Map::Tube

Lightweight Routing Framework

By

Mohammad S Anwar

(MANWAR)

London Perl Workshop 2017

(www.manwar.org)

Who am I?

- Perl & I, are in relationship for nearly **20** years.
- So far, published **78** CPAN modules, pause id "**MANWAR**".
- GitHub repository holds all the source codes, user name "**MANWAR**".
- Maintains some of the most popular distributions e.g. **PDF::Create**, **XML::XPath**, **SVG** etc.
- Contributed to over **300+** distributions e.g. **Dancer2**, **Dist::Zilla**, **Test::More** etc.
- Submitted over **600+** PullRequests in GitHub.
- Over **1000+** consecutive days of releasing to CPAN.

Overview

- Lightweight **Moo-based** role.
- Actively maintained for the last **8 years**.
- There have been **154** releases so far, last being Map::Tube **v3.42**.
- Supports the following plugins.
  - **Map::Tube::Plugin::Graph**
  - **Map::Tube::Plugin::FuzzyFind**
  - **Map::Tube::Plugin::Formatter**
- Supports command line tool '**map-tube**' supplied by **Map::Tube::CLI**
- Provides command line tool '**map-data-converter**', that can help you change the map data format between **JSON** and **XML**.

Contributors

- Michal Špaček (**SKIM**).
  - Created **32** maps (Bucharest, Budapest, Dnipropetrovsk, Kazan, Kharkiv, Kiev, Kuala Lumpur, Malaga, Minsk, Moscow, Nanjing, Nizhny Novgorod, Novosibirsk, Prague, Saint Petersburg, Samara, Singapore, Sofia, Tbilisi, Vienna, Warsaw, Yekaterinburg).
  - Created handy tools (**Map::Tube::Text::Shortest**, **Map::Tube::Text::Table** and **Task::Map::Tube**).
- Gisbert W. Selke (**GWS**).
  - Created **4** maps (Beijing, Glasgow, Köln Bonn, Lyon).
  - Created a handy plugin (**Map::Tube::Plugin::FuzzyFind**)
- Slaven Rezic (**SREZIC**).
  - Created **1** map (Berlin).

Maps Available

Barcelona	Beijing	Berlin	Bucharest	Budapest	Delhi
Dnipropetrovsk	Glasgow	Kazan	Kharkiv	Kiev	Koeln Bonn
Kolkatta	Kuala Lumpur	**London**	Lyon	Malaga	Minsk
Moscow	New York	Nanjing	Nizhny Novgorod	Novosibirsk	Prague
Saint Petersburg	Samara	Singapore	Sofia	Tbilisi	Tokyo
Vienna	Warsaw	Yekaterinburg			

