



Cisco Prime Performance Manager 1.7

REST API Guide

August 28 2015

Table of Contents

1	Purpose.....	3
2	Assumptions.....	3
2.1	Audience.....	3
2.2	HTTP Version	4
2.3	SSL and HTTP	4
2.4	Read Only API	4
2.5	Versioning	4
3	Prime Performance Manager REST API Installation	4
4	Reports Default Time Durations.....	4
5	REST API Components	5
5.1	Content-Type.....	5
5.1.1	HTTP “Accept” Header	5
5.2	Character Set Encoding	6
5.3	URI Syntax	6
5.4	HTTP Verb	7
5.5	Resources.....	7
5.5.1	Report Type.....	7
5.5.2	Report Listing (Report Categories and Report Names)	8
5.5.3	Reports	8
5.6	URL Encoding.....	9
5.7	Query String Parameters.....	9
5.7.1	FQDN.....	10
5.7.2	sgmid	10
5.7.3	startDate	11
5.7.4	endDate.....	12
5.7.5	durationSelect.....	13
5.7.6	intervalTypeKey.....	14
5.7.7	showSeconds	14
5.7.8	pageIndex	15
5.7.9	maxPageSize	15
5.7.10	outputType	16

5.7.11	csvHeader	16
5.7.12	includeDataTypes	16
5.7.13	includeMetadata	17
5.7.14	reportObjectFilter	17
5.7.15	filterAlgorithm	19
5.7.16	deviceTimeZone	19
5.7.17	showAllReports	20
5.7.18	sortDirection	20
5.7.19	sortedColumnId	21
5.7.20	precisionDigitLimit	21
5.7.21	columnHeaders	21
5.7.22	summary	22
5.7.23	seriesLimit	22
5.7.24	showViewReport	22
5.7.25	chartType	23
5.7.26	width	23
5.7.27	height	24
5.7.28	showLegend	24
5.7.29	showTitle	24
5.7.30	chartName	25
5.7.31	thresholdName	25
5.7.32	kpiName	25
5.7.33	kpiReportName	26
5.7.34	tcaType	26
5.7.35	tcaTenant	26
5.8	REST Parameter Rules	27
5.9	HTTP Request Headers	27
5.9.1	Accept	27
5.9.2	Authorization	28
5.10	HTTP Response Headers	28
5.11	HTTP Status Code	29
5.12	Error Responses	30
6	REST APIs	32

6.1	GET Report Listing	32
6.2	Report Listing Samples	32
6.2.1	CSV	32
6.2.2	JSONv2	33
6.2.3	JSON	33
6.2.4	XML	34
6.3	GET Report	36
6.4	Report Samples	36
6.4.1	CSV	36
6.4.2	JSONv2	37
6.4.3	JSON	39
6.4.4	XML	40
7	Pagination	41
8	References	42
9	Glossary	42

1 Purpose

This document describes the Cisco Prime Performance Manager Representational State Transfer (REST) API used to retrieve network/node level, dashboard and aggregated group reporting data from the Prime Performance Manager. The REST API is implemented as a web service front-end component on the Prime Performance Manager gateway. The API allows applications to retrieve performance data from the gateway using HTTP(s) GET requests using the URIs (Uniform Resource Identifier) specified in this document.

The data provided by the REST API can be in text format as XML (eXtensible Markup Language), JSON (JavaScript Object Notation), CSV (Comma Separated Value), image format as PNG (Portable Network Graphics), JPEG (Joint Photographic Experts Group), or PDF (Portable Document Format).

The report response format is chosen based on the request HTTP Accept Header or the value of the `outputType` query string parameter. If no specific data type is provided in either the HTTP Accept Header or in the `outputType` query string, the reporting data returned from the REST API defaults to the CSV text format.

2 Assumptions

2.1 Audience

This document is intended for developers who wish to implement the network management RESTful client to consume performance reporting data from the Prime Performance Manager application. The audience is assumed to be familiar with RESTful web services and the HTTP protocol.

2.2 HTTP Version

The Reports REST API embraces several aspects of the HTTP standards, including some HTTP headers, verbs, and status codes. The REST API uses status codes that are only supported in the HTTP/1.1 version. Therefore, it is expected that the client consuming this API complies with these HTTP/1.1 features.

2.3 SSL and HTTP

This API is served over an HTTPs secure connection when the Prime Performance Manager server has SSL installed and enabled. In this case the client side must support the HTTP/SSL protocol and have the Prime Performance Manager SSL public certificate installed on the client's trust store file.

2.4 Read Only API

This REST API is used for reading network performance reporting data. It supports the HTTP GET method which does not affect the state of the resource being queried in the server.

2.5 Versioning

The Prime Performance Manager Reports REST API complies with the following versioning rule: elements can be added when releasing a new version, but they may never be removed. A new release of this API is expected to be downward compatible with a previous version. An older version of this Reports REST API might appear to the client as if some elements were removed.

3 Prime Performance Manager REST API Installation

This document does not tell you how to install or configure the Prime Performance Manager. See the *Cisco Prime Performance Manager 1.7 User Guide* and *Cisco Prime Performance Manager 1.7 Quick Start Guide* for installation and configuration procedures. Also, the steps to configure SSL and authentication are not provided, as that pertains to the Prime Performance Manager server.

The Reports REST API is a Prime Performance Manager gateway component. All code needed to run and test the Reports REST API is included in the Prime Performance Manager installation. After you install the Prime Performance Manager, no additional Prime Performance Manager Reports REST API configuration is needed.

4 Reports Default Time Durations

The reporting data returned from the REST API defaults to the following durations, based off the `intervalTypeKey` parameter. If the `startDate` and `endDate` parameters are provided in the URL (refer to [Query String Parameters](#)), the duration is constrained by the time difference between the `endDate` and `startDate`.

Note: when the `intervalTypeKey` or `startDate/endDate` date parameters are not provided in the URL, the interval/duration defaults to “15 min / Last 12 hours” for node or group level reports, and “Hourly / Last 24 hours” for network level reports.

Table 1 - Report Default Durations

Interval	intervalTypeKey (parameter)	Duration
15 sec	FIFTEEN_SECS	Last 1 hour
30 sec	THIRTY_SECS	Last 1 hour
1 min	ONE_MINUTE	Last 1 hour
5 min	FIVE_MINUTE	Last 6 hours
15 min ¹	QUARTER_HOUR	Last 12 hours
Hourly ²	HOUR	Last 24 hours
Daily	DAY	Last 30 days
Weekly	WEEK	Last 6 months
Monthly	MONTH	Last 1 year

5 REST API Components

5.1 Content-Type

The reporting data provided by the REST API is formatted as text (XML, JSON, CSV), image (PNG, JPEG) or PDF. The data format type chosen for the responses is based on the `outputType` query string parameter or the value associated with the incoming HTTP Accept header. If no specific data type is found in the `outputType` query string parameter or the HTTP Accept Header is blank or `*/*`, the REST API defaults the response format and content-type to “text/csv”.

The `outputType` query string parameter has precedence over the HTTP Accept header. Therefore, when both the HTTP Accept header and the `outputType` query string parameter are found in the request, the response data type is based on the value defined in the `outputType` parameter.

5.1.1 HTTP “Accept” Header

The REST API implements the standard HTTP Accept header as specified in the Chapter 14.1 of the RFC 2616, Hypertext Transfer Protocol – HTTP/1.1. When the `outputType` parameter is not found in the request, the REST API parses and processes the incoming HTTP Accept header q-factors values to determine the best acceptable media type for the response.

Example of using the HTTP Accept header:

¹ Node or Group reports are by default rendered as 15 min interval reports. A node report requires either an `FQDN` Node or `sgmid` value to be provided in the `FQDN` or `sgmid` parameters in the REST API URL. A group report uses its own URL path “/groupreports” and the corresponding Group name defined in the `FQDN` parameter.

² Network level reports are by default rendered as hourly reports. Network level reports consist of data for the top node elements. Network level is the default report type when no `FQDN=Node` or `sgmid` query string parameter is provided in the REST API URL.

```
Accept=text/csv;q=0.5,application/pdf;q=0.7,application/xml;q=0.9,
application/vnd.cisco.ppm+json;v=2;q=0.95
```

For the above HTTP Accept header, the media-range with the highest q-factor is: application/vnd.cisco.ppm+json;v=2 followed by application/xml, application/pdf, and text/csv.

In this case, the REST API selects application/vnd.cisco.ppm+json;v=2 as the response format type which is the media-range with the highest q-factor.

The Accept header “application/vnd.cisco.ppm+json;v=2” can be used to select the newer JSON format, which is simpler to traverse through the reporting data than the previous JSON version. This HTTP Accept header has the same effect as passing outputType=jsonv2 as a query string parameter.

Note: a 406 status is returned for an Accept header that is not supported by the REST API.

5.1.1.1 Accept Header Preference

If the Accept header contains multiple supported media types, the REST API uses the HTTP Accept header “q” (quality factor) to determine the preferred client media type, as explained above. However, if the supported media-types have the same q (quality factor) values, the REST API selects the media range using the following order of precedence:

```
csv -> json -> xml -> png -> jpeg -> pdf
```

For example, if the following HTTP Accept header is sent to the REST API:

```
Accept=text/csv;q=0.5,application/pdf,application/xml
```

The selected media range should be “application/xml”.

5.2 Character Set Encoding

The REST API defaults all the text (XML, JSON, CSV) responses to the UTF-8 charset encoding, irrespective of the charset value requested by the client on the HTTP header request.

