Desenvolvimento de Software para Dispositivos Móveis

Aula 11 - REST

Professor: Anderson Almada

Revisão REST

Método	URI	Utilização
GET	/clientes	Recuperar os dados de todos os clientes.
GET	/clientes/id	Recuperar os dados de um determinado cliente.
POST	/clientes	Criar um novo cliente.
PUT	/clientes/id	Atualizar os dados de um determinado cliente.
DELETE	/clientes/id	Excluir um determinado cliente.

Retrofit

- Importe no build.gradle (Module: app)

```
dependencies {
    ...
    implementation 'com.squareup.retrofit2:retrofit:2.6.1'
    implementation 'com.squareup.retrofit2:converter-gson:2.6.1'
}
```

- Clique em **Sync now** para sincronização do projeto com as dependências

Crie uma instância de configuração

- Crie a classe **RetrofitClientInstance**
  - Singleton

- **Atributos**
  - **static** Retrofit retrofit = **null**;
  - **static** String BASE_URL = "https://jsonplaceholder.typicode.com";

retrofit - instância única
BASE_URL - local da api

Crie uma instância de configuração

- Crie o seguinte método

```
public static Retrofit getRetrofitInstance() {
    if (retrofit == null) {
        retrofit = new retrofit2.Retrofit.Builder()
                .baseUrl(BASE_URL)
                .addConverterFactory(GsonConverterFactory.create())
                .build();
    }
    return retrofit;
}
```

Crie uma instância de configuração

- O GsonConverterFactory representa a conversão dos objetos (existem vários tipos)

- Código da classe
  - https://pastebin.com/2WAM0KJS

Crie o modelo

- Cria a classe User
  - Atributos
    - Integer id
    - String name
    - String username

- Código da classe
  - https://pastebin.com/kBxk7ZQ2

Crie a interface do serviço

- Para utilizar o retrofit, é preciso definir a interface de comunicação dele com o servidor. Para o exemplo deve ser criado a interface **UserService**

- Essa interface contém todo os métodos de comunicação com a api
  - GET
  - POST
  - PUT
  - DELETE
  - ...

Crie a interface do serviço

- Métodos
  - **GET ALL**

```
@GET("/users")
Call<List<User>> getUsers();
```

  - **GET ID**

```
@GET("/users/{id}")
Call<User> getUser(@Path("id") Integer id);
```

Crie a interface do serviço

- Métodos
  - **POST**

```
@POST("/users")
Call<User> postUser(@Body User user);
```

  - **PUT**

```
@PUT("/users/{id}")
Call<User> putUser(@Body User user, @Path("id") Integer id);
```

  - **DELETE**

```
@DELETE("/users/{id}")
Call<User> deleteUser(@Path("id") Integer id);
```

Crie a interface do serviço

- Código da interface
  - https://pastebin.com/y9aFQyA1

Activity principal chama os métodos da interface

- A activity principal é responsável por chamar a interface para comunicação do cliente com o servidor

```
UserService service = RetrofitClientInstance
                      .getRetrofitInstance()
                      .create(UserService.class);
```

- Todos os métodos da interface possuem um padrão de chamada:

```
Call<List<User>> callGetAll = service.getUsers();
```

Activity principal chama os métodos da interface

```
Call<List<User>> callGetAll = service.getUsers();
callGetAll.enqueue(new Callback<List<User>>() {
  @Override
  public void onResponse(Call<List<User>> call, Response<List<User>> response) {   }

  @Override
  public void onFailure(Call<List<User>> call, Throwable t) {   }
});
```

Activity principal chama os métodos da interface

- Dentro do onResponse, você faz a devida manipulação que você queira:

```
public void onResponse(Call<List<User>> call, Response<List<User>> response) {
        for(User user : response.body()) {
                Log.i("GET ALL", user.toString());
        }
}
```

Activity principal chama os métodos da interface

- Código completo da Activity
  - https://pastebin.com/Hb39wsMs

Exercício

- Crie um cliente REST para uma api que retorna cursos. Segue os dados:
  - Ip: 192.168.0.114
  - Porta: 8080
  - API: /api/cursos

- Cada **Curso** é composto de:
  - Id
  - Nome
  - Duração

Exercício

- Crie o post para enviar os dados
  - Crie dois campos (nome e duração)
  - Crie um botão (POST)
  - Pressionou o botão, envia os dados

- Crie o get com id
  - Crie um campo (id)
  - Crie um botão (GET)
  - Pressionou o botão, aparece um Toast contendo os dados

Exercício

- Crie o put
  - Aproveite os campos já criados
  - Crie um botão (PUT)
  - Pressionou o botão, envia os dados e aparece um Toast contendo os dados atualizados

Exercício

- Crie o delete
  - Aproveite o campo id criado
  - Crie um botão (DELETE)
  - Pressionou o botão, aparece um Toast dizendo **OK** se foi sucesso, **FAIL** caso contrário

Exercício

- Crie o get all
  - Crie um botão (GET ALL)
  - Crie um recyclerView
  - Pressionou o botão, manda uma requisição para retornar todos os cursos e mostra na lista os nomes dos cursos

Link importante

- https://square.github.io/retrofit/

- https://blog.matheuscastiglioni.com.br/consumindo-web-service-no-android-com-retrofit/

Dúvidas??

E-mail: almada@crateus.ufc.br