

ECHONET Lite Web API ガイドライン
DR関連サービス仕様
Version 1.0.0



改定履歴

日付	版	説明
2023/6/30	Ver. 1.0.0	制定、一般公開

-
- エコーネットコンソーシアムが発行している規格類は、工業所有権(特許, 実用新案など)に関する抵触の有無に関係なく制定されています。
エコーネットコンソーシアムは、この規格類の内容に関する工業所有権に対して、一切の責任を負いません。
 - この書面の使用による、いかなる損害も責任を負うものではありません。
-

目次

- 1. はじめに
- 2. 参考文献
- 3. 用語
- 4. 本APIの扱いおよび本書の構成
- 5. ユースケース
 - 5.1 基本モデル
 - 5.2 ACとRA間での想定モデル
 - 5.3 RAと需要家向けサーバ間（HEMSクラウド）での想定モデル
- 6. 詳細仕様
 - 6.1 drResources
 - 6.1.1 DR対象エネルギーリソース登録 POST /elapi/v1/drResources
 - 6.1.2 DR対象エネルギーリソース一覧取得 GET /elapi/v1/drResources
 - 6.1.3 DR対象エネルギーリソースDescription取得 GET /elapi/v1/drResources/{drResourceId}
 - 6.1.4 DR対象エネルギーリソース内容取得 GET /elapi/v1/drResources/{drResourceId}/properties
 - 6.1.5 DR対象エネルギーリソース内容変更 PUT /elapi/v1/drResources/{drResourceId}/properties/{任意のプロパティリソース}
 - 6.2 drEvents
 - 6.2.1 DRイベント登録 POST /elapi/v1/drEvents
 - 6.2.2 DRイベント一覧取得 GET /elapi/v1/drEvents
 - 6.2.3 DRイベントDescription取得 GET /elapi/v1/drEvents/{drEventId}
 - 6.2.4 DRイベント内容取得 GET /elapi/v1/drEvents/{drEventId}/properties
 - 6.2.5 DRイベント内容変更 PATCH /elapi/v1/drEvents/{drEventId}/properties
 - 6.2.6 DRイベント応諾結果取得 POST /elapi/v1/drEvents/{drEventId}/actions/getOpts
 - 6.2.7 DRイベント中止 POST /elapi/v1/drEvents/{drEventId}/actions/abort
 - 6.2.8 DRイベント削除 DELETE /elapi/v1/drEvents/{drEventId}
 - 6.3 drReports
 - 6.3.1 DRレポート仕様登録 POST /elapi/v1/drReports
 - 6.3.2 DRレポート一覧取得 GET /elapi/v1/drReports
 - 6.3.3 DRレポートDescription取得 GET /elapi/v1/drReports/{drReportId}
 - 6.3.4 DRレポート内容取得 GET /elapi/v1/drReports/{drReportId}/properties
 - 6.3.5 DRレポート値取得 POST /elapi/v1/drReports/{drReportId}/actions/getValues
 - 6.3.6 DRレポート削除 DELETE /elapi/v1/drReports/{drReportId}
 - 6.4 resHistories (DR用指定例)
 - 6.4.1 GET /elapi/v1/resHistories
 - 6.4.2 GET /elapi/v1/resHistories/{resHistoryId}
 - 6.4.3 GET /elapi/v1/resHistories/{resHistoryId}/properties
 - 6.4.4 POST /elapi/v1/resHistories/{resHistoryId}/actions/prepareRetrieveData
 - 6.4.5 POST /elapi/v1/resHistories/{resHistoryId}/actions/retrieve
 - 6.5 dsrmss

- 6.5.1 DSR-MS IDの一覧取得 GET /elapi/v1/dsrms
 - 6.5.2 DSR-MS Description取得 GET /elapi/v1/dsrms/{dsrmssId}
 - 6.5.3 DSR-MSリソース内容取得 GET /elapi/v1/dsrms/{dsrmssId}/properties
 - 6.5.4 DSR-MSリソース内DR対象エネルギーリソース変更 PUT
/elapi/v1/dsrms/{dsrmssId}/properties/drResources
 - 6.5.5 (参考) DRに必要なDSR-MS情報の収集手順
- 7. おわりに

図目次

図5-1 基本構成

図5-2 DR操作シーケンス (DR対象エネルギーリソース)

図5-3 DR操作シーケンス (DRイベント)

図5-4 DR操作シーケンス (DRレポート)

図5-5 AG-RA間制御モデル

図5-6 RA-HEMSクラウド間制御モデル

図5-7 DR対象エネルギーリソースと非DR対象エネルギーリソース

図5-8 DR対象エネルギーリソースとコントローラの関係

図5-9 需要家を跨るDR対象エネルギーリソース (グループ化) 例

図6-1 DR対象エネルギーリソースの操作

図6-2 DRイベントの操作

図6-3 DRイベントの登録内容

図6-4 区間毎の応諾結果例

図6-5 DRレポートリソースの操作

図6-6 計測値 (type=measure) の例 : granularityとintervalが同一ケース

図6-7 計測値 (type=measure) の例 : granularityとintervalが異なるケース

図6-8 計測値におけるdataCacheDurationの例

図6-9 予測値 (type=projected) の例 : dataCacheDurationがintervalより短い場合

図6-10 予測値 (type=projected) の例 : dataCacheDurationがintervalより長い場合

図6-11 DSR-MSリソースの操作

1. はじめに

このドキュメントは、ECHONET Lite Web APIガイドライン（API仕様部[2]）をベースに、Demand Response（以降DRと称す）サービスを実現するためのAPIについて記述する。

DRはVirtual Power Plant（以降VPPと称す）の主要となる手法で、需要家側のエネルギーリソースを制御し需要を増減させることができる。VPPは、太陽光発電、蓄電池、電気自動車、ネガワット（節電した電力）といった需要家エネルギーリソースの他、電力系統に直接接続される発電設備・蓄電設備を含む、分散型エネルギーリソース（以降DERと称す）を対象に、高度なエネルギー監視・統合制御を実施することで仮想的な発電所のような機能を提供する。VPPにより、調整力提供やインバランス回避、出力抑制回避、電力料金削減など多岐にわたるサービスの提供が期待されている。国内では、DERを保有する需要家とVPPサービス契約を締結してリソース集約・制御するリソースアグリゲーター（以降RAと称す）やRAが制御する電力量を束ね一般送配電事業者や小売電気事業者と電力取引を行うアグリゲーションコーディネーター（以降ACと称す）といった事業者が存在し、両役割を兼ねる事業者も存在する。

RAが制御対象とするDERは、高圧から低圧含め多様な機器が候補となるが、DERをRAが直接収容するケースや、DERの管理単位・主体による監視・制御サーバ（クラウド）を経由して間接的に収容するケースがある。後者の場合は、HEMS、BEMSといった需要家向けサーバに対応する場合が多い。

本書では、基本的にACとRA間およびRAと需要家向けサーバ間で適用されるDR用APIを規定するが、適応先領域については限定せず、各種DR用途にて幅広く活用されるAPIとなることを目指す方針とする。

2. 参考文献

- [1] JEMA VPPにおける需要家エネルギーリソースの活用に関するガイドライン[第2.0版], https://www.jema-net.or.jp/Japanese/res/hems/data/VPP_guidelinev2.pdf
- [2] ECHONET Lite Web APIガイドライン API仕様部[Version 1.1.5], https://echonet.jp/web_api_guideline/
- [3] ECHONET Lite Web APIガイドライン 機器仕様部[Version 1.5.0], https://echonet.jp/web_api_guideline/
- [4] エネルギー・リソース・アグリゲーション・ビジネス検討会, デマンドリスポンス・インタフェース仕様書[第2.0版], https://www.enecho.meti.go.jp/category/saving_and_new/advanced_systems/vpp_dr/files/openadr_spec.pdf
- [5] 経済産業省資源エネルギー庁, エネルギー・リソース・アグリゲーション・ビジネスに関するガイドライン, <https://www.meti.go.jp/press/2020/06/20200601001/20200601001-1.pdf>
- [6] エネルギー・リソース・アグリゲーション・ビジネス検討会, OpenADR機器別実装ノート[第1.1版], https://www.enecho.meti.go.jp/category/saving_and_new/advanced_systems/vpp_dr/files/005_10_00.pdf