5.3 URI Syntax

The REST API calling URI syntax is defined as follows:

Table 2 – URL Syntax

URL Syntax
For network/node level web reports: <code>{scheme}://{host}:{port}/ppm/rest/{reportType}/{resource}?{parameters=...}..</code>

Table 3 – URL Parts

URL Parts	Sample Value	Description
scheme	<i>http or https</i>	Use https when Prime Performance Manager SSL is enabled; otherwise, it should be http.
host	For example: “ppm-clean.cisco.com”	The host name assigned to the Prime Performance Manager gateway server. Note: use your own domain/host name.

port	For example: 4440	HTTP port number assigned to the Prime Performance Manager gateway server. Note that the same port number can be used when Prime Performance Manager is running in SSL enabled mode.
path prefix	<i>/ppm/rest/</i>	All the REST API paths must be prefixed with this value. This is the entry point to the REST API.
reportType	<i>/dashboards</i> <i>/reports</i> <i>/groupreports</i> <i>/views</i> <i>/tcas</i>	This subpath identifies the report type. Note: The “/tcas” subpath identifies the TCAs (Threshold Crossing Alerts) report.
resources (listing)	<i>/{category}..</i>	A collection resource containing a list of all report categories, subcategories and/or corresponding report names URIs.
resources (reporting data)	<i>/{category}../{reportName}</i>	A document resource containing the report information (data) itself, which can be filtered using query string parameters.
parameters	<i>outputType, FQDN, sgmid, startDate, endDate, durationSelect, startDate, endDate...</i>	Refer to Query String Parameters

5.4 HTTP Verb

The Reports REST API is read-only. It only supports the HTTP GET verb. The following table shows an example of a URL and corresponding GET requests to retrieve a listing of {reportType} elements in JSON format. Notice that `jsonv2` corresponds to the newer JSON report format

Table 4 – HTTP URL Report Listing Sample

<code>http://ppm-clean.cisco.com:4440/ppm/rest/{reportType}?outputType=jsonv2</code>
<code>GET /ppm/rest/{reportType}?outputType=jsonv2</code>

5.5 Resources

5.5.1 Report Type

Reports within Prime Performance Manager are grouped into the following reporting types {reportType}:

Dashboards: The dashboard provides a way to group different reports within a single page. The REST API path to dashboard reports is `/ppm/rest/dashboards`.

Reports: Prime Performance Manager Web Reports. The REST API path to Prime Performance Manager web reports is `/ppm/rest/reports`.

Grouped Reports: Prime Performance Manager allows users to generate reports based on groups of network objects. The REST API path to grouped reports is `/ppm/rest/groupreports`.

Views Reports: Prime Performance Manager allows the user to create views and copy/paste reports to separate views. The REST API path to view reports is `/ppm/rest/views`.

TCAs Reports: Prime Performance Manager allows the user to retrieve TCA (Threshold Crossing Alerts) data. The REST API path to the TCA report is `/ppm/rest/tcas`.

Note: *When querying views from the REST API, the user is recommended to name that view uniquely to differentiate it from other views. For example, the user might consider prepending the word “REST” to the view name to help identify that view as being used from the REST API; for example – “REST <some view name>”*

5.5.2 Report Listing (Report Categories and Report Names)

Prime Performance Manager reports are grouped by {category}. A category is a unique Prime Performance Manager name that identifies a group of dashboard reports, network/node level web reports, and group reports.

Categories consist of subcategories that form a hierarchical path to various reports in the system. Categories and sub-categories are followed by a report name to identify a specific report in the system.

To retrieve a list of categories or report names, end the GET URL at the root level or at a specific category or sub-category name. For example:

GET `/ppm/rest/{reportType}/` or GET `/ppm/rest/{reportType}/{category}`

Table 5 – List of Report Examples

Resource Name	Description
<code>/ {reportType}</code>	The list of all top-level report categories for the {reportType}.
<code>/ {reportType}/ {category}..</code>	The list of report subcategories or report names available for that category. An empty {category} implies the root category that shows all the top {reportType} categories.
<code>/dashboards/Application+Dashboards/</code>	List all dashboard reports subcategories and dashboard reports names (if any) under the Application Dashboards category.
<code>/reports/Availability</code>	List all reports subcategories and reports (if any) under the “Availability” category.
<code>/groupreports/Transport+Statistics</code>	List all group reports subcategories and group reports names (if any) under the “Transport Statistics” category.
<code>/views</code>	List all the top views.
<code>/views/Data+Center</code>	List all the sub-views under the “Data Center” view.

5.5.3 Reports

The REST API allows users to retrieve all the reports available within the Prime Performance Manager system using HTTP GET URLs containing the corresponding full path to that report name. The Prime Performance Manager reports are uniquely identified by {reportType}, {category} and a {reportName}.

The REST API report response contains the reporting data associated with the requested report filtered by any query string parameters found in the URL.

Example: GET /ppm/rest/{reportType}/{category}../{reportName}?{queryString}

Note: {queryString} parameters like FQDN, sgmid, startDate, endDate, intervalTypeKey, duration, and others are used to filter out the reporting data returned from the system. The FQDN parameter or sgmid query string parameters are used when a specific node level report or group level report is required.

Table 6 – Path for a Network Level Report

Resource Name	Description
/reports/Availability/ICMP+Ping/ICMP+Ping+Availability	A report identified by the {reportName} “ICMP Ping Availability” which is part of the “ICMP Ping” sub {category} and “Availability” {category}.

5.6 URL Encoding

Encode the URL according to the Internet URL RFC 3986 standards. Some characters are not considered safe for URL³ use.

For example, if a query string parameter value should contain the positive sign (+), URL encode the “+” character to its corresponding URL escaped sequence “%2B”; otherwise, the “+” will be interpreted as a space by the REST API. Other characters such as “{”, “}”, “[”, “\”, “^”, “~”, “[“, “]”, and “” are unsafe because gateways and other transport agents are known to sometimes modify these characters.

When in doubt about the safety of a character within a query string parameter value, encode the query string parameter value before submitting a request.

The server side of the REST API always decodes the passed in query string parameter values according to RFC 3986 standards. Therefore, the client side can safely encode the parameter values using the same standards.

Note: The “+” (plus-sign) represents a space in the URL. If you want to pass a positive sign plus (+) character as part of a parameter value (for example, 2006-08-25T15:30+0800), encode the “+” to its respective URL encoded form “%2B”.

5.7 Query String Parameters

The following tables show the REST API request query string parameters. The REST API uses a subset of the HTTP request parameters found in the Prime Performance Manager web application. The parameter names found in the Prime Performance Manager web application are also found in the REST API. This allows the REST API and Prime Performance Manager web application to have consistent naming and also allows the same values used in the Prime Performance Manager web application to be passed to the REST API.

Notes:

- Parameter names/values are case-sensitive.
- Parameter values should be encoded to escape unsafe HTTP characters.
- Some parameters are mutually exclusive.

³ For more information, see section 2.2 URL Character Encoding Issues in RFC 1738.

5.7.1 FQDN

Table 7 – FQDN Parameter

Parameter	Notes
FQDN	<p>The FQDN (Fully Qualified Domain Name) value as defined in the Prime Performance Manager Server. It may contain the Node or Group, or the corresponding “Var Name (key)” found in the ProcessDBSummary section of the report definition XML file along followed by a specific value. A tenant report is specified by prepending the FQDN with a Tenant value.</p> <p>Format: Node=[host or IP],[other keys as defined in the corresponding report XML file]</p> <p>Node=[host or IP],Group=[group name],[other keys as defined in the corresponding report XML file]</p> <p>Group=[group name]</p> <p>[Var name as defined in the corresponding report XML file]=[value].</p> <p>Examples of valid formatted FQDN for some reports: FQDN=Node=em1941kbf.cisco.com FQDN=Tenant=MyTenant,Node=10.74.125.210 FQDN=Node=em1941kbf.cisco.com,ifDescr=FastEthernet0/0 FQDN=Node=10.74.125.210,CPUslot=0,CPUNum=0,processorIndex=1 FQDN=Node=NAM-ppm-nam01,Group=MY_GROUP FQDN=Group=CableModem FQDN=IPSLAOwner=Rubens FQDN=DatastoreName=Local+Datastore+sjo-smf-esxi-8</p> <p><i>Refer to your Prime Performance Manager server or network to determine your own FQDN values.</i></p>
Additional Notes	<ul style="list-style-type: none"> • Only supported for “GET /{reportType}/{category}..{reportName}” requests. • If an FQDN or sgmid is not provided, the REST API defaults to a top network level report. A network level report consists of information for the top samples in that report metric. • Refer to the Prime Performance Manager documentation for more information about network level reports. • Not supported by views.

5.7.2 sgmid

Table 8 – sgmid Parameter

Parameter	Notes
sgmid	<p>A positive integer number that uniquely identifies a node in the Prime Performance Manager server</p> <p>Format: Positive integer number greater than 0 (e.g. 1002, 13001, ..). -1 is also accepted, and it is optional. See note (3) below.</p> <p>Examples of valid formatted sgmid:</p>

```
sgmid=1002
sgmid=13001
```

Refer to your Prime Performance Manager server or network to determine your own sgmid values

Additional Notes

- Only supported in the “GET /{reportType}/{category}../{reportName}” requests.
- If an FQDN or sgmid is not provided, the REST API defaults to a top network level report. A network level report consists of information for the top samples in that report metric. Refer to the Prime Performance Manager documentation for more information about network-level reports.
- For consistency with the Prime Performance Manager web based application a sgmid equal to -1 means a network level report. However, this sgmid parameter value is not required as a network level report is already assumed when FQDN and sgmid are not given.
- Not supported by views.

