API Documentation

The MANRS benchmarking tool comes with an API that exposes the data gathered in the DataBase.

Routes

The following API routes are defined:

/reports/

This API route returns a collection of reports that meet the following criteria:

- Reports with `period_start` the same or later than the `period_start` specified on the request AND with `period_end` the same or sooner than the `period_end` specified on the request;
- Reports which contain any of the given ASNs specified with `asns` on the request (Optional);
- Reports with `type` the same as the one specified on the request (Optional and defaults to `auto`).

API call

```
POST /reports/ HTTP/1.1
Content-Type: application/json

{
    "period_start": "YYYY-MM-DD",
    "period_end": "YYYY-MM-DD",
    "asns": [1234,3456,678],
    "type": "<manual/auto>"
}
```

API reply

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "message": "OK",
    "data": [
        {
            "id": <report_id>,
            "period_start": <datetime.isoformat>,
            "period_end": <datetime.isoformat>,
            "type": <manual/auto>,
            "date_started": <datetime.isoformat>,
            "date_finished": <datetime.isoformat>,
        },
        ...
    ]
}
```

/reports/<id>/

This API route returns data for a specific report based on the report's `id`.

API call

```
GET /reports/<id>/ HTTP/1.1
```

API reply

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "message": "OK",
    "data": {
        "id": <report_id>,
        "period_start": <datetime.isoformat>,
        "period_end": <datetime.isoformat>,
        "type": <manual/auto>,
        "date_started": <datetime.isoformat>,
        "date_finished": <datetime.isoformat>,
    }
}
```

/reports/<id>/results/

This API route returns data for a specific report based on the report's `id` along with any results and statistics. Statistics are precalulated for all the data present in the report.

API call

```
GET /reports/<id>/results/ HTTP/1.1
```

API reply

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "message": "OK",
    "data": {
        "id": <report_id>,
        "period_start": <datetime.isoformat>,
        "period_end": <datetime.isoformat>,
        "type": <manual/auto>,
        "date_started": <datetime.isoformat>,
        "date_finished": <datetime.isoformat>,
        "results" : [
            {
                "asn": <ASN>,
                "m1": <float>,
                "m1_data": [
                    {
                        "prefix": <prefix>,
                        "weight": <float>,
                        "duration": <integer>,
                        "bgpstream_eventid": <integer>,
                    },
                    ...
                ],
                "m1c": <float>,
                "m1c_data": [
                    {
                        "prefix": <prefix>,
                        "weight": <float>,
                        "duration": <integer>,
                        "bgpstream_eventid": <integer>,
```

```
        },
        ...
    ],
    "m2": <float>,
    "m2_data": [
        {
            "prefix": <prefix>,
            "weight": <float>,
            "duration": <integer>,
            "bgpstream_eventid": <integer>,
        },
        ...
    ],
    "m2c": <float>,
    "m2c_data": [
        {
            "prefix": <prefix>,
            "weight": <float>,
            "duration": <integer>,
            "bgpstream_eventid": <integer>,
        },
        ...
    ],
    "m3": <float>,
    "m3_data": [
        {
            "prefix": <prefix>,
            "dates": [
                "YYYY-MM-DD",
                "YYYY-MM-DD",
                ...
            ],
        },
        ...
    ],
    "m6": <true/false>,
    "m6_data": {
        "checked_on": "YYYY-MM-DD",
        "has_exports": <true/false>,
        "has_imports": <true/false>,
    },
    "m7irr": <float>,
    "m7irr_data": {
        "checked_on": "YYYY-MM-DD",
        "total_routes_num": <integer>,
        "unregistered_routes_num": <integer>,
```

```
                    "unregistered_routes": [
                        "<prefix>",
                        "<prefix>",
                        ...
                    ],
                },
                "m8": <true/false>,
                "m8_data": {
                    "checked_on": "YYYY-MM-DD",
                    "has_contact_info": <true/false>,
                },
            },
            ...
        ],
        "stats": {
            "m1_mean": <float>,
            "m1_median": <float>,
            "m1c_mean": <float>,
            "m1c_median": <float>,
            "m2_mean": <float>,
            "m2_median": <float>,
            "m2c_mean": <float>,
            "m2c_median": <float>,
            "m3_mean": <float>,
            "m3_median": <float>,
            "m6_mode": <true/false>,
            "m7irr_mean": <float>,
            "m7irr_median": <float>,
            "m8_mode": <true/false>,
        }
    }
}
```

/results/

This API route returns a collection of results per ASN and the global statistics that meet the following criteria:

- Results with `period_start` the same or later than the `period_start` specified on the request AND with `period_end` the same or sooner than the `period_end` specified on the request;
- Results for the given ASNs specified with `asns` on the requests (Optional);
- If `metrics` is given only the specified `metrics` will be included in the results and statistics (Optional);

- If `only_metrics` is specified and `false` the results will also contain the accompanying data to the metrics (Optional defaults to `true`).

API call