3. 用語

用語	説明
DER	Distributed Energy Resources。分散型エネルギーリソース。需要家側にて分散配置される太陽光発電、蓄電池などのエネルギー源
DR	Demand Response。電力の需要と供給のバランスをとるために、電力需要を制御すること
VPP	Virtual Power Plant。仮想発電所。DERなどを相互に接続し、エネルギー制御することで一つの発電機のように機能させる仕組み

用語	説明
DR対象エネルギーリソース	DR制御されるDERを複数まとめた群
DRイベント	本APIを用いて指定されるDR指示内容
DRレポート	DR対象エネルギーリソースに関する、DR制御結果の計測値や将来の予測値（供給可能容量など）

4. 本APIの扱いおよび本書の構成

ECHONET Lite Web APIガイドラインにおける「API仕様部[2]」では汎用的なユースケースや基本・応用API仕様などを規定し、「機器仕様部[3]」ではECHONET Lite機器（もしくはECHONET Lite Web API仕様マッピング可能なIoT機器）を対象とした各種機器用API仕様を規定している。

本「DR関連サービス仕様」は、「API仕様部[2]」や「機器仕様部[3]」とは別に、DR制御実現のために導入する特定のサービス種である`drResources/drEvents/drReports/dsrms`の仕様について解説する（各サービス種の説明は後述）。

本DR関連サービス種を搭載したサーバでは、“`/elapi/v1`”パスへのGETメソッド呼び出しに応じて、サーバが対応しているサービス種一覧が返却される。下記のレスポンス例では、`drResources/drEvents/drReports/dsrms`の各サービス種に対応していることを示している。なお、サーバは対応サービス種を自由に選択・組み合わせることができ、DRレポートの履歴値に対応する場合は`resHistories`を追加したり、「API仕様部[2]」記載の`devices/bulks/groups/histories`など他のサービス種を併記することも自由に設定できる。

リクエスト例

```
GET /elapi/v1 HTTP/1.1
```

レスポンス例

```
{
  "v1": [
    {
      "name": "drResources",
      "descriptions": {
        "ja": "DRリソースの説明文",
        "en": "DR resources"
      }
    },
    {
      "name": "drEvents",
      "descriptions": {
        "ja": "DRイベントの説明文",
        "en": "DR events"
      }
    },
    {
      "name": "drReports",
      "descriptions": {
        "ja": "DRレポートの説明文",
        "en": "DR report"
      }
    }
  ]
}
```

```
    }  
  },  
  {  
    "name": "dsrmss",  
    "descriptions": {  
      "ja": "DSR-MSSの説明文",  
      "en": "DSR-MSS"  
    }  
  }  
]  
}
```

以降、本書の構成について説明する。5章にて想定ユースケースを示し、いくつか代表的なDR制御モデルについて検討する。6章にて、DRにかかわる各種サービス種について詳細なAPI仕様を示す。

5. ユースケース

本DR用ECHONET Lite Web APIを用いた基本的なDR制御モデルについて説明する。

DR指令・制御など、共通的なDR操作ベースモデルについて5.1にて示し、AGとRA間での想定モデルを5.2、RAと需要家向けサーバ間（HEMSクラウド）での想定モデルを5.3に示す。

5.1 基本モデル

図5-1のように、DR制御を実施する上位システム（クライアント）は、DR対象エネルギーリソースを束ねて監視・制御するサーバに対して同サーバ上に実装されるDR用ECHONET Lite Web APIを呼び出す形で、各種DR操作を行う。同サーバからDR制御されるDERは複数まとめた群（DR対象エネルギーリソースと呼ぶ）として管理可能であり、クライアントはこのDR対象エネルギーリソース単位でDR制御を行う。

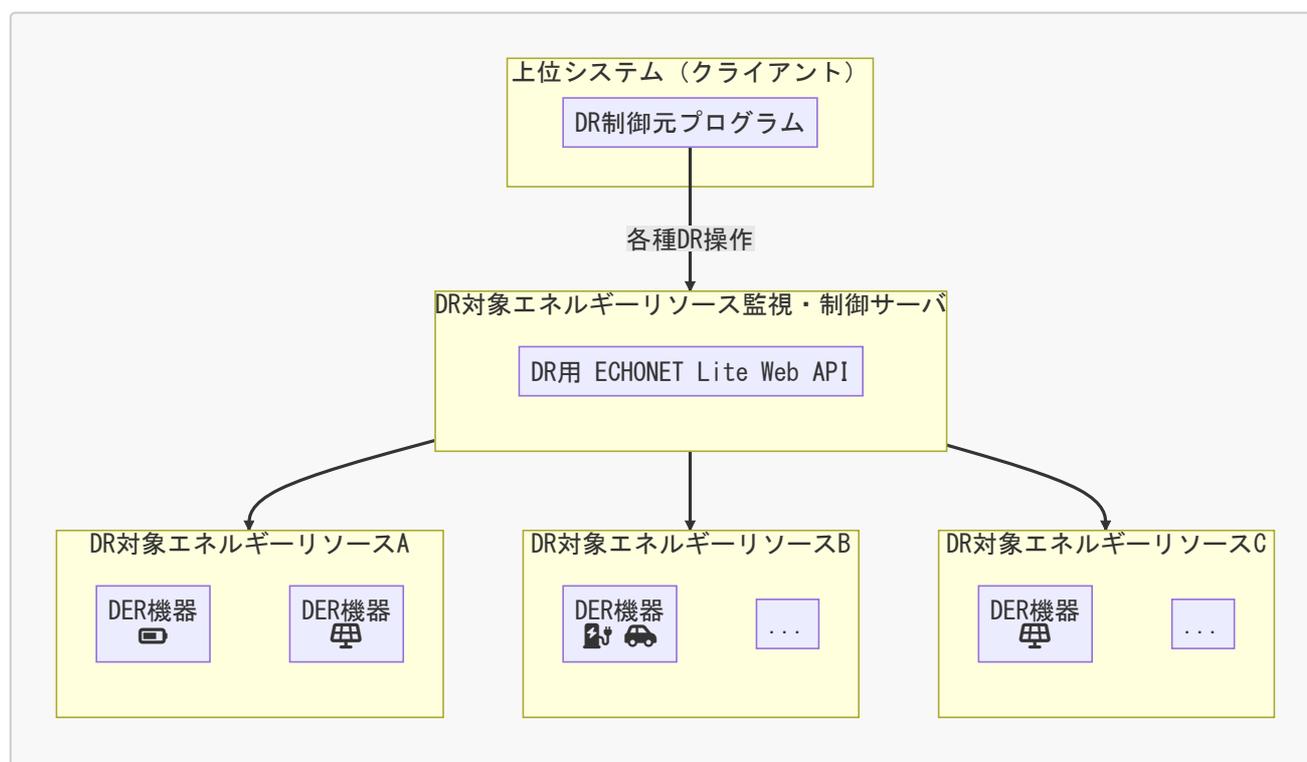


図5-1 基本構成

クライアントが実施する各種DR操作に関連するシーケンスについて説明する。

DR対象エネルギーリソースの登録

DR対象エネルギーリソースをサーバへ登録するには、サーバ自身が手動でDR対象エネルギーリソースを自身のDBなどへ登録する方法や、クライアントがDR対象エネルギーリソースを把握した上で、サーバに対して登録する方法が考えられる（図5-2）。

後述する各種DR操作は、DR対象エネルギーリソースIDを指定できれば実施可能なため、DR対象エネルギーリソースをサーバに対して登録するAPIはオプション扱いとする。