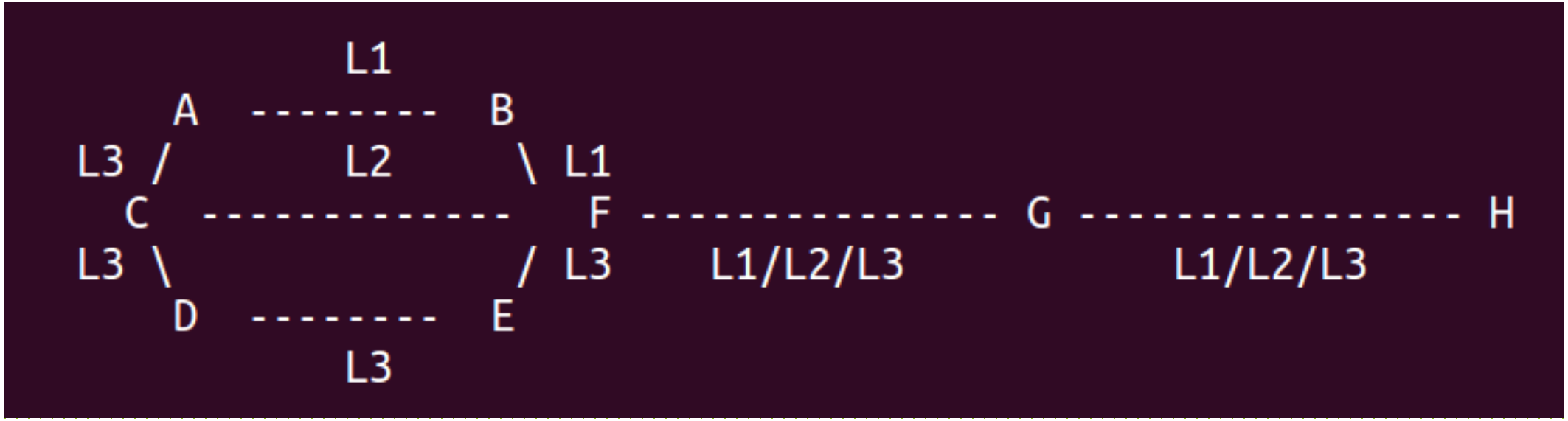
Setup Environment

```
$ sudo cpanm -v Map::Tube

$ sudo cpanm -v Map::Tube::Plugin::Graph

$ sudo cpanm -v Map::Tube::Plugin::FuzzyFind

$ sudo cpanm -v Map::Tube::Plugin::Formatter
```

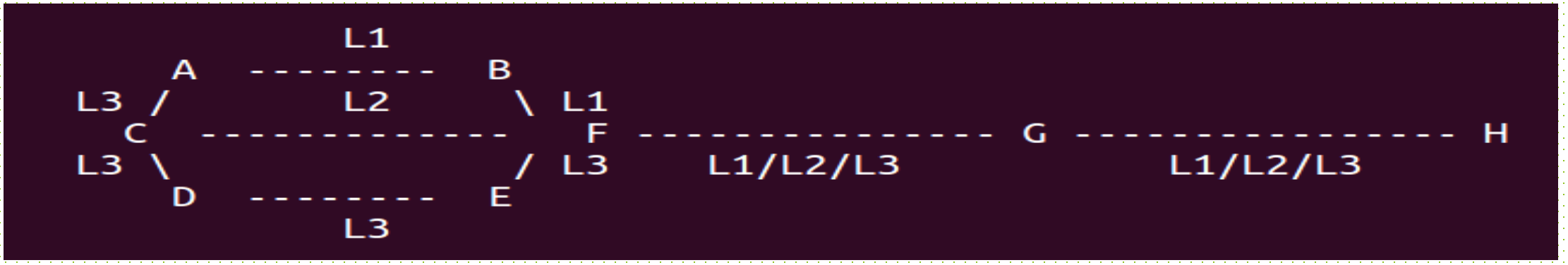
Lets build a new map

- Step 1: Collect the source data of the new map.
- Step 2: Decide the format of map data. e.g. **JSON** or **XML**.
- Step 3: Build map data in the selected format.
- Step 4: Create package to consume the role **Map::Tube**.

Step 1: Collect the map data.

