AEMO
AUSTRALIAN ENERGY MARKET OPERATOR

Guide to Reallocation APIs

1.02 Final

March 2023

Provides details for Electricity Reallocation APIs

Important Notice

PURPOSE

This Guide to Reallocation APIs (Guide), prepared by AEMO, provides guidance for Reallocations under the National NER (Rules).

NO RELIANCE OR WARRANTY

This document does not constitute legal or business advice, and should not be relied on as a substitute for obtaining detailed advice about the National Gas or Electricity Law, the Rules or any other applicable laws, procedures or policies. While AEMO has made every effort to ensure the quality of the information in this Guide, neither AEMO, nor any of its employees, agents and consultants make any representation or warranty as to the accuracy, reliability, completeness, currency or suitability for particular purposes of that information.

LIMITATION OF LIABILITY

To the maximum extent permitted by law, AEMO and its advisers, consultants and other contributors to this Guide (or their respective associated companies, businesses, partners, directors, officers or employees) are not liable (whether by reason of negligence or otherwise) for any errors, omissions, defects or misrepresentations in this document, or for any loss or damage suffered by persons who use or rely on the information in it.

TRADEMARK NOTICES

DISTRIBUTION

Available to the public.

DOCUMENT IDENTIFICATION

Business custodian: Team Lead, Prudentials
IT custodian: Team Lead, Settlements and Prudential Systems
Prepared by: Technology, Technical Writers and 5MS Settlements Team

VERSION HISTORY

Version 1.02. Updates the API web portal links .
Last update: Monday, 27 March 2023 12:50 PM

DOCUMENTS MADE OBSOLETE

The release of this document changes any previous versions of Guide to Reallocation APIs.

FEEDBACK

Your feedback is important and helps us improve our services and products. To suggest improvements, please contact AEMO's Support Hub.

Contents

Introduction

Chapter 4 Energy & FCAS Bids

Purpose

Provides assistance to developers and users of the Bidding API.

Audience

This guide is for Reallocations Registered Participants who have an executed Letter of Agreement with AEMO to use Electricity Reallocations APIs or the web portal.

Participant administrators wanting to know the user rights management (URM) entity to grant for participant users to access Electricity Reallocations are a secondary audience.

Assumed Knowledge

This guide assumes you have knowledge of:

- JSON basics
- REST API standards
- Web-based technologies

- The operating system you are using
- Connecting to AEMO’s Electricity IT Systems

Related rules and procedures

The following rules and procedures relate to Reallocations:

Name	Purpose
MNSP Convexity Rule	AEMC website > NER Clause 3.8.6A (e)
Introduction to Market Rules	AEMC website > NER Chapter 3
Market Floor Price	AEMC website > NER 3.9.6
Market Price Cap	AEMC website > NER 3.9.4
Rebidding and Technical Parameters Guideline	**https://www.aer.gov.au/wholesale-markets/market-guidelines-reviews/rebidding-and-technical-parameters-guideline-amendments-for-5-minute-settlement-2019**

How to use this guide

- This guide is written in plain language for easy reading.
- Where there is a discrepancy between the NER, and information or a term in this document, the NER takes precedence.
- Where there is a discrepancy between the Procedures, and information or a term in this document, the Procedures take precedence.
- **Text in this format** indicates there is a resource on AEMO’s website, for details, see References.
- Text in this format indicates a link to a related resource.

- Actions to complete in the web portal interface are **bold and dark grey**.
- Rules terms used throughout this guide are capitalised and listed in Rules Terms on page 1. You can find definitions in the NER.
- Glossary terms are capitalised and have the meanings listed against them in the Glossary on page 83.
- References to time are Australian Eastern Standard Time (AEST).

What’s in this guide

Chapter 1 Need to know on page 1 explains what you need to know before you start using Reallocations.

Chapter 2 About Electricity Reallocation APIs on page 7 explains the Reallocations web application, who it is for, and how to access it, and how to use the common interface features such as selecting a unit ID, date and so on.

Chapter 3 Reallocation APIs on page 13 explains the new Reallocations APIs, validation rules, error messages, request and response structure along with examples for the APIs.

Appendix Reallocation JSON Schema on page 69 provides the JSON schema for the Reallocation APIs.

Needing Help on page 1 provides information to assist participants with IT related issues, requesting assistance from AEMO, and using the Set Participant option.

References on page 78 contains a list of resources mentioned throughout this guide.

Glossary on page 83 explains the terms and abbreviations used throughout this guide.

Chapter 1 Need to know

Updates to Reallocations for Five-minute Settlements

For five-minute settlements Reallocations, the following section lists the changes listed in the document:

- Participants can enter five-minute reallocations from the Reallocations go-live date for days after the commencement of the 5MS rule change date.
- Participants can continue to enter 30-minute reallocations fromthe commencement of the 5MS rule change date for days before the 5MS rule change date until the ex-post timetable allows it.
- For reallocations spanning the commencement of the 5MS rule change date, participants need to submit two separate reallocations requests:
    - One that starts on the start date and ends on 30 September 2021 with the interval length of 30-minutes.
    - And a second one starting on the 1 October 2021 and ends on the end date with the interval length of five-minutes.
- New Reallocation APIs that allow you to create, authorise, or cancel reallocations.
- Updated web portal and new screens for creating, viewing, and managing reallocations requests. For more information, see Guide to Electricity Reallocations.

Who can use Reallocations?

Participants having access to Reallocations include Reallocations Registered Participants (usually Market Participants and reallocators).

Only participants who have an executed Letter of Agreement with AEMO can access Reallocations:

- Participants registering as reallocators complete the Letter of Agreement during registration.
- Market Participants wanting to use Reallocations can complete the Letter of Agreement found at the back of the relevant Procedure:
    - Reallocation Procedure Energy and Dollar Offset Reallocations.
    - Reallocation Procedure Swap and Option Offset Reallocations.
- **Cap**, **floor**, and **swap agreement types** are only available to Market Customer, Market Generator, or MNSP participant registered for these categories.

The Letter of Agreement is only required once and both parties to a Reallocation must have a signed Letter of Agreement. Once your Letter of Agreement is executed, AEMO enables Reallocations so you can submit Reallocation requests.

Chapter 2 About Electricity Reallocation APIs

A Reallocation Request Service (Reallocations) is a financial arrangement supported by the Rules in which two participants registered as reallocators request AEMO to make matching debits and credits to their settlement position. Reallocations is responsible for calculating the debit and credit amounts based on the contracts.

The Reallocation APIs allow you to create, authorise, cancel reallocations. You can also search and retrieve reallocations, calendars, participants, profile types, agreement types, and market price cap. For more information on creating and using reallocations using the web portal, see Guide to Electricity Reallocations.

For information on access and using APIs, see **Guide to AEMO's e-Hub APIs**.

Credentials, paths, schemas, validation, and examples

- For individual schemas and API paths, see Reallocations schemas.
- For details about AEMO's API standards, SSL certificates, response codes, security, and authentication details, see **Guide to AEMO's APIs**.

Common header parameters

Parameter	Required	Format
Content-type	Yes	Text/xml
Accept	Yes	Text/xml
Authorization	Yes	Two-way SSL and basic auth - participant user URM username and password
X-initiatingParticipantID	Yes	Participant ID
X-market	Yes	NEM

Example error response

```
{
 "transactionId": "e7d0b3db-de42-4f38-9e26-b60bd6114367",
 "data": {
  "reallocationId": null,
  "reallocationSuccessful": false
 },
 "errors": [
  {
   "code": "ERROR_SUBMIT_REALLOCATION",
   "title": "INVALID_INDICATOR",
   "detail": "Request does not have a valid credit/debit indicator.",
   "source": null
  }
 ]
}
```

The title and detail changes based on which validation fails.

