

EZapp (Owners: Ben, Beshoi)

EZapp is the endpoint for input data from outside of EZstack ecosystem. The majority of data communication from within EZapp to other EZstack services leverages Kafka streams to create a producer/consumer ecosystem.

The data transferred between EZapp and the Query Aggregator known hereafter as the Aggregator and our denormalizer consist of three general data types. This type of data gets placed on different Kafka topics so raw JSON data without any

Type 1: Input Data

Type 2: Query

Type 3: Denormalized Data

Logging (Owners: Beshoi)

The underlying logging system is a REST interface.

POST /logger/insert

Example 1:

INPUT:

```
{
  "type" : "info",
  "message" : "info message regarding your system"
}
```

OUTPUT:

```
{
  "result" : "success"
}
```

Example 2:

INPUT:

```
{
  "type" : "warning",
  "message" : "warning message regarding your system"
}
```

OUTPUT:

```
{
  "result" : "success"
}
```

Example 3:

INPUT:

```
{
  "type" : "error",
  "message" : "error message regarding your system"
}
```

OUTPUT:

```
{
  "result" : "success"
}
```

POST /logger/bulk

Example:

INPUT:

```
[[
  {
    "type" : "info",
    "message" : "info message regarding your system"
  },
  {
    "type" : "info",
    "message" : "info message regarding your system"
  },
  {
    "type" : "info",
    "message" : "info message regarding your system"
  }
]]
```

OUTPUT:

```
{
  "result" : "success"
}
```

GET /logger

| Attribute | Definition |
|-----------|--|
| type | The type of info that should be returned. Can be left blank to return all types. |
| last | Specifies the amount of records to be |

| | |
|--|-----------|
| | returned. |
|--|-----------|

Example:

/logger?type=info&last=5

OUTPUT:

```
[{
  "type" : "info",
  "message" : "info message regarding your system"
},
{
  "type" : "info",
  "message" : "info message regarding your system"
} ... ]
```

Samza Mesos (Owners: Beshoi)

The following application is utilized by the development team to run Samza jobs. The denormalizer and Query aggregation components both leverage the following interface to start there different Samza Jobs.

The configuration files developed need to follow the format specified in the samza mesos repo on git which is located here:

<https://github.com/gw-cs-sd/sd-18-ezstack/tree/samza-mesos/samza-mesos>

The jobs can be run and status can be retrieved using Samza StreamJob interface located:

<http://samza.apache.org/learn/documentation/0.13/jobs/job-runner.html>

Denormalizer (Owners: Ben)

Query Aggregation (Owners: Sam)