

HTTP Client Console

This example for a Yún device shows how create a basic HTTP client that connects to the internet and downloads content. In this case, you'll connect to the Arduino website and download a version of the logo as ASCII text. This version uses Console and shows the output on your Arduino Software (IDE) Console through a WiFi connection and not the USB one.

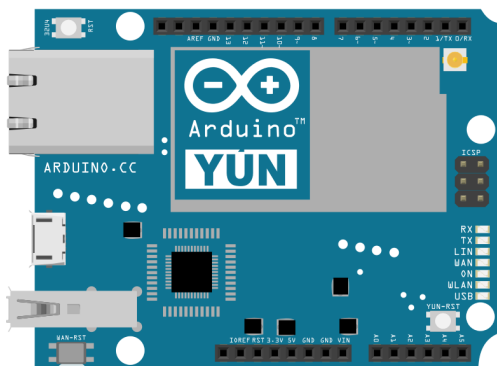
Select the IP port as connection to your board and open the Serial Monitor in the IDE once you've programmed the board.

Hardware Required

- Yún board or shield
- a wireless network connection to the internet

Circuit

There is no circuit for this example.



(//www.arduino.cc/en/uploads/Tutorial/Yun_Fritzing.png)

image developed using Fritzing (<http://www.fritzing.org>). For more circuit examples, see the Fritzing project page (<http://fritzing.org/projects/>)

Code

Include the Bridge, HttpClient and Console libraries

```
#include <Bridge.h>
#include <HttpClient.h>
#include <Console.h>
```

[Get Code] ([//www.arduino.cc/en/Tutorial/HttpClientConsole?action=sourceblock&num=2](http://www.arduino.cc/en/Tutorial/HttpClientConsole?action=sourceblock&num=2))

In `setup()` start Bridge, and wait for a serial connection before going into `loop()` .

```
void setup() {
  pinMode(13, OUTPUT);
  digitalWrite(13, LOW);
  Bridge.begin();
  Console.begin();
  while(!Console);
}
```

[Get Code] ([//www.arduino.cc/en/Tutorial/HttpClientConsole?action=sourceblock&num=3](http://www.arduino.cc/en/Tutorial/HttpClientConsole?action=sourceblock&num=3))

In `loop()` , create a named instance of HttpClient, and call a URL with `client.get(url)` .

```
void loop() {
  HttpClient client;
  client.get("http://www.arduino.cc/asciilogo.txt");
```

[Get Code] ([//www.arduino.cc/en/Tutorial/HttpClientConsole?action=sourceblock&num=4](http://www.arduino.cc/en/Tutorial/HttpClientConsole?action=sourceblock&num=4))

As long as there are bytes from the server in the client buffer, read the bytes and print them to the serial monitor. Repeat every 5 seconds.
&SCOPE=PUBLIC&PROFILE=PUBLIC&RESPONSE_TYPE=JSON&REDIRECT_URI=HTTPS%3A%2F%2FWWW.ARDUINO.CC)

```
while (client.available()) {  
  char c = client.read();  
  Console.print(c);  
}  
Console.flush();  
  
delay(5000);  
}
```

[Get Code] ([//www.arduino.cc/en/Tutorial/HttpClientConsole?action=sourceblock&num=5](https://www.arduino.cc/en/Tutorial/HttpClientConsole?action=sourceblock&num=5))

The complete sketch is below :

```
/*  
  Yún HTTP Client Console version for Arduino Uno and Mega using Yún Shield  
  
  This example for the YunShield/Yún shows how create a basic  
  HTTP client that connects to the internet and downloads  
  content. In this case, you'll connect to the Arduino  
  website and download a version of the logo as ASCII text.  
  
  created by Tom igoe  
  May 2013  
  modified by Marco Brianza to use Console  
  
  This example code is in the public domain.  
  
  http://www.arduino.cc/en/Tutorial/HttpClient  
  
  */  
  
#include <Bridge.h>  
#include <HttpClient.h>  
#include <Console.h>  
  
void setup() {  
  // Bridge takes about two seconds to start up  
  // it can be helpful to use the on-board LED  
  // as an indicator for when it has initialized  
  pinMode(13, OUTPUT);  
  digitalWrite(13, LOW);  
  Bridge.begin();  
  digitalWrite(13, HIGH);  
  
  Console.begin();  
  
  while (!Console); // wait for a serial connection  
}  
  
void loop() {  
  // Initialize the client library  
  HttpClient client;  
  
  // Make a HTTP request:  
  client.get("http://www.arduino.cc/asciilogo.txt");  
  
  // if there are incoming bytes available  
  // from the server, read them and print them:  
  while (client.available()) {  
    char c = client.read();  
    Console.print(c);  
  }  
  Console.flush();  
  
  delay(5000);  
}
```

