Take a look at the call(s) that you will need for your activity to get a better idea of what's going on and if there are any issues you can let me(Jasper) know and I'll fix them asap. General rule of thumb when using the calls:

If you're just calling a set/update call then the only thing you really need from the response is whether it was successful, and if not the err message that goes along with.

If you are making a call to fetch data then the data will be returned to the on response listener, and any functionality that needs that data will need to be coded within that listener.

If the data is being stored as an object(AKA a class inside firebase) then the return key for that object will be formatted as: ClassName_obj(see second example below).

If the data is not being stored as an object, but rather as key value pairs then it will be returned as 'data' and once you pull that from the response it can be treated as a hasMap.

If your call gets a specific value rather than a document as a whole there is no specific way for me to know the best way to name the return value, so in that case just check the function before implementing it to figure out what you need to get from response

All errors that are returned in response will be labeled as "err".

**Make sure you initialize the APIMain class before attempting to make a call

below are a couple of working examples

```
APIMain APIServer = new APIMain();
                         // player id↓↓↓
APIServer.getPlayerInfo("3ou2WaY8MwpJbeesN0UO", new ResponseCallback() {
    @Override
    public void onResponse(Map<String, Object> response) {
         Log.i("login", response +""); // check console to see what response
looks like
         Log.i("login", response.get("success")+""); // this is how you get the
success value(true/false)
         Map<String, Object> data = (Map<String, Object>) response.get("data");
// this is how you access the data(if it's stored as key value pairs)
```

```
         mEmail.setText(data.get("email")+""); // anything that relies on this
info must be put inside this function, as the data DNE outside this scope*
    }
});
```

- *technically the data does "exist" outside of the onResponse function, however if you init a var outside of onResponse, set the value of that var inside, and then later try to access that value it will still apear as uninitialized because the callback is being called after the async proccess from firebase has ended, meaning that in that time your code has already finished running and the values that were fetched will not have been present during that runtime

```
APIMain APIServer = new APIMain();
APIServer.getQRcode(code_id, new ResponseCallback() {
    @Override
    public void onResponse(Map<String, Object> response) {
         Log.i("login", response +""); // check console to see what response
looks like
         QRcode code = (QRcode) response.get("QRcode_obj"); // this is how you
access the data(if it's stored as key value pairs)
         int score = code.getScore() // useable as a QRcode object from now on
   }
});
```

I know this seems like a strange setup, but it's the best way to deal with async functions in java without having to deal with multithreading(which I'm not even sure would work for this case), but if anyone has a better idea/way to deal with this then PLEASE let me know