SPP LEP/TROLIE API
Data Exchange Guide

Contents

About This Document

This document describes interaction with the SPP implementation of GEV's LEP (General Electric Vernova's Limit Exchange Portal), utilizing the TROLIE (Transmission Ratings and Operating Limits Information Exchange) API specification. This document describes the background for key messages and lays out the format and construction of REST function calls.

Change Summary

Rev	*Author*	*Date*	*Comments*
0.1	L. Reaves	9/20/24	Initial conceptual draft.
0.2	L. Reaves	10/3/24	Updates from GEV in sections 5.1 and 5.2.
0.3	L. Reaves	10/18/24	Updates from GEV in sections 5.1 and 5.2, added appendix.
1.0	L. Reaves	11/8/24	Updates from GEV in sections 4.5, 5, and 7.

Scope and Prerequisite Knowledge

This document is intended to be used by SPP Members, customers, and participating vendors as an aid in developing interfaces to access the SPP implementation of GEV's LEP.

The following topics are covered by this document:

- An introduction to the technologies (REST) used by the LEP APIs.
- An introduction on how to use the API specifications, including TROLIE.
- A description on how TOs and other users will authenticate with SPP to access the LEP APIs.
- An overview of the concepts included in the APIs.
- A subset of usage examples of the TROLIE API, to demonstrate the most frequent needs and patterns of usage.

1. Introduction

1.1 Purpose

FERC order 881 poses a significant data exchange challenge for all grid operators in North America that make operations decisions against the Bulk Electric System (BES). While TOs and TSPs have experience in exchanging real-time ratings dynamically over ICCP, the FERC order makes this data exchange much more complex. To summarize, the following changes are necessary:

- AARs are now required on every transmission line under the SPP Tariff, as well as certain other equipment.
- Traditionally, dynamic ratings / AARs have only been provided over ICCP in "real-time", to represent the current state. The FERC order now requires these to be forecast at least at hourly granularity over 10 days in the future, with the forecast itself being updated at least hourly.

The above is an oversimplification. For all this data related to ratings, a data exchange is required to solve the following problems:

- SPP, their neighboring reliability coordinators, and their member TOs must make operating decisions against the same set of ratings.
- Per the FERC order, SPP must store and make available not only the ratings themselves but the inputs and decisions that went into them, such as operator overrides and temporary ratings, for a period of 5 years.

LEP includes an API using REST technology to facilitate the above data exchange.

1.2 Document References

The following documents provide supporting information for items listed in this document. These are also listed in the Order 881 Member Impacting Program Overview document found in the Change User Forum reference documents at https://spp.org/spp-documents-filings/?id=466584.

DOCUMENT	LOCATION	DESCRIPTION
SPP Connectivity Specification	SPP Documents > Stakeholder Group Documents > Change User Forum > CUF Reference Documents > Technical Reference Documents > SPP Environments	SPP Connectivity Specifications updated to include LEP system

DOCUMENT	LOCATION	DESCRIPTION
TROLIE API specification and articles	https://trolie.energy	Transmission Ratings and Operating Limits Information Exchange (TROLIE) specifications and usage articles for interfacing with LEP
Web Services (API technical specifications)	SPP Documents > Stakeholder Group Documents > Change User Forum > CUF Reference Documents > Technical Reference Documents > Technical Specifications > Tech Specs > Future Tech Specs	Summary of TROLIE API Technical specifications for interfacing with LEP
Business Validations	SPP Documents > Stakeholder Group Documents > Change User Forum > CUF Reference Documents > Technical Reference Documents > Business Rule Guides > Future Business Rule Guides	A reference document to provide members a list of validations that the SPP Portal, LEP, and RST will perform upon submission or query of rating data
UI Guides	SPP Documents > Stakeholder Group Documents > Change User Forum > CUF Reference Documents > Technical Reference Documents > User Interface Guides > Future UI Guides	A reference document to provide members a guide for submitting and viewing data through the SPP Portal, LEP, and RST User Interfaces
SPP Operational Ratings Methodology	SPP Documents > Stakeholder Group Documents > Ambient Adjusted Ratings Implementation Task Force	A reference document to provide members with information on rating data exchange

1.3 Terminology

This section will be added to and refined as needed in future document versions.

AAR – acronym meaning Ambient Adjusted Ratings.

API – acronym meaning Application Programming Interface.

BES – acronym meaning Bulk Electric System.

Digital Certificate – an electronic object that acts as an identity card that establishes your credentials for doing business or other transactions over the Internet. A digital certificate is issued by a certification authority and it contains your name, the name of your organization, serial number, and Support End Date.

A digital certificate has a copy of your public key (generated when the certificate is assigned) which is used to translate and understand a channel of communication which is encrypted using your private key.

DLR – acronym for Dynamic Line Rating.

EMS – acronym for Energy Management System.

GEV – acronym meaning General Electric Vernova, the software vendor for the LEP.

HTTP – acronym meaning Hyper-text Transfer Protocol.

HTTPS – acronym meaning Hyper-text Transfer Protocol Secure. HTTPS is a secure protocol where the security is established by SSL (see terminology entry). HTTPS does not define nor does it add new communications features to HTTP, it is merely a secure version of HTTP. The HTTPS name is used to specify the protocol in the URL declaration to identify it as being secured by SSL.

ICCP – acronym meting Inter-Control Center Communications Protocol.

ISO – acronym meaning Independent System Operator.

JSON – acronym for JavaScript Object Notation.

LEP - GEV's Limit Exchange Portal.

LSA – acronym meaning Local Security Administrator.

RC – acronym meaning Reliability Coordinator.

REST – acronym Representational State Transfer, defined in Section 1.4 below.

RTO – acronym meaning Regional Transmission Organization.

SSL – acronym meaning Secure Sockets Layer. SSL is a protocol standard used to establish a secure, encrypted, connection between a given client and server.

TCP/IP – acronym meaning Transport Control Protocol over Internet Protocol.

TO - acronym meaning Transmission Owner.

TOP - acronym meaning Transmission Operator.

TROLIE – acronym meaning Transmission Ratings and Operating Limits Information Exchange.

TSP - acronym meaning Transmission Service Provider.

1.4 Why REST?

While REST isn't new[1], it may imply a learning curve for typical operations technology (OT) teams that need to integrate with LEP. This is because LEP must be integrated with EMS systems, where older methods have frequently been used

[1] REST was first defined in academic literature circa 2000- https://ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

for data exchanges, such as SCADA protocols, custom TCP-based protocols, and ad-hoc file exchanges.