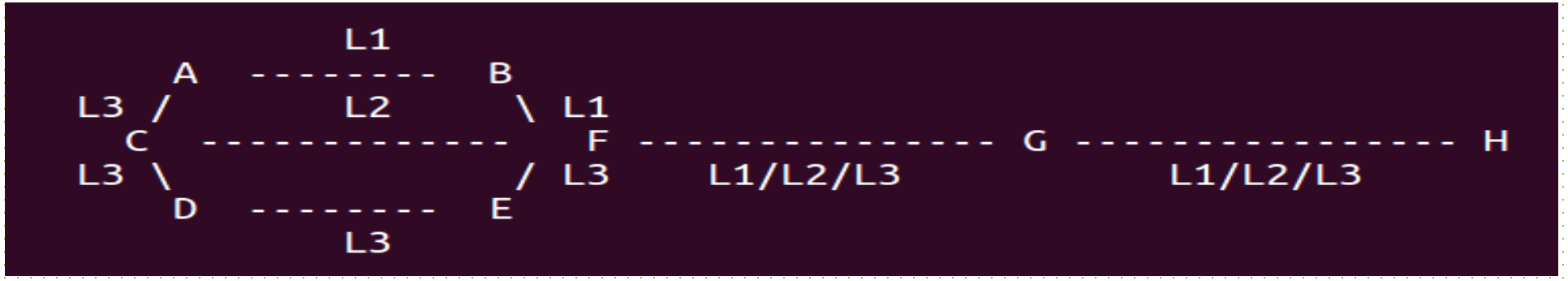


Step 2: Decide the format of map data.

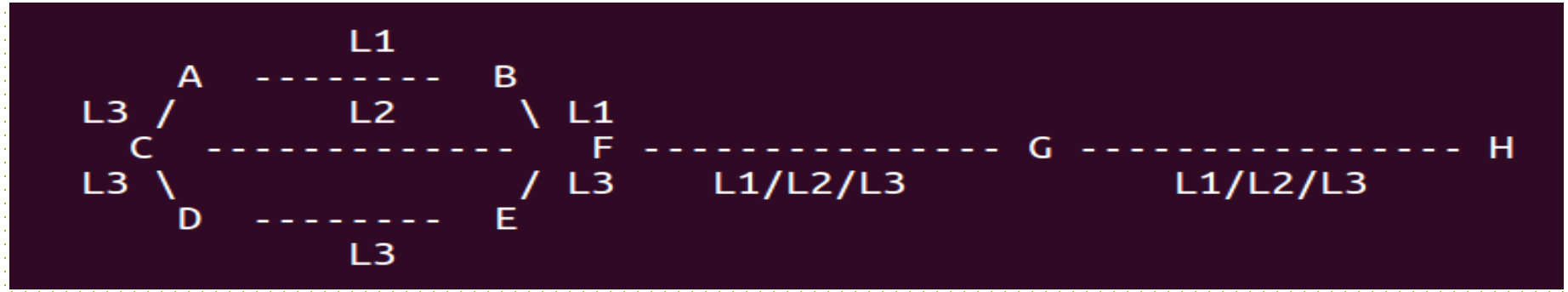



```
{
        "name": "Trial",
        "lines":
        {
                "line":
                [
                ]
        },
        "stations":
        {
                "station":
                [
                ]
    }
}
```

Step 3: Build map data in selected format




```
{
    "name": "Trial",
    "lines":
    {
        "line":
        [
            { "id" : "L1", "name" : "L1" },
            { "id" : "L2", "name" : "L2" },
            { "id" : "L3", "name" : "L3" }
        ]
    },
    "stations":
    {
        "station":
        [
        ]
    }
}
```




```
{
    "name": "Trial",
    "lines":
    {
        "line":
        [
            { "id": "L1", "name": "L1" },
            { "id": "L2", "name": "L2" },
            { "id": "L3", "name": "L3" }
        ]
    },
    "stations":
    {
        "station":
        [
            { "id": "A", "name": "A", "line": "L1,L3",     "link": "B,C"       },
            { "id": "B", "name": "B", "line": "L1",       "link": "A,F"       },
            { "id": "C", "name": "C", "line": "L2,L3",    "link": "A,D,F"     },
            { "id": "D", "name": "D", "line": "L3",       "link": "C,E"       },
            { "id": "E", "name": "E", "line": "L3",       "link": "D,F"       },
            { "id": "F", "name": "F", "line": "L1,L2,L3", "link": "B,C,E,G"   },
            { "id": "G", "name": "G", "line": "L1,L2,L3", "link": "F,H"       },
            { "id": "H", "name": "H", "line": "L1,L2,L3", "link": "G"         }
        ]
    }
}
```

