perfis

Vamos criar agora perfis de acesso para que nossos usuários tenham acesso a alguns Endpoints e sejam barrados em outros



[isi]

04

Configurando o CORS

Quando usamos Spring Security o CORS não é apenas uma nova anotação nos controllers. Precisamos configurar o acesso e a nossa API para “liberar” o CORS