5.7.3 startDate

Table 9 – startDate Parameter

Parameter	Notes
startDate	<p>The start date for the reporting data.</p> <p>Format: yyyy-MM-ddTHH:mmZ</p> <p>where, yyyy: year (e.g. 2010, 2011, 2012...) MM: month in year (e.g., 1-12) dd: day in month (e.g. 1, 2,...,31) T: charater to separate date from time. HH: hour in day (0-23) mm: minute in hour (0-59) Z: time zone represented as follows (+/-)HHmm “+” to add or “-“ to subtract the “HHmm” from the GMT time.</p> <p>Examples of valid formatted start date: startDate=2012-3-23T12:30%2B0800 startDate=2012-12-23T2:05-0500</p> <p><i>The timezone “+” sign needs to be URL encoded to “%2B”.</i></p>
Additional Notes	<ul style="list-style-type: none"> • Only supported in the “GET /{reportType}/{category}../{reportName}” requests. • The startDate/endDate and durationSelect parameters are mutually exclusive. • If the startDate is not given, it will default to the calculated value based on durationSelect and endDate (or system time if endDate is not given). • If both startDate and endDate are given, their time zone must match. For example: <pre>startDate is not given. endDate="2012-03-23T12:30+0800" duration="last24Hours" intervalTypeKey="HOUR" The startDate is 24 hours less than the endDate; that is "2012-03-22T12:30+0800"</pre> • The startDate may be calculated to be a date in the future. In that case, keep in mind that no reporting information is available in the system for a future date. • If startDate and endDate are not given, the current server time is used for the endDate. The

`startDate` is calculated by subtracting the `durationSelect` from the current time. In this case, the date/times will all be set according to the server time zone settings.

- Provide either the start or end date with proper GMT offset to ensure the reporting data received are in the expected time zone.
- Not supported by views.

5.7.4 endDate

Table 10 – endDate Parameter

Parameter	Notes
endDate	<p>The end date for the reporting data.</p> <p>Format: yyyy-MM-ddTHH:mmZ</p> <p>where, yyyy: year (e.g. 2010, 2011, 2012...) MM: month in year (e.g., 1-12) dd: day in month (e.g. 1, 2,...,31) T: character to separate date from time. HH: hour in day (0-23) mm: minute in hour (0-59) Z: time zone represented as follows (+/-)HHmm “+” to add or “-“ to subtract the “HHmm” from the GMT time.</p> <p>Examples of valid formatted end date: endDate=2012-04-23T02:30%2B0800 endDate=2012-07-01T03:00-0600</p> <p><i>The timezone “+” sign needs to be URL encoded to “%2B”.</i></p>
Additional Notes	<ul style="list-style-type: none"> • Only supported for the “GET /{reportType} /{category}../{reportName}” requests. • The “startDate”/“endDate” and “durationSelect” parameters are mutually exclusive. • If the endDate is not given, it will default to the calculated value based on durationSelect and startDate (or system time if startDate is not given). • If both startDate and endDate are given, their time zone must match. For example, endDate is not given. startDate=”2012-03-22T12:30+0800” duration=”last24Hours” intervalTypeKey=”HOUR” The endDate is 24 hours plus the startDate; that is “2012-03-23T12:30+0800” • The endDate may be calculated to be a date in the future. In that case, the user should keep in mind that no reporting information is available in the system for a future date. • If startDate and endDate are not given, the REST API will determine the startDate and endDate using the time zone set in the Prime Performance Manager server. In this case, the dates/times where REST client application is running and the dates/times set in the server may be different. • The user should provide either the start or end date with proper GMT offset to ensure the reporting data received are in the expected time zone. • Not supported by views.

5.7.5 durationSelect

Table 11 – durationSelect Parameter

Parameter	Notes
durationSelect	<p>The duration interval used for the reporting data.</p> <p>Possible values:</p> <ul style="list-style-type: none"> last5Years previousYear last1Year thisYear last6Months last90Days last12Weeks last60Days last8Weeks previousMonth thisMonth last30Days last4Weeks last21Days last14Days previousWeek thisWeek last7Days last3Days previousDay lastDay last24Hours today workShift last12Hours last6Hours previousHour lastHour <p>Example: durationSelect=last30Days</p>
<p>Additional Notes</p> <ul style="list-style-type: none"> • Only supported for the “GET /{reportType}/{category}../{reportName}” requests. • The startDate/endDate and durationSelect parameters are mutually exclusive. • The default value is <u>last24Hours</u> for network level report, unless the intervalTypeKey parameter is provided (refer to document section 4 Reports Default Time Durations). • The default value is <u>last12Hours</u> for a specific node report, unless the intervalTypeKey parameter is provided (refer to document section 4 Reports Default Time Durations). • The durationSelect must be greater than intervalTypeKey. • Not supported by views. <p>Default values for durationSelect are based on intervalTypeKey -- refer to document section 4 Reports Default Time Durations.</p>	

5.7.6 intervalTypeKey

Table 12 – intervalTypeKey Parameter

Parameter	Notes
intervalTypeKey	<p>The average interval type for the measurements in the reporting data.</p> <p>Possible values: FIFTEEN_SECS THIRTY_SECS ONE_MINUTE FIVE_MINUTE QUARTER_HOUR HOUR DAY WEEK MONTH</p> <p>Example: intervalTypeKey=MONTH</p>
<p>Additional Notes</p> <ul style="list-style-type: none"> • Only supported in the “GET /{reportType}/{category}../{reportName}” requests. • By default, reports queried using second intervals (FIFTEEN_SECS, THIRTY_SECS) will display the seconds in the timestamp. In that case the format for the timestamp will be: yyyy-MM-ddTHH:mm:ssZ (notice that the “ss” has been added to display seconds) • HOUR is the default for network reports; that is, when FQDN or sgmid are not given. • QUARTER_HOUR is the default for device reports; that is, either FQDN or sgmid is given. • Not supported by views. 	

5.7.7 showSeconds

Table 13 - showSeconds Parameter

Parameter	Notes
showSeconds	<p>A boolean parameter that display seconds on the report timestamp column.</p> <p>By default, except for (FIFTEEN_SECS, THIRTY_SECS) seconds interval, reports do not show seconds in the timestamp column. If the user passes this parameter with a value of “true”, the report shows the seconds, and the timestamp is formatted as yyyy-MM-ddTHH:mm:ssZ</p> <p>where, yyyy: year (e.g. 2010, 2011, 2012...) MM: month in year (e.g., 1-12) dd: day in month (e.g. 1, 2,...,31) T: character to separate date from time. HH: hour in day (0-23) mm: minute in hour (0-59) ss: seconds in minute(0-59) Z: time zone represented as follows (+/-)HHmm “+” to add or “-“ to subtract the “HHmm” from the GMT time.</p>

Examples of formatted timestamp:
 2013-04-23T02:30:15%2B0800
 2013-07-01T03:00:30-0600

Format:
 true or false

Example:
 showSeconds=false
 showSeconds=true

Additional Notes

- Only supported in the “GET /{reportType}/{category}../{reportName}” requests.
- By default, reports queried using the seconds interval (FIFTEEN_SECS or THIRTY_SECS) render the seconds on the report timestamp column. If the user does not want to see the seconds in the timestamp, this parameter (showSeconds) can be used with a value of false.

5.7.8 pageIndex

Table 14 – pageIndex Parameter

Parameter	Notes
pageIndex	<p>The report page number being requested for multi-page reports. This parameter only applies for pagination; that is, for partial report response (HTTP status response code=206).</p> <p>Format: Positive integer number greater than 0.</p> <p>Example: pageIndex=2</p>
<h4>Additional Notes</h4> <ul style="list-style-type: none"> • Only supported in the “GET /{reportType}/{category}../{reportName}” requests. 	

5.7.9 maxPageSize

Table 15 – maxPageSize Parameter

Parameter	Notes
maxPageSize	<p>The maximum total number of report records that can be found in the response.</p> <p>Format: a positive integer number greater than 0.</p> <p>Example: maxPageSize=10000</p>
<h4>Additional Notes</h4> <ul style="list-style-type: none"> • Only supported in the “GET /{reportType} /{category}../{reportName}” requests. • The default is <u>1000</u>. • This parameter represents the maximum number of records that may be found on the report response. 	

5.7.10 outputType

Table 16 – outputType Parameter

Parameter	Notes
outputType	<p>The content-type determines the expected format of the data in the response. This parameter, when provided, overrides the value defined in the HTTP “Accept” header.</p> <p>Possible values:</p> <ul style="list-style-type: none"> csv xml json jsonv2 jpeg pdf png <p>Example:</p> <pre>outputType=jsonv2 outputType=png</pre>
<p>Additional Notes</p> <ul style="list-style-type: none"> • This parameter is supported in all requests. • The default value is <u>csv</u>. 	

5.7.11 csvHeader

Table 17 – csvHeader Parameter

Parameter	Notes
csvHeader	<p>The csvHeader can be used to disable headers in CSV responses. By default CSV content contains headers in the first line. If the user does not want to see headers in the first line, this parameter should be set to “false”.</p> <p>Possible values:</p> <ul style="list-style-type: none"> true (default) false <p>Example:</p> <pre>csvHeader=false</pre>
<p>Additional Notes</p> <ul style="list-style-type: none"> • This parameter is supported in all requests. • The default value is <u>true</u>. 	