Step 4: Create package to consume the role Map::Tube

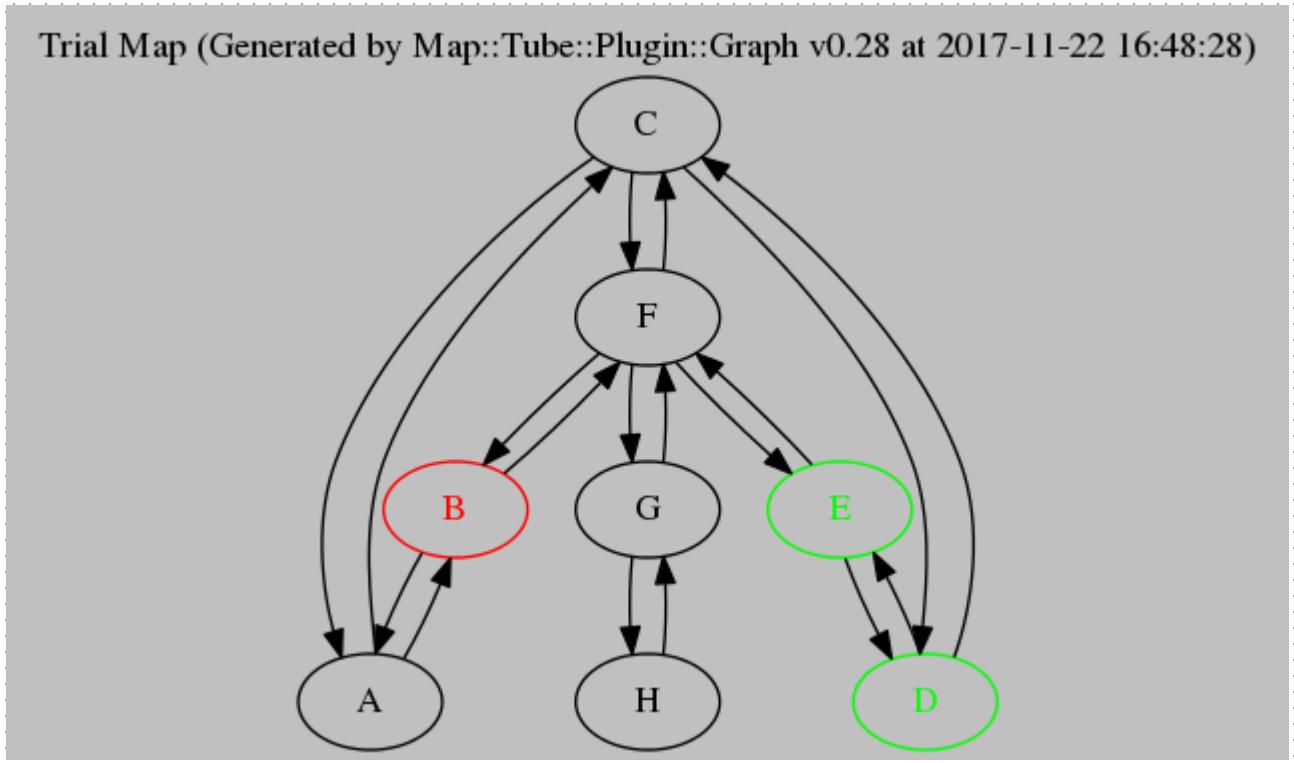
```
package Map::Tube::Trial;

use Moo;
use namespace::autoclean;

has json => (is => 'ro', default => sub { 'trial.json' });
with 'Map::Tube';

1;
```

Find shortest route

```
#!/usr/bin/perl

use strict; use warnings;
use MIME::Base64;
use Map::Tube::Trial;

my $map = Map::Tube::Trial->new;
print $map->get_shortest_route('A', 'D'), "\n";
```

