Project OverView: Cron Expression Parser

Overview

The Cron Expression Parser is a JavaScript project designed to parse and interpret cron expressions, providing functionality to expand cron fields and validate input expressions. This project is structured with modularity in mind, making it easy to understand and extend.

Project Structure

```
cron-expression-parser
|-- src
|   |-- expandCronField.js
|   |-- main.js
|   |-- tableFormatter.js
|   |-- inputValidator.js
|   |-- parseCronExpression.js
|-- test
|   |-- expandCronField.test.js
|   |-- main.test.js
|   |-- inputValidator.test.js
|   |-- parseCronExpression.test.js
|-- package.json
|-- Readme.md
```

Source Code

1. **expandCronField.js**
   - Responsible for expanding a specific field in a cron expression.
2. **main.js**
   - Main entry point for the Cron Expression Parser application.
3. **tableFormatter.js**
   - Provides utility functions for formatting data into tables.
4. **inputValidator.js**
   - Implements functions to validate the input cron expressions.
5. **parseCronExpression.js**
   - Core module for parsing and interpreting cron expressions.

Test Suite

1. **expandCronField.test.js**
   - Test cases for the `expandCronField` module.
2. **main.test.js**
   - Test cases for the main application logic.
3. **inputValidator.test.js**
   - Test cases for the `inputValidator` module.

4. parseCronExpression.test.js

- Test cases for the core `parseCronExpression` module.

Additional Files

1. package.json

- Contains project metadata and dependencies.

2. Readme

- Project documentation providing an overview, installation instructions, and usage guidelines.

3. GitHub Repository

- Source code can be found here: GitHub Repo.

Getting Started

1. **Installation:**

   ```
   npm install
   ```

2. **Run Tests:**

   ```
   npm test
   ```

3. **Usage:**

   - Refer to the documentation in Readme.md for detailed instructions on using the Cron Expression Parser.

Usage/Examples

A cron expression is a string conforming to a pre-defined format that tells a computer program (or a person) how often a task needs to run.

```
* * * * * *
| | | | | |--- command
| | | | +----- Day of the week (0 - 6) (Sunday to Saturday)
| | | +------- Month (1 - 12)
| | +--------- Day of the month (1 - 31)
| +----------- Hour (0 - 23)
+------------- Minute (0 - 59)
```

Cron Expression Components

Asterisk (*)

The asterisk indicates that the cron expression matches for all values of the field. E.g., using an asterisk in the 4th field (month) indicates every month.

Slash (/)

Slashes describe increments of ranges. For example 3-59/15 in the minute field indicate the third minute of the hour and every 15 minutes thereafter. The form */... is equivalent to the form “first-last/...”, that is, an increment over the largest possible range of the field.

Comma (,)

Commas are used to separate items of a list. For example, using 1,3,5 in the 5th field (day of week) means Mondays, Wednesdays and Fridays.

Hyphen (-)

Hyphens define ranges. For example, 2000-2010 indicates every year between 2000 and 2010, inclusive.

For more details on CRON expressions, you can refer to the Wikipedia page.

Deployment

1. Link the Cron Expression Parser globally using npm:

```
sudo npm link
```

This makes the **cep** command available system-wide,

2. Run the Cron Expression Parser with your desired cron expression:

   ```
   cep <cron_expression>
   ```

Replace `<cron_expression>` with the specific cron string you want to parse.

Example:

```
cep "*/15 2 5/3 3-5 2,4 /usr/bin/find"
```

Tech Stack

- Node.js
- JavaScript
- Jest

Thank you for using the Cron Expression Parser!