5.7.12 includeDataTypes

Table 18 – includeDataTypes Parameter

Parameter	Notes
includeDataTypes	<p>The includeDataTypes can be used to add data type headers to CSV, XML, JSON, and JSONv2 responses. By default content does not contain data type headers in the second line (or, for CSV only, in the first line if csvHeader=false). If the user wants to see data type headers, this parameter</p>

should be set to “true”.

Possible values:

true
false (default)

Example:

```
includeDataTypes=true
```

Additional Notes

- This parameter is supported in all CSV, XML, JSON, and JSONv2 requests.
- The default value is false.

5.7.13 includeMetadata

Table 19 – includeMetadata Parameter

Parameter	Notes
includeMetadata	<p>The includeMetadata can be used to add metadata to facilitate automated threshold creation to CSV, XML, JSON, and JSONv2 responses. These data include data type headers plus columnName and kpiName values for each thresholdable column, along with reportKey and kpiReport values for the entire report. By default content does not contain any of these data. If the user wants to receive metadata, this parameter should be set to “true”. NOTE: setting includeMetadata to true sets includeDataTypes to true, even if the URL has includeDataTypes=false.</p> <p>Possible values: true false (default)</p> <p>Example: includeMetadata=true</p>
<h4>Additional Notes</h4> <ul style="list-style-type: none"> • This parameter is supported in all CSV, XML, JSON, and JSONv2 requests. • The default value is <u>false</u>. 	

5.7.14 reportObjectFilter

Table 20 – reportObjectFilter Parameter

Parameter	Notes
reportObjectFilter	<p>The reportObjectFilter parameter is used to filter the reporting data using a specific report column name.</p> <p>The reportObjectFilter parameter consists of that report column name “var name” defined in the corresponding report XML ProcessDBSummary section followed by a macro or operator and a value.</p> <p>If you are looking at a dashboard report XML, you would first need to find the baseTable name for the corresponding WebReport TableView. Then, you would need to go to the Prime Performance Manager shell system prompt, and search and find the corresponding ProcessDBSummary web report XML that contains that table name in the folder “/opt/CSCOppm-</p>

gw/etc/pollers/system”. For example, let’s assume that the baseTable name is “SNMP”:

```
$ pwd
/opt/CSCOppm-gw/etc/pollers/system
$ grep -i "\bSNMP\b" *.xml | grep ProcessDBSummary
snmp.xml: <ProcessDBSummary baseTableName="SNMP"
dbnum="3">
```

In the above case, the file “snmp.xml” contains the ProcessDBSummary with the “Var name” used for that dashboard report.

Also, another way to find a valid column “Var name” for a given report is to send a REST query to a report (not dashboard) by passing an invalid column name followed by macro/operator and value. The REST API should respond with an error and the listing of valid column names for that report.

Formats:

- supported macro: contains
- supported operators: > < == != => <=

```
[Var name].contains("[some text value]")
not([Var name].contains("[some text value]"))
```

```
[Var name]=="[some text value]"
[Var name]!="[some text value]"
```

```
[Var name]==[number]
[Var name]!=[number]
```

```
[Var name]>[number]
[Var name]>=[number]
[Var name]<[number]
[Var name]<=[number]
```

Examples:

```
reportObjectFilter=IPSLAOwner.contains("some+text+value")
reportObjectFilter=not(IPSLAOwner.contains("some+text+value"))
reportObjectFilter=IPSLAOwner=="some+text+value"
```

```
reportObjectFilter=ResponseTimeAvg=="7"
reportObjectFilter=ResponseTimeAvg==7
reportObjectFilter=ResponseTimeAvg!="8"
reportObjectFilter=ResponseTimeAvg!=8
reportObjectFilter=ResponseTimeAvg==%2B7
reportObjectFilter=ResponseTimeAvg==7
```

```
reportObjectFilter=ResponseTimeAvg>2
reportObjectFilter=ResponseTimeAvg>-2
reportObjectFilter=ResponseTimeAvg>=2
reportObjectFilter=ResponseTimeAvg<2
reportObjectFilter=ResponseTimeAvg<=2
```

Additional Notes

- Only supported in the “GET /{reportType} /{category}../{reportName}” requests
- Not supported by views.

5.7.15 filterAlgorithm

Table 21 – filterAlgorithm Parameter

Parameter	Notes
filterAlgorithm	<p>The <code>filterAlgorithm</code> parameter is used to filter the reporting data using complex algorithms. This parameter is similar to the <code>reportObjectFilter</code> parameter, except that this parameter accepts complex algorithms. A complex algorithm takes combinations of “or”, “and” and any macro implemented by Prime Performance Manager reports. For more information about the Prime Performance Manager macro refer to Prime Performance Manager Integration Document.</p> <p>You may use combinations of “or” and “and” along with any accepted macro available in the Prime Performance Manager system to construct your complex algorithm. You should notice that we only validate if the algorithm can be parsed; that is, if Prime Performance Manager Algorithm Parser is able to parse the given algorithm. We do NOT validate the names of the table columns names and whether the algorithm is truly valid. Therefore, the user may get an empty response if Prime Performance Manager is applying the given algorithm on a set of columns that are not found for that report, or if the algorithm does not make any sense.</p> <p>Examples: <code>filterAlgorithm=CPUSlots==0 @and CPUNum==769 @and CPUUtilAvg5min>0.8</code></p>

Additional Notes

- Only supported in the “GET /{reportType} /{category}../{reportName}” requests
- Not supported by views.

5.7.16 deviceTimeZone

Table 22 – deviceTimeZone Parameter

Parameter	Notes
deviceTimeZone	<p>The <code>deviceTimeZone</code> parameter allows node level reports to display timestamps using the device (node) time zone. By default node level reports use timestamps displayed on the server configured time zone</p> <p>The <code>deviceTimeZone</code> parameter is a boolean text value that can be set for node level reports request.</p> <p>Possible values: <code>true</code> <code>false</code></p> <p>Examples: <code>deviceTimeZone=true</code> <code>deviceTimeZone=false</code></p>

Additional Notes

- The `deviceTimeZone` query string parameter is only acceptable for node level report queries.

5.7.17 showAllReports

Table 23 – showAllReports Parameter

Parameter	Notes
showAllReports	<p>The showAllReports parameter is used to display all the report names available in the Prime Performance Manager for a given category or sub-category.</p> <p>The showAllReports may be used to list all reports within a given category or sub-category. This parameter only applies when listing the reports available in the Prime Performance Manager system.</p> <p>Possible values: This parameter is an empty parameter. It does not accept a value.</p> <p>Example: <code>showAllReports</code></p>

Additional Notes

- This parameter is only applicable to generate a listing of reports for a given report category.
- There is no default value, as this parameter stands on its own

5.7.18 sortDirection

Table 24 - sortDirection Parameter

Parameter	Notes
sortDirection	<p>The sort direction to be applied on the report sorted column: ascending or descending.</p> <p>The sortDirection can be used on text formatted reports to define the sort order (descending or ascending) to be applied on the sortedColumnId. If the sortedColumnId is not provided, the default sorted column for that report will use the corresponding sort direction.</p> <p>Possible values: asc desc (default)</p> <p>Example: <code>sortDirection=asc</code></p>

Additional Notes

- Only supported in the “GET /{reportType} /{category}../{reportName}” requests
- The default value is based on user preference settings. If the user preference setting is not available, it defaults to “desc” (Descending)

5.7.19 sortedColumnId

Table 25 - sortedColumnId Parameter

Parameter	Notes
sortedColumnId	<p>The report column ID used to sort the reporting data based on a specific report column.</p> <p>The sortedColumnId consists of that report column name “Var name” defined in the corresponding report XML ProcessDBSummary. Or you can use the word “timestamp”. The “timestamp” is always an accepted sortedColumnId value.</p> <p>Possible values: Need to consult the report column “Var name” or use “timestamp”</p> <p>Example: sortedColumnId=timestamp sortedColumnId=CPUSlot</p>
<p>Additional Notes</p> <ul style="list-style-type: none"> • Only supported in the “GET /{reportType} /{category}../{reportName}” requests. • Defaults to sorting in descending order. To change the order, use the “sortDirection” parameter. 	

5.7.20 precisionDigitLimit

Table 26 - precisionDigitLimit Parameter

Parameter	Notes
precisionDigitLimit	<p>The precisionDigitLimit parameter rounds and formats report columns containing double and float numbers.</p> <p>Possible values: A positive integer number.</p> <p>Example: precisionDigitLimit=1 precisionDigitLimit=3</p>
<p>Additional Notes</p> <ul style="list-style-type: none"> • Only supported in the “GET /{reportType} /{category}../{reportName}” requests. • By default if the precisionDigitLimit parameter is not provided, the rounding of double or float decimal numbers uses the decimal precision defined for that report column. If a report column does not have a decimal precision defined, the float/double is rounded to 2 decimal places. 	

5.7.21 columnHeaders

Table 27 - columnHeaders Parameter

Parameter	Notes
columnHeaders	<p>The columnHeaders parameter allows users to alter the columns shown and the order in which they are presented. Headers (the column headers returned in the report) may be included more than once, in which case that column will be repeated. Headers that don’t match any header in the report are silently ignored. If you want to include a header that contains a comma, that comma</p>

should be replaced by “%1F”.

Possible values:

Any comma-separated list of URL-encoded column headers.

Additional Notes

- Only supported in the “GET /{reportType} /{category}../{reportName}”requests.