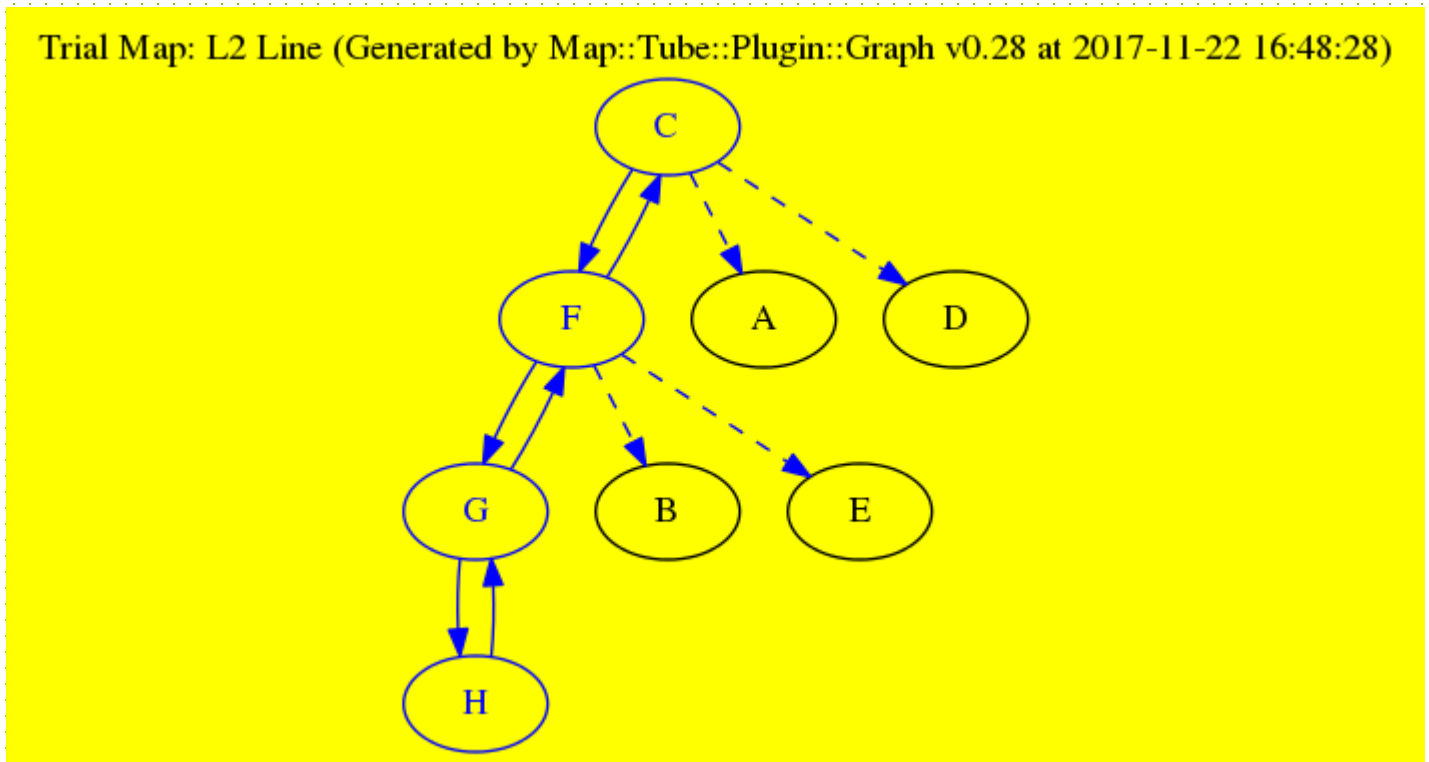
Create entire map

```
my $name = $map->name;
open(my $MAP_IMAGE, ">$name.png");
binmode($MAP_IMAGE);
print $MAP_IMAGE decode_base64($map->as_image);
close($MAP_IMAGE);
```