受電点および機器点の扱い

1つのDR対象エネルギーリソースとしてまとめられるDER群に関する種類として、受電点（需要家側リソース群）、機器点（本書では蓄電池群を規定。将来EV充放電器群やPV群を検討予定）を扱う。共にDR対象エネルギーリソースIDによって識別・選択される対象となるが、後述のスケジュール制御や計測値・予測値の取得などにて、指定可能な（扱える）値の種類が異なる。

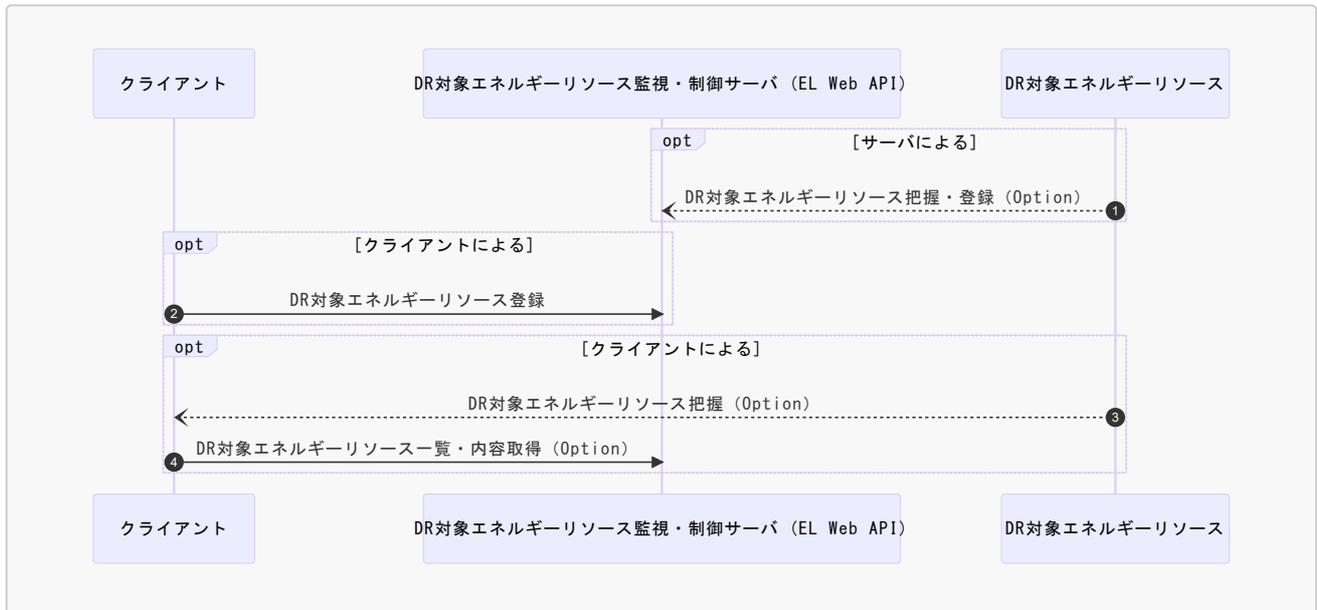


図5-2 DR操作シーケンス (DR対象エネルギーリソース)

DR対象エネルギーリソースのスケジュール制御と応諾状況確認

クライアントは、各DR対象エネルギーリソースに対してDR開始時刻や任意のタイムスロットおよび同スロット内でのエネルギー制御量の組などをDR用ECHONET Lite Web APIに対して指定することで、所望のDR指令・制御指示を行う（図5-3）。このとき指定されるDR指示内容をDRイベントと呼ぶ。DR対象エネルギーリソース監視・制御サーバは、指示されたDRイベントの内容に基づき、各DR対象エネルギーリソース内の各機器に対して（必要に応じて）制御量の配分・スケジュールリングなどを決定し、適切に制御を実施する。具体的な配分・計画の算出方法については本仕様では規定しない。

制御量については、「上げ/下げDR指示電力量 (kWh)」、「上げ/下げDR指示電力 (kW)」を指定可能とし、受電点が対象の場合は負荷増減制御 (Δ kW/ Δ kWh制御) もしくは直接負荷制御 (kW/kWh制御)、機器点（蓄電池群）が対象の場合は充放電制御 (kW/kWh/%) による制御を可能とする。また、個々のDRイベントは「DRイベントID」によって指定・識別でき、クライアントは、要求したDRイベントが受領されたか/否認されたか確認が必要な場合、サーバの応諾状況を取得できる。

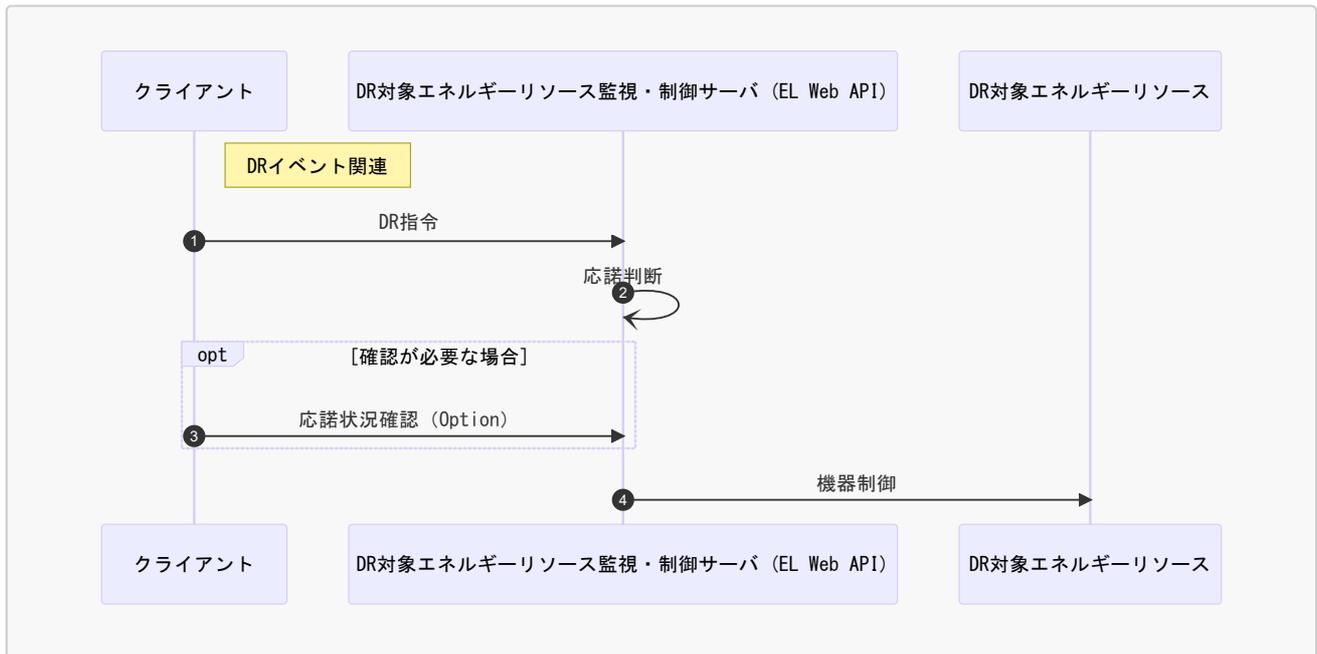


図5-3 DR操作シーケンス (DRイベント)

DR対象エネルギーリソースのDR制御結果（計測値）の取得

DRイベントによるDR制御実施に伴う、各DR対象エネルギーリソース内でのエネルギー制御結果・計測値の取得もDR用ECHONET Lite Web APIを用いて提供される（図5-4）。同制御結果・計測値の内容をDRレポートと呼ぶ。計測値としては、計測値の種類や計測間隔などを指定することができ、クライアントがDRイベントの実行に合わせてDRレポート用APIを用いてDR実行期間中または実行直後に取得することが典型的な使用ケースとなる。