5.7.22 summary

Table 28 - `summary` Parameter

Parameter	Notes
summary	<p>The summary parameter generates a summarized report. That is, only the top N (N=10 by default) results are reported. The <code>seriesLimit</code> parameter can be used with the summary to specify the number of top rows to appear on the summary.</p> <p>Possible values: This parameter is an empty parameter. It does not accept a value.</p> <p>Example: <code>summary</code></p>
<p>Additional Notes</p> <ul style="list-style-type: none"> • Only supported in the “GET /{reportType} /{category}../{reportName}”requests . 	

5.7.23 seriesLimit

Table 29 - `seriesLimit` Parameter

Parameter	Notes
seriesLimit	<p>The <code>seriesLimit</code> parameter is used with the summary parameter to allow the user to specify the number of top rows to appear on the summary report.</p> <p>Possible values: Positive integer number.</p> <p>Example: <code>seriesLimit=5</code></p>
<p>Additional Notes</p> <ul style="list-style-type: none"> • Only supported in the “GET /{reportType} /{category}../{reportName}”requests . • It is only supported for summary reports. 	

5.7.24 showViewReport

Table 30 - `showViewReport` Parameter

Parameter	Notes
showViewReport	<p>A view might have a subview or a view report. By default, when you are in a view, a report is displayed if available. If there is no report and a subview is found, the REST API responds with a list of the subviews. If the user wants to</p>

<p>always see a list of subviews, use <code>showViewReport=false</code>.</p> <p>Possible values: true false</p> <p>Example: <code>showViewReport=false</code> <code>showViewReport=true</code> (default)</p>
<p>Additional Notes</p> <ul style="list-style-type: none"> • Only supported in the “GET /views/{category}../{viewName}” requests. • The <code>showViewReport</code> cannot be used with the <code>showAllReports</code> parameter. They are mutually exclusive. • It is only supported by views.

5.7.25 chartType

Table 31 - chartType Parameter (charts image reports)

Parameter	Notes
chartType	<p>chartType selects the type of chart for image reports. The chartType allows the user to select the type of chart to be rendered for the image reports.</p> <p>Possible values: bar line percentageArea percentageColumn stackedArea stackedColumn</p> <p>Example: <code>chartType=percentageArea</code></p>
Additional Notes	<ul style="list-style-type: none"> • Only supported in the “GET /{reportType} /{category}../{reportName}” graph image reports requests. • The default value is “line”

5.7.26 width

Table 32 - width Parameter (charts image reports)

Parameter	Notes
width	<p>The width parameter is used for the chart width on image reports. The width allows the user to specify the width in pixels of image report charts.</p> <p>Possible values: Positive integer value.</p> <p>Example: <code>width=800</code></p>
Additional Notes	<ul style="list-style-type: none"> • Only supported in the “GET /{reportType} /{category}../{reportName}” graph image reports requests.

- The default value is 1000

5.7.27 height

Table 33 - **height** Parameter (charts image reports)

Parameter	Notes
height	<p>Specifies the chart height on image reports. The height allows the user to specify the height in pixels of image report charts.</p> <p>Possible values: Positive integer value.</p> <p>Example: height=800</p>
<p>Additional Notes</p> <ul style="list-style-type: none"> • Only supported in the “GET /{reportType} /{category}../{reportName}” graph image reports requests. • The default value is 1000. 	

5.7.28 showLegend

Table 34 - **showLegend** Parameter (charts image reports)

Parameter	Notes
showLegend	<p>When set to false, hides the legend on image reports charts.</p> <p>Possible values: true false.</p> <p>Example: showLegend=false</p>
<p>Additional Notes</p> <ul style="list-style-type: none"> • Only supported in the “GET /{reportType} /{category}../{reportName}” graph image reports requests. • The default value is true. That is, legend is always displayed. 	

5.7.29 showTitle

Table 35 - **showTitle** Parameter (charts image reports)

Parameter	Notes
showTitle	<p>If set to false, hides the title on image reports charts.</p> <p>Possible values: true false.</p> <p>Example: showTitle=false</p>
<p>Additional Notes</p> <ul style="list-style-type: none"> • Only supported in the “GET /{reportType} /{category}../{reportName}” graph image reports requests. • The default value is true. That is, title is always displayed. 	

5.7.30 chartName

Table 36 - **chartName** Parameter (charts image reports)

Parameter	Notes
chartName	<p>Allows users to specify a chart where there is more than one chart per report.</p> <p>For example, under the Resources -> Disk -> Disk Space Statistics Hourly, one can find 4 (four) different charts: Disk Space Utilization Disk Space Total Bytes Disk Space Available Bytes Disk Space Used Bytes</p> <p>In the above example, if the chartName parameter is not provided, the first chart (Disk Space Utilization) is rendered. If the user wants other charts to be rendered, the corresponding chart name must be provided in the chartName parameter</p> <p>Example: chartName=Disk+Space+Total+Bytes</p>
<p>Additional Notes</p> <ul style="list-style-type: none"> • Only supported in the “GET /{reportType} /{category}../{reportName}” graph image reports requests. • By default if the chartName is not provided, the first chart for that report is displayed. 	

5.7.31 thresholdName

Table 37 - **thresholdName** Parameter

Parameter	Notes
thresholdName	<p>For threshold crossing alerts (TCA), the threshold name assigned by the user who created the threshold. This parameter may be used to filter out the TCA (threshold crossing alerts) report.</p> <p>Example: thresholdName=IPSLADNS_ReachabilityPercentage</p>
<p>Additional Notes</p> <ul style="list-style-type: none"> • Only supported in the “GET /tcas” TCA report requests. 	

5.7.32 kpiName

Table 38 - **kpiName** Parameter

Parameter	Notes
kpiName	<p>The KPI name associated with a threshold. This parameter may be used to filter out the TCA (threshold crossing alerts) report.</p> <p>Example: kpiName=Reachability Percentage</p>
<p>Additional Notes</p> <ul style="list-style-type: none"> • Only supported in the “GET /tcas” TCA report requests. 	

5.7.33 kpiReportName

Table 39 - kpiReportName Parameter

Parameter	Notes
kpiReportName	<p>The PPM XML fully qualified report name associated with a given threshold. This parameter may be used to filter out the TCA (threshold crossing alerts) report.</p> <p>Example: kpiReportName=REPORT, level1IPSLA, level2DNSQuery, wrnIPSLADNSDashboard</p>
<p>Additional Notes</p> <ul style="list-style-type: none"> Only supported in the “GET /tcas” TCA report requests. 	

5.7.34 tcaType

Table 40 - tcaType Parameter

Parameter	Notes
tcaType	<p>By default the TCA records returned in a TCA report only contains TCA non-alarm events. If the user wants to see both non-alarm and alarm events, or if the user wants to see only alarms, then the parameter tcaType should be used.</p> <p>Possible values: alarms events (default) both</p> <p>Example: tcaType=alarms tcaType=both tcaType=events</p>
<p>Additional Notes</p> <ul style="list-style-type: none"> Only supported in the “GET /tcas” TCA report requests. By default the tcaType is events. 	

5.7.35 tcaTenant

Table 41 - tcaTenant Parameter

Parameter	Notes
tcaTenant	<p>By default the TCA records returned in a TCA report are not filtered by tenant. If the user wants to see only records for a certain tenant, then the parameter tcaTenant should be used with the value set to that tenant’s name.</p> <p>Example: tcaTenant=MyTenantName</p>
<p>Additional Notes</p> <ul style="list-style-type: none"> Only supported in the “GET /tcas” TCA report requests. By default the tcaType is events. 	

5.8 REST Parameter Rules

The following request parameters rules are enforced by the REST API. An error response can be returned if any of these rules is violated.

- Only parameters identified in the Query String Parameters table above are accepted.
- Parameter names and values are case sensitive.
- Parameter values must be properly formatted.
- FQDN and sgmid are mutually exclusive.
- sgmid must correspond to a valid node.
- An FQDN Node must correspond to a valid node.
- startDate, endDate and durationSelect are mutually exclusive.
- endDate must be greater than startDate.
- durationSelect must be greater than intervalTypeKey.
- showAllReports should not contain a value.
- showViewReport and showAllReports are mutually exclusive.
- Chart parameters (e.g., height, width) are only accepted for report graphs requests
- TCA parameters (e.g., thresholdName, kpiName) are only accepted for TCA report requests

5.9 HTTP Request Headers

The following table shows the supported HTTP Request headers.

5.9.1 Accept

Table 42 – Accept HTTP Header

Header Name	Notes
Accept	<p>The requested content-type for the response. This header indicates to the REST API server the media type of the response that the client expects. When this header is present, the REST API attempts to respond to the client using the client's Accept header preferred media type; unless, that content-type is not supported.</p> <p>The REST API implements the Accept header as specified per the Internet HTTP standards defined in the chapter 14.1 of the RFC 2616, Hypertext Transfer Protocol – HTTP/1.1.</p> <p>Note: the “outputType” parameter overrides the Accept header. This means that if the “outputType” parameter is found in the request, the Accept header is not processed.</p> <p>If the Accept header contains multiple media types supported by the REST API, the corresponding q-factor value is used to determine the best media type. If there is a match, the REST API chooses the first media range type found on the Accept header.</p> <p>The “application/vnd.cisco.ppm+json;v=2” can be used to select the json version 2 report formats. The JSON version 2 is a simpler data</p>

structure that is easier to traverse than the previous version.

Example to select the JSON version 2 (simpler JSON format):
`Accept=application/vnd.cisco.ppm+json;v=2`

Example to select PDF from the Accept header:
`Accept=application/pdf`

Example to select CSV:
`Accept=application/pdf;q=0.5, text/csv;q=0.8`

Example to select JSON v2
`Accept= application/pdf;q=0.5, text/csv;q=0.8, application/vnd.cisco.ppm+json;v=2`

Additional Notes:

- The outputType query string parameter overrides the Accept header.
- A 406 error status code is returned for an Accept header not supported by the REST API.