Common response

Error Code	Description
200	OK
422	Invalid Indicator
500	No payload

User rights access

The user rights access used in the Prudential APIs, and Prudential web screens use a new URM entity:

- **EMMS - Settlements - Reallocations**

By default, this enables the read-only access. The PA needs to tick the following rights to allow you access to Reallocations in the MSATS web portal:

Reallocations function/API	Access Rights
Submit Reallocation	Create
Authorise or Cancel Reallocations	Update
All other functions	Read

API account passwords are reset every 90 days.

The steps to set up URM rights for API access are:

1. If required, the Participant Administrator (PA) creates a new Participant User in MSATS.
2. The PA assigns the URM entity to the Participant User.

For help with user rights access, see Guide to User Rights Management.

API format

API URLs are in the following format:

```
https://<host>/ws/<business_
function>/<APIversion>/<resource>?querystring parameters
```

For example:

```
https://apis.prod.aemo.com.au:9319/ws/NEMWholesale/reallo
cations/v1/submitReallocation
```

Table 1 API Definition

Parameter	Description
<protocol>	HTTPS
<host>	Names the server hosting the service or an external proxy Internet web service host: apis.prod.aemo.com.au:9319 MarketNet web service host: apis.prod.marketnet.net.au:9319
<Business_function>	API Name - The AEMO system providing the services e.g. GeneratorRecall
<verb><resource>	Entities of a Business Function e.g. /submitReallocation
?querystring parameters	Query string parameters for GET method

API e-Hub addresses

API details, including Swagger files, are available via AEMO's developer portal. For more information about APIs, see **API Reference.**

The following AEMO API Gateways are available:

Table 2 API Gateway Addresses

Environment	Internet Address	MarketNet Address
5MS Staging	**https://apis.5msstaging.aemo.com.au:9319**	**https://apis.5msstaging.marketnet.net.au:9319**
5MS Staging API portal	**https://dev.preprod.aemo.com.au** You need to register and sign in to access the 5MS APIs in this portal.	**Unavailable until further notice**
Pre-production	**https://apis.preprod.aemo.com.au:9319**	**https://apis.preprod.marketnet.net.au:9319**
Pre-production Developer Portal	**https://developer-portal-ppd.aemo.com.au/** You need to register and sign in to access the 5MS APIs in this portal.	**Unavailable until further notice**
Production	**https://apis.prod.aemo.com.au:9319**	**https://apis.prod.marketnet.net.au:9319**
Production Developer Portal	**https://dev.aemo.com.au/**	**Unavailable until further notice**

Submission and throttling

Payload size	Object
Submission frequency	250 messages per minute. For any messages sent beyond the limit, there is a denial of service until that one minute is complete.
Throttling limit	Allowed
Throttling limit	1 per submission

Chapter 3 Reallocation APIs

The following chapter lists the new Reallocations APIs, the validation rules, and the request and response structure for the Reallocations APIs.

submitReallocation API

The submitReallocation API creates a reallocation and sets the status to SUBMIT. The API creates an ID in the format “<YYYYMMDD>.RS<NNNN>“ where:

- <YYYYMMDD> is the current system date
- <NNNN> is the consistently increasing sequence number for reallocations created on the specific date including leading zeros.

For example, the first reallocation submitted on April 24, 2019, would be 20190424.RS0001, and so on.

Participants cannot submit a Reallocations request for 30-minute data for dates post July 2021 after the 5MS Production date.

API request/response structure

API	submitReallocation
Method	POST
URL	/NEMWholesale/reallocations/v1/submitReallocation
Header	Standard request header attributes to include: • Authorisation: Basic …. • Content-Encoding: Should be one of [gzip, deflate, compress] • Accept-Encoding: Should be one or more of [gzip, deflate, compress]
Authorisation	Participant specific

<table>
<tr><th>API</th><th>submitReallocation</th></tr>
<tr><td>Request Content</td><td>

Content Body:

```
{
  "reallocation":
  {
    "startDate": [datetime],
    "endDate": [datetime],
    "submittingParticipantId": [string],
    "counterpartyParticipantId": [string],
    "agreementTypeId": [string],
    "profileTypeId": [string],
    "regionId": [string],
    "creditDebitIndicator": [string],
    "intervalLength": [integer],
    "submittingParticipantReference": [string],
    "calendarId":[sting],
    "reallocationProfile":
    [
      {
        "periodId": [integer],
        "reallocationValue": [decimal],
        "nrp": [decimal, optional]
      }
    ]
  }
}
```

</td></tr>
<tr><td>Success response</td><td>

Content Body:

</td></tr>
</table>

API	submitReallocation

```
{
 "transactionId": [string]
 "data":
 {
  "reallocationId": [string]
 }
}
```

In the Request content, the submitting and counterparty participant IDs are mapped to the credit and debit participant IDs as follows:

creditDebitIndicator	creditParticipantId	debitParticipantId
C	submittingParticipantId	counterpartyParticipantId
D	counterpartyParticipantId	submittingParticipantId

Parameters

Name	Required	Datatype	Description	Example
startDate	Y	Datetime	First day of the reallocation agreement	2000-12-31T00:00:00
endDate	Y	Datetime	Last day of the reallocation agreement	2020-12-31T00:00:00

Name	Required	Datatype	Description	Example
submittingParticipantId	Y	String	Unique identifier of the participant submitting the reallocation request	AEMO
counterpartyParticipantId	Y	String	Unique identifier of the counterparty participant in the reallocation request	AEMO
agreementType	Y	String	$ (price), MWh (Quantity), SWAP, CAP, or FLOOR.	DOLLAR
profileTypeId	Y	String	The day type profile for which the reallocation applies over the start and end date range. Valid entries are BUSINESS, NON-BUSINESS, or FLAT.	BUSINESS

Name	Required	Datatype	Description	Example
regionId	Y	String	Region identifier, being the spot price reference node for this reallocation.	QLD1
creditDebitIndicator	Y	String	Denotes that the submitting party is the Credit or Debit party with a value of "C" or "D" respectively.	C
intervalLength	Y	Integer	Either 30 or 5	30
submittingParticipantReference	N	String	Reference detail for the initiating party	REALLOC1
calendarId	Y	String	Unique identifier for the Settlements Calendar	SETT_ REGIONAL
periodId	Y	Integer	Period identifier (Reallocation occurring before 5MS implementation - 1....48)	1,2,3 48

Name	Required	Datatype	Description	Example
			(Reallocation occurring after 5MS implementation -1....288)	
reallocationValue	Y	Decimal	Reallocation value in $ if the agreement type is $ or MWh of the agreement type	30
nrp	N	Decimal	Nominated Reallocations Price, only used in agreement types of SWAP, CAP and FLOOR, being the contract strike price in $/MWh	500

Validation checks and error messages

Validation check	Error message
Company IDs associated with submitting participant and the counterparty must be different	The system cannot create your reallocation because your credit and debit participant IDs belong to the same company ID.

Validation check	Error message
A prudential run must be available with a prudential date equal to the current day	The reallocation cannot be processed. AEMO's initial prudential assessment for the current day is not complete at this time.
The submitting participant must pass the prudential check.	AEMO cannot process this Reallocation at this time. Please check your trading limit.
The submitting participant must be registered for reallocations.	The system cannot create your reallocation because the participant <PID> is not registered for reallocations.
The agreement type must be a valid Reallocation Agreement Type.	The agreement type is not valid.
The profile type must be a valid Reallocation Profile Type.	The profile type is not valid.
The credit/debit indicator must be a valid Reallocation Credit or Debit indicator.	The credit/debit indicator is not valid.
The region identifier must be a valid NEM region identifier for all days between the start and end dates (inclusive) of the reallocation.	The region ID is not valid.
The number of trading intervals in the reallocation profile must be equal to (1440 / intervalLength), where intervalLength is either 5-minutes or 30-minutes depending on the date of the Reallocation request.	Please check the number of trading intervals in your reallocation.