DR対象エネルギーリソースの対象が、受電点（需要家側リソース群）の場合、基準値（直前計測方式）、実績電力値、実績電力量値が取得可能であり、機器点（蓄電池群）の場合、蓄電池の正常・異常状態、充電（放電）電力（量）、充電（放電）可能量、蓄電池残量が取得可能とする。

受電点の各値については、スマートメータの値を活用する。2025年度以降、導入が計画されている次世代スマートメータが利用できる場合は、計測値に関して1分値といった時間粒度（granularity）の指定も可能となる。

未来で利用可能なDR対象エネルギーリソース量（予測値）や過去の計測値（履歴値）の取得

DRレポートは、DR対象エネルギーリソースに関するエネルギー量の予測値（供給可能容量など）についても利用できる仕様としている。DRレポート用APIを用いて、将来DR対象エネルギーリソースより提供可能なエネルギー量を予測し、クライアントへ返却する。具体的な予測方法については本仕様では規定しない。

DR対象エネルギーリソースの対象が受電点（需要家側リソース群）の場合、制御可能量、基準値（直前予測方式）が取得可能となる。

DRレポートの一種として、ECHONET Lite Web APIガイドラインAPI仕様部V.1.1.5[2]にて導入されたresHistoriesを活用して、過去の計測値などの履歴値を取得可能とする。

ベースラインの取得

DR対象エネルギーリソースの対象が受電点（需要家側リソース群）の場合、DRレポートの計測値にて直前計測方式による基準値を取得することができ、また、DRレポートの予測値では直前予測方式による基準値を取得することができる。各

基準値に関して、具体的な算出・予測方法については規定しない。

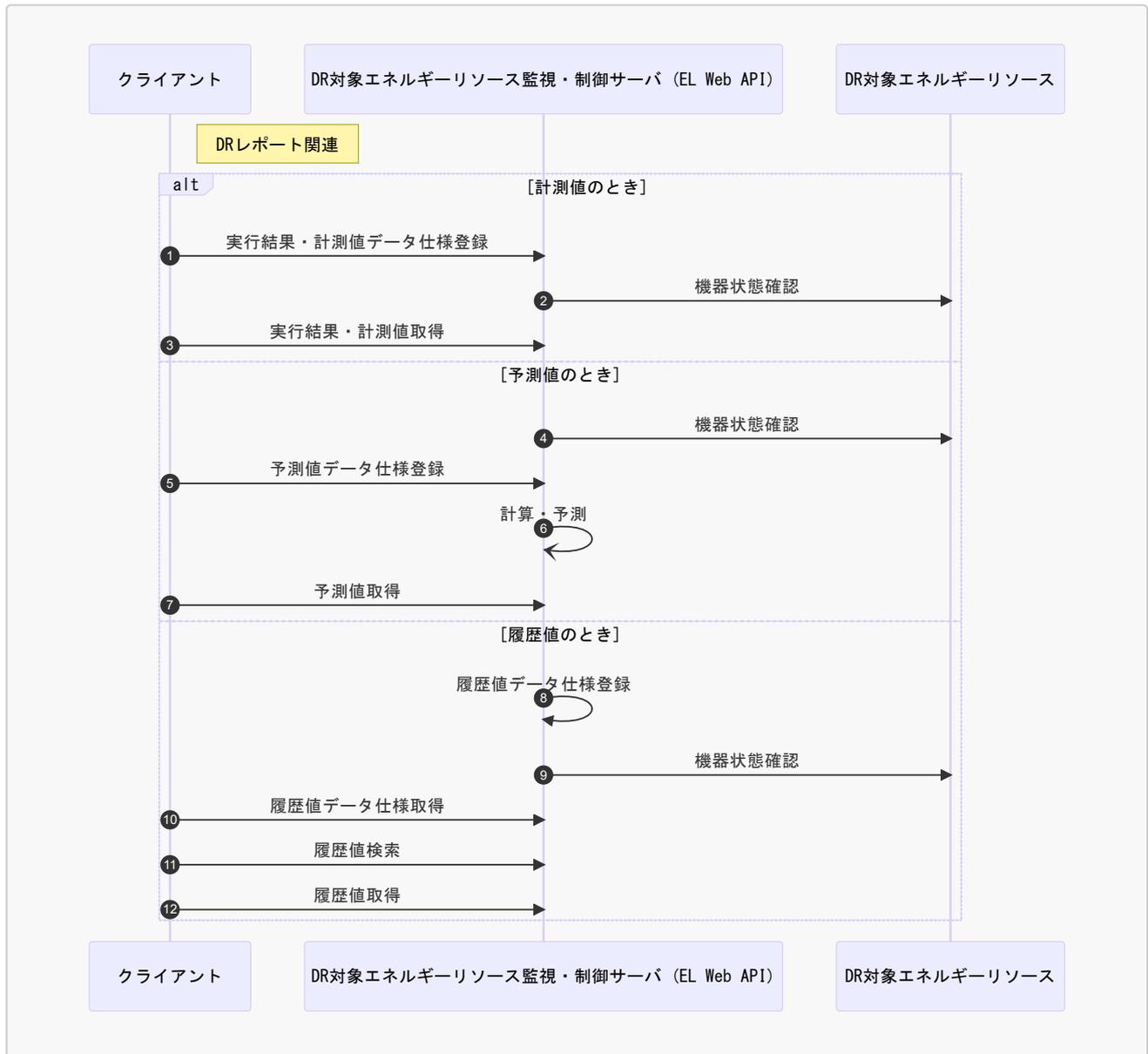


図5-4 DR操作シーケンス (DRレポート)

離脱情報の把握

制御対象となるDR対象エネルギーリソースのうち、一部の機器が故障や需要家ら利用者都合などにより制御対象から外れた際、状況把握できることが望ましい。離脱情報を上位システム（クライアント）にて取得し、なんらかのフィードバック制御につなげることが考えられる。DR対象エネルギーリソースに内包される機器リスト内の各機器IDに紐づく形で各機器の状態確認ができるようにする。

5.2 ACとRA間での想定モデル

DR用ECHONET Lite Web APIを搭載したサーバがRAとなる場合、同サーバはクライアント（AC）の指示にもとづき、各エリア単位に存在するDR対象エネルギーリソースを個別にDR操作することになる。DR対象エネルギーリソースは通常同エリア内にある大量のDER機器を内包した形で表現されるが、個々のDER機器はRAが個別に制御量やタイミングなどを割り振った

上で適切にDR制御される。DR対象エネルギーリソースは、同一エリアであっても、DRサービス種（三次②下げDRなど）が異なれば、別のリソースとして扱われる。



図5-5 AC-RA間制御モデル

5.3 RAと需要家向けサーバ間（HEMSクラウド）での想定モデル

RAは、DR用ECHONET Lite Web APIを搭載したHEMSクラウド（サーバ）を介して、需要家内のDR対象エネルギーリソースを制御する（図5-6）。なお、HEMS Cloud（サーバ）と需要家間のI/Fについては本仕様の対象外とする。

需要家内にあるDR対象エネルギーリソースは、ECHONET Lite対応コントローラーを介するもの（図中需要家A, B）や、介さないもの（図中需要家C）のいずれのケースも対応可能とする。DR対象エネルギーリソース内のDER機器は、HEMSクラウドから監視・制御可能となる。