5.9.2 Authorization

Table 43 – Authorization HTTP header

Header Name	Notes						
Authorization	<p>This request header should be sent to the REST API when the underlying Prime Performance Manager server has user access enabled. The REST API follows the HTTP Basic Authentication Scheme standards. The specifics of how this header should be formatted are defined in the RFC 2616 HTTP 1.1 standards – section 14.8 Authorization of RFC 2616, and in the RFC 2617 HTTP Authentication: Basic and Digest Access Authentication – section 2 Basic Authentication Scheme.</p> <p>That is, when the Prime Performance Manager has user access enable, the REST API requests should be sent over an HTTPS connection. And the HTTP Authorization header should contain the text “Basic username:password”, where the “username:password” should be encoded according to the Base64 format.</p> <p>For example, if the user agent wishes to send user id “rubens” and password “r1b2ns”, it would Base64 encode the text “rubens:r1n2ns”. And it would send the following header field:</p> <table border="0"> <tr> <td>Header Key</td> <td> </td> <td>Header Value</td> </tr> <tr> <td>Authorization</td> <td> </td> <td>Basic cnViZW5zOnIxYjJucw==</td> </tr> </table> <p>Format:</p> <p>Refer to section 2 RFC 2617 Basic Authentication Scheme.</p>	Header Key		Header Value	Authorization		Basic cnViZW5zOnIxYjJucw==
Header Key		Header Value					
Authorization		Basic cnViZW5zOnIxYjJucw==					
Additional Notes:	<ul style="list-style-type: none"> • Verify that this header is sent with the request if Prime Performance Manager has user access enabled • The userId and password defined in the Prime Performance Manager server must be base64 encoded and added to the header as defined in the HTTP standards. 						

5.10 HTTP Response Headers

The following table shows the HTTP Response headers supported.

Table 44 – HTTP Response Headers

Header Name	Sample Value	Description
Allow (HTTP/1.1)	For example, “GET”	This header is generated along with status code “405” in response to request with a method (HEAD, POST, PUT, DELETE) that is not supported for that resource.
Cache-Control (HTTP/1.1)	For example, “no-cache” OR “max-age=3600”	When “no-cache” is found, the client should not cache the response. Otherwise if max-age is found, response can be cached by the client. The max-age value tells the client how long in seconds the response should be cached.
Content-Type	For example. “application/xml;charset=UTF-8”	The content-type of the response followed by the charset encoding.
Date	For example. “Date Fri, 27 Jan 2012 20:29:02 GMT”	The date the response was created.
WWW-Authenticate	“Basic realm=RESTSecureArea”	This header is sent back to the client when Prime Performance Manager has user access (authentication) enabled, and the system failed to authenticate the user’s request (see the Authorization Request header above). This response header conforms to the HTTP Basic WWW-Authenticate header defined in HTTP 1.0 standards.
Content-Range	pages [page number]/[total pages] For example. “pages 2/10” In the example the current response contains page 2 of a report that has 10 pages total.	This response header is sent back when more than one page is available for the requested report.

5.11 HTTP Status Code

The following table shows the HTTP Status codes supported.

Table 45 – HTTP Status Codes

Status Code	Description
200	OK response. The request was correctly processed, and a response with a content body was produced.
204	No content found. That is, the request was a valid request; however, the system did not have any content available for the resource requested. For example, this would be the case for a valid report that has no data in it.

	<i>Notice that no message-body (no content) is found in an HTTP response that has 204 as the status code.</i>
206	Partial content. This status code is returned when a partial set of the report data has been returned. It is sent along with a “Content-Range” header. This status code indicates that pagination is required to retrieve the remaining data for the requested report.
400	Bad Request. This happens when the request has an invalid parameter, or it is missing a required parameter. Or it could also be the case when the requested resource path is not in the expected syntax.
401	Authentication Failed. This status code is generated when the Prime Performance Manager server is running with user access (authentication) enabled, and the REST API server was unable to authenticate the user.
403	Forbidden or Access Denied. This status code is generated when the requested report is currently disabled in the Prime Performance Manager system. A report can be disabled when its “poller” is disabled in Prime Performance Manager.
404	Resource Not Found. This status code is generated when the requested resource does not exist in the Prime Performance Manager REST API server. For example, if the request is for a report that is not found on the Prime Performance Manager server.
405	Method Not Allowed. This status code is generated when the requested method (HEAD, POST, PUT or DELETE) is not supported for that resource.
406	Not Acceptable. This status code is generated when the “Accept” header sent in the request is not supported by the REST API.
415	Unsupported Media Type. This status code is generated when the requested content-type is not supported for that resource. For example, let’s say a given report is only be available in CSV, and the client request that report using XML in the content-type. In this case, the 415 error is generated.
500	Internal Error. This error code is sent back for unexpected error on the server. If this error occurs, the user should retry the same request. If the error continues the user should verify the server logs.

5.12 Error Responses

In addition to HTTP header status code, the error response contains a content component. The following table shows error response samples. All the other error responses follow a similar format.

Table 46 - CSV Request / Error Response Sample – Invalid HTTP Verb

Request / Error Response (CSV) - Invalid HTTP Verb: POST
Request
POST /ppm/rest/reports/ HTTP/1.1 Content-Length: 0 Content-Type: text/plain; charset=UTF-8 Host: ppm-clean.cisco.com:4440 User-Agent: RestClient-Tool
Response
HTTP/1.1 405 Method Not Allowed Transfer-Encoding: chunked Date: Wed, 28 Mar 2012 16:12:11 GMT

```

Allow: GET
Content-Type: text/csv;charset=UTF-8
Server: Apache-Coyote/1.1
Cache-Control: no-cache

error_code,error_message
UnsupportedHttpVerb,HTTP method [POST] is not supported.

```

Table 47 - JSON Request / Error Response Sample - Invalid HTTP Verb

Request / Error Response (Json) – Invalid HTTP Verb: POST
Request – Notice, the Accept header is set for “application/json”.
<pre> POST /ppm/rest/reports/ HTTP/1.1 Accept: application/json Content-Length: 0 Content-Type: text/plain; charset=UTF-8 Host: ppm-clean.cisco.com:4440 User-Agent: RestClient-Tool </pre>
Response
<pre> HTTP/1.1 405 Method Not Allowed Transfer-Encoding: chunked Date: Sun, 01 Apr 2012 07:50:28 GMT Allow: GET Content-Type: application/json;charset=UTF-8 Server: Apache-Coyote/1.1 Cache-Control: no-cache { "error": { "code": "UnsupportedHttpVerb", "message": "HTTP method [POST] is not supported." } } </pre>

Table 48 - XML Request / Error Response Sample - Invalid HTTP Verb

Request / Error Response (XML) – Invalid HTTP Verb: POST
Request – Notice, the Accept header is set for “application/xml”.
<pre> POST /ppm/rest/reports/ HTTP/1.1 Accept: application/xml Content-Length: 0 Content-Type: text/plain; charset=UTF-8 Host: ppm-clean.cisco.com:4440 Connection: Keep-Alive User-Agent: RestClient-Tool </pre>
Response
<pre> HTTP/1.1 405 Method Not Allowed Transfer-Encoding: chunked Date: Wed, 28 Mar 2012 18:47:05 GMT Allow: GET Content-Type: application/xml;charset=UTF-8 Server: Apache-Coyote/1.1 Cache-Control: no-cache <?xml version="1.0" encoding="UTF-8" standalone="yes" ?> <error xmlns:ppm="http://www.cisco.com/ppm/rest/reports"> <code>UnsupportedHttpVerb</code> </pre>


```
<message>HTTP method [POST] is not supported.</message>
</error>
```

6 REST APIs

The APIs are composed of a GET URI + resource + parameter (or parameters) combination.

6.1 GET Report Listing

This API is used to retrieve a listing of sub-categories and report names. An empty {category} implies a list of all top level categories. Reports are defined in a hierarchical grouping of categories and subcategories.

Ex: GET /{reportType} /{category}..

The following table shows a sample request for the top level web report categories. Notice that if the Prime Performance Manager server has SSL enabled, you should also include the Authorization header.

Table 49 - - HTTP List of Reports Request Sample – Top Level Category

HTTP Request Sample – List of Reports

```
GET /ppm/rest/reports/ HTTP/1.1
Host: ppm-clean.cisco.com:4440
Connection: Keep-Alive
User-Agent: RestClient-Tool
```

6.2 Report Listing Samples

The following samples were collected from a Cisco development server to illustrate the data structure of the different data format types: XML, JSON, and CSV. The data provided by your server will likely be different.

