How to import retrofit in android studio

I'm not robot

reCAPTCHA

Continue

I'm not robot

reCAPTCHA

Many sites have their own APIs to facilitate access to their data. Currently, the most common version is JSON. Data in the form of XML and other formats can also be found. The renovation library simplifies the interaction with the site's REST API by taking on some of the household chores. The authors of the renovation library are square developers, who have written many useful libraries, such as Picasso, Okhttp, Otto. Homepage - Library is convenient for requesting various web services with GET, POST, PUT, DELETE commands. It can work in asynchronous mode, eliminating unnecessary code. Basically, you will have to work with GET and POST methods. If you create your own API, you'll use other commands. Connects the standard. Resam 2.x automatically connects the OkHttp library and does not need to be written separately. The library can work with GSON and XML using special converters that need to be specified separately. The converter is then added using addConverterFactory in the code. addConverterFactory (GsonConverterFactory.create:com.squareup.retrofit2:converter-gson-jackson: com.squareup.retrofit2:converter-jackson Moshi: com.squareup.retrofit2:converter-moshi Protobuf: com.squareup.retrofit2:converter-protobuf Wire: com.squareup.renovation2:converter-wire-simple-XML: com.squareup.retrofit2:converter-simplexml Scalars and String: com.squareup.retrofit2:converter-scalars Also you can create your own converter, by implementing the interface based on the abstract class Converer.Factory. You can connect multiple converters (order is important). Renovation renovation - Retrofit.Builder() .baseUrl (; .addConverterFactory (GsonConverterFactory.create))build(); If you want to change the format of a JSON object, you can do so with GsonConverterFactory.create (): Gson gson q new GsonBuilder () .setDateFormat (yyyyy-MM-Dd'THH:mm:ss) .create(); Renovation of the renovation - new renovation. Builder.baseUrl (.addConverterFactory(GsonConverterFactory.create(gson)) .build(); service - retrofit.create (APIService.class); The basic URL always ends with a slash/. It is defined in the basicUrl method. You can specify the full URL in the query, and then the basic URL will be ignored: API, Public Interface @GET (Call-l;Users (); POJO (Plain Old Java Object) or Model Class - json-response from the server should be implemented as model Retrofit - a class for processing results. It must specify the basic address in the basic methodUrl () Interface - the interface for managing addresses, using GET commands, POST, etc. The first task. POJO Task first is to examine the response structure of the site in the form of JSON (or other formats) and create a Java class based on it in the form of POJO. POJO is more convenient to create with ready-to-use web services in automatic mode. Or you can create a class yourself if the structure is not too complex. Annotations are often used in the classroom. Sometimes they are necessary, sometimes they can be missed. In some cases, annotations help avoid errors. The list of annotations depends on the type of converter used, and the list can be accessed in the relevant documentation. The second task. The Task Two interface is to create an interface and specify the name of the method. Add the necessary options if you need them. The interface asks for query commands for the server. The team is combined with 'lt;/Users'(baseUrl) and the full path to the page. The code can be simple and complex. You can see the examples in the documentation. Requests are placed in the generalized call class with the desired type. import renovation2. Call; Public interface APIService @GET call-lt;Repo-repo (); In most cases, you'll return a Call-t-item with the right type, such as User.'Call. Dynamic settings use curly hooks (users/user/rest) that match the right values get_all_cats @GET. / / Team on the list getAllCats server (); / your AbstractSation annotation code @GET ()GET request for a basic address @POST. @Path Sperm to replace the @GET ()GET request for a basic address @POST. @Path Sperm to replace the @GET ()GET request for a basic address @POST. @Path Sperm to replace the @GET ()GET request for a basic address @POST. @Path Sperm to replace the @GET annotation code ()GET request of a For example, The username will be set in the endpoint address @Query Sals the name of the query key with the @Body Uses in POST calls (from the Java object in JSON-string) @Header Shos the title with the @Headers Sals @Multipart all titles together Uses When downloading files or images @FormUrlEncoded Uses a pair of name/value in POST queries @FieldMap uses when using a pair of name/value in POST @Url D queries to support dynamic addresses @Query Abstract is @Query useful when queries are useful Response User Tt./RepoSettings. Suppose the site has an additional option to the query, which displays a list of items under a sorted form: . This is a simple example and we can put a query with the option in the unchanged interface. @GET call getAllCats (category roduits/cats?); If you don't need to manage the sorting, you can leave it in the code and it will be used by default. This option can be annotated and managed by program in the code. @GET call getAllCats (@Query (category) int categoryId; Sort we leave as is, and the category is transferred to the method settings under the