Console. h

```
#include <Blynk/BlynkConsole.h>

BlynkConsole    edgentConsole;

void console_init()
{
  edgentConsole.init(BLYNK_PRINT);

  edgentConsole.print("\n>");

  edgentConsole.addCommand("reboot", []() {
    edgentConsole.print(R"json({"status":"OK","msg":"resetting device"})json"
"\n");
    delay(100);
    restartMCU();
  });

  edgentConsole.addCommand("config", []() {
    edgentConsole.print(R"json({"status":"OK","msg":"entering configuration
mode"})json" "\n");
    BlynkState::set(MODE_WAIT_CONFIG);
  });

  edgentConsole.addCommand("devinfo", []() {
    edgentConsole.printf(
        R"json({"board":"%s","tmpl_id":"%s","fw_type":"%s","fw_ver":"%s"})json"
"\n",
        BLYNK_DEVICE_NAME,
        BLYNK_TEMPLATE_ID,
        BLYNK_FIRMWARE_TYPE,
        BLYNK_FIRMWARE_VERSION
    );
  });

  edgentConsole.addCommand("netinfo", []() {
    char ssidBuff[64];
    getWiFiName(ssidBuff, sizeof(ssidBuff));

    edgentConsole.printf(
        R"json({"ssid":"%s","bssid":"%s","mac":"%s","rssi":%d})json" "\n",
        ssidBuff,
        WiFi.softAPmacAddress().c_str(),
        WiFi.macAddress().c_str(),
        WiFi.RSSI()
    );
  });

}

BLYNK_WRITE(InternalPinDBG) {
  String cmd = String(param.asStr()) + "\n";
  edgentConsole.runCommand((char*)cmd.c_str());
}
```