6.2.1 CSV

Table 50 - CSV Request / Response Sample – List of Reports

Request / Response (CSV) – List of Reports

Request

```
GET /ppm/rest/reports/ HTTP/1.1
Host: ppm-cls-vm05.cisco.com:4440
Connection: Keep-Alive
User-Agent: RestClient-Tool
```

Response

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Date: Wed, 09 May 2012 15:34:16 GMT
Content-Type: text/csv;charset=UTF-8
Server: Apache-Coyote/1.1
Cache-Control: 3600
```

```
name,uri
Application Traffic,http://ppm-cls-
vm05.cisco.com:4440/ppm/rest/reports/Application+Traffic/
Availability,http://ppm-cls-
vm05.cisco.com:4440/ppm/rest/reports/Availability/
```

```

IP Protocols,http://ppm-cls-
vm05.cisco.com:4440/ppm/rest/reports/IP+Protocols/
IP QoS,http://ppm-cls-vm05.cisco.com:4440/ppm/rest/reports/IP+QoS/
IP SLA,http://ppm-cls-vm05.cisco.com:4440/ppm/rest/reports/IP+SLA/
Mobile IOS Statistics,http://ppm-cls-
vm05.cisco.com:4440/ppm/rest/reports/Mobile+IOS+Statistics/
Mobile StarOS Statistics,http://ppm-cls-
vm05.cisco.com:4440/ppm/rest/reports/Mobile+StarOS+Statistics/
Resources,http://ppm-cls-vm05.cisco.com:4440/ppm/rest/reports/Resources/
Security,http://ppm-cls-vm05.cisco.com:4440/ppm/rest/reports/Security/
Transport Statistics,http://ppm-cls-
vm05.cisco.com:4440/ppm/rest/reports/Transport+Statistics/
Video Broadcast,http://ppm-cls-
vm05.cisco.com:4440/ppm/rest/reports/Video+Broadcast/

```

6.2.2 JSONv2

JSONv2 is a simpler and easier to traverse data structure reports in the JSON format. It can be obtained by passing the `outputType=jsonv2` query string parameter or the HTTP Accept header `“application/vnd.cisco.ppm+json;v=2”`.

6.2.3 JSON

Table 51 - JSON Request / Response Sample – List of Reports

Request / Response (JSON) – List of Reports
<p>Request – Notice, the Accept header is set for “<i>application/json</i>”.</p> <pre> GET /ppm/rest/reports/ HTTP/1.1 Accept: application/json Host: ppm-cls-vm05.cisco.com:4440 Connection: Keep-Alive User-Agent: RestClient-Tool </pre>
<p>Response</p> <pre> HTTP/1.1 200 OK Transfer-Encoding: chunked Date: Wed, 09 May 2012 15:31:05 GMT Content-Type: application/json;charset=UTF-8 Server: Apache-Coyote/1.1 Cache-Control: 3600 { "reports": { "reportInfo": [{ "name": "Application Traffic", "uri": "http://ppm-cls- vm05.cisco.com:4440/ppm/rest/reports/Application+Traffic/" }, { "name": "Availability", "uri": "http://ppm-cls- vm05.cisco.com:4440/ppm/rest/reports/Availability/" }, { "name": "IP Protocols", </pre>

```

    "uri": "http://ppm-cls-
vm05.cisco.com:4440/ppm/rest/reports/IP+Protocols/"
  },
  {
    "name": "IP QoS",
    "uri": "http://ppm-cls-vm05.cisco.com:4440/ppm/rest/reports/IP+QoS/"
  },
  {
    "name": "IP SLA",
    "uri": "http://ppm-cls-vm05.cisco.com:4440/ppm/rest/reports/IP+SLA/"
  },
  {
    "name": "Mobile IOS Statistics",
    "uri": "http://ppm-cls-
vm05.cisco.com:4440/ppm/rest/reports/Mobile+IOS+Statistics/"
  },
  {
    "name": "Mobile StarOS Statistics",
    "uri": "http://ppm-cls-
vm05.cisco.com:4440/ppm/rest/reports/Mobile+StarOS+Statistics/"
  },
  {
    "name": "Resources",
    "uri": "http://ppm-cls-
vm05.cisco.com:4440/ppm/rest/reports/Resources/"
  },
  {
    "name": "Security",
    "uri": "http://ppm-cls-
vm05.cisco.com:4440/ppm/rest/reports/Security/"
  },
  {
    "name": "Transport Statistics",
    "uri": "http://ppm-cls-
vm05.cisco.com:4440/ppm/rest/reports/Transport+Statistics/"
  },
  {
    "name": "Video Broadcast",
    "uri": "http://ppm-cls-
vm05.cisco.com:4440/ppm/rest/reports/Video+Broadcast/"
  }
]
}
}

```

6.2.4 XML

Table 52 - XML Request / Response Sample – List of Reports

Request / Response (XML) – List of Reports

Request – Notice, the Accept header is set for “application/xml”.

```

GET /ppm/rest/reports/ HTTP/1.1
Accept: application/xml
Host: ppm-cls-vm05.cisco.com:4440
Connection: Keep-Alive
User-Agent: RestClient-Tool

```

Response

```

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Date: Wed, 09 May 2012 15:41:10 GMT
Content-Type: application/xml;charset=UTF-8
Server: Apache-Coyote/1.1
Cache-Control: 3600

<?xml version="1.0" ?>
<reports>
  <reportInfo>
    <name>Application Traffic</name>
    <uri>http://ppm-cls-
vm05.cisco.com:4440/ppm/rest/reports/Application+Traffic/</uri>
  </reportInfo>
  <reportInfo>
    <name>Availability</name>
    <uri>http://ppm-cls-
vm05.cisco.com:4440/ppm/rest/reports/Availability/</uri>
  </reportInfo>
  <reportInfo>
    <name>IP Protocols</name>
    <uri>http://ppm-cls-
vm05.cisco.com:4440/ppm/rest/reports/IP+Protocols/</uri>
  </reportInfo>
  <reportInfo>
    <name>IP QoS</name>
    <uri>http://ppm-cls-vm05.cisco.com:4440/ppm/rest/reports/IP+QoS/</uri>
  </reportInfo>
  <reportInfo>
    <name>IP SLA</name>
    <uri>http://ppm-cls-vm05.cisco.com:4440/ppm/rest/reports/IP+SLA/</uri>
  </reportInfo>
  <reportInfo>
    <name>Mobile IOS Statistics</name>
    <uri>http://ppm-cls-
vm05.cisco.com:4440/ppm/rest/reports/Mobile+IOS+Statistics/</uri>
  </reportInfo>
  <reportInfo>
    <name>Mobile StarOS Statistics</name>
    <uri>http://ppm-cls-
vm05.cisco.com:4440/ppm/rest/reports/Mobile+StarOS+Statistics/</uri>
  </reportInfo>
  <reportInfo>
    <name>Resources</name>
    <uri>http://ppm-cls-vm05.cisco.com:4440/ppm/rest/reports/Resources/</uri>
  </reportInfo>
  <reportInfo>
    <name>Security</name>
    <uri>http://ppm-cls-vm05.cisco.com:4440/ppm/rest/reports/Security/</uri>
  </reportInfo>
  <reportInfo>
    <name>Transport Statistics</name>
    <uri>http://ppm-cls-
vm05.cisco.com:4440/ppm/rest/reports/Transport+Statistics/</uri>
  </reportInfo>

```

```

<reportInfo>
  <name>Video Broadcast</name>
  <uri>http://ppm-cls-
vm05.cisco.com:4440/ppm/rest/reports/Video+Broadcast/</uri>
</reportInfo>
</reports>

```

6.3 GET Report

The GET API retrieves a specific network or node/group report from the Prime Performance Manager server. The query string parameters (sgmid, startDate, endDate and FQDN) are used to constrain the reporting data returned.

Examples:

```
GET /reports/{category}../{reportName}
```

```
GET /reports/{category}../{reportName}?FQDN=Node=[some node]
```

```
GET /groupreports/{category}../{reportName}?FQDN=Group=[some group]
```

Following is a sample request for the Interface Availability network level report. If the Prime Performance Manager server has SSL enabled, the HTTP Authorization header should be included.

Table 53 – HTTP Report Request Sample – “Interface Availability” Network Level Report

HTTP Request Sample – “Interface Availability” Network Level Report

```

GET /ppm/rest/reports/Availability/Interfaces/Interface++Availability
HTTP/1.1
Host: ppm-clean.cisco.com:4440
Connection: Keep-Alive
User-Agent: RestClient-Tool

```

6.4 Report Samples

The following samples were collected from a Cisco development server to illustrate the data structure of the different data format types: XML, JSON, and CSV. The data provided by your server will likely be different. The sample reports might only show a subset of the report. The report uses “. . .” to indicate that further data is available but not displayed.

6.4.1 CSV

Table 54 - CSV Request / Response Sample – “Interface Availability” Network Level Report

Request / Response (CSV) - “Interface Availability” Network Level Report

Request

```

GET /ppm/rest/reports/Availability/Interfaces/Interface++Availability
HTTP/1.1
Host: ppm-clean.cisco.com:4440
Connection: Keep-Alive
User-Agent: RestClient-Tool

```

Response – *Notice only a portion of the report is being displayed*

```

HTTP/1.1 206 Partial Content
Transfer-Encoding: chunked
Date: Fri, 05 Oct 2012 21:22:11 GMT
Content-Type: text/csv;charset=UTF-8
Server: Apache-Coyote/1.1

```

```
Content-Range: pages 1/8
Cache-Control: no-cache
```

```
Device,Interface,Timestamp,Interface Index,Interface Availability Percentage
Down,Interface Availability Percentage Up,Interface Availability Percentage
Timeout
```