To summarize the benefits of REST:

- REST is significantly easier to secure than either SCADA protocols or file-based exchanges. This aligns well with GE's "security built-in" strategy with GridOS, but also provides several security benefits and architectural flexibility to SPP.
- Data integrity and non-repudiation is much easier to achieve with REST, especially vs file exchanges.
- REST has established patterns for modeling complex data structures that would be awkward in SCADA-style exchanges like ICCP.
- REST has established patterns for making backward compatibility significantly easier than it would be for file exchanges without developing complex custom conventions.
- It will be significantly easier to scale the system in terms of both load and performance with REST than it would be with the more traditional exchange methods used in EMS.

While there are many good materials available for understanding the REST architecture in detail, users of the LEP APIs do not need to be experts in REST. From a user perspective, REST could be seen as a minor cognitive leap from file exchanges, which this document will illustrate through examples.

1.4.1 Why not ICCP?

The Inter-Control Center Communications Protocol (ICCP), part of IEC standard 60870-6, is used to connect SPP's SCADA system with member TOs' SCADA systems to capture operational data. It has traditionally been used for dynamic line ratings / real-time AARs for selected lines as an emerging technology. ICCP works well for this use case, as the rating of the line in real-time is functionally much like any other telemetry.

ICCP is reliable, mature, and well-known, and will continue to be supported by SPP as a method for submitting DLRs. However, ICCP won't scale well to handle the forecasted data. There are a couple reasons for this:

- The data model is a significant mismatch. ICCP is designed around SCADA concepts, such as points and quality flags. It is difficult to create arbitrarily complex data structures, like a forecast, and link them together. There are also needs to provide various strings into this data, such as override reasons.
- Using ICCP for forecasts would result in a significant increase in the count of SCADA points. The increased configuration overhead and load on potentially overtaxed SCADA systems makes ICCP less attractive, and

the model size increases poses a risk to the stability of the SPP ICCP systems.

This is all in addition to considering the advantages of REST discussed above.

1.4.2 Why not File Exchanges?

Unlike ICCP, file exchanges using common formats such as CSV, XML, and JSON do allow for more complex data structures. The advantages to REST however over file exchanges are all outlined above.

In addition to these more abstract advantages, there is a trend to move away from file-based data exchange for frequently updated data items, and to move towards an API based data exchange. File based interactions not only are more difficult to maintain a multi-site high availability posture, but also bring file lock and contention issues for both producers and consumers of the files.

1.4.3 REST is a Document Exchange Over the Web

To understand this API, REST could simply be seen as the exchange of JSON "files" over HTTP (or more precisely for LEP, HTTPS). LEP hosts a server, which may be accessed over the HTTPS port (443). As a user, one sends a JSON document to a URL, much like a website. This act of sending, called a "request", includes the following:

- A URL, much like one would enter a web browser to visit a website. This URL determines the operation of the API being invoked, as well as identifying information for the resource being edited. The URL may also include query parameters.
- A set of headers. These are key-value pairs that also provide metadata about the request.
- A "method" to be used against the resource. Methods are a built-in part of the HTTP protocol, and the typical standard methods are used by the LEP API. These include:
    - GET fetches a resource. GETs are generally queries.
    - HEAD only fetches HTTP response headers associated with a resource one would otherwise GET. This is for more advanced usage, often associated with caching. HEAD requests may be used to determine whether data returned by GET has changed without calling GET.
    - POST is used to create a new resource.
    - PUT is used to update a resource.

    - PATCH is used to partially update a resource. This is like PUT but implies that users don't have to provide the whole document. Rather, the JSON document would include only the changes.
    - DELETE, intuitively, deletes a resource.
- A JSON document to be used as input, referred to as a request "body". This is only applicable for POST, PUT and PATCH requests.

In return, the server will respond back with a response. The response includes:

- A numeric status code, indicating success or failure of the request. LEP uses standard HTTP status codes.
- A set of response headers. These are key-value pairs providing metadata of the response, much like the request headers.
- A JSON document, referred to as a response "body", including either the data requested, or a list of errors or warnings. Not all operations return bodies. HEAD requests, for example, never return bodies on success.

While toolkits are available in most modern languages for executing these requests in a high-performance way in-memory, it may be convenient for users to learn the API through file exchange. This is possible using the free open-source tool "curl" (https://curl.se/). Curl is a command-line tool that may be used to interact with REST APIs, including LEP, entirely using files. Curl is included with most Linux distributions as well as MacOS. It may be freely downloaded for Windows from the Curl website linked above.

1.5 TROLIE

GEV is a driving supporter of the TROLIE initiative (https://lfenergy.org/projects/trolie/). TROLIE (Transmission Ratings and Operating Limits Information Exchange) is an open-source REST API specification for rating exchange between TOs, RCs, and TSPs, managed by LF Energy, the energy subgroup of the Linux Foundation. TROLIE aims to create a vendor-neutral standard REST API specification for ratings exchange pursuant to FERC 881 to ease the integration burden across North American power grid operators.

LEP implements the TROLIE specification, which defines a certain set of methods intended specifically for interop. In addition, LEP provides additional API methods *not* included in TROLIE, primarily intended for internal usage by SPP or to drive user interfaces. However, the examples in this document are all focused on TROLIE usage, and reference the TROLIE specification itself, which may be downloaded from GitHub at https://github.com/TROLIE/spec.

2. OpenAPI Specifications and TROLIE

In practice, modern REST APIs are documented using OpenAPI specifications. These specifications dictate the operations that are available in the API, along expected parameters, headers, response codes, and JSON schema for requests and responses. This also includes documentation and descriptions of schema, operations, and parameters. OpenAPI specifications are specified in files, formatted using either YAML or JSON. LEP provides OpenAPI specification using YAML files, the more easily human-readable format.

2.1 Programming Against REST with OpenAPI

Many tools are available to work with OpenAPI, both commercial and open source. A good reference may be found at https://openapi.tools/. Libraries are available to work with REST and OpenAPI in most programming languages grid operators are familiar with, including Python, Java, C#/.NET and others. GE recommends TOs use these tools to integrate their systems with LEP.

As mentioned above, it is also possible to simply work with files using curl and similar utilities. GE recommends this be used primarily for testing and exploration purposes. Likewise, it is possible to invoke the GET operations of the API simply by putting the appropriate URLs in a web browser.

2.2 TROLIE Specification

The TROLIE OpenAPI specification may be downloaded from https://trolie.energy/openapi.yaml. Generated documentation is also available at https://trolie.energy/spec.

3. Authentication and Authorization

3.1 Users

The users of the LEP will primarily be Transmission Owners, Transmission Operators, peer RCs, peer TSPs, peer RTOs/ISOs, Market Monitors, and SPP internal systems.

3.2 SPP Portal Integration

Both the LEP API and LEP UI will be exposed over the SPP Portal.