Create a particular line map

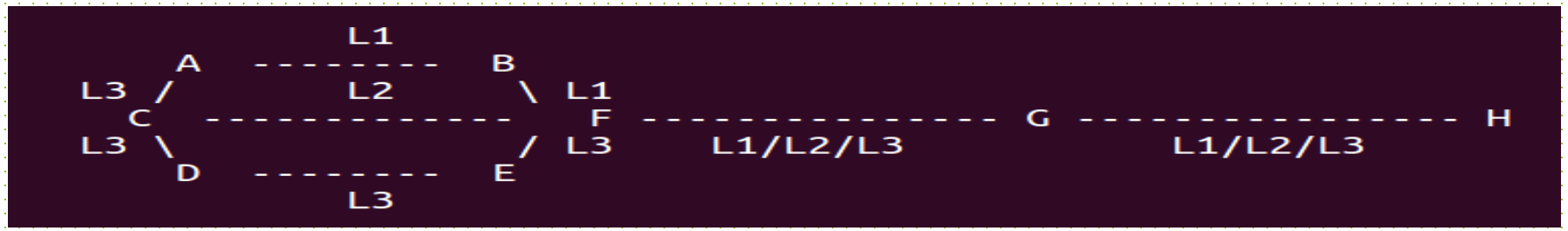
```
my $line = 'L2';
open(my $LINE_IMAGE, ">$line.png");
binmode($LINE_IMAGE);
print $LINE_IMAGE decode_base64($map->as_image($line));
close($LINE_IMAGE);
```



Bonus Feature #1: Color code line

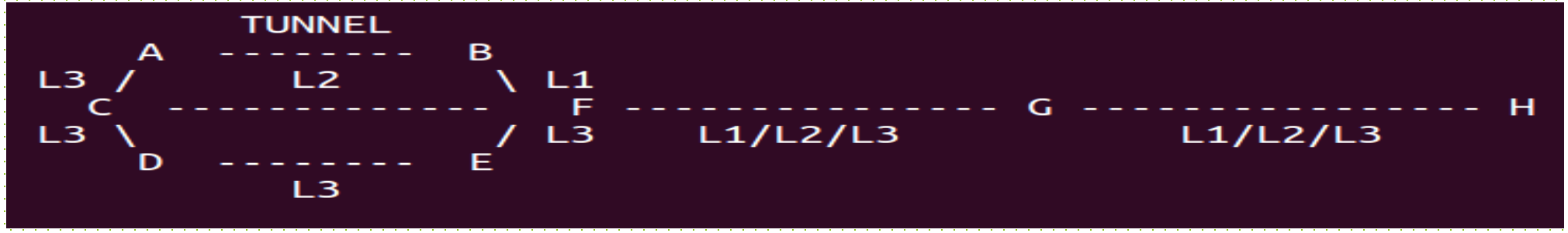
```
{
    "name": "Trial",
    "lines":
    {
        "line":
        [
            { "id": "L1", "name": "L1", "color": "red"   },
            { "id": "L2", "name": "L2", "color": "blue"  },
            { "id": "L3", "name": "L3", "color": "green" }
        ]
    },
    "stations":
    {
        "station":
        [
            { "id": "A", "name": "A", "line": "L1,L3",     "link": "B,C"       },
            { "id": "B", "name": "B", "line": "L1",       "link": "A,F"       },
            { "id": "C", "name": "C", "line": "L2,L3",    "link": "A,D,F"     },
            { "id": "D", "name": "D", "line": "L3",       "link": "C,E"       },
            { "id": "E", "name": "E", "line": "L3",       "link": "D,F"       },
            { "id": "F", "name": "F", "line": "L1,L2,L3", "link": "B,C,E,G"   },
            { "id": "G", "name": "G", "line": "L1,L2,L3", "link": "F,H"       },
            { "id": "H", "name": "H", "line": "L1,L2,L3", "link": "G"         }
        ]
    }
}
```

