# About AvdtaJSONSerializer

AvdtaJSONSerializer is a class you can use to serialize information about links in a network into Json format. The file which contains the outputted JSON will be in AVDTA/google\_maps\_project/maps.js. This file is then consumed by maps.html and shown to the client.

# Using AvdtaJSONSerializer

AvdtaJsonSerializer has only one public method to use, which is write(). Write takes three parameters and uses those parameters to write JSON to AVDTA/google\_maps\_project/maps.js. The three parameters are as follows:

1. A network object – after you run a simulation, pass the resulting network object here.
2. A name – this is the display name shown to the client.
3. An array of Metric objects – this tells AvdtaJsonSerializer which information to write. An explanation of the Metric class is below.

# The Metric Class

**Using the Metric Class**

The metric class is used by AvdtaJsonSerializer to decide what information to write. You should only ever create instances of the Metric class and pass them into AvdtaJsonSerializer.write(). The Metric class is not used for any other purpose, and you should never directly call any function of the Metric class. The constructor of the Metric class takes the following arguments:

1. metricType: This is an enumerated Type which currently supports only four values: CAPACITY, FLOW\_IN, AVERAGE\_TT, FREE\_FLOW\_SPEED.
2. minValue: This is the minimum value displayed on the legend to the client
3. c2Value: This is the middle value displayed on the legend to the client
4. maxValue: This is the maximum value displayed on the legend to the client
5. unitOfMeasurement: This is the unit label displayed on the legend to the client

**Modifying the Metric Class**

The Metric class currently only supports four metrics. If you wish to add a metric, there are two steps to take:

1. Add an enumerated Type to the Type enum which exists in Metric.java
2. Add a switch statement in the getValue() method which handles your new enumerated type. Your switch statement should return the value for your new metric at the given ast.

# Basic Customization of AvdtaJSONSerializer

**Changing the colors of the Legend**

The legend consists of three main colors. You can change the RGB of these three colors by modifying the public static variables C1\_R, C1\_G, C1\_B etc.

# Example Usage of AvdtaJSONSerializer and Metric

A screen shot of a computer

Description automatically generated with low confidence