3.3 SPP UAA Integration

3.3.1 Authentication and Authorization

Authentication is performed using Digital Certificates (and related technologies). A certificate identifies a particular User, and a User is related to a set of associated Roles by the SPP Authentication and Authorization system.

Every connection to the LEP is associated with a Role. When the connection is made, UAA first authenticates the user. After authentication, UAA looks up the user's permissions, or authorization, to LEP function, and the Scope of the user.

Assignment of the User's roles is necessary to perform API calls to the LEP.

3.3.2 Details of new Roles for the LEP application.

This information will be listed in this section until Roles and Scopes are finalized, and the "SPP Portal Application/Role Reference Guide" has been updated on the SPP Portal, at which time this section will simply reference that document.

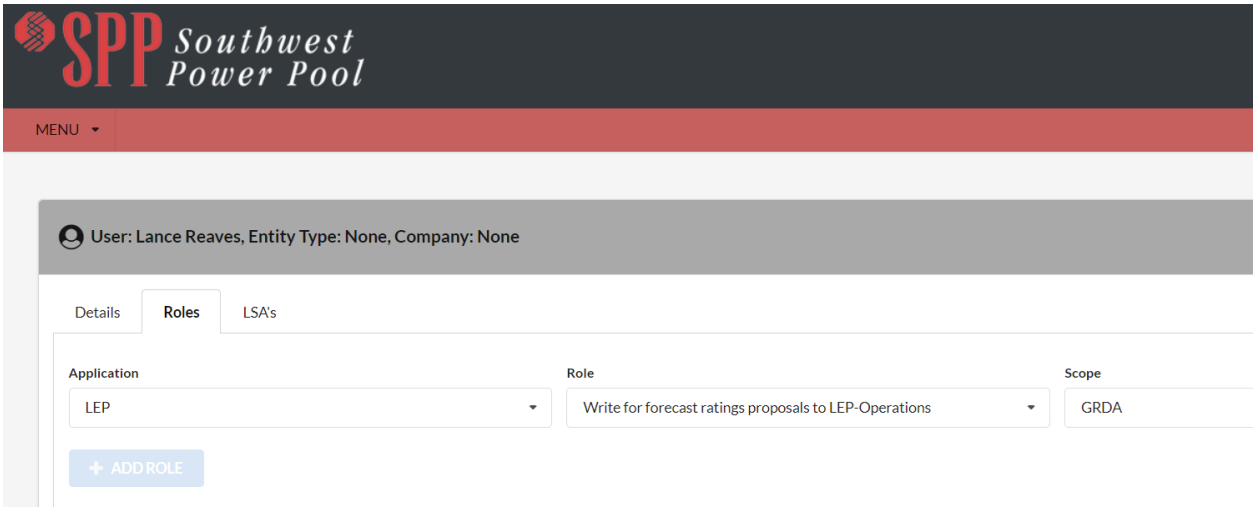
LSA's will have access to assign users access to the LEP application, and associated Roles and Scope.

Role	Only for the Rating Provider	Description
Read access to LEP-History data		Read access to LEP History data. Only includes global ratings stored in history. This role is intended to cover

Role	Only for the Rating Provider	Description
		market monitors and OASIS users as specified in the FERC order.
Read access for forecast snapshots in LEP-Operations		Read access for forecast snapshots (i.e. global ratings) in LEP-Operations. Will be needed by TOs, but also could be visible for other operations related use cases to various users.
Read-only access to raw rating submissions in LEP-Operations		Read-only access to raw rating submissions (i.e. internal ratings) in LEP-Operations. Likely only needed by users monitoring the submissions of others for support purposes. Also implies read access to temporary AAR exceptions.
Write for forecast ratings proposals to LEP-Operations	Y	Write for forecast ratings proposals. Also implies read access for proposals, temporary AAR exceptions.
Read access for limit real-time snapshots		Read access for limit real-time snapshots
Read-only access to real-time rating submissions		Read-only access to real-time rating submissions. Also implies read access to temporary AAR exceptions.
Read access for seasonal rating snapshots		Read access for seasonal rating snapshots. Also implies access to temporary seasonal ratings.

Role	Only for the Rating Provider	Description
Write monitoring sets	Y	Allows edit access for monitoring sets

Example of LEP application in the SPP Portal User Administration.



4. TROLIE API Concepts

Before reviewing examples, users may benefit from a basic understanding of the key constructs used by the LEP and TROLIE APIs. Definitions of these concepts are included here.

4.1 Transmission Facilities

A Transmission Facility is a logical part of the electrical network that may have a rating, whether simply seasonal or an AAR. This most often represents a transmission line but could also include transformers and other large pieces of equipment, or perhaps even logical points on the network such as interfaces. Most importantly, these are points at which SPP needs rating values to operate against.

Rating snapshots are always done against Transmission Facilities. Transmission Facilities include one or more Segments. The Transmission Facilities included in LEP must be pre-coordinated ahead of time through the modeling process.

4.2 Segments

Segments represent a component of some Transmission Facility that may affect its overall rating. All Transmission Facilities must have at least one Segment. In terms of TROLIE, ratings providers (TOs) are obligated to provide rating data, in the form of Proposals, against segments. On jointly owned lines or tie lines for example, each stakeholder in the line (the Transmission Facility) will be responsible for submitting Proposals against a different Segment in the model allocated to that stakeholder.

4.3 Proposals

Proposals are forecasted or real-time ratings values submitted to LEP/TROLIE against a particular Segment. They are referred to as “Proposals”, as they are inputs to the limit “clearing” process that will integrate them into a final in-use rating set. LEP, along with SPP’s EMS, will validate the proposals in coordination with other inputs, such as ICCP data, planned or current topology conditions, conflicting proposals and operator overrides to determine the ratings for each Transmission Facility in Snapshots. Snapshots are a distinct data set from Proposals. Proposals may be queried as well as submitted, so that the TO’s original input data is always kept separately from the in-use ratings.

4.4 Snapshots

As implied above, Snapshots are generated in LEP from the clearing process within the SPP EMS, which integrates all relevant proposals with other inputs to

generate in-use ratings for each Transmission Facility. TROLIE allows for TOs to fetch the latest snapshot, aka the latest "version" of the ratings data. LEP APIs will allow fetch of specific snapshots, including the latest.