Bonus Feature #2: Index station per line




```
{
    "name": "Trial",
    "lines":
    {
        "line":
        [
            { "id": "L1", "name": "L1", "color": "red"   },
            { "id": "L2", "name": "L2", "color"; "blue"  },
            { "id": "L3", "name": "L3", "color": "green" }
        ]
    },
    "stations":
    {
        "station":
        [
            { "id": "A", "name": "A", "line": "L1:1,L3:1",       "link": "B,C"     },
            { "id": "B", "name": "B", "line": "L1:2",           "link": "A,F"     },
            { "id": "C", "name": "C", "line": "L2:1,L3:2",      "link": "A,D,F"   },
            { "id": "D", "name": "D", "line": "L3:3",           "link": "C,E"     },
            { "id": "E", "name": "E", "line": "L3:4",           "link": "D,F"     },
            { "id": "F", "name": "F", "line": "L1:3,L2:2,L3:5", "link": "B,C,E,G" },
            { "id": "G", "name": "G", "line": "L1:4,L2:3,L3:6", "link": "F,H"     },
            { "id": "H", "name": "H", "line": "L1:5,L2:4,L3:7", "link": "G"       }
        ]
    }
}
```

Bonus Feature #3: Link station by “other link”




```
{
    "name": "Trial",
    "lines":
    {
        "line":
        [
            { "id": "L1", "name": "L1", "color": "red"   },
            { "id": "L2", "name": "L2", "color"; "blue"  },
            { "id": "L3", "name": "L3", "color": "green" }
        ]
    },
    "stations":
    {
        "station":
        [
            { "id": "A", "name": "A", "line": "L3:1",           "link": "C", "other_link": "tunnel:B" },
            { "id": "B", "name": "B", "line": "L1:1",           "link": "F", "other_link": "tunnel:A" },
            { "id": "C", "name": "C", "line": "L2:1,L3:2",      "link": "A,D,F"       },
            { "id": "D", "name": "D", "line": "L3:3",           "link": "C,E"         },
            { "id": "E", "name": "E", "line": "L3:4",           "link": "D,F"         },
            { "id": "F", "name": "F", "line": "L1:2,L2:2,L3:5", "link": "B,C,E,G"     },
            { "id": "G", "name": "G", "line": "L1:3,L2:3,L3:6", "link": "F,H"         },
            { "id": "H", "name": "H", "line": "L1:4,L2:4,L3:7", "link": "G"           }
        ]
    }
}
```

Need more information?

- I would recommend **Map::Tube::Cookbook** documentation for detailed description of internals of Map::Tube.
- For all other details, please refer to the documentation of **Map::Tube**.
- In case you still have any questions/suggestions, then please free to contact me by email (**mohammad.anwar@yahoo.com**).

Mini Challenge

- I would like to give you all a mini challenge to create simple map, having at least 2 lines with stations for now, and release it to CPAN.
- Whoever do this first by end of today's workshop will receive a gift from me.
- To help you in your challenge, I have picked few maps that are still missing:
  - Paris Metro
    - Download sample data (http://www.manwar.org/talks/paris-metro.json)
  - Madrid Metro
    - Download sample data (http://www.manwar.org/talks/madrid-metro.json)
  - Mexico City Metro
    - Download sample data (http://www.manwar.org/talks/mexico-city-metro.json)
- Download source: (http://www.manwar.org/talks/Map-Tube-Trial-0.01.tar.gz)

Any Questions ?

Before I end my talk ...

I would like to thank all my friends and families, especially ...




And finally Thank you to the sponsors, without which the London Perl Workshop would not be possible:

Eligo, Perl Careers,
CV-Library, WCN,
Adzuna, Bytemark, OpusVL, Booking.com,
SureVoIP, Magnum Solutions,
Perl 6, Geekuni,
University of Westminster, Cogendo,
Science Photo Library,
The Enlightened Perl Organisation,
Evozon, O'Reilly.