

---

## title: LeadingAgile Code Analysis Tool Data Files

### 1. Language Analysis Data Files

- 1. Language Analysis Data Files
  - 1.1. data.json
  - 1.2. statistics.json
  - 1.3. trend.json
- 

#### 1.1. data.json

A list of dictionaries where each item is the data from a step in the data collection. For a gather run for 52 steps (`-t 52`) where the steps are (roughly) 7 days each, there will be 52 dictionary items in the list.

```
[
  {
    "revision_hash": "359aec0af3003bc2ce17b9b2c209e1e744a6a543",
    "hash_date": "2021-02-11T14:41:43+00:00",
    "extracted_data": {}
  },
  {
    "revision_hash": "e9507ca91d57b995a71adb1d0a2ca07d2e44d919",
    "hash_date": "2021-02-04T12:53:00+00:00",
    "extracted_data": {}
  }
]
```

Each dictionary item in the list has

- "revision\_hash": "359aec0af3003bc2ce17b9b2c209e1e744a6a543" The git repository commit hash for that step. This hash can be used to return to the code repository at that specific point in time to inspect the code.
- "hash\_date": "2021-02-11T14:41:43+00:00" The date from the commit hash.
- "extracted\_data": {} A dictionary of items for each thing being examined (methods, modules, files, framework, etc.). There will be as many items in the dictionary as there are unique things being examined. "Unique" is defined as having unique location information (for however much location information is available for a metric).

Extracted Data item format:

```
"57cd9568e357c3fbac9a4f712f7c6e3": {
  "location": {
    "file":
      "broker/src/main/resources/io/zeebe/broker/exporter/debug/mustache-3.0.0.min.js",
    "line": 1,
    "character": 1
  },
  "cyclomatic_complexity": 2,
  "method_length": 1,
  "method_name": "anonymous function"
}
```

Each item is identified with a key which is a hash of the location information. This is used to keep metrics collected from various options or tools to be stored together.

The item value is a dictionary with these fields:

- location A dictionary that has as much location information as available ( file , line, character, framework). All items are optional, but there will be at least one.
- Metric information. A metric name with the data from the metric. All metric name items are optional. The metrics present will depend on the metric collection tools used. Possible metric names are:
  - cyclomatic\_complexity
  - method\_length
  - fan\_out\_complexity
  - lines\_of\_code
  - number\_of\_comments
  - percentage\_of\_comments
  - fan\_in
  - fan\_out
  - instability
  - number\_of\_abstracts
  - number\_of\_concretes
  - abstractness
  - distance\_from\_main\_sequence
  - number\_of\_methods
  - number\_of\_tests
  - number\_of\_imports
- When available, a method\_name item which has the name of the method or function in the form reported by the tool used. Sometimes these will include the class hierarchy and the parentheses. Sometimes they will be a generic description of the method (e.g. for lambdas and anonymous functions). method\_name is optional data.

## 1.2. statistics.json

A collection of standard statistic measures calculated from the collected data (see above).

The top-level dictionary is a collection of items with keys that are the metric names (see above). There will be one item in the list for each metric type present in the collected data.

The value for each of these metric items is a list of dictionary items containing the calculated statistics. There will be one item for each step in the collected data.

```
{
  "complexity": [
    {
      "datetime": "2021-02-11T14:41:43+00:00",
      "mean": 1.8574479444741057,
      "median": 1,
      "mode": 1,
      "min": 1,
      "max": 38,
      "stdev": 1.8515008778840008
    },
    {
      "datetime": "2021-02-04T12:53:00+00:00",
      "mean": 1.8589375448671932,
      "median": 1,
      "mode": 1,
      "min": 1,
      "max": 38,
      "stdev": 1.8496301623884799
    }
  ],
  "fan_out": [],
  "method_lines": []
}
```

The keys for the possible metric items match the names of the metrics from the collected data. See above for the possible metric names.

The statistics data for each step contains:

- datetime: "2021-02-11T14:41:43+00:00", The date time from the commit hash in ISO format.
- mean: 1.8574479444741057,
- median: 1,
- mode: 1,
- min: 1,
- max: 38,
- stdev: 1.8515008778840008

Other than the date, the values are calculated from the metric data from all the items in a step in the collected data.

### 1.3. trend.json

The trend data is a collection of calculated data based on the data in the Statistic.Json.

This is basically the smoothing fit values and the forecast values derived using the Statistics data.

The top-level dictionary is a collection of items with keys that are the metric names (see above). There will be one item in the list for each metric type present in the collected data. This collection will exactly match the data from the source statistics json file.

The value for each of these metric items is a dictionary of items containing the calculated smooth-fit values for each statistic measure.

```
{
  "complexity": {
    "meta": {
      "type": "Exponential Smoothing",
      "source": "statsmodels.tsa.holtwinters",
      "function": "Holt",
      "params": {
        "damped_trend": true,
        "initialization_method": "estimated",
        "smoothing_level": 0.2,
        "smoothing_trend": 0.2,
        "steps_to_forecast": 10
      }
    },
    "means": {
      "fit_data": [
        -2.327344288200841,
        5.524836197788926,
        10.152443502581576,
        12.383136245463,
        18.487193791233775,
        22.79071266755824
      ],
      "forecast_data": [
        26.000093593112442,
        26.000022407199747,
        25.99996545846959,
        25.999919899485466,
        25.999883452298164,
        25.999854294548324,
        25.999830968348455,
        25.999812307388556,
        25.999797378620638,
        25.999785435606302
      ]
    }
  },
}
```

```

        "forecast_slope": -3.081575061401054e-05
    },
    "medians": {},
    "modes": {},
    "mins": {},
    "maxs": {},
    "stdevs": {}
},
"fan_out": {},
"method_lines": {}
}

```

The keys for the possible metric items match the names of the metrics from the collected data. See above for the possible metric names.

The items in each metric value will be a meta item and a pluralized keys for each of the statistic measures.

The meta item stores some information about what was used to generate the smooth-fit and forecast data.

Each of the statistic items contains:

- `fit_data` A list of fitted values (y-axis values) for the statistic type. There will be one value in this list for each step in the data collection and statistics.
- `forecast_data` A list of forecast values (y-axis values) projecting the statistic a few steps forward using the fitted data as the basis of the projection. There will be a small number of these items (currently 10).
- `forecast_slope` The slope of the forecast segment from the first forecast point to the last forecast point.