4.5 Monitoring Sets

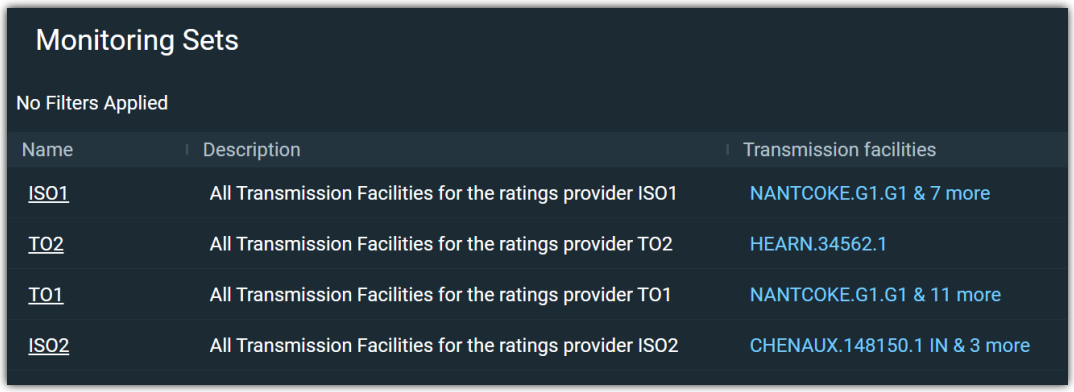
Monitoring Sets are arbitrarily defined sets of transmission facilities that may be used to filter ratings and limits returned by queries against these APIs. These are stored sets that may be defined arbitrarily by users of the API, so long as those users have permission to view relevant components.

4.5.1 Default Monitoring Sets

The system will also include some built-in Monitoring Sets, representing geographic regions, for instance.

The built-in Sets are defined in the Source and a user can choose to modify the list of elements that comprises the Monitoring Set via UI or LEP API.

An example of the LEP UI displaying built-in Monitoring Sets:



4.5.2 Creating and Maintaining Monitoring Sets

Besides the built-in sets, a user can create a Monitoring Set or update or delete an existing one.

Refer to Section 5.1.2 for an example on creating a monitoring set.

4.5.3 Using Monitoring Sets

Several operations support specifying a Monitoring Set as a parameter, e.g., getRealTimeLimits, getLimitsForecastSnapshot, getSeasonalRatingsSnapshot. Refer to Section 5.2.5.2 for an example on using a monitoring set.

4.6 Operating Limit

System Operating Limit is a well-defined industry term, but the salient point for TROLIE is that a Limit satisfies the most limiting of any provided reliability criteria. Limits for Transmission Facilities are determined by the Clearinghouse after considering all Ratings Proposals for the Power System Resources associated with that Transmission Facility during a particular Period of an Operational Window.

4.7 Forecast Limit

Forecast limits refer to the 240-hour-ahead forecasted AAR data set mandated by FERC order 881. While this is a forecast of limits as they would be in real-time, in practice these are often used for various processes involved in near-term planning of transmission services, including day-ahead markets and other look-ahead resource commitment processes, transmission scheduling and outage coordination.

4.8 Real-Time Limit

In addition to forecasts, TROLIE supports the exchange ratings within the current hour. SPP will not utilize this function. Forecast ratings sent during the current hour for the next hour will become the Real-Time ratings when the next hour occurs.

4.9 Naming Schemes

SPP will utilize the EMS Key as the identifier in LEP API calls. This key is available on the SPP Ratings Submission Tool on the EMS Elements tab. More information will be added to this section in future document versions.

5. Data Exchange Overview

This chapter runs through some simple examples of API usage for typical use cases. The examples focus on forecasted ratings. Real-time and seasonal ratings use very similar constructs and patterns.

NOTE: The URL information is provided for example only. Refer to the SPP Connectivity Specification document linked in section 1.2 Document References.

5.1 Monitoring Sets

The latest details about the operation can be found at:

TROLIE OpenAPI Specification

(https://trolie.energy/spec#tag/Monitoring-Sets)

5.1.1 Get Monitoring Set

5.1.1.1 Query Parameter

Use the following to get a single monitoring-set by Id.

Field	Type	Required	Example Value
Id		yes	test-monitoring-set
	Description:		

Use the following for returning multiple monitoring-sets.

Field	Type	Required	Example Value
transmissionFacilitiesSize	<int32>	yes	2
	Description:		

5.1.1.2 Example

Use the following content for input.json to query a monitoring-set.

```
{
    "id": "test-monitoring-set"
}
```

The following command will get a monitoring-set by id:

```
curl -d @input.json -X GET \
-H "Content-Type: application/ application/vnd.lep.monitoring-
set.v1+json" \
-o output.json -k \
--header "Authorization: Bearer <AUTH-TOKEN>" \
https://<baseURL>/lep-operations/...
```

This will return the Monitoring Set if present.


```
[{"name":"test-monitoring-set","description":"Example Monitoring
Set","transmission-facilities-count":2,"transmission-
facilities":["TOPOLOGY.TOP1.TOP2.LINE_NAME.B","TOPOLOGY.TOP1.TOP2.LINE_NAME.A"]}]
```


5.1.2 Create a Monitoring Set

To create a monitoring set, use the createMonitoringSet operation.

5.1.2.1 Query Parameter

Field	Type	Required	Example Value
id		yes	test-monitoring-set
	Description:		
description	string	no	Example Monitoring Set
	Description:		
Transmission-facilities-count	Integer <int32>	no	2
	Description:		
Transmission-facilities	Array of strings	yes	TOPOLOGY.TOP1.TOP2.LINE_NAME.A, TOPOLOGY.TOP1.TOP2.LINE_NAME.B
	Description:		

5.1.2.2 Example

Use the following content for input.json to create a monitoring-set.

```
{
    "id": "test-monitoring-set",
    "description": "Example Monitoring Set",
    "transmission-facilities": [
      "TOPOLOGY.TOP1.TOP2.LINE_NAME.A",
      "TOPOLOGY.TOP1.TOP2.LINE_NAME.B"
    ]
}
```

The following CURL command will create a monitoring set:

```
curl -d @input.json -X POST \
-H "Content-Type: application/vnd.lep.monitoring-set.v1+json" \
-o output.json -k \
--header "Authorization: Bearer <AUTH-TOKEN>" \
https://<baseURL>/lep-operations/monitoring-sets
```

This will return the newly created monitoring-set into output.json.


```
{"id":"test-monitoring-set","description":"Example Monitoring Set","transmission-facilities-count":2,"transmission-facilities":["239050c3-b40a-48c0-9191-9f7677608b60","2d9301d5-415f-46e2-91f3-c1f25d19d44b"]}
```


5.1.3 Update a Monitoring Set

5.1.3.1 Query Parameters

Field	Type	Required	Example Value
id		yes	test-monitoring-set
	Description:		
description	string	no	Example Updating Monitoring Set
	Description:		
Transmission-facilities-count	Integer <int32>	no	2
	Description:		
Transmission-facilities	Array of strings	yes	TOPOLOGY.TOP1.TOP2.LINE_NAME.A, TOPOLOGY.TOP1.TOP2.LINE_NAME.B, TOPOLOGY.TOP1.TOP1.OTHER_NAME.A
	Description:		