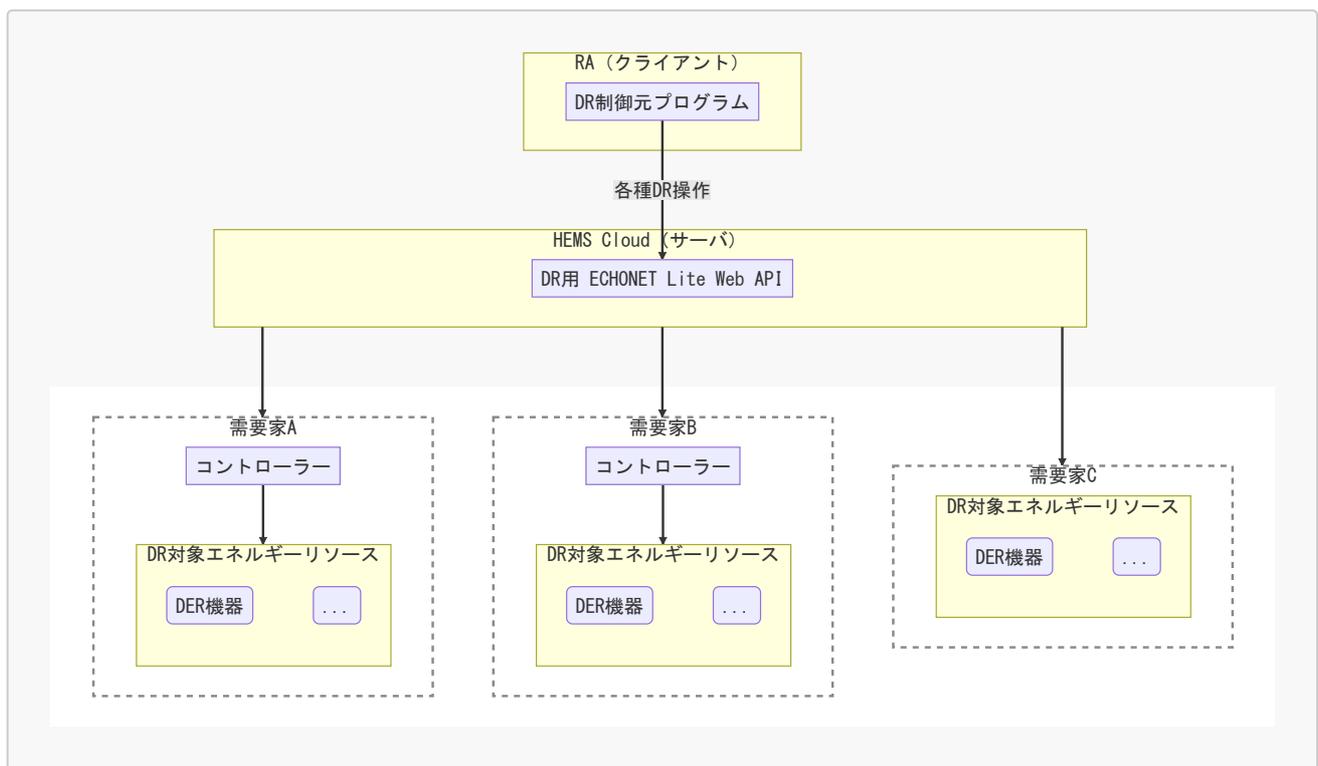


図5-6 RA-HEMSクラウド間制御モデル

DSR-MS

JEMA VPPガイドライン[1]では、RAが、需要家リソースをアグリゲーションビジネスに活用する場合の基本システム構成として、

1. コントローラーを介して単一あるいは複数の分散型エネルギーリソースを直接指定して制御する場合
2. 配下にある分散型エネルギーリソースを自ら制御する機能を有するコントローラーを指定する場合

の2つのうち、2.の構成、すなわち、需要家サイドの機器を制御するコントローラーを制御対象とし、審査内容関連に係る対応を軽減すべく、「需要家リソースをDR制御する機能を具備するEMS等」をDSR-MS (Demand Side Resources- Energy Management System) と定義している。なお、1.の構成においても、DSR-MSがRAからの指示情報をスルーすることで、2.の構成と同様にRAが分散型リソース機器を制御対象とする場合にも対応することが可能としている。

また、HEMSの定義と同様に、DSR-MSのサービス連携機能については、サービス連携機能がサーバに配置される場合と需要家側の機器に配置される場合を想定している。

制御対象機器の選択・機器リスト管理

需要家が保有する制御可能な機器は、DR対象エネルギーリソースに使用できるものと、そうでないものが存在する。RAに対しては、制御可能な機器の中から、電力・電力量指定による制御を実施したい機器を選択・登録・変更できる機器リスト管理機能を提供する。選択した機器リストがDR対象エネルギーリソースとなる。

JEMA VPPガイドライン[1]では、ある需要家が保有する制御可能な機器のうち、電力または電力量の制御が可能なネットワーク接続機器を列挙したものを「制御可能機器リスト」と定義している。また、制御可能機器リストのうち、実際にDR制御対象となる機器を列挙したものを「制御対象機器指定リスト」と定義している。

また、同ガイドラインにおけるDSR-MSは、制御対象機器指定リストを含む制御可能機器リストをひとまとまりのリソース単位として扱うモデルであり、本書では同リソースをDSR-MSリソースと呼ぶ。

さらに、DR対象エネルギーリソースは、DRサービス種（三次②下げDRなど）に応じて、管理が分かれる。概念的には図5-7のように表現される。図では、需要家AはDRサービス種の異なる2つのDR対象エネルギーリソースを保有している。

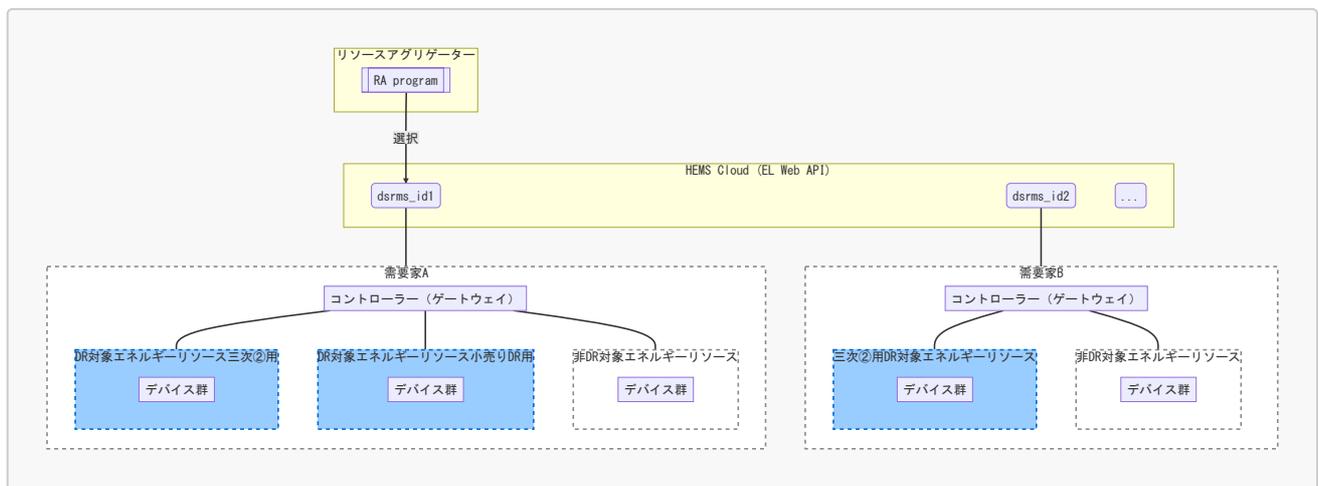


図5-7 DR対象エネルギーリソースと非DR対象エネルギーリソース

DSR-MSとDR対象エネルギーリソースの関係

DSR-MS IDや需要家内の各DR対象エネルギーリソース、DER機器全体にIDを付与した形態を図5-8に示す。

DSR-MS IDは、基本的にある需要家が保有するDR対象エネルギーリソース（「制御対象機器指定リスト」）に加え、非DR対象エネルギーリソースを含めたすべての「制御可能機器リスト」を構成管理できるIDとなる。

図ではコントローラー内に各IDを保持する形となっているが、実際にはHEMS Cloud内に保持するIDであるため、コントローラーを介さない形式であっても良い。

なお、図中、概念的にDR対象エネルギーリソースと区別がしやすいように非DR対象エネルギーリソースはグループIDによってまとめているが、実際のAPI定義では使い勝手を考慮しグループIDではなく、DR対象エネルギーリソースと非DR対象エネルギーリソースとを加えた全デバイスの機器IDリスト（「制御可能機器リスト」）を保持する形式を想定する。