Validation check	Error message
The period identifiers in the reallocation profile must start at 1 and consistently increase.	The period identifiers must start at 1 and increase by 1 for each period.
The trading interval in the reallocation must be equal to the settlement interval length for all days between the start and end dates (inclusive) of the reallocation.	The reallocation interval length must be equal to the trading interval length for for all days between the start and end dates (inclusive).
The start date of the reallocation must be less than or equal to the end date of the reallocation.	The start date must be less than or equal to the end date.
The start date of the reallocation must be greater than or equal to the start date cut-off.	The start date must be greater than or equal to the ex post cut-off date.
The Calendar ID must be a valid calendar.	The calendar ID is not valid

Example request

```
POST /NEMWholesale/reallocations/v1/submitReallocation
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip
Content-Length: nnn
Content-Encoding: gzip
Authorisation: Basic dGVzdG1hbnVhbSpOZK1tY28wMw==
X-market: NEM
X-initiatingParticipantID: <PARTICIPANTID>
{
   "reallocation": {
      "startDate": "2000-12-31T00:00:00",
      "endDate": "2020-12-31T00:00:00",
      "submittingParticipantId": "<PID1>",
      "counterPartyParticipantId": "<PID2>",
      "agreementTypeId": "MWh",
      "profileTypeId": "FLAT",
      "regionId": "VIC1",
      "creditDebitIndicator": "C",
      "intervalLength": 5,
      "submittingParticipantReference": "<SUBMITTINGPARTY_
REFERENCE>",
      "calendarId":"SETT_REGIONAL",
      "reallocationProfile": [
         {
            "periodId": 1,
            "reallocationValue": 0,
            "nrp": 12
         },
         {
            "periodId": 2,
            "reallocationValue": 0,
            "nrp": 0
         },
         ……
         {
            "periodId": 287,
            "reallocationValue": 0,
            "nrp": 0
         },
         {
            "periodId": 288,
            "reallocationValue": 0,
            "nrp": 0
         }
```

```
    ]
  }
}
```

Example response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnn
Content-Encoding: gzip
{
  "transactionId": "76f4db27-0de3-4d5b-9e4f-81059190614c",
  "data": {
    "reallocationId": "20200210.RS002",
    "reallocationSuccessful": true
  },
  "errors": []
}
```

authoriseReallocation API

The authoriseReallocation API sets the state of the reallocation to AUTHORISE, signalling that both parties have agreed to the reallocation.

API request/response structure

API	authoriseReallocation
Method	PUT
URL	/NEMWholesale/reallocations/v1/authoriseReallocation
Authorisation	Participant specific
Request Content	Content Body:

API	authoriseReallocation

	`{`
	`  "reallocationId": [string],`
	`  "counterpartyReference": [string]`
	`}`

Success response	Content Body:
	`{`
	`  "transactionId": [string]`
	`  "data":`
	`  {`
	`  }`
	`}`

Parameters

Name	Required	Datatype	Description	Example
reallocationId	Y	String	Unique reallocation identifier	20190820.RS0001
counterPartyReference	N	String	Reference detail for the counterparty	ReallocRef

Validation checks and error messages

Validation check	Error message
The reallocationId parameter must be a valid reallocation ID.	The reallocation ID is not valid.
The participant ID of the Counterparty in the reallocation must match the ID of the request participantId.	The requesting participant is not a party to this reallocation.
The counterparty must be registered for reallocations.	The system cannot create your reallocation because the counterparty participant ID is not registered for reallocations.
The reallocation must be in the SUBMIT state.	You cannot authorise this reallocation as it is not in submit status.
A prudential run must be available with a prudential date equal to the current day.	The reallocation cannot be processed. AEMO's initial prudential assessment for the current day is not complete at this time.
The Credit and Debit participants must both pass a prudential check at the time of authorisation.	(Authorising party at authorisation state) AEMO cannot process this Reallocation at this time. Please check your trading limit.

Example request

```
PUT /NEMWholesale/reallocations/v1/authoriseReallocation
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip
Content-Length: nnn
Content-Encoding: gzip
Authorisation: Basic dGVzdG1hbnVhbSpOZK1tY28wMw==
X-market: NEM
X-initiatingParticipantID: <PARTICIPANTID>
{
   "reallocationId”: "20130108.RS003",
   "counterpartyReference": "<COUNTERPARTY_REFERENCE>"
}
```

Example response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnn
Content-Encoding: gzip
{
  "transactionId": "0598cd72-2a81-4cf5-9809-2a7c81d28610",
  "data": {   “
    "responseAppStatus": "Successful",
    "messageList": []
  },
  "errors": []
}
```

cancelReallocation API

The cancelReallocation API sets the state of a reallocation to CANCEL, signalling to the Prudential system to exclude the reallocation from processing further.

API request/response structure

API	cancelReallocation
Method	PUT
URL	/api/NEMReallocations/v1/cancelReallocation
Authorisation	Participant specific

Request Content	Content Body:
	{
	  "reallocationId": [string],
	  "counterpartyReference": [string]
	}

Success response	Content Body:
	{
	  "transactionId": [string]
	  "data":
	  {
	  }
	}

Parameters

Name	Required	Datatype	Description	Example
reallocationId	Y	String	Unique reallocation identifier	20190820.RS0001
counterPartyReference	N	String	Reference detail for the counterparty	ReallocRef

Validation checks and error messages

Validation check	Error message
The reallocationId parameter must be a valid reallocation ID.	The reallocation ID is not valid.
Either the submittingParticipantId or counterpartyParticipantId property of the reallocation must match the ID of the request participantId.	The credit or debit participant ID is not valid.
The reallocation must be in the SUBMIT state.	You cannot cancel this reallocation as it is not in submit status.

Example request

```
PUT /NEMWholesale/reallocations/v1/cancelReallocation
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip
Content-Length: nnn
Content-Encoding: gzip
Authorisation: Basic dGVzdG1hbnVhbSpOZK1tY28wMw==
X-market: NEM
X-initiatingParticipantID: <PARTICIPANTID>
{
   "ReallocationId": "20181214.RS019",
   "CounterpartyReference": "<COUNTERPARTY_REFERENCE>"
}
```

Example response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnn
Content-Encoding: gzip
{
  "transactionId": "0598cd72-2a81-4cf5-9809-2a7c81d28610",
  "data": {   "
    "responseAppStatus": "Successful",
    "messageList": []
  },
  "errors": []
}
```

getReallocation API

The getReallocation API retrieves a single reallocation along with its associated properties and reallocation profile data. The reallocationId request parameter is used to retrieve the reallocation matching the supplied ID.

API request/response structure

API	getReallocation
Method	GET
URL	/NEMWholesale/reallocations/v1/getReallocation
Authorisation	Participant specific
Request Content	reallocationId

API	getReallocation
Success response	Content Body:

API	getReallocation