5.1.3.2 Example

Use the following content for input.json to update a monitoring-set.

```
{
    "id": "test-monitoring-set",
    "description": "Example Updating Monitoring Set",
    "transmission-facilities": [
      "TOPOLOGY.TOP1.TOP2.LINE_NAME.A",
      "TOPOLOGY.TOP1.TOP2.LINE_NAME.B",
      "TOPOLOGY.TOP1.TOP1.OTHER_NAME.A"
    ]
}
```

The following CURL command will update a monitoring set:

```
curl -d @input.json -X PUT \
-H "Content-Type: application/vnd.lep.monitoring-set.v1+json" \
-o output.json -k \
--header "Authorization: Bearer <AUTH-TOKEN>" \
https://<baseURL>/lep-operations/monitoring-sets
```

This will return the updated monitoring-set into output.json.

5.1.4 Delete a Monitoring Set

5.1.4.1 Query Parameter

Field	Type	Required	Example Value
id	<int32>	yes	test-monitoring-set
	Description:		

5.1.4.2 Example

Use the following content for input.json to delete a monitoring-set.

```
{
    "id": "test-monitoring-set"
}
```

The following command will delete a monitoring-set by id:

```
curl -d @input.json -X DELETE \
-H "Content-Type: application/ application/vnd.lep.monitoring-
set.v1+json" \
-o output.json -k \
--header "Authorization: Bearer <AUTH-TOKEN>" \
https://<baseURL>/lep-operations/...
```

This will delete the monitoring-set if present.

5.2 Forecast Ratings

The latest details about the operation can be found at:

TROLIE OpenAPI Specification

(https://trolie.energy/spec#tag/Forecasting)

5.2.1 Send Forecast Ratings

For most TOs, the most important operation will be to regularly (at least hourly) send forecasted AARs to SPP. This is done by updating that TO's forecasted rating Proposal. More specifically, this may be achieved using the *patchRatingForecastProposal* operation in TROLIE.

5.2.1.1 Query Parameters

Proposal-Header: Populated by the Ratings Provider in a request to submit forecast ratings. The Clearinghouse implementation should record this information, and the detail limits snapshot should reflect the appropriate metadata from the Clearinghouse Provider's perspective.

Ratings: Forecasted Ratings as an array of objects (resource-forecast-proposal), up to 50,000 items.

Field	Type	Required	Example Value
resource-id	string (resource-id) ^(.){0,250}$	yes	TOPOLOGY.TOP1.TOP2.LINE_NAME.A
	Description: Up to 250 characters. Contains a unique identifier for a power system resources, such as a transmission facility, segment, interface, etc.		

Periods: Array of Forecasted Period objects up to 300 items.

Field	Type	Required	Example Value
period-start	string <date-time>	yes	2025-11-01T01:00:00-05:00

Field	Type	Required	Example Value
	Description: Up to 25 characters. RFC 3339 date-time string with no fractional seconds component that designates a start or end to an operating period (such as an hour) that starts at a specified time. This will frequently be at the start of an hour, but may be finer-grained, such as every 30 minutes, should the Clearinghouse Provider choose. If the Transmission Provider is operating in EST, these are valid and equivalent values: • 2023-01-01T06:00Z • 2023-01-01T01:00-5:00 • 2023-01-01T00:00-6:00 • 2023-01-01T11:30+5:30 The server should uniformly represent date-times in the operational time zone of the Clearinghouse Provider.		
period-end	string <date-time>	yes	2025-11-01T02:00:00-05:00
	Description: Up to 25 characters. RFC 3339 date-time string with no fractional seconds component that designates a start or end to an operating period (such as an hour) that starts at a specified time. This will frequently be at the start of an hour, but may be finer-grained, such as every 30 minutes, should the Clearinghouse Provider choose. If the Transmission Provider is operating in EST, these are valid and equivalent values: • 2023-01-01T06:00Z • 2023-01-01T01:00-5:00 • 2023-01-01T00:00-6:00 • 2023-01-01T11:30+5:30 The server should uniformly represent date-times in the operational time zone of the Clearinghouse Provider.		

continuous-operating-limits: Active-power (object) or apparent-power (object) or current (object) or reactive-power (object) or overvoltage-threshold-pu (object) or overvoltage-threshold (object) or undervoltage-threshold-pu (object) or undervoltage-threshold (object) (Limit).

Field	Type	Required	Example Value
mva	integer <int32> [0 … 1440]	yes	160
	Description:		

Emergency-operating-limits: Array of objects (limit-value-set) between 1 and 10 items.

A set of general limit or rating values, each mapped to the various limit bands defined by the Transmission Provider's operating manual. This typically consists of a "normal" limit, as well as limits for various levels of emergency conditions (typically 2-4) defined in the operations manual.

Field	Type	Required	Example Value

duration-name	string ^[A-Za-z\-]{3,10}$	no	emergency
	Description: Up to 20 characters.		

limit: active-power (object) or apparent-power (object) or current (object) or reactive-power (object) or overvoltage-threshold-pu (object) or overvoltage-threshold (object) or undervoltage-threshold-pu (object) or undervoltage-threshold (object) (Limit). Defines the limit. In practice, most exchanges will only support one kind of limit for proposals and snapshots. However, the specification supports defining limits on a per resource basis as well as limit types that are not anticipated to be used to implement Order 881. TROLIE server implementations must support at least one of these limit types and should return a well-known application/problem+json response if they receive a proposal in an unsupported but valid limit type as defined here.			
Field	**Type**	**Required**	**Example Value**
mva		yes	165
	Description:		

