

## Tutorial 2 – Introduction to Bash Scripting

*Disclaimer: Subject to updates as corrections are found*

Version 0.10

Scoring: 20 pts maximum

The purpose of this tutorial is to introduce Bash scripting under the Linux operating system, while also introducing web services, and the use of curl as a command-line HTTP REST web service client. Complete this tutorial leveraging your Linux environment set up for Tutorial 1. Please review sections 1 – 8 of the online Bash Scripting Tutorial as needed to complete the scripting activity:

Bash Scripting Tutorial:

<https://ryantutorials.net/bash-scripting-tutorial/>

Tutorial Sections include:

1. What is a BASH script?
2. Variables
3. Input
4. Arithmetic
5. If Statements
6. Loops
7. Functions
8. User Interface

At the conclusion of the online Bash tutorial, please complete the Bash scripting task described below. As needed, search the Internet to find BASH examples beyond Ryan’s tutorial to help with the overall programming task. Submit your completed operational Bash script as a file called **weather.sh** online via Canvas. While it is possible to perform the implementation in Python, the goal here is to gain experience using Bash and curl.

### BASH WEATHER FORECAST TOOL

Write a short Bash script that makes use of two web services to obtain a localized weather forecast based on the latitude and longitude of your computer’s Internet connection. To complete the script it is recommended to use the following commands:

| Command     | Description                                                                                                                                                                                                                                                                            |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>curl</b> | Command line http REST client for performing GET PUT POST requests, etc.<br>If curl has not already been installed, it can be installed with:<br>sudo apt install curl<br><br>When using curl, please use the “-s” flag to request silent output without continuous status information |
| <b>cut</b>  | Cut is simple parsing tool in Bash. Simply pipe text to cut, and specify a custom column delimiter with “-d”, then specify the desired column with “-f”.                                                                                                                               |
| <b>jq</b>   | Jq is a Bash JSON parser. You will need to install jq as follows:                                                                                                                                                                                                                      |

|            |                                                                                                                        |
|------------|------------------------------------------------------------------------------------------------------------------------|
|            | <code>sudo apt install jq</code>                                                                                       |
| <b>awk</b> | Awk can easily parse individual columns of a file: (for column 2)<br><code>cat myfile.txt   awk '{ print \$2 }'</code> |

Use the following two web services to obtain a weather forecast using the geolocation (latitude and longitude coordinates) of your internet providers IP address.

**IP Address Location API:** <https://ipinfo.io/developers>

The “IP Address Location API” provides latitude and longitude based on the IP address of your incoming web service request. The API is free to use and does not require registration. The web service never sees your internal IP address inside the “firewall” of your business, school, or home. For example, at home you may receive an IP address from a Comcast modem of “10.0.0.50”. This is an internal IP address. Internal IP addresses are described in the table:

### Private IPv4 addresses

| RFC1918 name | IP address range              | host id size |
|--------------|-------------------------------|--------------|
| 24-bit block | 10.0.0.0 – 10.255.255.255     | 24 bits      |
| 20-bit block | 172.16.0.0 – 172.31.255.255   | 20 bits      |
| 16-bit block | 192.168.0.0 – 192.168.255.255 | 16 bits      |

The “IP Address Location API” will provide metadata regarding your Internet service provider in JSON format as follows:

```
{
  "ip": "71.209.4.118",
  "hostname": "c-71-209-4-118.hsd1.wa.comcast.net",
  "city": "Tacoma",
  "region": "Washington",
  "country": "US",
  "loc": "47.2529,-122.4417",
  "postal": "98402",
  "org": "AS33650 Comcast Cable Communications, LLC"
}
```

Using the “jq” function, parse the latitude and longitude, and pass this to the Weatherbit API.

Without authentication calls to the ipinfo.io endpoint are limited to ~1,000 free requests per day. To avoid calling ipinfo.io too often, you should cache this information using a local hidden file called “.myipaddr”. In Bash, at the start of your script, check for the existence of a file called “.myipaddr”. If the file does not exist, call the service to obtain the JSON object. Save the JSON object to the disk. Read the JSON object into a Bash variable using the “cat” command. Then process the JSON with “jq”.

Print out the following messages based on whether you're using a cached IP from ".myipaddr", or whether your script had to call the service to obtain the JSON describing your IP and location:

```
echo "CALLING API TO QUERY MY IP"
```

```
echo "IP READ FROM CACHE"
```

See the online documentation for more information: <https://ipinfo.io/developers>

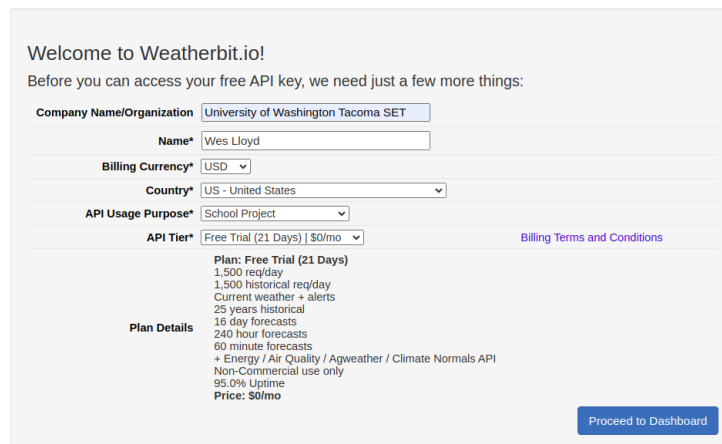
Use the Weatherbit API to obtain a 7-day weather forecast with High and Low temperature information. The Weatherbit API provides up to a detailed 16-day forecast on request (if paying for a license), and a 7-day forecast for the free account. This API is limited to 50 calls per day. If you are concerned that debugging your script will exceed 50 calls per day, please consider caching the JSON object as described above for the IP Address Location API. Then write your parsing code using the cached JSON object to save calls to the web service.