```
{
 "transactionId": [string]
 "data":
 {
  "reallocation":
  {
   "reallocationId": [string],
   "startDate": [datetime],
   "endDate": [datetime],
   "submittingParticipantId": [string],
   "counterpartyParticipantId": [string],
   "agreementTypeId": [string],
   "profileTypeId": [string],
   "regionId": [string],
   "currentStepId": [string],
   "creditDebitIndicator": [string],
   "submittingParticipantReference": [string],
   "counterpartyReference": [string],
   "intervalLength": [integer],
   “calendarId”:[string]
   "lastChanged": [datetime],
   "reallocationProfile":
   [
    {
     "periodId": [integer],
     "reallocationValue": [decimal],
     "nrp": [decimal]
    }
   ]
  }
 }
}
```

Parameters

Name	Required	Datatype	Description	Example
reallocationId	Y	String	Unique reallocation identifier	20190820.RS0001

Validation checks and error messages

Validation check	Error message
Either the submittingParticipantId or counterpartyParticipantId property of the reallocation must match the ID of the request participantId.	The reallocation ID is not valid.

Example request

```
GET
/NEMWholesale/reallocations/v1/getReallocation?ReallocationId=2013
0108.RS004
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip
Authorisation: Basic dGVzdG1hbnVhbSpOZK1tY28wMw==
X-market: NEM
X-initiatingParticipantID: <PARTICIPANTID>
```

Example response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnn
Content-Encoding: gzip
{
 "transactionId": "9cc007d7-c8da-436a-9fe7-d6a618b1d8fa",
 "data": {
  "reallocation": {
   "reallocationId": "20130108.RS004",
   "startDate": "2013-01-20T00:00:00",
   "endDate": "2013-01-26T00:00:00",
   "submittingParticipantId": "<PID1>",
   "counterPartyParticipantId": "<PID2>",
   "agreementTypeId": "MWh",
   "profileTypeId": "NON_BUSINESS",
   "regionId": "VIC1",
   "currentStepId": "AUTHORISE",
   "creditDebitIndicator": "C",
   "submittingParticipantReference":
"<SUBMITTINGPARTYREFERENCE>",
   "counterPartyReference": null,
   "intervalLength": 5,
   "calendarId": null,
   "lastChanged": "2013-01-10T11:06:16",
   "exAnteDueDate": "2013-01-10T00:00:00",
   "exPostDueDate": "2013-02-19T00:00:00",
   "reallocationProfile": [
    {
     "periodId": 1,
     "reallocationValue": 34.08,
     "nrp": 0.0
    },
    {
     "periodId": 2,
     "reallocationValue": 31.74,
     "nrp": 0.0
    },
……
```

```
        {
          "periodId": 287,
          "reallocationValue": 43.26,
          "nrp": 0.0
        },
        {
          "periodId": 288,
          "reallocationValue": 40.42,
          "nrp": 0.0
        }
      ]
    }
  },
  "errors": []
}
```

getReallocations API

The getReallocations API retrieves a list of reallocations and associated properties that fulfil the following criteria:

- Either the submittingParticipantId or counterpartyParticipantId property of the reallocation must match the ID of the request participantId.
- If the regionId filter parameter is in the request, the regionId property of the reallocation must match the value.
- If the startDate filter parameter is in the request, the endDate property of the reallocation must be greater than or equal to the value.
- If the endDate filter parameter is in the request, the startDate property of the reallocation must be less than or equal to the value.
- If the agreementType filter parameter is in the request, the agreementType property of the reallocation must match one of the comma-separated values.
- If the steps filter parameter is in the request, the current_StepId property of the reallocation must match one of the comma-separated values.
- If the lastChangedGreaterThan parameter is present in the request, the lastChanged property of the reallocation must be greater than or equal to the value.
- If the expiringWithinDays parameter is present in the request, the exPostDueDate property of the reallocation must be greater than or equal to the current system date plus the number of days equal to the parameter value.

The filter criteria are additive, that is they collectively represent a series of AND operations on the filter.

API request/response structure

API	getReallocations
Method	GET
URL	/NEMWholesale/reallocations/v1/getReallocations
Authorisation	Participant specific
Request Content	

API	getReallocations
Success response	Content Body:

```
{
 "transactionId": [string]
 "data":
 {
  "reallocations":
  [
   {
     "reallocationId": [string],
     "startDate": [datetime],
     "endDate": [datetime],
     "submittingParticipantId": [string],
     "counterpartyParticipantId": [string],
     "agreementTypeId": [string],
     "profileTypeId": [string],
     "regionId": [string],
     "currentStepId": [string],
     "creditDebitIndicator": [string],
     "submittingParticipantReference": [string],
     "counterpartyReference": [string],
     "intervalLength": [integer]
     "lastChanged": [integer]
   }
  ]
 }
```

}

Parameters

Name	Required	Datatype	Description	Example
startDate	N	Datetime	Start date parameter of the reallocation filter.	2000-12-31T00:00:00
endDate	N	Datetime	End date parameter of the reallocation filter.	2020-12-31T00:00:00
agreementType	N	String	Agreement types parameter of the reallocation filter, containing a comma-separate listed of agreement type IDs.	MWh

Name	Required	Datatype	Description	Example
steps	N	String	Steps parameter of the reallocation filter, containing a comma-separated list of reallocation step IDs.	SUBMIT
regionId	N	String	Region ID parameter of the reallocation filter.	QLD1
calendarId	N	String	Calendar ID parameter of the reallocation filter.	SETT_ CALENDAR
expiringWithinDays	N	Integer	Expiring within X days parameter of the reallocation filter.	5
lastChangedGreaterThan	N	Integer	Last changed greater than X parameter of the reallocation filter.	3

Example request

```
GET
/NEMWholesale/reallocations/v1/getReallocations?StartDate&EndDate&RegionId&LastChangedGreaterThan&AgreementTypes&Steps&ExpiringWithinDays&CalendarId&ExPostDueDateStart&ExPostDueDateEnd
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip
Authorisation: Basic dGVzdG1hbnVhbSpOZK1tY28wMw==
X-market: NEM
X-initiatingParticipantID:<PARTICIPANTID>
```

Example response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnn
Content-Encoding: gzip
```

```
{
  "transactionId": "a531c7a4-b49e-48c4-91f2-cb8ab486e670",
  "data": {
    "responseAppStatus": "Successful",
    "reallocations": [
      {
        "reallocationId": "20130108.RS004",
        "startDate": "2013-01-20T00:00:00",
        "endDate": "2013-01-26T00:00:00",
        "submittingParticipantId": "<SUBMITTING_PARTICIPANTID>",
        "counterPartyParticipantId": "<COUNTERPARTY_
PARTICIPANTID>",
        "agreementTypeId": "MWh",
        "profileTypeId": "NON_BUSINESS",
        "regionId": "VIC1",
        "currentStepId": "AUTHORISE",
        "creditDebitIndicator": "C",
        "submittingParticipantReference": "APG Week 04-13 NWD",
        "counterPartyReference": null,
        "intervalLength": 0,
        "calendarId": null,
        "lastChanged": "2013-01-10T11:06:16",
        "exAnteDueDate": "2013-01-10T00:00:00",
        "exPostDueDate": "2013-02-19T00:00:00",
        "reallocationProfile": []
      },
      {
        "reallocationId": "20130108.RS003",
        "startDate": "2013-01-20T00:00:00",
        "endDate": "2013-01-26T00:00:00",
        "submittingParticipantId": "<SUBMITTING_PARTICIPANTID>",
        "counterPartyParticipantId": "<COUNTERPARTY_
PARTICIPANTID>",
        "agreementTypeId": "MWh",
        "profileTypeId": "BUSINESS",
        "regionId": "VIC1",
        "currentStepId": "AUTHORISE",
        "creditDebitIndicator": "C",
        "submittingParticipantReference": "APG Week 04-13WD",
        "counterPartyReference": null,
        "intervalLength": 0,
        "calendarId": null,
        "lastChanged": "2013-01-10T11:07:03",
        "exAnteDueDate": "2013-01-10T00:00:00",
        "exPostDueDate": "2013-02-19T00:00:00",
        "reallocationProfile": []
      },
      {
        "reallocationId": "20121120.RS001",
        "startDate": "2012-12-02T00:00:00",
        "endDate": "2012-12-08T00:00:00",
        "submittingParticipantId": "<SUBMITTING_PARTICIPANTID>",
        "counterPartyParticipantId": "<COUNTERPARTY_
PARTICIPANTID>",
        "agreementTypeId"
        "profileTypeId": "BUSINESS",
        "regionId": "VIC1",
        "currentStepId": "AUTHORISE",
        "creditDebitIndicator": "C",
        "submittingParticipantReference": "REF Week 04-13WD",
        "counterPartyReference": null,
        "intervalLength": 0,
        "calendarId": null,
        "lastChanged": "2013-01-10T11:07:03",
        "exAnteDueDate": "2013-01-10T00:00:00",
        "exPostDueDate": "2013-02-19T00:00:00",
        "reallocationProfile": []
      }
     ]
    }
  },
  "errors": []
}
```

getProfileTypes API

The getProfileTypes API retrieves a list of reallocation Day Types and associated descriptions. Possible values are:

- Flat - everyday including Business and Non-Business days.
- Business - Business days do not include weekends or public holidays (based on the chosen PH calendar).
- Non-business - Any day that is not a business day.

API request/response structure

API	getProfileTypes
Method	GET
URL	/NEMWholesale/reallocations/v1/getProfileTypes
Authorisation	Not authenticated
Request Content	None
Success response	Content Body:

```
{
 "transactionId": [string]
 "data":
 {
  "profileTypes":
  [
   {
     "profileTypeId": [string],
     "description": [string]
   }
  ]
 }
}
```

Example request

```
GET /NEMWholesale/reallocations/v1/getProfileTypes
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip
Authorisation: Basic dGVzdG1hbnVhbSpOZK1tY28wMw==
X-market: NEM
X-initiatingParticipantID: <PARTICIPANTID>
```

Example response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnn
Content-Encoding: gzip
{
 "transactionId": "d7c837ae-f752-4d0d-a961-4bce0e47103b",
 "data": {
  "profileTypes": [
   {
    "profileTypeId": "FLAT",
    "description": "Apply to profile to all days"
   },
   {
    "profileTypeId": "BUSINESS",
    "description": "Apply to profile to business days only"
   },
   {
    "profileTypeId": "NON_BUSINESS",
    "description": "Apply to profile to non-business days only"
   }
  ]
 },
 "errors": []
)
```

getReallocationSteps API

The getReallocationSteps API retrieves a list of reallocation step IDs and associated descriptions.

API request/response structure

API	getReallocationSteps
Method	GET
URL	/NEMWholesale/reallocations/v1/getReallocationSteps
Authorisation	Not authenticated
Request Content	None
Success response	Content Body:

```
{
  "transactionId": [string]
  "data":
  {
    "reallocationSteps":
    [
      {
        "stepId": [string],
        "description": [string]
      }
    ]
  }
}
```

Example request

```
GET /NEMWholesale/reallocations/v1/getReallocationSteps
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip
Content-Length: nnn
Content-Encoding: gzip
Authorisation: Basic dGVzdG1hbnVhbSpOZK1tY28wMw==
X-market: NEM
X-initiatingParticipantID: <PARTICIPANTID>
```

Example response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnn
Content-Encoding: gzip
{
 "transactionId": "c084a0c1-633e-4d55-bcde-1f7600b978a1",
 "data": {
  "reallocationSteps": [
   {
    "stepId": "AUTHORISE",
    "description": "Authorised"
   },
   {
    "stepId": "CANCEL",
    "description": "Cancelled"
   },
   {
    "stepId": "EXPIRED",
    "description": "Expired"
   },
   {
    "stepId": "SUBMIT",
    "description": "Submitted"
   }
  ]
 },
 "errors": []
}
```

getRegions API

The getRegions API retrieves a list of regions and associated descriptive names. Possible values are as follows:

- New South Wales
- Queensland
- South Australia
- Tasmania
- Victoria

API request/response structure

API	getRegions
Method	GET
URL	/NEMWholesale/reallocations/v1/getRegions
Authorisation	Not authenticated
Request Content	None

<table>
<tr><th>API</th><th>getRegions</th></tr>
<tr><td>Success response</td><td>

Content Body:

```
{
  "transactionId": [string]
  "data":
  {
    "regions":
    [
      {
        "regionId": [string],
        "name": [string]
      }
    ]
  }
}
```

</td></tr>
</table>

Example request

```
GET /NEMWholesale/reallocations/v1/getRegions
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip
Content-Length: nnn
Content-Encoding: gzip
Authorisation: Basic dGVzdG1hbnVhbSpOZK1tY28wMw==
X-market: NEM
X-initiatingParticipantID: <PARTICIPANTID>
```

Example response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnn
Content-Encoding: gzip
{
  "transactionId": "21674758-ae35-46eb-88a9-14c0bcbc7b5f",
  "data": {
    "regions": [
      {
        "regionId": "NSW1",
        "name": "New South Wales"
      },
      {
        "regionId": "QLD1",
        "name": "Queensland Region"
      },
      {
        "regionId": "SA1",
        "name": "South Australia Region"
      },
      {
        "regionId": "SNOWY1",
        "name": "Snowy Regions"
      },
      {
        "regionId": "TAS1",
        "name": "Tasmanian Region"
      },
      {
        "regionId": "VIC1",
        "name": "Victoria"
      }
    ]
  },
  "errors": []
}
```

getCalendars API

The getCalendars API retrieves a list of Calendar IDs and associated descriptive names fulfilling the following criteria:

- The Calendar is configured for reallocations.
- If the regionId filter parameter is in the request, there must be at least one calendar entry matching the given region.
- If the startDate filter parameter is in the request, there must be at least one calendar entry with a calendar date greater than or equal to the start date.
- If the endDate filter parameter is in the request, there must be at least one calendar entry with a calendar date less than or equal to the end date.

The filter criteria are additive, that is they collectively represent a series of AND operations on the filter.

API request/response structure

API	getCalendars
Method	GET
URL	/NEMWholesale/reallocations/v1/getCalendars
Authorisation	No authorisation
Request Content	

<table>
<tr><th>API</th><th>getCalendars</th></tr>
<tr><td>Success response</td><td>Content Body:<pre>
{
  "transactionId": [string]
  "data":
  {
    "calendars":
    [
      {
        "calendarId": [string],
        "description": [string]
      }
    ]
  }
}
</pre></td></tr>
</table>

Parameters

Name	Required	Datatype	Description	Example
regionId	N	String	Unique region identifier	QLD1
startDate	N	Datetime	Start date of the reallocation	2013-01-20T00:00:00
endDate	N	Datetime	End date of the reallocation	2013-03-20T00:00:00

Example request

```
GET /NEMWholesale/reallocations/v1/getCalendars
Accept: application/json
Accept-Encoding: gzip
Authorisation: Basic dGVzdG1hbnVhbSpOZK1tY28wMw==
X-market: NEM
X-initiatingParticipantID: <PARTICIPANTID>
```

Example response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnn
Content-Encoding: gzip
{
  "transactionId": "187bd455-0108-4411-ab8f-e993448cfde5",
  "data": {
    "calendars": [
      {
        "calendarId": "SETT_REGIONAL",
        "description": "Includes public holidays by NEM region"
      }
    ]
  },
  "errors": []
}
```

getCalendar API

Weekends are implicitly assumed to be non-business days and are not explicitly included in the calendar information returned.

The getCalendar API retrieves a list of public holidays from a given calendar. The calendar non-business days are taken from the NEM database.

API request/response structure

API	getCalendar
Method	GET
URL	/NEMWholesale/reallocations/v1/getCalendar
Authorisation	Not authenticated
Request Content	See Parameters below.
Success response	Content Body:

```
{
  "transactionId": [string]
  "data":
  {
    “calendarId”: [string],
    “regions”:
    [
      {
        “regionId”: [string],
        “nonBusinessDays”:
        [
          [datetime],
          [datetime],
          [datetime],
          [datetime],
          [datetime]
        ]
      }
    ]
  }
}
```

Parameters

Name	Required	Datatype	Description	Example
calendarId	Y	String	Unique identifier of the calendar	SETT_ REGIONAL

Example request

```
GET /NEMWholesale/reallocations/v1/getCalendar?calendarId=SETT_
REGIONAL
Accept: application/json
Accept-Encoding: gzip
Authorisation: Basic dGVzdG1hbnVhbSpOZK1tY28wMw==
X-market: NEM
X-initiatingParticipantID: <PARTICIPANTID>
```

Example response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnn
Content-Encoding: gzip
{
  "transactionId": "c6e37c00-bd13-47e1-a6e9-d434754316dd",
  "data": {
    "calendars": {
      "calendarId": "SETT_REGIONAL",
      "Regions": [
        {
          "regionId": "NSW1",
          "nonBusinessdays": [
            "2016-01-01T00:00:00",
            "2016-01-26T00:00:00",
            ……
            "2021-12-27T00:00:00",
            "2021-12-28T00:00:00"
          ]
        },
        {
          "regionId": "QLD1",
          "nonBusinessdays": [
            "2016-01-01T00:00:00",
            "2016-01-26T00:00:00",
            ……
            "2021-12-27T00:00:00",
            "2021-12-28T00:00:00"
          ]
        },
        {
          "regionId": "SA1",
          "nonBusinessdays": [
            "2016-01-01T00:00:00",
            "2016-01-26T00:00:00",
           ……
      "2021-12-27T00:00:00",
      "2021-12-28T00:00:00"
     ]
    },
    {
     "regionId": "TAS1",
     "nonBusinessdays": [
      "2016-01-01T00:00:00",
      "2016-01-26T00:00:00",
      ......
      "2021-12-27T00:00:00",
      "2021-12-28T00:00:00"
     ]
    },
    {
     "regionId": "VIC1",
     "nonBusinessdays": [
      "2016-01-01T00:00:00",
      "2016-01-26T00:00:00",
      ......
      "2021-12-27T00:00:00",
      "2021-12-28T00:00:00"
     ]
    }
   ]
  }
 },
 "errors": []
}
```

getParticipants API

The getParticipants API retrieves a list of NEM Participant IDs and associated descriptive names fulfilling the following criteria:

- Have a participant class of Market Participant.
- Participants cannot create reallocation requests between participant IDs owned by the same Company ID.

API request/response structure

API	getParticipants
Method	GET
URL	/NEMWholesale/reallocations/v1/getParticipants
Authorisation	Not authenticated
Request Content	None
Success response	Content Body:

```
{
  "transactionId": [string]
  "data":
  {
    "participants":
    [
      {
        "participantId": [string],
        "name": [string]
      }
    ]
  }
}
```

Example request

```
GET /NEMWholesale/reallocations/v1/getParticipants
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip
Content-Length: nnn
Content-Encoding: gzip
Authorisation: Basic dGVzdG1hbnVhbSpOZK1tY28wMw==
X-market: NEM
X-initiatingParticipantID: <PARTICIPANTID>
```

Example response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnn
Content-Encoding: gzip
{
 "transactionId": "796249d9-f4c8-4889-927a-24e55a5c79dc",
 "data": {
  "participants": [
   {
    "participantId": "Participant1",
    "name": " Participant1",
    "enable_Derivatives": false
   },
   {
    "participantId": " Participant2",
    "name": " Participant2,
    "enable_Derivatives": false
   },
.........
   {
    "participantId": " Participant_n-1",
    "name": " Participant_n-1",
    "enable_Derivatives": false
   },
   {
    "participantId": " Participant_n",
    "name": " Participant_n ",
    "enable_Derivatives": false
   }
  }
  ]
 }
 "errors": []
}
```

getAgreementTypes API

The getAgreementTypes API retrieves a list of reallocation agreement types and associated descriptions. Possible values are:

- $ - Dollar offset, a set dollar value for each interval.
- MWh - Energy Offset, a set energy quantity for each interval, values at that interval's RRP.
- Swap - Swap offset, a quantity and price.
- Cap - Cap offset, a quantity and price.
- Floor - Floor offset, a quantity and price.

API request/response structure

API	getAgreementTypes
Method	GET
URL	/NEMWholesale/reallocations/v1/getAgreementTypes
Authorisation	Not authenticated
Request Content	None

API	getAgreementTypes
Success response	Content Body:

```
{
  "transactionId": [string]
  "data":
  {
    "agreementTypes":
    [
      {
        "agreementTypeId": [string],
        "description": [string]
      }
    ]
  }
}
```

Example request

```
GET /NEMWholesale/reallocations/v1/getAgreementTypes
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip
Content-Length: nnn
Content-Encoding: gzip
Authorisation: Basic dGVzdG1hbnVhbSpOZK1tY28wMw==
X-market: NEM
X-initiatingParticipantID: <PARTICIPANTID>
```

Example response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnn
Content-Encoding: gzip
{
  "transactionId": "623aa230-d2aa-4218-9447-eb65364888e5",
  "data": {
    "agreementTypes": [
      {
        "agreementTypeId": "$",
        "description": "Dollar"
      },
      {
        "agreementTypeId": "MWh",
        "description": "Quantity"
      }
    ]
  },
  "errors": []
}
```

getMarketPriceCap API

The getMarketPriceCap API retrieves the VOLL (Value of Lost Load) price at the given effective date in the request.

The specific value retrieved is the one that:

- Has an effectiveDate value that is the greatest effectiveDate value, less than or equal to the supplied effective date in the request, where the record is authorised.
- Has a versionNo value that is greater than the versionNo value, for the corresponding effectiveDate value, where the record is authorised.

API request/response structure

API	getMarketPriceCap
Method	GET
URL	/NEMWholesale/reallocations/v1/getMarketPriceCap
Authorisation	Not authenticated
Request Content	
Success response	Content Body:

```
{
  "transactionId": [string]
  "data":
  {
    "vollPrice": [number]
  }
}
```

Parameters

Name	Required	Datatype	Description	Example
effectiveDate	Y	Datetime	The date for which to retrieve the Market Price Cap in yyyy-mm-dd format	2016-01-01T00:00:00

Example request

```
GET /NEMWholesale/reallocations/v1/getMarketPriceCap
Content-Type: application/json
Accept: application/json
Accept-Encoding: gzip
Content-Length: nnn
Content-Encoding: gzip
Authorisation: Basic dGVzdG1hbnVhbSpOZK1tY28wMw==
X-market: NEM
X-initiatingParticipantID: <PARTICIPANTID>
```

Example response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnn
Content-Encoding: gzip
{
 "transactionId": "3d681ce4-57e5-4446-a6d9-74a874963e84",
 "data": {
  "vollPrice": 11100.0
 },
 "errors": []
}
```

Appendix Reallocation JSON Schema

This section provides the JSON schema for the Reallocation APIs.

The following is used to describe a fields requirement:

- Mandatory (M) – The field must be provided, in respect to its parent element.
- Optional (O) – You may provide the field, if not provided a default is assumed. To indicate no value, you must remove the entire attribute.
- Conditional (C) - The field is normally optional but may be mandatory under certain conditions.

submitReallocation

