

Criando a Call de inserção de aluno

Agora que temos o Retrofit configurado precisamos criar o service que ficará responsável em criar as Calls para realizar as requisições baseadas na entidade aluno da nossa App. Para isso crie a interface `AlunoService` no pacote `br.com.alura.agenda.service`. Em seguida, declare a assinatura `insere()` que retorna uma `Call`, conforme abaixo:

```
package br.com.alura.agenda.service;

import retrofit2.Call;

public interface AlunoService {

    Call insere();
}
```

Definindo o endereço da URL

Após definir a assinatura para a `Call` é necessário definir a URL que ela realizará a requisição. Porém, conforme visto em aula, para declararmos um endereço para a URL precisamos primeiro indicar o verbo do HTTP. Neste cenário, queremos inserir uma informação no servidor, portanto, o verbo que utilizaremos será o **POST**. Para isso utilize a annotation `@POST` e envie o endereço via parâmetro da annotation por meio de uma String. Vimos que para enviar o aluno precisamos utilizar o endereço http://seu_ip:8080/api/aluno (http://seu_ip:8080/api/), sabemos que a URL base é http://seu_ip:8080/api/ (http://seu_ip:8080/api/), portanto, basta apenas enviar a String `"aluno"`.

Adicionando aluno na requisição

Embora a `Call` já tenha uma URL definida em nenhum momento estamos enviando o aluno. Portanto envie o parâmetro `Aluno aluno` na assinatura `insere()`. Em seguida, é necessário indicar o que esse parâmetro significa para a requisição, ou seja, se faz parte do cabeçalho (header), caminho da URL (path) ou corpo (body). Neste caso, o aluno será transformado em JSON e será enviado no corpo da requisição! Adicione a annotation `@Body` para indicar que o parâmetro fará parte do corpo da requisição.

Definindo o retorno da requisição

As `Call`s do Retrofit para serem executadas com sucesso exigem que seja informado o retorno que se espera da requisição conforme vimos em aula. Portanto, vamos adicionar o retorno! Para adicionar um retorno basta apenas indicar via **generics**. Neste caso, não esperamos nenhum retorno, ou seja, adicione a classe `Void` como valor do generics.