You will need to create an account and obtain an API Key to use the Weatherbit API. Keys are generally made available within a couple of minutes. The weatherbit API is actually quite extensive. They provide a variety of APIs to provide daily and hourly forecasts and also APIs for current weather, severe weather, historical weather, etc. To create an account, visit the website:

**Weatherbit API signup:** <https://www.weatherbit.io/api>

You will receive a confirmation email to finalize setup of your account.

Once the account is setup, login, and then set the API Tier to **"Free Trial (21 Days) | \$0/mo"**



The screenshot shows the 'Welcome to Weatherbit.io!' page. It prompts the user to provide information before accessing their free API key. The form includes fields for 'Company Name/Organization' (University of Washington Tacoma SET), 'Name\*' (Wes Lloyd), 'Billing Currency\*' (USD), 'Country\*' (US - United States), 'API Usage Purpose\*' (School Project), and 'API Tier\*' (Free Trial (21 Days) | \$0/mo). A 'Billing Terms and Conditions' link is also present. Below the form, 'Plan Details' are listed: Plan: Free Trial (21 Days), 1,500 req/day, 1,500 historical req/day, Current weather + alerts, 25 years historical, 16 day forecasts, 240 hour forecasts, 60 minute forecasts, + Energy / Air Quality / Agweather / Climate Normals API, Non-Commercial use only, 95.0% Uptime, Price: \$0/mo. A 'Proceed to Dashboard' button is at the bottom right.

**The Free Trial supports 1,500 requests per day with 16-day forecasts. This downgrades to ~50 requests per day and 7-day forecasts after the Free Trial expires.**

**BE SURE TO INCLUDE THE CORRECT APIKEY IN THE SCRIPT FOR TESTING !**

Once having a key, you can check your daily usage quota with the following call:

```
curl -s -g "https://api.weatherbit.io/v2.0/subscription/usage?key=[YOUR-API-KEY]"
```

Specify your actual apikey (a combination of letters and numbers) in place of “[YOUR-API-KEY]”.

To obtain a 7-day weather forecast, use the following API described in the documentation [HERE](https://www.weatherbit.io/api/weather-forecast-16-day):  
<https://www.weatherbit.io/api/weather-forecast-16-day>

Note after the trial expires, the API will only provide a 7-day weather forecast.

It is recommended to parse the fields “max\_temp” and “min\_temp” for the weather forecast.

For the assignment, **it is okay** if the license has expired and only a 7-day forecast is produced.

**\*\*\* To call the weatherbit services with CURL put the URL in “quotes” \*\*\***

The following online JSON formatter is helpful to paste JSON into a web browser to rapidly make it more readable: <https://jsonlint.com/>

Your weather.sh script should produce output as below:

#### **Sample Output – First Call:**

```
CALLING API TO QUERY MY IP
Forecast for my lat=47.2529°, lon=-122.4443°
Forecast for 2025-10-07 HI: 23°C LOW: 9.5°C
Forecast for 2025-10-08 HI: 16.3°C LOW: 10.1°C
Forecast for 2025-10-09 HI: 16.8°C LOW: 8°C
Forecast for 2025-10-10 HI: 16.8°C LOW: 9.2°C
Forecast for 2025-10-11 HI: 15.1°C LOW: 10.3°C
Forecast for 2025-10-12 HI: 14°C LOW: 6.8°C
Forecast for 2025-10-13 HI: 10.5°C LOW: 7.5°C
```

#### **Sample Output – Subsequent Calls:**

```
IP READ FROM CACHE
Forecast for my lat=47.2529°, lon=-122.4443°
Forecast for 2025-10-07 HI: 23°C LOW: 9.5°C
Forecast for 2025-10-08 HI: 16.3°C LOW: 10.1°C
Forecast for 2025-10-09 HI: 16.8°C LOW: 8°C
Forecast for 2025-10-10 HI: 16.8°C LOW: 9.2°C
Forecast for 2025-10-11 HI: 15.1°C LOW: 10.3°C
Forecast for 2025-10-12 HI: 14°C LOW: 6.8°C
Forecast for 2025-10-13 HI: 10.5°C LOW: 7.5°C
```

#### **How to type the degree symbol in Ubuntu/BASH:**

To produce the degrees symbol in Ubuntu/BASH simply type [CONTROL]-[SHIFT]-[U] as a three-key combination and then quickly type “b0” and press [ENTER]. The degree symbol should appear: °

[CONTROL]-[SHIFT]-[U] enables Unicode characters to be entered such (4E91) as: ☁

#### **Generative AI Policy**

Use of generative AI is permitted, though not encouraged as this is a relatively simple scripting assignment. If using generative AI, use of ipinfo.io and the weatherbit.io APIs are required, and all output requirements and caching aspects should be implemented properly as described. Use of generative AI should be declared in comments at the top of the script.

### **Grading**

The Bash weather script (weather.sh) will be scored out of 20 points. Submit your completed script called **weather.sh** online via Canvas. Your script should include a valid weather bit API KEY for testing.

- 10 points for being functionally correct. It should obtain and display either a 16-day or 7-day weather forecast for the lat/long of the internet service provider depending on the age of your weatherbit API KEY. Caching of the IP address to a hidden file on the disk should be implemented correctly.
- 10 additional points for proper formatting. Formatting should match the output shown in the tutorial.