5.2.1.2 Example

An example input could be done with the following JSON file below, assumed to be called *input.json*:

```
{
 "proposal-header" : {
   "source": {
     "last-updated": "2025-10-31T15:05:43.044267100-07:00",
     "provider": "UTILITY-A",
     "origin-id": "5aeacb25-9b65-4738-8a00-ac10afa63640" /* This is up to the
               proposal submitter and more for tracking/debugging purposes */
   },
   "begins": "2025-11-01T01:00:00-05:00",
   "default-emergency-durations": [
     {
       "name": "emergency",
       "duration-minutes": 30
     }
   ],
   "power-system-resources": [
     { "resource-id": "TOPOLOGY.TOP1.TOP2.LINE_NAME.A", /* This can be the
                                                          NETMOM id or MRID */
       "alternate-identifiers": [ /* Users can use aliases for ease of
                                     identification */
         {"name": "segmentX", "authority": "TO-NERC-ID"},
         {"name": "LINE1 SEG-X", "authority": "RC-NERC-ID", "mrid":
```

```
                                                "8badf00d"}
      ]
    }
  ]
},
"ratings": [
  {/* This object is an array.  One entry
       should be provided for each resource */
    "resource-id": "TOPOLOGY.TOP1.TOP2.LINE_NAME.A",
    "periods": [
      {/* One entry in this array should be provided for each period
           in the upcoming 240 hours.  The format of the times is based on
           date-times as specified in RFC 3339
           (https://www.rfc-editor.org/rfc/rfc3339) */
        "period-start": "2025-11-01T01:00:00-05:00",
        "period-end": "2025-11-01T02:00:00-05:00",
        "continuous-operating-limit": {
          "mva": 160
        },
        "emergency-operating-limits": [
          {
            "duration-name": "emergency",
            "limit": {
              "mva": 165
            }
          }
        ]
      },
      {
        "period-start": "2025-11-01T01:00:00-06:00",
        "period-end": "2025-11-02T02:00:00-06:00",
        "continuous-operating-limit": {
          "mva": 160
        },
        "emergency-operating-limits": [
          {
            "duration-name": "emergency",
            "limit": {
              "mva": 165
            }
          }
        ]
      }
    ]
  }
```

```
 ]
}
```

Given *input.json*, run the following command to send it to LEP:

```
curl -d @input.json \
-X PATCH \
-H "Content-Type: application/vnd.trolie.rating-forecast-
proposal.v1+json" \
-o output.json -k \
--header "Authorization: Bearer <AUTH-TOKEN>"
\"https://<baseURL>/rating-proposals/forecasts"
```

On success, output.json will contain a copy of the data just uploaded, with additional metadata on update time and status. An example of this response format is given below:

```
{
  "forecast-provider": {
    "provider": "UTILITY-A",
    "last-updated": "2023-07-12T15:05:43.044267100-07:00",
    "origin-id": "5aeacb25-9b65-4738-8a00-ac10afa63640"
  },
  "incomplete-obligation-count": 0,
  "incomplete-obligations": [],
  "invalid-proposal-count": 0,
  "proposal-validation-errors": []
}
```

5.2.2 Invalid Forecasts for Individual Resources

The TROLIE spec supports allowing some individual resource forecasts to be invalid without rejecting the entire proposal. For example, suppose the Ratings Provider submits a Forecast Proposal for two resources – `TOPOLOGY.TOP1.TOP2.LINE_NAME.A` and `TOPOLOGY.TOP1.TOP1.OTHER_NAME.A`. Further suppose that there's nothing wrong at all with the `TOPOLOGY.TOP1.TOP1.OTHER_NAME.A` forecast, but the `TOPOLOGY.TOP1.TOP2.LINE_NAME.A` forecast is missing an hour, with everything else about the request being valid. In this case TROLIE should a return response like the following:

```
HTTP/1.1 202 Accepted
Content-Type: application/vnd.trolie.rating-forecast-proposal-status.v1+json
Server: trolie.example.com
```

```
Date: Wed, 29 Feb 2024 12:03:20 GMT
ETag: "123e4567e89b12d3a456426614174000"
X-Rate-Limit-Limit: 100
X-Rate-Limit-Remaining: 97
X-Rate-Limit-Reset: 3400

{
  "forecast-provider": {
    "provider": "UTILITY-A",
    "last-updated": "2023-07-12T15:05:43.044267100-07:00",
    "origin-id": "5aeacb25-9b65-4738-8a00-ac10afa63640"
  },
  "begins": "2025-11-01T01:00:00-05:00",
  "incomplete-obligation-count": 1,
  "incomplete-obligations": [
      { "resource-id": "TOPOLOGY.TOP1.TOP2.LINE_NAME.A",
        "alternate-identifiers": [
          {"name": "segmentX", "authority": "TO-NERC-ID"},
          {"name": "LINE1 SEG-X", "authority": "RC-NERC-ID", "mrid": "8badf00d"}
        ]
      }
  ],
  "invalid-proposal-count": 1,
  "proposal-validation-errors": [
    {
      "message": "The `resource-forecast-proposal` for
`TOPOLOGY.TOP1.TOP2.LINE_NAME.A` is
                  incomplete.",
      "resource-id": "TOPOLOGY.TOP1.TOP2.LINE_NAME.A"
    }
  ]
}
```

Bear in mind the proposal must always be on-time. Moreover, there are other client errors that are not tolerated, including:

- Malformed requests, i.e., the JSON provided is not valid according to the media type schema.
- Unprocessable content error: when the Forecast Proposal is well-formed, but the units provided in any of the forecasts are invalid.
- Unprocessable content error: when none of the individual resource Forecast Proposals are valid, but the request is otherwise well-formed.

Additional client errors are identified in the patchRatingForecastProposal spec.

The flip-side of this accommodative approach is that clients will not receive an error response when one of their resource forecasts is invalid, so the spec defines `incomplete-obligation-count`:

The number of facilities for this provider whose Ratings Obligation has not been met in this forecast window. This number may be larger than the size of `incomplete-facilities`, *since the latter has a pre-defined upper bound for performance and application security reasons*.

NOTE: The Ratings Provider should check that this value is zero when they believe they have completed their submission process.

5.2.3 Multiple Submissions per Forecast Window

In every Forecast Window, a new area-wide Forecast Proposal is created on the TROLIE server of the Clearinghouse Provider. Each Ratings Provider then PATCHes the area-wide proposal with the forecasts for their respective Ratings Obligations. Any unmet Ratings Obligations will result in the Clearinghouse Provider using an appropriate Recourse Rating for those unmet obligations.

For Ratings Providers with a natural split in their Ratings Obligations, e.g., geographic or control areas, the PATCH semantics afford the ability to submit multiple Forecast Proposals containing just proposals for the relevant resources, if they choose to do so. This affordance can also be leveraged to split a large proposal into one or more parts in cases where that is advantageous from a performance or reliable delivery perspective.

5.2.4 Jointly Owned Facilities

In a jointly owned facility, there may be one or more Ratings Providers for a given facility. This is expected to be fairly typical on seams. From a submittal perspective, this is inconsequential: Each Ratings Provider simply submits their own Forecast Proposal for Ratings Obligation using the appropriate resource-id. As with all resource-id uses, the TROLIE spec is agnostic as to which kind of Power System Resource is nominated by the identifier, but it will typically be a Segment in the case of a Jointly Owned Facility with multiple Ratings Providers.

5.2.5 Get Limit Forecast Snapshot

The TO will be interested to obtain the Limits Forecast the Transmission Provider is currently using in Operations, relative to the current time.