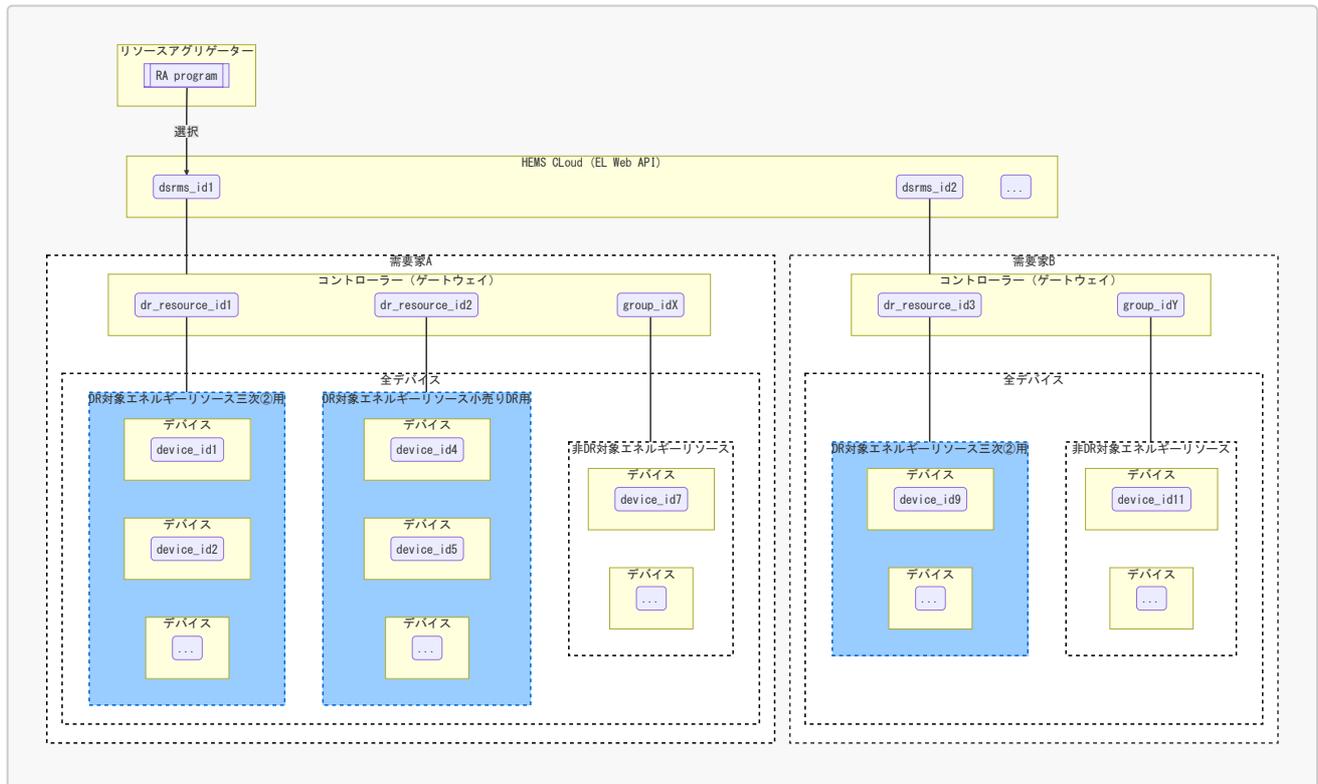


図5-8 DR対象エネルギーリソースとコントローラーの関係

需要家を跨るリソースの扱い

DR対象エネルギーリソースは需要家を跨って、DRサービス種が同一となるリソースをひとつのDR対象エネルギーリソースとして扱っても良い（図5-9）。

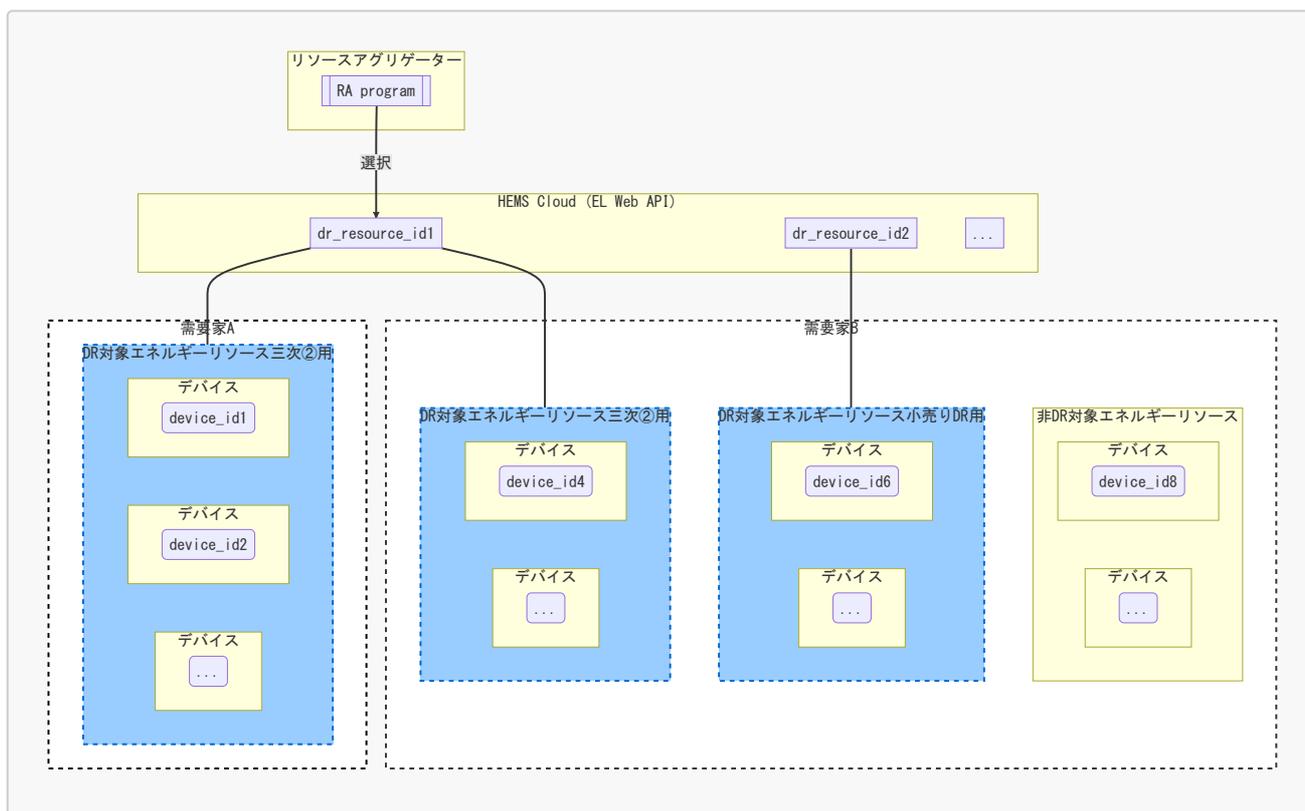


図5-9 需要家を跨るDR対象エネルギーリソース（グループ化）例

6. 詳細仕様

DR対象エネルギーリソース、DRイベント、DRレポートの各サービス種に関するAPI仕様について解説する。

どのサービス種やAPIを必須化／オプション化するかは、サーバが採用するDRモデルに基づき自由に指定可能とする。ただし、各APIで規定されるリクエストおよびレスポンスの各値に関して必須項目については実装されることが望ましい。

6.1 drResources

DR対象エネルギーリソースを扱う。DR対象エネルギーリソースの登録や、DR対象エネルギーリソースの一覧取得、登録済みDR対象エネルギーリソースの内容確認などが可能となる。