nameId, with the ability to provide an annotation. Call getAllCats () catAPIService.getAllCats (5); The request will be made in the form of a . In a method, you can specify multiple query settings. @Path request may have modifiable parts of @Path path. that will be set to go. @GET (/users/username) Call getUser (@Path (username) String userName); @Headers Annotation example @Headers that allows you to put all the titles together. @Headers (Hide-Control: max-age 640,000, User-Agent: My-App-Name @GET (some/endpoint) @Multipart Example of @Multipart annotation when downloading files or images: @Multipart/CatsCall-Response (@Part (description) Chain description, @Part (image) Request image) @FormUrlEncoded example of @FormUrlEncoded annotation: @FormUrlEncoded @POST @FieldMap (some/endpoint) Call some Names of Call, String, @FieldMap @Url Example of @Url annotation: Public interface UserService q @GET public call @Url Renovation For a synchronous request, use the call.execute method.), call.enqueue for asynchronous. The object of the server request is created in the simplest case as follows the public static final chain BASE_URL the Renovation - new renovation. Builder.baseUrl(BASE_URL).addConverterFactory(GsonConverterFactory.create()) .build(); Therefore, we received a Retrofit object that contains a basic URL and the ability to convert JSON data using the specified Gson converter. Next in its create() method we specify our interface class with the requests on the site. UserService userService - retrofit.create (UserService.class); We then receive the call and call the enqueue method (for asynchronous call) and create Callback for it. The request will be made in a separate thread, and the result will come to Callback in the main stream. Therefore, the renovation library will apply, receive a response and analyze the response, putting the data on the shelves. All you have to do is call the right model methods to extract the data. Most of the work takes place in onResponse(), errors appear on LaFailure (wrong server address, incorrect data format, incorrect model class format, etc.). The HTTPS server (e.g. 404) is not a mistake. OnResponse() is always called, even if the request has failed. The Response class has a practical isSuccessful method for successfully processing an application (codes 200s). In erroneous situations, you can deal with an error in the errorBody method./File,/String, SomeResponseResponseBody. Other useful response methods. code()- http body response code()- the answer itself in the form of a line, without serialization of headers ()- () - (or null) raw)- raw HTTP-response Can write such a design. code 200 si (response.isSuccessful()) q … / code for a successful case q else q switch (response.code(response.code)q case 404: / page is not found. See pause below; box 500: / error on the server. Interceptors can be entered into the library to modify the headers using the Interceptor class from OkHttp. First, create an interceptor object and transfer it to OkHttp, which in turn must be explicitly connected to Retrofit.Builder via the customer method. For example, interceptors try to manage the query title, for example, to work with authorization tokens in the Authorization header. Customer OkHttpClient - new OkHttpClient(); customer.interceptors ().add (new Interceptor)and @Override public response (Chain chain chain) launches IOException q Request original q chain.request ();// Set up requests for the original. NewBuilder () .header (Accept, application/json).header (Authorization, auth-token). Answer - chain.proceed (request); return response; } }); Renovation renovation - Retrofit.Builder() .baseUrl (; .client .build(); HttpLoggingInterceptor Library HttpLoggingInterceptor is part of OkHttp, but comes separately from it. The interceptor should be used when you really need to study the server's response logs. In fact, the library is the network analog of the usual LogCat. Connect 'com.squareup.okhttp3:journal-interceptor:3.9.0' Connect the interceptor to the web client. Add it after other interceptors to catch all the messages. There are several levels of data interception: NONE, BASIC, HEADERS, BODY. The last option is the most informative, use it carefully. With large data streams, the information will forget the entire screen. Use intermediate options. HttpLoggingInterceptor loggingInterceptor - new HttpLoggingInterceptor(); Only in debugging mode if (BuildConfig.DEBUG) q loggingInterceptor.setLevel (HttpLoggingInterceptor.Level.BODY); OkHttpClient okClient - new OkHttpClient.Builder() .addInterceptor (new ResponseInterceptor).addInterceptor (loggingInterceptor) .build(build); RxJava Library developers are very fond of responsive programming and have made many efforts to integrate into the RxJava library. build.gradle implementation 'com.squareup.retrofit2:adapt-rxjava:2.5.0' Renovation renovation - new renovation. Builder.baseUrl (baseUrl); .addCallAdapterFactory (RxJavaCallAdapterFactory.()) .addConverterFactory (GsonConverterFactory.create()); Improved additional 2.x playback. Examples for GitHub Renovation 2.x. POST, PUT, DELETE Retrofit 2.x. Converter XML Retrofit 2.x. Converter ScalarsConverterFactory and POST Currency Rate of the Central Bank of Russia. Renovation, RENOVATION XML on Kotlin with the use of corutina. JSONPlaceHolder Retrofit, Reddit (Kotlin) TheCatAPI - Cats as a Service, Everyday is Caturday (Retrofit, Kotlin) Advertising