```
{
  "definitions": {},
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "http://nemweb.com.au/schemas/json/submitReallocation.json",
  "type": "object",
  "description": "An explanation about the purpose of this instance.",
  "additionalProperties": false,
  "required": [
    "reallocation"
  ],
  "properties": {
    "reallocation": {
      "$id": "#/properties/reallocation",
      "type": "object",
      "description": "Reallocation details of the request",
      "additionalProperties": false,
      "required": [
        "startDate",
        "endDate",
        "submittingParticipantId",
        "counterPartyParticipantId",
        "agreementTypeId",
        "profileTypeId",
        "regionId",
        "creditDebitIndicator",
        "intervalLength",
        "submittingParticipantReference",
        "calendarId",
        "reallocationProfile"
      ],
      "properties": {
        "startDate": {
          "$id": "#/properties/reallocation/properties/startDate",
          "type": "string",
          "description": "Start date of reallocation request.",
          "format": "date-time"
        },
        "endDate": {
          "$id": "#/properties/reallocation/properties/endDate",
          "type": "string",
          "description": "End date of reallocation request.",
          "format": "date-time"
        },
        "submittingParticipantId": {
          "$id": "#/properties/reallocation/properties/submittingParticipantId",
          "type": "string",
          "description": "submitting participant id of reallocation request.",
          "minLength": 1,
          "maxLength": 20
        },
        "counterPartyParticipantId": {
          "$id":
    "#/properties/reallocation/properties/counterPartyParticipantId",
          "type": "string",
          "description": "counter party id of reallocation request.",
          "minLength": 1,
          "maxLength": 20
        },
        "agreementTypeId": {
          "$id": "#/properties/reallocation/properties/agreementTypeId",
          "type": "string",
          "description": "Agreementtypeid of reallocation request.",
          "minLength": 1,
          "maxLength": 10
        },
        "profileTypeId": {
          "$id": "#/properties/reallocation/properties/profileTypeId",
          "type": "string",
          "description": "Profiletype of reallocation request.",
          "minLength": 1,
          "maxLength": 20
        },
        "regionId": {
          "$id": "#/properties/reallocation/properties/regionId",
          "type": "string",
          "description": "Region of reallocation request.",
          "minLength": 1,
          "maxLength": 20
        },
        "creditDebitIndicator": {
          "$id": "#/properties/reallocation/properties/creditDebitIndicator",
          "type": "string",
          "description": "Credit/ Debit indicator - C/D of reallocation request.",
          "maxLength": 1
        },
        "intervalLength": {
          "$id": "#/properties/reallocation/properties/intervalLength",
          "type": "integer",
          "description": "Interval length 5 or 30 minutes of reallocation
    request.",
         "minimum": 5,
         "maximum": 30
        },
        "submittingParticipantReference": {
         "$id":
    "#/properties/reallocation/properties/submittingParticipantReference",
         "type": "string",
         "description": "Submitting participant reference of reallocation
    request.",
         "minLength": 1,
         "maxLength": 400
        },
        "calendarId": {
         "$id": "#/properties/reallocation/properties/calendarId",
         "type": "string",
         "description": "CalendarId available in the configurable data of
    reallocation request.",
         "minLength": 1,
         "maxLength": 30
        },
        "reallocationProfile": {
         "$id": "#/properties/reallocation/properties/reallocationProfile",
         "type": "array",
         "description": "An explanation about the purpose of this instance.",
         "uniqueItems": true,
         "additionalItems": false,
         "minItems": 48,
         "maxItems": 288,
         "items": {
          "$id":
    "#/properties/reallocation/properties/reallocationProfile/items",
          "type": "object",
          "description": "Reallocation interval of reallocation request.",
          "additionalProperties": false,
          "required": [
           "periodId",
           "reallocationValue",
           "nrp"
          ],
          "properties": {
           "periodId": {
            "$id":
    "#/properties/reallocation/properties/reallocationProfile/items/properties
    /periodId",
            "type": "integer",
            "description": "Period Id of reallocation request.",
            "minimum": 1,
            "maximum": 288
           },
           "reallocationValue": {
            "$id":
    "#/properties/reallocation/properties/reallocationProfile/items/properties
    /reallocationValue",
            "type": "number",
            "description": "Reallocation value of reallocation request."
           },
           "nrp": {
            "$id":
    "#/properties/reallocation/properties/reallocationProfile/items/properties
    /nrp",
            "type": "number",
            "description": "NRP value of reallocation request."
           }
          }
         }
        }
       }
      }
     }
    }
```

authoriseReallocation

```
{
 "definitions": {},
 "$schema": "http://json-schema.org/draft-07/schema#",
 "$id":
"http://nemweb.com.au/schemas/json/authoriseReallocation.json",
 "type": "object",
 "description": "Schema of authorise reallocation request",
 "additionalProperties": false,
 "required": [
  "reallocationId",
  "counterpartyReference"
 ],
 "properties": {
  "reallocationId": {
   "$id": "#/properties/reallocationId",
   "type": "string",
   "description": "Reallocation id of the request",
   "minLength": 1,
   "maxLength": 20
  },
  "counterpartyReference": {
   "$id": "#/properties/counterpartyReference",
   "type": "string",
   "description": "Counterparty reference of the details.",
   "minLength": 1,
   "maxLength": 400
  }
 }
}
```

cancelReallocation