```
ems3825d,GigabitEthernet0/0,2012-10-05T05:00-0400,1,0.0,100.0,0.0
ems3825j,GigabitEthernet0/0,2012-10-05T05:00-0400,1,0.0,100.0,0.0
ems7606d,GigabitEthernet5/2,2012-10-05T05:00-0400,22,0.0,100.0,0.0
ems7606d,EoBC0/0,2012-10-05T05:00-0400,41,0.0,100.0,0.0
ems7606d,EoBC0/2,2012-10-05T05:00-0400,42,0.0,100.0,0.0
ems7606d,SPAN RP Interface,2012-10-05T05:00-0400,44,0.0,100.0,0.0
ems3825e,GigabitEthernet0/0,2012-10-05T05:00-0400,1,0.0,100.0,0.0
ppm5580a,Adaptive Security Appliance management interface,2012-10-05T05:00-
0400,2,0.0,100.0,0.0
ppm5580a,Adaptive Security Appliance Virtual254 interface,2012-10-05T05:00-
0400,8,0.0,100.0,0.0
. . . .
. . . .
SimLab-30.4.1.90,My_ifAlias_string-GigabitEthernet01,2012-10-05T12:00-
0400,3,0.0,100.0,0.0
```

6.4.2 JSONv2

The JSON report format version 2 is a simpler and easier to traverse JSON data structure. It can be obtained by passing the “outputType=jsonv2” query string parameter or passing the HTTP Accept header “application/vnd.cisco.ppm+json;v=2”.

Table 55 – JSONv2 Request / Response Sample – “Interface Availability” Network Level Report

Request / Response (JSON) - “Interface Availability” Network Level Report
<p>Request – Notice, the outputType query string parameter is set “json”.</p> <pre>GET /ppm/rest/reports/Application+Traffic/AAA/AAA+Accounting+Dashboard?outputType =jsonv2 HTTP/1.1 Host: ppm-clean.cisco.com:4440 Connection: Keep-Alive User-Agent: RestClient-Tool</pre>
<p>Response</p> <pre>HTTP/1.1 206 Partial Content Transfer-Encoding: chunked Date: Fri, 05 Oct 2012 21:18:30 GMT Content-Type: application/json;charset=UTF-8 Server: Apache-Coyote/1.1 Content-Range: pages 1/1 Cache-Control: no-cache</pre> <pre>{ "report": { "name": "AAA Accounting Dashboard", "header": ["Device", "Timestamp", "Server", "Protocol", "Requests Requests",</pre>

```

    "Requests Timeouts",
    "Requests Percent",
    "Requests Response Time Millisecs",
    "Responses Server Error",
    "Responses Incorrect",
    "Successful Transactions Success",
    "Successful Transactions Success\/Sec",
    "Failed Transactions Failed",
    "Failed Transactions Percent"
  ],
  "data": [
    [
      "ems2941a",
      "2013-02-17T16:00-0500",
      "10.74.125.10",
      "radius",
      "0",
      "0",
      "0.0",
      "0.0",
      "0",
      "0",
      "0",
      "0.0",
      "0",
      "0.0"
    ],
    [
      "ems2941a",
      "2013-02-17T17:00-0500",
      "10.74.125.10",
      "radius",
      "0",
      "0",
      "0.0",
      "0.0",
      "0",
      "0",
      "0",
      "0.0",
      "0",
      "0.0"
    ],
    ...
    ...
    [
      "ems2941a",
      "2013-02-18T15:00-0500",
      "10.74.125.10",
      "radius",
      "0",
      "0",
      "0.0",
      "0.0",
      "0",
      "0",
      "0",
      "0.0",
      "0",
      "0"
    ]
  ]

```

```

        "0.0",
        "0",
        "0.0"
    ]
}
}
}

```

6.4.3 JSON

Table 56 - JSON Request / Response Sample – “Interface Availability” Network Level Report

Request / Response (JSON) - “Interface Availability” Network Level Report
<p>Request – Notice, the <i>outputType</i> query string parameter is set “json”.</p> <pre> GET /ppm/rest/reports/Availability/Interfaces/Interface++Availability?outputType= json HTTP/1.1 Host: ppm-clean.cisco.com:4440 Connection: Keep-Alive User-Agent: RestClient-Tool </pre>
<p>Response</p> <pre> HTTP/1.1 206 Partial Content Transfer-Encoding: chunked Date: Fri, 05 Oct 2012 21:18:30 GMT Content-Type: application/json;charset=UTF-8 Server: Apache-Coyote/1.1 Content-Range: pages 1/8 Cache-Control: no-cache </pre> <pre> { "report": { "reportHeaders": { "reportHeader": ["Device", "Interface", "Timestamp", "Interface Index", "Interface Availability Percentage Down", "Interface Availability Percentage Up", "Interface Availability Percentage Timeout"] }, "reportData": { "reportDataItems": [{ "reportDataItem": ["ems3825d", "GigabitEthernet0/0", "2012-10-05T05:00-0400", "1", "0.0", "100.0", "0.0"] }] } } } </pre>


```

    <reportHeader>Interface Availability Percentage Down</reportHeader>
    <reportHeader>Interface Availability Percentage Up</reportHeader>
    <reportHeader>Interface Availability Percentage
Timeout</reportHeader>
  </reportHeaders>
  <reportData>
    <reportDataItems>
      <reportDataItem>ems3825d</reportDataItem>
      <reportDataItem>GigabitEthernet0/0</reportDataItem>
      <reportDataItem>2012-10-05T05:00-0400</reportDataItem>
      <reportDataItem>1</reportDataItem>
      <reportDataItem>0.0</reportDataItem>
      <reportDataItem>100.0</reportDataItem>
      <reportDataItem>0.0</reportDataItem>
    </reportDataItems>
    . . .
    . . .
    <reportDataItems>
      <reportDataItem>SimLab-30.4.1.90</reportDataItem>
      <reportDataItem>My_ifAlias_string-
GigabitEthernet0'1</reportDataItem>
      <reportDataItem>2012-10-05T12:00-0400</reportDataItem>
      <reportDataItem>3</reportDataItem>
      <reportDataItem>0.0</reportDataItem>
      <reportDataItem>100.0</reportDataItem>
      <reportDataItem>0.0</reportDataItem>
    </reportDataItems>
  </reportData>
</report>

```

7 Pagination

The reports retrieved from the REST API has an upper limit set to 1000 records per page unless modified by the “maxPageSize” parameter. If a report has more than 1000 records (or the “maxPageSize”), the report is divided into pages. The first REST response contains the first 1000 records along with a “Content-Range” header (refer to HTTP Response Headers section of this document).

When the report is broken into multiple pages, use the “pageIndex” parameter to navigate to the next pages. The “Content-Range” HTTP header value provides the current page number and number of pages the report contains.

The following table shows a report sample that displays the “pageIndex” request parameter set to page number “3”. The “Content-Range” header response displays page number 3 of 8 pages total.

Table 58 - CSV Request / Response Sample Showing Pagination (Content-Range Header)

Request
GET /ppm/rest/reports/Availability/Interfaces/Interface++Availability?pageIndex=3 HTTP/1.1 Host: ppm-clean.cisco.com:4440
Response – Notice the “Content-Range” header indicating pagination.
HTTP/1.1 206 Partial Content Transfer-Encoding: chunked Date: Fri, 05 Oct 2012 21:08:04 GMT

```
Content-Type: text/csv;charset=UTF-8
Server: Apache-Coyote/1.1
Content-Range: pages 3/8
Cache-Control: no-cache
```

```
Device,Interface,Timestamp,Interface Index,Interface Availability Percentage
Down,Interface Availability Percentage Up,Interface Availability Percentage
Timeout
SimLab-30.4.1.25,My_ifAlias_string-My_ifDescr_string,2012-10-05T16:00-
0400,2,0.0,100.0,0.0
SimLab-30.4.1.54,My_ifAlias_string-My_ifDescr_string,2012-10-05T16:00-
0400,2,0.0,100.0,0.0
SimLab-30.4.1.81,My_ifAlias_string-My_ifDescr_string,2012-10-05T16:00-
0400,2,0.0,100.0,0.0
SimLab-30.4.1.120,My_ifAlias_string-My_ifDescr_string,2012-10-05T16:00-
0400,2,0.0,100.0,0.0
SimLab-30.4.1.49,My_ifAlias_string-My_ifDescr_string,2012-10-05T16:00-
0400,2,0.0,100.0,0.0
SimLab-30.4.1.96,My_ifDescr_string,2012-10-05T16:00-0400,3,0.0,100.0,0.0
SimLab-30.4.1.223,My_ifAlias_string-My_ifDescr_string,2012-10-05T16:00-
0400,2,0.0,100.0,0.0
SimLab-30.4.1.223,My_ifDescr_string,2012-10-05T16:00-0400,3,0.0,100.0,0.0
SimLab-30.4.1.91,My_ifAlias_string-My_ifDescr_string,2012-10-05T16:00-
0400,3,0.0,100.0,0.0
. . . .
```

8 References

Fielding, R. T. “Architectural Styles and the Design of Network-based Software Architectures”. Ph.D. dissertation, University of California, Irvine, 2000.

Richardson, L. & Ruby, S. (2007). *RESTful Web Services*. Sebastopol, CA : O'Reilly Media.

Berners-Lee, T., Fielding, R. T., et al. *Uniform Resource Identifier (URI): Generic Syntax, RFC 3986*, RFC Editor, 2005 (<http://www.rfc-editor.org/rfc/rfc3986.txt>).

Request for Comments (RFC) <http://www.ietf.org/rfc.html>

9 Glossary

API: Application Programming Interface

CSV: Comma Separated Values

HTTP: Hypertext Transfer Protocol

HTTPS: Hypertext Transfer Protocol over Transport Layer Security (TLS)

JPEG: Joint Photographic Experts Group

JSON: JavaScript Object Notation

PDF: Portable Document Format

PNG: Portable Network Graphics

PPM: Prime Performance Manager

REST: Representational State Transfer

RFC: Request for Comment

SSL: Secure Socket Layer

XML: Extensible Markup Language