[Get Code] ([//www.arduino.cc/en/Tutorial/HttpClientConsole?action=sourceblock&num=1](https://www.arduino.cc/en/Tutorial/HttpClientConsole?action=sourceblock&num=1))

See Also

- Bridge Library ([//www.arduino.cc/en/Reference/YunBridgeLibrary](https://www.arduino.cc/en/Reference/YunBridgeLibrary)) - Your reference to the Bridge Library
- Bridge ([//www.arduino.cc/en/Tutorial/Bridge](https://www.arduino.cc/en/Tutorial/Bridge)) – Simple REST style calls to access analog and digital pins
- Console Ascii Table ([//www.arduino.cc/en/Tutorial/ConsoleAsciiTable](https://www.arduino.cc/en/Tutorial/ConsoleAsciiTable)) – A complete ASCII table printed to the Console
- Console Pixel ([//www.arduino.cc/en/Tutorial/ConsolePixel](https://www.arduino.cc/en/Tutorial/ConsolePixel)) – Turn an LED on and off through the Console

- Console Read ([//www.arduino.cc/en/Tutorial/ConsoleRead](http://www.arduino.cc/en/Tutorial/ConsoleRead)) - Read data coming from bridge using the Console.read() function
- Data Logger ([//www.arduino.cc/en/Tutorial/YunDataLogger](http://www.arduino.cc/en/Tutorial/YunDataLogger)) - Log data from three analog sensors to an SD card.
- File Write ([//www.arduino.cc/en/Tutorial/FileWriteScript](http://www.arduino.cc/en/Tutorial/FileWriteScript)) - How to write file into the Yún filesystem.
- Http Client ([//www.arduino.cc/en/Tutorial/HttpClient](http://www.arduino.cc/en/Tutorial/HttpClient)) - A basic HTTP client that connects to the internet and downloads content.
- Mailbox Read Message ([//www.arduino.cc/en/Tutorial/MailboxReadMessage](http://www.arduino.cc/en/Tutorial/MailboxReadMessage)) - How to read the messages queue, called Mailbox, using the Bridge library.
- Process ([//www.arduino.cc/en/Tutorial/Process](http://www.arduino.cc/en/Tutorial/Process)) - How to run linux processes using an Yún.
- Remote Due Blink ([//www.arduino.cc/en/Tutorial/RemoteDueBlink](http://www.arduino.cc/en/Tutorial/RemoteDueBlink)) - How to upload remotely a sketch on DUE boards.
- Shell Commands ([//www.arduino.cc/en/Tutorial/ShellCommands](http://www.arduino.cc/en/Tutorial/ShellCommands)) - How to run linux shell commands using a Yún.
- Temperature Web Panel ([//www.arduino.cc/en/Tutorial/TemperatureWebPanel](http://www.arduino.cc/en/Tutorial/TemperatureWebPanel)) - How to serve data from an analog input via the Yún's built-in webserver.
- Time check ([//www.arduino.cc/en/Tutorial/TimeCheck](http://www.arduino.cc/en/Tutorial/TimeCheck)) - Gets the time from Linux via Bridge then parses out hours, minutes and seconds.
- WiFi Status ([//www.arduino.cc/en/Tutorial/WiFiStatus](http://www.arduino.cc/en/Tutorial/WiFiStatus)) - Prints information about the status of your wifi connection.
- Yún First Configuration ([//www.arduino.cc/en/Tutorial/YunFirstConfig](http://www.arduino.cc/en/Tutorial/YunFirstConfig)) - Easily configure your Yún device using Serial Monitor and USB port.
- Serial Terminal ([//www.arduino.cc/en/Tutorial/YunSerialTerminal](http://www.arduino.cc/en/Tutorial/YunSerialTerminal)) - Use the Yún's 32U4 processor as a serial terminal for the Linux side on the Yún.

Last revision 2016/05/25 by SM


Share





NEWSLETTER

ENTER YOUR EMAIL TO SIGN UP

SUBSCRIBE

 Copyright Notice (<http://www.arduino.cc/en/Main/CopyrightNotice>)

 Contact Us (<http://www.arduino.cc/en/Main/ContactUs>)

 About Us (<http://www.arduino.cc/en/Main/AboutUs>)

 Careers (<http://www.arduino.cc/Careers>)

(<https://www.arduino.cc/en/Main/AboutUs>)