```
{
  "definitions": {},
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "http://nemweb.com.au/schemas/json/CancelReallocation.json",
  "type": "object",
  "description": "Schema of Cancel reallocation request",
  "additionalProperties": false,
  "required": [
    "reallocationId",
    "counterpartyReference"
  ],
  "properties": {
    "reallocationId": {
      "$id": "#/properties/reallocationId",
      "type": "string",
      "description": "Reallocation id of the request",
      "minLength": 1,
      "maxLength": 20
    },
    "counterpartyReference": {
      "$id": "#/properties/counterpartyReference",
      "type": "string",
      "description": "Counterparty reference of the details.",
      "minLength": 1,
      "maxLength": 400
    }
  }
}
```

Needing Help

AEMO's Support Hub

IT assistance is requested through one of the following methods:

- Phone: 1300 AEMO 00 (1300 236 600)

  For non-urgent issues, normal coverage is 8:00 AM to 6:00 PM on weekdays, Australian Eastern Standard Time (AEST).

- The Contact Us form on AEMO's website.

Information to provide

AEMO recommends participants call AEMO's Support Hub for all urgent issues, whether or not you have logged a call using the contact us form.

Please provide the following information when requesting assistance from AEMO:

- Your contact details
- Company name
- Company ID
- System or application name
- Environment: production or pre-production
- Problem description
- Screenshots

For AEMO software-related issues please also provide:

- Participant ID (if Data Interchange (DI) problem)
- Version of software
- Properties or log files
- PDR Monitor support dump and DI instance name (if DI problem)

References

You can find resources on AEMO's website.

Data Interchange Framework and Glossary: provides important information about upgrading your Data Interchange (DI) environment, explains DI terms, and DI related resources. Please read this guide in conjunction with this technical specification.

Guide to AEMO’s e-Hub APIs: Provides details about using AEMO’s e-Hub as an interface to communicate information with AEMO. It assists Wholesale electricity and gas participants developing their own APIs.

Guide to Electricity Reallocations: Provides instructions for Reallocations Registered Participants to use the Reallocations Request Service (Reallocations) in the Energy Market Systems Web Portal (Web Portal).

Guide to Electricity Information Systems: Provides guidance for Registered Participants and interested parties about AEMO's participant electricity market systems.

Guide to User Rights Management: Assists participant administrators (PAs) to use the user rights management functions in the MSATS Web Portal.

Connecting to AEMO’s Electricity IT Systems: Provides guidance for Registered Participants about using the IT interfaces available and how to access them.

Rules and Procedures Terms

This section lists the terms defined in the National Electricity Rules (NER) and the following Reallocations procedures:

- Reallocation Procedure Energy and Dollar Offset Reallocations.
- Reallocation Procedure Swap and Option Offset Reallocations.

Term
AEMO
Agreement Type
Billing Period
Business
Calendar
Cap Offset
Counterparty
Credit Participant
Day Type
Debit Participant
Dollar Offset

Term
Energy Offset
Ex Ante Reallocation
Ex Post Reallocation
Flat
Floor Offset
Letter of Agreement
Market Customer
Market Participant
MWh Offset
NEM
Non-business
Participant ID
Period ID
Profile Type
Prudential Requirements
Reallocation Request Service

Term
Reallocation Timetable
Region ID
Registered Participant
Settlements
Strike Price
Swap Offset
Trading Day
Trading Interval

Glossary

30-min period
New rules term replacing Trading Interval, where the period remains as 30 minutes

AEMO API Gateway
The gateway on AEMO's side providing participant communication options, accessible over the internet or MarketNet. It uses resources and methods to push messages to Participants' API Gateways .

AEST
Australian Eastern Standard Time

API
Application Programming Interface. A set of clearly defined methods of communication between various software components.

API Portal
Where you can view available APIs, manage your API Keys, and obtain OAS files.

API Protocol
An e-Hub delivery method.

aseXML
A standard developed by Australian energy industries to facilitate the exchange of information between energy industry participants using XML.

Bids/Offers
Relates to the following Dispatch Bids: 1. Energy (Scheduled Loads) The term Offer relates to the following Dispatch Offers: 1. Energy (Generation Dispatch Offer) 2. Frequency Control Ancillary Service (FCAS) 3. Market Network Service Provider (MNSP - Network Dispatch Offer)

CSR
Certificate Signing Request is a block of encoded text given to a Certificate Authority when applying for an SSL Certificate. It also contains the Public Key to include in the certificate. Usually, a Private Key is created at the same time, making a Key Pair.

csv
Comma-separated values; a file format for exchanging data.

Curl
A command line utility used to interact with REST API endpoints.

Data Model
The definition of the interface to participants of data published by AEMO for gas or electricity. A database conforming to the Data Model can contain a local copy of all current participant-specific data recorded in the main database. The Data Model includes database tables, indexes, and primary keys

DUID
Dispatchable Unit or MSNP Interconnector ID

e-Hub
Consists of the API Portal and the API Gateway for both electricity and gas.

EDM
Electricity Data Model

EMMS
Wholesale Electricity Market Management System; software, hardware, network and related processes to implement the energy market.

Endpoint
Where the API request is sent and where the response comes from.

FCAS
Frequency Control Ancillary Services

Fixed Load
Optional MW, not greater than the max. availability

FTP
File transfer protocol; a standard network protocol used for the transfer of computer files between a client and server on a computer network.

Header Parameters
Parameters included in the request header.

JSON
Java Standard Object Notation. An agreed format for text files and data exchange. This is now used by AEMO to receive Bids and Offers and provide responses

JSON Schema
Defines the structure and content of the bidding details.

Key Pair
SSL uses a technique called public-key cryptography, based on the concept of a Key Pair. The Key Pair consists of encrypted Public and Private Key data. It is only possible to decrypt the Public Key with the corresponding Private Key.

Link ID
Identifies the MNSP interconnector link in AEMO's systems. A property (linkId) in the MNSPBidLink object in the JSON bidding schema

MACK
Message Acknowledgement

MarketNet
AEMO’s private network available to participants having a participant ID

Markets Portal
Web portal for access to AEMO's wholesale web-based applications.

Method
The allowed operation for a resource, e.g. GET, POST, PUT, DELETE, and so on. These operations determine whether you’re reading information, creating new information, updating existing information, or deleting information.

MNSP
Market Network Service Provider

MSATS
Retail Market Settlement and Transfer Solution

MSATS Web Portal
MSATS web-based interactive interface

MW
Megawatt

NACK
Negative Acknowledgement (Rejection)

NER
National Electricity Rules

OAS
OpenAPI specification

OpenAPI specification document
The file, either in YAML or JSON, describing your REST API. Follows the OpenAPI specification format.

PA
Participant Administrator who manages participant company's user access and security. The initial PA is set up by the AEMO system administrator as part of the registration process.

Parameters
Parameters are options you pass with the endpoint (such as specifying the response format or the amount returned). There are four types of parameters: header parameters, path parameters, query string parameters, and request body parameters. The different types of parameters are often documented in separate groups on the same page. Not all endpoints contain each type of parameter. See Parameters for more details.

Participant API Gateway
The interface implemented by participants where AEMO pushes messages.

Participant File Server
The publishing point from AEMO systems to participant systems. Each participant is allocated an account and access to private and public areas. Participants are responsible for interfacing with the Participant File Server. If uncollected, files are moved to the archive folder after a couple of days. If your Data Interchange environment is configured properly it automatically retrieves the missing files from the archive. Files are kept in the archive for approximately six months. AEMO’s production and pre-production environments are independently operated, so each environment has its own IP address for its Participant File Server. For help, see Connection to AEMO's IT Systems.

Participant ID
Registered participant identifier

Participant User ID
The user ID you used to login to the system.

Participant Users
Set up by the company's Participant Administrator.

Path
Parameters in the path of the endpoint, before the query string (?). Path parameters are usually set off within curly braces.

Payload
The data sent by a POST request. The Payload section sits after the header.

PID
Participant ID

Pre-production
AEMO's test system available to participants

Private Key
The secret Private Key is a text file used initially to generate a Certificate Signing Request (CSR), and later to secure and verify connections.

Production
AEMO's live system

Public Key
The Public Key is included as part of your SSL certificate, and works together with your Private Key to make sure your data is encrypt Key (i.e. the certificate) can verify the digital signature is authentic without having to know the secret Private Key.

Query String Parameters
Parameters in the query string of the endpoint, after the ?.

Request
The way information is returned from an API. In a request, the client provides a resource URL with the proper authorization to an API server. The API returns a response with the information requested.

Request Body Parameters
Parameters in the request body. Usually submitted as JSON.

Response
The information returned by an API after a request is made. Responses are usually in JSON or XML format.

Response Example
The response example shows a sample response from the request example; the response schema defines all possible elements in the response. The response example is not comprehensive of all parameter configurations or operations, but it should correspond with the parameters passed in the request example. The response lets developers know if the resource contains the information they want, the format, and how that information is structured and labeled. The description of the response is known as the response schema. The response schema documents the response in a more comprehensive, general way, listing each property that could possibly be returned, what each property contains, the data format of the values, the structure, and other details.

REST
The Representational State Transfer API architecture

Service Type
ENERGY, MNSP, RAISE6SEC, RAISE60SEC, RAISE5MIN, RAISEREG, LOWER6SEC, LOWER60SEC, LOWER5MIN, LOWERREG

SSL
Secure Sockets Layer, cryptographic protocol providing API communication security

Submission
A Bid/Offer submission can have: 1. Multiple Trading Days 2. Multiple DUIDs/Interconnector IDs 3. All Service Types in the same Submission

Swagger file
Refers to the OpenAPI specification

TACK
Transaction Acknowledgement

Throttling
AEMO uses API throttling to prevent overwhelming the API Gateway.

TLS
Transport Layer Security, cryptographic protocol providing API communication security

Unit
Generating Unit

URM
User Rights Management; see the Guide to URM on AEMO's website

zip
The file compression format used for exchanging data with AEMO.

Index