

## Disparando eventos com Event Bus

Agora que estamos salvando os alunos no SQLite assim que os recebemos, temos que avisar a activity que ela pode atualizar a lista que existe novas informações!

Para isso, usaremos o Event Bus que é uma biblioteca que permite entidades da App como Services, Activities, Fragments entre outras se comunicarem entre elas por meio de eventos.

### Adicionando o Event Bus no projeto

Para inserir o Event Bus podemos fazer o mesmo procedimento que foi feito nas demais bibliotecas que adicionamos no decorrer do curso, ou seja, por meio do menu **File > Project Structure > App > Dependencies**. Em seguida clique na cruzinha verde e escolha a opção **Library dependency**. Então, insira o valor "eventbus" e escolha a biblioteca `org.greenrobot:eventbus:3.0.0`. Clique em **OK** e depois em **OK** novamente e aguarde o gradle sincronizar com o projeto.

### Disparando evento

Agora que temos o Event Bus no nosso projeto podemos iniciar o processo para disparar o evento para a activity que tem a lista de alunos. Para isso crie um objeto do tipo `EventBus` por meio do método estático `getDefault()` da própria classe `EventBus`. Em seguida, com o objeto do `EventBus` chame o método `post()` enviando como parâmetro o seguinte argumento `new AtualizaListaAlunoEvent()`. Essa classe ainda não existe! Portanto, crie ela no pacote `br.com.alura.agenda.event` por meio do atalho **Alt + Enter**. Lembrando que essa classe não possui nenhuma implementação, ou seja, servirá apenas como parâmetro para indicar que o evento foi disparado.

### Recebendo evento

Após enviar o evento precisamos configurar a `ListaAlunosActivity` para que ela se inscreva nesse evento e o receba! Portanto, vá até o `onCreate()` da `ListaAlunosActivity` crie uma instância do `EventBus` conforme fizemos anteriormente, ou seja, pelo método `getDefault()`, e então, chame o método `register()` enviando como argumento o parâmetro `this`, dessa forma, fazemos com que a activity seja registrada para receber eventos do Event Bus.

### Se inscrevendo no evento

Agora que a activity foi registrada no Event Bus, crie o método `atualizaListaAlunoEvent()` que devolva o retorno `void` e que receba como parâmetro a classe `AtualizaListaAlunoEvent`, além disso, anote o método com a annotation `@Subscribe`. Agora quando o evento for disparado, todo o código dentro desse método será executado! Portanto, chame agora o método que carrega a lista de alunos, ou seja, o `carregaLista()`.

### Detalhe da modificação de View em Thread paralela

Um detalhe importante para o código ao qual implementamos, é que estamos modificando uma view e, quando fazemos uso do event bus, os métodos que são inscritos não necessariamente são executados na thread principal, ou seja, na thread da activity. Em outras palavras, apenas a activity pode modificar a view e sua hierarquia, portanto, da forma que está implementado atualmente, o Android lançará uma exception! Para evitar esse problema precisamos apenas indicar que esse método deve ser executado pela thread principal, isto é, a thread main. ou seja, adicione o parâmetro `threadMode = ThreadMode.MAIN` na annotation `@Subscribe`.

### Testando a App

Agora que o event bus foi implementado, execute novamente a App, e então, faça alguma alteração ou inserção no servidor e veja se é replicada na App sem ao menos mexê-la.

