Charakteristik and Services Tables

Mittwoch, 22. März 2023

12:32

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Actions** | **Admin**  **/User**  **Code** | **Receiving/Sending JSON Code to ESP** | **Android Method**  **BleutoothLeService.** | **Service Constant** | **Characteristic Constant** | **Description** | **Test** |
| Initial User | User | {"cmd": "init\_user", "name": "test-user"} | initUser(String name) | BluetoothLeService.SERVICE\_READ\_WRITE | BluetoothLeService.CHARACTERISTIC\_MESSAGE\_USER | register as a new user and receive a user ID | ok |
| Make Recipe | User | {"cmd": "make\_recipe", "user": 8858, "recipe": "radler"} | makeRecipe(float user, String recipe) | BluetoothLeService.SERVICE\_READ\_WRITE | BluetoothLeService.CHARACTERISTIC\_MESSAGE\_USER | mixes the recipe | ok |
| Reset | User | {"cmd": "reset", "user": 9650} | reset(float user) | BluetoothLeService.SERVICE\_READ\_WRITE | BluetoothLeService.CHARACTERISTIC\_MESSAGE\_USER | Reset the machine so that it can make a new cocktail |  |
| Define Pump | Admin | {"cmd": "define\_pump", "user": 0, "liquid": "water", "volume": 1000, "slot": 1} | adminDefinePump(String liquid, float volume, int slot) | BluetoothLeService.SERVICE\_READ\_WRITE | BluetoothLeService.CHARACTERISTIC\_MESSAGE\_ADMIN | add a new pump to ESP | ok |
| Add liquid | User/Admin | {"cmd": "add\_liquid", "user": \*/0, "liquid": "water", "volume": 30} | addLiquid(float user, String liquid, float volume) | BluetoothLeService.SERVICE\_READ\_WRITE | BluetoothLeService.CHARACTERISTIC\_MESSAGE\_User | Adds liquid to the cocktail | ok |
| Define Recipe | User/Admin | {"cmd": "define\_recipe", "user": \*/0, "name": "radler", "liquids": [["beer", 250], ["lemonade", 250]]} | defineRecipe(float user, String name, ArrayList<Pair<String, Float>> liquids) | BluetoothLeService.SERVICE\_READ\_WRITE | BluetoothLeService.CHARACTERISTIC\_MESSAGE\_USER | defines a new recipe or changes an existing recipe | ok |
| Edit Recipe | User/Admin | {"cmd": "edit\_recipe", "user": \*/0, "name": "radler", "liquids": [["beer", 250], ["lemonade", 250]]} | editRecipe(float user,String name, ArrayList<Pair<String, Float>> liquids) | BluetoothLeService.SERVICE\_READ\_WRITE | BluetoothLeService.CHARACTERISTIC\_MESSAGE\_USER | edit an existing recipe | ok |
| Calibrate Pumps | Admin | {"cmd": "calibrate\_pumps", "user": 0} | adminCalibratePumps() | BluetoothLeService.SERVICE\_READ\_WRITE | BluetoothLeService.CHARACTERISTIC\_MESSAGE\_ADMIN | Calibrate all pumps | ok |
| restart | Admin | {"cmd": "restart", "user": 0}  {"cmd": "restart", "user": 0, "factory\_reset": true} | adminRestart(Boolean restFactory) | BluetoothLeService.SERVICE\_READ\_WRITE | BluetoothLeService.CHARACTERISTIC\_MESSAGE\_ADMIN | restarts the machine. If factory\_reset is set to true, all settings will also be deleted. | at this point factroy\_recet and without it behave same |
| clean | Admin | {"cmd": "clean", "user": 0} | adminClean() | BluetoothLeService.SERVICE\_READ\_WRITE | BluetoothLeService.CHARACTERISTIC\_MESSAGE\_ADMIN | Clean the machine | ok |
| Status Liquids | Admin | {"d":970.0,"c":1000.0,"a":200.0,"b":1000.0} | getCharacteristicValue(BluetoothLeService.SERVICE\_READ\_STATE, BluetoothLeService.CHARACTERISTIC\_READ\_LIQUIDS); | BluetoothLeService.SERVICE\_READ\_STATE | BluetoothLeService.CHARACTERISTIC\_READ\_LIQUIDS | Read Pumps Volume | ok |
| Status State | Admin | "ready" | getCharacteristicValue(BluetoothLeService.SERVICE\_READ\_STATE, BluetoothLeService.CHARACTERISTIC\_READ\_STATE); | BluetoothLeService.SERVICE\_READ\_STATE | BluetoothLeService.CHARACTERISTIC\_READ\_STATE | Read current State | ok |
| Status Cocktail | Admin |  | getCharacteristicValue(BluetoothLeService.SERVICE\_READ\_STATE, BluetoothLeService.CHARACTERISTIC\_READ\_COCKTAIL); | BluetoothLeService.SERVICE\_READ\_STATE | BluetoothLeService.CHARACTERISTIC\_READ\_COCKTAIL | Read Current Cocktail | error |
| Status Recipes | Admin | {"abc":[["c",5.0],["b",5.0],["a",5.0]],"ab":[["b",5.0],["a",5.0]]} | getCharacteristicValue(BluetoothLeService.SERVICE\_READ\_STATE,  BluetoothLeService.CHARACTERISTIC\_STATUS\_RECIPES); | BluetoothLeService.SERVICE\_READ\_STATE | BluetoothLeService.CHARACTERISTIC\_STATUS\_RECIPES | Read saved Recipes from Device | ok |

**Wirte Characteristic/Send Message to ESP:** to use the Methods it should be make an Instance of **BluetoothLeService** Class and for every Listener it should calls proper Method in **BluetoothLeService** ofcourse in try catch exception with **JSONException**. To receive Response from ESP call **getCharacterisitcValue(String Service, String Characterisitc)**

**Read Charactersitic/Receive Message from ESP:** To read Characteristic from ESP call **getCharacterisitcValue(String Service, String Characterisitc)**