

Enviando token para o servidor

Embora tenhamos gerado o token do FCM na nossa App Android, para que o nosso servidor consiga notificar a App, é necessário enviar o token para ele. Em outras palavras, precisamos implementar uma funcionalidade que pegará o token assim que ele for gerado e enviará para o servidor. Para isso, no final do `onTokenRefresh()`, chame o método `enviaTokenParaServidor()` e mande o token como parâmetro. Lembrando que o Android Studio irá reclamar, ou seja, use o atalho **Alt + Enter** para que o Android Studio crie automaticamente o método com o parâmetro já settado.

Criando a call que vai enviar o token

Com o método criado, basta apenas criar a call que ficará responsável em enviar o token para o servidor. Entretanto, essa call refere-se ao recurso do dispositivo, ou seja, ao invés de utilizar o service do aluno crie a interface `DispositivoService` que ficará responsável em criar essa call. Em seguida, crie a assinatura `enviaToken()` que enviará o token por meio de um **POST** com a URL `"firebase/dispositivo"`, o token será identificado pelo parâmetro `String token`.

Adicionando header no Retrofit

A definição do token nesta requisição é um **header**, para identificá-lo como um header basta apenas utilizar a annotation `@Header`, porém, da mesma forma como vimos no `@Path`, podemos mandar diversos headers na requisição, ou seja, é necessário adicionar uma identificação. Para isso, envie como parâmetro a `String "token"`. Por fim, basta apenas definir o retorno, nesse caso, não esperamos nenhum retorno, logo, pode ser do tipo `Void`.

Fazendo uso da DispositivoService

Agora que temos a call, faça a chamada dela dentro do método `enviaTokenParaServidor()`. Para isso, utilize a classe `RetrofitInicializador` novamente, porém, ao invés de utilizar o método `getAlunoService()`, crie agora o método `getDispositivoService()` e faça o mesmo processo que fizemos para criar e retornar `AlunoService`, mas, agora retorne a interface `DispositivoService`.

Executando a call

Agora chame o método `enviaToken()` e atribua o seu retorno para uma `Call`. Com a call em mãos, basta apenas realizar executar o método `enqueue()`. Por fim, adicione um log dentro do `onResponse()` que exibirá o token que enviamos, dessa forma veremos se o token foi realmente enviado. Além disso, adicione também o log no `onFailure()` que mostrará a mensagem `getMessage()` do parâmetro `Throwable`.

Testando o envio do token

Embora estejamos realizando todos os passos, para que seja chamada essa call será necessário criar um novo token do FCM. Para isso, desinstale a App e execute novamente. Após execução, veja se o token é gerado e se aparece o log do `onResponse()`.