```
POST /results/ HTTP/1.1
Content-Type: application/json

{
    "period_start": "YYYY-MM-DD",
    "period_end": "YYYY-MM-DD",
    "asns": [1234,3456,678],
    "metrics": ["m1", "m2"],
    "only_metrics": <true/false>
}
```

API reply

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "message": "OK",
    "data": {
        "asns": {
            "<ASN>": [
                {
                    "period_start": <datetime.isoformat>,
                    "period_end": <datetime.isoformat>,
                    "m1": <float>,
                    "m1_data": [
                        {
                            "prefix": <prefix>,
                            "weight": <float>,
                            "duration": <integer>,
                            "bgpstream_eventid": <integer>,
                        },
                        ...
                    ],
                    "m1c": <float>,
                    "m1c_data": [
                        {
                            "prefix": <prefix>,
                            "weight": <float>,
```

```
            "duration": <integer>,
            "bgpstream_eventid": <integer>,
        },
        ...
    ],
    "m2": <float>,
    "m2_data": [
        {
            "prefix": <prefix>,
            "weight": <float>,
            "duration": <integer>,
            "bgpstream_eventid": <integer>,
        },
        ...
    ],
    "m2c": <float>,
    "m2c_data": [
        {
            "prefix": <prefix>,
            "weight": <float>,
            "duration": <integer>,
            "bgpstream_eventid": <integer>,
        },
        ...
    ],
    "m3": <float>,
    "m3_data": [
        {
            "prefix": <prefix>,
            "dates": [
                "YYYY-MM-DD",
                "YYYY-MM-DD",
                ...
            ],
        },
        ...
    ],
    "m6": <true/false>,
    "m6_data": {
        "checked_on": "YYYY-MM-DD",
        "has_exports": <true/false>,
        "has_imports": <true/false>,
    },
    "m7irr": <float>,
    "m7irr_data": {
        "checked_on": "YYYY-MM-DD",
```

```
                          "total_routes_num": <integer>,
                          "unregistered_routes_num": <integer>,
                          "unregistered_routes": [
                               "<prefix>",
                               "<prefix>",
                               ...
                          ],
                     },
                     "m8": <true/false>,
                     "m8_data": {
                          "checked_on": "YYYY-MM-DD",
                          "has_contact_info": <true/false>,
                     },
                },
                {
                     "period_start": <datetime.isoformat>,
                     "period_end": <datetime.isoformat>,
                     ...
                },
                ...
           ],
           ...
     },
     "stats": [
          {
                "period_start": <datetime.isoformat>,
                "period_end": <datetime.isoformat>,
                "m1_mean": <float>,
                "m1_median": <float>,
                "m1c_mean": <float>,
                "m1c_median": <float>,
                "m2_mean": <float>,
                "m2_median": <float>,
                "m2c_mean": <float>,
                "m2c_median": <float>,
                "m3_mean": <float>,
                "m3_median": <float>,
                "m6_mode": <true/false>,
                "m7irr_mean": <float>,
                "m7irr_median": <float>,
                "m8_mode": <true/false>,
          },
          {
                "period_start": <datetime.isoformat>,
                "period_end": <datetime.isoformat>,
                ...
```

```
            },
            ...
        ],
    }
}
```

General Responses

The following general responses can also be returned while calling the API:

Bad Request

A call to the API contains invalid input.

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
 "title": "Invalid Input",
 "description": "<Description of the invalid input>"
}
```

Not Found

A requested resource is not found on the server.

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
 "message": "Not Found",
 "data": []
}
```

Service Unavailable

A resource could not be retrieved from the database. Retry later.

```
HTTP/1.1 503 Service Unavailable
Content-Type: application/json
Retry-After: <seconds>
```

```
{
 "title": "Service Outage",
 "description": "Resource currently unavailable."
}
```