

# **Cron Expression Parser**

08.05.2022

Likhith Pallagani pallaganilikhith@gmail.com

## **Overview**

A command line application or script which parses a cron string and expands each field to show the times at which it will run.

## Goals

1. Given a cron string, Parse it and display the times which it will run

For example, the following input argument:

```
*/15 0 1,15 * 1-5 /usr/bin/find
```

Should yield the following output:

```
minute 0 15 30 45
hour 0
day of month 1 15
month 1 2 3 4 5 6 7 8 9 10 11 12
day of week 1 2 3 4 5
command /usr/bin/find
```

## **Installing Dependencies:**

These dependencies are based on the assumption, we are using a linux machine

1. Install Node is to run the node is files

Command: sudo apt-get install npm

2. Using npm(node package manager) install the dependencies in the CronExpressionParser Folder

Command: npm install

```
MHYDFA0B7509:Desktop likhith.pallagani$ cd CronExpressionParser/
MHYDFA0B7509:CronExpressionParser likhith.pallagani$ npm install
```

# **Usage**

#### Command to Run:

After the dependencies are installed, run the following command, in the proj directory

#### Command:

likhith.pallagani\$ node <FileName> <Cron String>

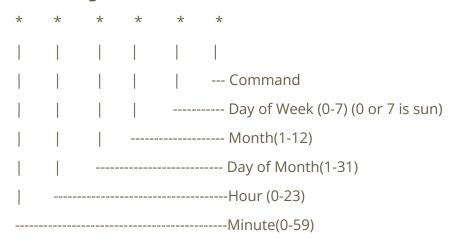
#### node:

the key word to run the node js files

#### FileName:

parser.js contains the script which takes the input of the cron string and invokes the script which parses the cron string.

#### **Cron String:**



# **Cron Expression Parser Library Usage:**

```
var CronExpression = require('./expression');
var expression = "*/15 0 1,15 * 1-5 /usr/bin/find";
var cronOutput = CronExpression.parse(expression);
```

## **Cron Expression Parser Test:**

Refer to the test folder to find all the tests which are run against the cron library

- For each field in the cron expression, all the different test cases are run.
- Eg: For minute field only ranges from 0-59 are allowed
- Eg: For hour field only ranges from 0-23 are allowed
- Eg: For day of Month field only ranges from 1-31 are allowed
- For Month field only ranges from 1-12 are allowed

Example1: Minute value out of range

```
test('minute value out of the range', function(t) {
   t.throws(function() {
        CronExpression.parse('32,72 * * * * /user/bin');
   }, new Error('Constraint error, got value 72 expected range 0-59'));
   t.end();
});
```

Example2: Day of the week out of range

```
test('day of the week value out of the range', function(t) {
    t.throws(function() {
        CronExpression.parse('* * * * 9 /user/bin');
    }, new Error('Constraint error, got value 9 expected range 0-7'));
    t.end();
});
```

## Example3: April doesnt have 31 days

```
test('April doesnt have 31 days', function(t) {
    t.throws(function() {
        const iter = CronExpression.parse('0 0 31 4 * /user/bin');
        iter.next();
    }, new Error('Invalid explicit day of month definition'));
    t.end();
});
```