The getLimitsForecastSnapshot operation can be used to obtain the Forecast snapshot.

5.2.5.1 Query Parameters

The query supports several parameters.

Field	Type	Example Value
offset-period-start	string <date-time>	2025-07-12T03:00:00-05:00
	Description: Up to 25 characters. Rather than returning the entire forecast from the beginning, i.e., the next operating period, instead return a subset of the forecast starting with the period starting at offset-period-start.	
period-end	string <date-time>	2025-07-12T03:00:00-05:00
	Description: Specifies the end of a period for which a filter queries. Only the periods that start prior to this time will be returned.	
monitoring-set	string (generic-identifier) ^(.){0,250}$	X-AMPL - A pre-coordinated id that nominates a `monitoring-set`.
	Description: Up to 250 characters. Only return ratings or limits for facilities of the associated monitoring-set. The identifier for a monitoring-set is pre-coordinated, but using the NERC id of the associated Ratings Provider is recommended.	
transmission-facility	string (generic-identifier) ^(.){0,250}$	86753_1 - A typical AC line identifier
	Description: Up to 250 characters. Only return limits for this transmission facility.	
static-only	boolean	true/false
	Description: Default: false. Only return limits determined by static ratings. Without this parameter a forecast limits snapshot would include limits determined after considering dynamic ratings, like AARs or DLRs, when those are available for a particular forecast period. With static-only=true, the forecast limits snapshot would only include limits that were determined using static ratings, such as seasonal ratings, seasonal overrides, temporary AAR exceptions, operator overrides, etc.	

5.2.5.2 Example

Assuming a MonitoringSet “test-monitoring-set” was created to contain the TransmissionFacilities of interest (see chapter 4.5 for more information on MonitoringSets) as below:

```
{
    "id": "test-monitoring-set",
    "description": "Example Monitoring Set",
    "transmission-facilities": [
        "TOPOLOGY.TOP1.TOP2.LINE_NAME.A"
    ]
}
```

Then the current in-use forecasted limits may be fetched with the following command:

```
curl -H "Accept: application/vnd.trolie.forecast-limits-snapshot-slim.v1+json" \
-o output.json -k \
--header "Authorization: Bearer <AUTH-TOKEN>" \
"https://<baseURL>/rating-proposals/forecasts?monitoring-set=test-monitoring-set"
```

This will return the current version of the in-use ratings for the next 240 hours into output.json. See the following for an example:


```
{
    "snapshot-header": {
        "begins": "2023-07-12T16:00:00-07:00",
        "source": {
            "provider": "X-AMPL",
            "last-updated": "2023-07-12T16:00:00-07:00",
            "origin-id": "5aeacb25-9b65-4738-8a00-ac10afa63640"
        },
        "default-emergency-durations": [
            {
                "name": "emergency",
                "duration-minutes": 30
            }
        ],
    },
    "limits": [
        {/* This object is an array.  One entry for each facility */
            "resource-id": "TOPOLOGY.TOP1.TOP2.LINE_NAME.A",
            "periods": [
                {
                    "period-start": "2023-07-12T16:00:00-07:00",
                    "period-end": "2023-07-12T17:00:00-07:00",
                    "continuous-operating-limit": {
                        "mw": 160,
                        "pf": 1.0
                    },
                    "emergency-operating-limits": [
                        {
                            "duration-name": "emergency",
                            "limit": {
                                "mw": 170,
                                "pf": 1.0
                            }
```

```
                        }
                    ]
                }
            ]
        }
    ]
}
```


NOTE: the above example assumes the next 240 hours *as determined by the computer clock where LEP is running*.

Given that there are edge cases in time and the user's clocks are likely slightly off from the LEP clock, it is recommended to specify the times more explicitly to ensure that users are getting what is expected. This may be done by specifying the "offset-period-start" parameter, like in the following example:

`https://<baseURL>/rating-proposals/forecasts?monitoring-set=test-monitoring-set&offset-period-start=2025-11-02T02:00:00-06:00`

NOTE: This query is an example of an HTTP GET. In addition to curl, the same URL may also be placed in a web browser to see the data.

5.3 Real-time Ratings

The latest details about the operation can be found at

TROLIE OpenAPI Specification

(https://trolie.energy/spec#tag/Real-Time)

5.3.1 Get Real-time Ratings

To obtain the Real-time Limits the Transmission Provider is currently using in Operations, the getRealTimeLimitsSnapshot operation can be used.

5.3.1.1 Query Parameters

The query supports several parameters.

Field	Type	Example Value
monitoring-set	string (generic-identifier) ^(.){0,250}$	X-AMPL - A pre-coordinated id that nominates a `monitoring-set`.
	Description: Up to 250 characters. Only return ratings or limits for facilities of the associated monitoring-set. The identifier for a monitoring-set is pre-coordinated, but using the NERC id of the associated Ratings Provider is recommended.	

transmission-facility	string (generic-identifier) ^(.){0,250}$	86753_1 - A typical AC line identifier
	Description: Up to 250 characters. Only return limits for this transmission facility.	

5.3.1.2 Example

Assuming a MonitoringSet “test-monitoring-set” was created to contain the TransmissionFacilities of interest (see chapter 4.5 for more information on MonitoringSets) as below:

```
{
    "id": "test-monitoring-set",
    "description": "Example Monitoring Set",
    "transmission-facilities": [
        "TOPOLOGY.TOP1.TOP2.LINE_NAME.A",
        "TOPOLOGY.TOP1.TOP2.LINE_NAME.B"
    ]
}
```

Then the current in-use forecasted limits may be fetched with the following command:

```
curl -H "Accept: application/vnd.trolie.realtime-limits-
snapshot.v1+json" \
-o output.json -k \
--header "Authorization: Bearer <AUTH-TOKEN>"
\"https://<baseURL>/rating-proposals/realtime?monitoring-
set=test-monitoring-set"
```

This will return the current version of the in-use ratings or limits for facilities of the associated monitoring-set into output.json.

```
{
  "snapshot-header": {
    "source": {
      "provider": "X-AMPL",
      "last-updated": "2023-07-12T15:05:43.044267100-07:00",
      "origin-id": "5aeacb25-9b65-4738-8a00-ac10afa63640"
    },
    "default-emergency-durations": [
      {
        "name": "emergency",
        "duration-minutes": 240
```

```
      }
    ],
    "power-system-resources": [
      {
        "resource-id": "TOPOLOGY.TOP1.TOP2.LINE_NAME.A",
        "alternate-identifiers": [
          {
            "name": "segmentX",
            "authority": "TO-NERC-ID"
          },
          {
            "name": "LINE1 SEG-X",
            "authority": "RC-NERC-ID",
            "mrid": "TOPOLOGY.TOP1.TOP2.LINE_NAME.A"
          }
        ]
      }
    ]
  },
  "limits": [
    {
      "resource-id": "TOPOLOGY.TOP1.TOP2.LINE_NAME.A",
      "continuous-operating-limit": {
        "mva": 160
      },
      "emergency-operating-limits": [
        {
          "duration-name": "emergency",
          "limit": {
            "mva": 165
          }
        }
      ]
    }
  ]
}
```


5.4 Seasonal Ratings

The latest details about the operation can be found at:

TROLIE OpenAPI Specification

(https://trolie.energy/spec#tag/Seasonal)

5.4.1 Get Seasonal Ratings

To obtain the Seasonal Limits the Transmission Provider is currently using in Operations, the getSeasonalRatingSnapshot operation can be used.

Field	Type	Example Value
monitoring-set	string (generic-identifier) ^(.){0,250}$	X-AMPL - A pre-coordinated id that nominates a `monitoring-set`.
	Description: Up to 250 characters. Only return ratings or limits for facilities of the associated monitoring-set. The identifier for a monitoring-set is pre-coordinated, but using the NERC id of the associated Ratings Provider is recommended.	
transmission-facility	string (generic-identifier) ^(.){0,250}$	86753_1 - A typical AC line identifier
	Description: Up to 250 characters. Only return limits for this transmission facility.	

5.4.1.1 Example

Assuming a MonitoringSet “test-monitoring-set” was created to contain the TransmissionFacilities of interest (see chapter 4.5 for more information on MonitoringSets) as below:

```
{
    "id": "test-monitoring-set",
    "description": "Example Monitoring Set",
    "transmission-facilities": [
        "TOPOLOGY.TOP1.TOP2.LINE_NAME.A"
    ]
}
```

Then the current in-use forecasted limits may be fetched with the following command:

```
curl -d @input.json -X GET \
-H "Content-Type: application/vnd.lep.???.v1+json" \
-o output.json -k \
--header "Authorization: Bearer <AUTH-TOKEN>" \
https://<baseURL>/lep-operations/...
```

This will return the current version of the in-use ratings or limits for facilities of the associated monitoring-set into output.json.

5.5 Using the Conditional GET

LEP API supports the Conditional GET pattern and clients should use it to determine when limits snapshots are available. In this article we will discuss this pattern in the context of obtaining a Forecast Limits Snapshot.

By employing the Conditional GET pattern, the client can efficiently determine when a new version of the resource is available without having to repeatedly download the entire resource during polling. This helps in reducing unnecessary network traffic and conserving bandwidth.

This pattern involves using the HTTP GET method along with conditional headers such as If-None-Match and If-Modified-Since to check if the resource located at /limits/forecast-snapshot has been modified since the last request.

Let's start with this initial request:

```
GET /limits/forecast-snapshot HTTP/1.1
Host: trolie.example.com
User-Agent: TROLIE-Examples-Client
Accept: application/vnd.trolie.forecast-limits-snapshot.v1+json
Accept-Encoding: br
```

Suppose this is the response from the LEP server:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.trolie.forecast-limits-snapshot.v1+json
Server: trolie.example.com
Date: Wed, 29 Feb 2024 12:00:00 GMT
ETag: "d41d8cd98f00b204e9800998ecf8427e"
X-Rate-Limit-Limit: 100
X-Rate-Limit-Remaining: 98
X-Rate-Limit-Reset: 3600

{... response body in Brotli compressed format ...}
```

When using the Conditional GET pattern, the client includes the previously received ETag and/or Last-Modified timestamp in the request headers. If the resource has not been modified since the provided ETag or Last-Modified timestamp, the server responds with a 304 Not Modified status code, indicating that the client's cached version is still valid. If the resource has been modified, the server responds with a 200 OK status code and provides the updated resource.

The client can then issue a Conditional GET:

```
GET /limits/forecast-snapshot HTTP/1.1
Host: trolie.example.com
User-Agent: TROLIE-Examples-Client
Accept: application/vnd.trolie.forecast-limits-snapshot.v1+json
Accept-Encoding: br
If-None-Match: "d41d8cd98f00b204e9800998ecf8427e"
```

Assuming a new snapshot hasn't been generated, we should see a response like the following:

```
HTTP/1.1 304 Not Modified
ETag: "d41d8cd98f00b204e9800998ecf8427e"
Server: trolie.example.com
Date: Wed, 29 Feb 2024 12:03:20 GMT
X-Rate-Limit-Limit: 100
X-Rate-Limit-Remaining: 97
X-Rate-Limit-Reset: 3400
```

Note: The X-Rate-Limit-Remaining is decremented here, since rate limiting is applied in all circumstances, including 3XX and 4XX responses.

Otherwise, we would have gotten a new limit forecast:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.trolie.forecast-limits-snapshot.v1+json
Server: trolie.example.com
Date: Wed, 29 Feb 2024 12:03:20 GMT
ETag: "123e4567e89b12d3a456426614174000"
X-Rate-Limit-Limit: 100
X-Rate-Limit-Remaining: 97
X-Rate-Limit-Reset: 3400
Content-Encoding: br

{... response body in Brotli compressed format ...}
```

Note to Implementors:

The use of If-None-Match with ETag headers is strongly encouraged to implement the Conditional GET pattern. The ETag should not be a simple hash of a particular representation of a resource, i.e., do not compute a hash of the JSON document returned to the client. Instead, the ETag should be unique to the logical state of the resource. One method to do that is illustrated in this pseudocode:

```
ETag: hash("{resource internal id}+{last modified timestamp}")
```

This has several desirable properties for an ETag:

- Unique: Minimizing the chance of collisions for different resources.
- Opaque: Clients cannot infer anything about the resource content or structure from the ETag.

Stable: Changes only when the resource is updated.

6. Versioning

Changes to the SPP LEP API will be communicated via normal Change User Forum method. More details may be added to this section in future document versions.

7. Appendix

7.1 Parameter Use Cases

This section lists the SPP specific use cases for various parameters.

Parameter Name	SPP Use Case
resource-id	SPP has chosen to use the RTNET.NETMOM Key. All records sent to LEP will be validated against SPP's current NETMOM model.

7.2 HTTP Response Codes

Code	Description
200	OK
201	OK Created
304	Not modified
400	Malformed request
401	Unauthorized
403	Access denied
406	Not Acceptable (Content negotiation failed)
410	Gone
429	Too many requests (rate limiting)
500	An unexpected error occurred

7.3 Ratings Timeline

For the TOs to visualize how the Ratings exchange happens relative to the SPP Real-time and Near-term time line, the diagram below can provide some perspective.