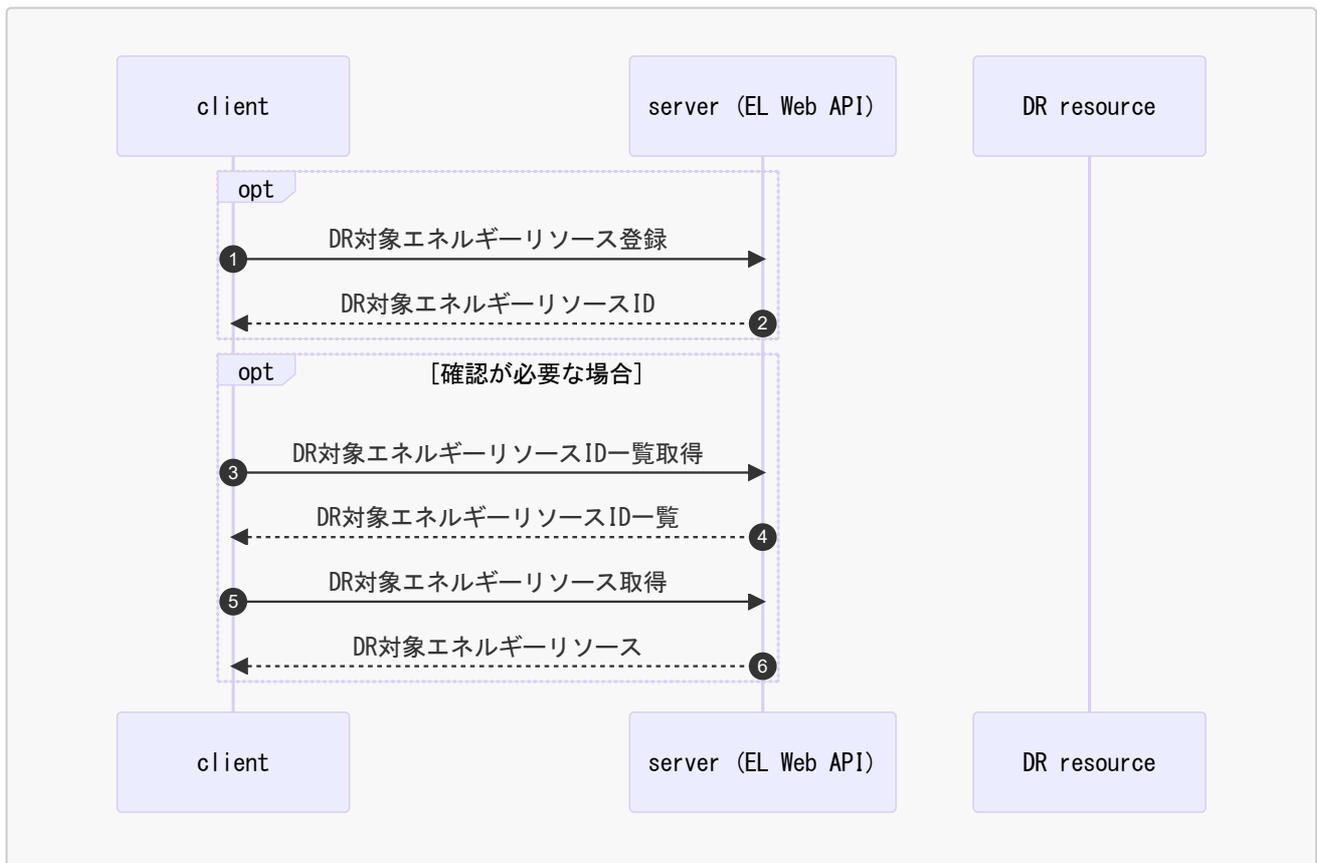


図6-1 DR対象エネルギーリソースの操作

6.1.1 DR対象エネルギーリソース登録 POST /elapi/v1/drResources

対象となるDRサービス、アグリゲーター、エリア、（サブエリア）、DERタイプ、（グループID）を指定してDR対象エネルギーリソースを登録する。登録に成功すると、DR対象エネルギーリソースIDが返却され、以降、このIDを用いてDR対象エネルギーリソースの参照やDRイベント操作などが可能となる。

REQUEST

REQUEST BODY - application/json

Property	Type	Required	Description
descriptions	object	Yes	DR対象エネルギーリソースの名称

Property	Type	Required	Description
descriptions.ja	string	Yes	日本語記述
descriptions.en	string	Yes	英語記述
drService	string	Yes	DRサービス種 。 <ul style="list-style-type: none"> “secondary2DownDr” 二次②下げDR, “secondary2UpDr” 二次②上げDR, “tertiary1DownDr” 三次①下げDR, “tertiary1UpDr” 三次①上げDR, “tertiary2DownDr” 三次②下げDR, “tertiary2UpDr” 三次②上げDR, “powerSupplyDr” 発動指令電源, “marketLinkedDr” 市場価格連動DR, “manualDr” 手動DR のいずれか。
aggregator	string	Yes	上位システムの名称。 任意の文字列
area	string	Yes	管轄電力エリア。 <ul style="list-style-type: none"> “hokkaido” 北海道電力エリア, “tohoku” 東北電力エリア, “tokyo” 東京電力エリア, “chubu” 中部電力エリア, “hokuriku” 北陸電力エリア, “kansai” 関西電力エリア, “chugoku” 中国電力エリア, “shikoku” 四国電力エリア, “kyushu” 九州電力エリア, “okinawa” 沖縄電力エリア, のいずれか。
subArea	string	No	サブエリア。任意の文字列にて指定可能
derType	string	Yes	DER群の種類。 <ul style="list-style-type: none"> “demandGroup” 需要家側リソース群, “storageBatteryGroup” 蓄電池群, 将来、“evChargerDischargerGroup” EV群, “pvPowerGenerationGroup” PV群などサポート予定 のいずれか。
devices	array	No	デバイスIDのリスト

Property	Type	Required	Description
devices[]	string	No	デバイスID

【例】

```
{
  "descriptions": {
    "ja": "低圧リリース群 0001",
    "en": "low-voltage resource group 0001"
  },
  "drService": "manualDr",
  "aggregator": "X_Company_Ra",
  "area": "hokkaido",
  "derType": "demandGroup",
  "devices": ["1", "3", "4"]
}
```

RESPONSE

STATUS CODE - 201:

RESPONSE MODEL - application/json

Property	Type	Required	Description
id	string	Yes	DR対象エネルギーリソースID

【例】

```
{
  "id": "1"
}
```

6.1.2 DR対象エネルギーリソース一覧取得 GET /elapi/v1/drResources

DR対象エネルギーリソースのID一覧を取得する。

REQUEST

No request parameters

RESPONSE

STATUS CODE - 200:

RESPONSE MODEL - application/json

Property	Type	Required	Description
registrationLimit	number	No	登録数制限。サーバ上に登録可能なDR対象エネルギーリソース数の上限。
drResources	array	Yes	DR対象エネルギーリソースのIDリスト。descriptionsを伴う。DR対象エネルギーリソースが登録されていない場合は空（“[]”）を返す。
drResources[].id	string	No	DR リソースID
drResources[].descriptions	object	No	DR対象エネルギーリソースの名称
drResources[].descriptions.ja	string	No	日本語記述
drResources[].descriptions.en	string	No	英語記述

【例】

```
{
  "registrationLimit": 100,
  "drResources": [
    {
      "id": "1",
      "descriptions": {
        "ja": "低圧リソース群 0001",
        "en": "low-voltage resource group 0001"
      }
    },
    {
      "id": "2",
      "descriptions": {
        "ja": "低圧リソース群 0002",
        "en": "low-voltage resource group 0002"
      }
    }
  ]
}
```

6.1.3 DR対象エネルギーリソースDescription取得 GET /elapi/v1/drResources/{drResourceId}

DR対象エネルギーリソースIDを指定して、DR対象エネルギーリソースDescriptionを取得する。

REQUEST

PATH PARAMETERS

Property	Type	Required	Description
drResourceId	string	Yes	DR resource ID

【例】

```
"1"
```

RESPONSE

STATUS CODE - 200:

RESPONSE MODEL - application/json

Property	Type	Required	Description
properties	object	Yes	「6.1.4 DR対象エネルギーリソース内容取得」におけるプロパティリソースに関する定義

【例】

```
{
  "properties": {
    "descriptions": {
      "descriptions": {
        "ja": "DR対象エネルギーリソースの名称 | DR対象エネルギーリソースを特定するための名称",
        "en": "name of the energy resource for Demand Response (DR)"
      },
      "writable": true,
      "observable": false,
      "schema": {
        "type": "object",
        "properties": {
          "ja": {
            "type": "string"
          },
          "en": {
            "type": "string"
          }
        }
      }
    },
    "drService": {
      "descriptions": {
        "ja": "DRサービス種",
        "en": "DR service type"
      },
      "writable": true,
      "observable": false,
      "schema": {
        "type": "string",
        "enum": [
          "secondary2DownDr",
          "secondary2UpDr",
          "tertiary1DownDr",
          "tertiary1UpDr",
          "tertiary2DownDr",
          "tertiary2UpDr",
        ]
      }
    }
  }
}
```

```
        "powerSupplyDr",
        "marketLinkedDr",
        "manualDr"
    ]
  }
},
"aggregator": {
  "descriptions": {
    "ja": "上位システムの名称",
    "en": "name of aggregator"
  },
  "writable": true,
  "observable": false,
  "schema": {
    "type": "string"
  }
},
"area": {
  "descriptions": {
    "ja": "管轄電力エリア",
    "en": "distribution area"
  },
  "writable": true,
  "observable": false,
  "schema": {
    "type": "string",
    "enum": [
      "hokkaido",
      "tohoku",
      "tokyo",
      "chubu",
      "hokuriku",
      "kansai",
      "chugoku",
      "shikoku",
      "kyushu",
      "okinawa"
    ]
  }
},
"subArea": {
  "descriptions": {
    "ja": "サブエリア",
    "en": "sub area"
  },
  "writable": true,
  "observable": false,
  "schema": {
    "type": "string"
  }
},
"derType": {
  "descriptions": {
    "ja": "DER群の種類",
    "en": "type of DER group"
  },
  "writable": true,
  "observable": false,
  "schema": {
    "type": "string",
```

```

        "enum": [
            "demandGroup",
            "storageBatteryGroup"
        ]
    },
    "devices": {
        "descriptions": {
            "ja": "device ID list",
            "en": "device ID list"
        },
        "writable": true,
        "observable": false,
        "schema": {
            "type": "array",
            "items": {
                "type": "string"
            }
        }
    },
    "status": {
        "descriptions": {
            "ja": "各デバイスID対応状態リスト",
            "en": "status list for each device ID"
        },
        "writable": false,
        "observable": false,
        "schema": {
            "type": "array",
            "items": {
                "type": "string",
                "enum": [
                    "active",
                    "inactive"
                ]
            }
        }
    }
}

```

6.1.4 DR対象エネルギーリソース内容取得 GET /elapi/v1/drResources/{drResourceId}/properties

DR対象エネルギーリソースIDを指定して、登録済みのDR対象エネルギーリソース内容を取得する。

REQUEST

PATH PARAMETERS

Property	Type	Required	Description
drResourceId	string	Yes	DR resource ID

【例】

```
"1"
```

RESPONSE

STATUS CODE - 200:

RESPONSE MODEL - application/json

Property	Type	Required	Description
descriptions	object	Yes	DR対象エネルギーリソースの名称
descriptions.ja	string	Yes	日本語記述
descriptions.en	string	Yes	英語記述
drService	string	Yes	DRサービス種。 <ul style="list-style-type: none"> “secondary2DownDr” 二次②下げDR, “secondary2UpDr” 二次②上げDR, “tertiary1DownDr” 三次①下げDR, “tertiary1UpDr” 三次①上げDR, “tertiary2DownDr” 三次②下げDR, “tertiary2UpDr” 三次②上げDR, “powerSupplyDr” 発動指令電源, “marketLinkedDr” 市場価格連動DR, “manualDr” 手動DR のいずれか。
aggregator	string	Yes	上位システムの名称。 任意の文字列
area	string	Yes	管轄電力エリア。 <ul style="list-style-type: none"> “hokkaido” 北海道電力エリア, “tohoku” 東北電力エリア, “tokyo” 東京電力エリア, “chubu” 中部電力エリア, “hokuriku” 北陸電力エリア, “kansai” 関西電力エリア, “chugoku” 中国電力エリア, “shikoku” 四国電力エリア, “kyushu” 九州電力エリア, “okinawa” 沖縄電力エリア, のいずれか。
subArea	string	No	サブエリア。任意の文字列にて指定可能

Property	Type	Required	Description
			DER群の種類。
derType	string	Yes	<ul style="list-style-type: none"> “demandGroup” 需要家側リソース群, “storageBatteryGroup” 蓄電池群, 将来、“evChargerDischargerGroup” EV群, “pvPowerGenerationGroup” PV群などサポート予定 のいずれか。
devices	array	No	デバイスIDのリスト
devices[]	string	No	デバイスID
status	array	No	devices記載の各デバイス対応状態リスト
status[]	string	No	devices記載の各デバイス対応状態。 <ul style="list-style-type: none"> “active” 利用可能, “inactive” 利用不可

【例】

```
{
  "descriptions": {
    "ja": "低圧リソース群 0001",
    "en": "low-voltage resource group 0001"
  },
  "drService": "manualDr",
  "aggregator": "X_Company_Ra",
  "area": "hokkaido",
  "derType": "demandGroup",
  "devices": ["1", "3", "4"],
  "status": ["active", "active", "inactive"]
}
```

離脱機器の確認

本APIを定期的呼び出しDR対象エネルギーリソース内容を確認する、もしくは、下記呼び出しにより定期的に“status”を確認することで、デバイスの離脱状況が把握できる。

```
GET /elapi/v1/drResources/<drResourceId>/properties/status HTTP/1.1
```

さらに、“statusが”inactive”となるデバイスに対しては、デバイスIDを指定して異常内容（faultDescription）プロパティリソースを下記のように指定しGETすることで、より詳細な情報が得られる可能性がある。Appendix Release Rでは、機器の定義によっては、復帰可能な異常（電源入れ直しや電池交換など）や修理が必要な異常（センサー異常、機能部品異常など）といった要因が確認できる（詳細は機器毎の仕様参照）。

```
GET /elapi/v1/devices/<device id>/properties/faultDescription HTTP/1.1
```

離脱機器発生時の通知

また、drResource descriptionにて“status”の“observable”を“true”とした上で、“status”の値が変更された時点で、即座にクライアント側へ通知が発行される仕組みを追加しても良い。具体的には、API仕様部[2]の「5.10 機器オブジェクトのプロパティ値通知（INF）」を参考に、適切な通知機構を選択・実装することで対応可能となる。

6.1.5 DR対象エネルギーリソース内容変更 PUT /elapi/v1/drResources/{drResourceId}/properties/{任意のプロパティリソース}

DR対象エネルギーリソース内の任意のプロパティリソース（“status”を除く）を変更する。変更する値をリクエストボディにて指定し、本APIによって上書きすることで実現できる。

以下にて、“devices”の内容変更を行う事例を示す。

REQUEST

【例】

```
PUT /elapi/v1/devices/<drResourceId>/properties/devices HTTP/1.1
```

PATH PARAMETERS

Property	Type	Required	Description
drResourceId	string	Yes	DR resource ID

【例】

```
"1"
```

REQUEST BODY – application/json

Property	Type	Required	Description
devices	array	Yes	デバイスIDの配列
devices[]	string	Yes	デバイスID

【例】

```
{  
  "devices": [  
    "1", "3", "4", "5"  
  ]  
}
```

```
]
}
```

RESPONSE

STATUS CODE - 200:

RESPONSE MODEL - application/json

Property	Type	Required	Description
devices	array	Yes	デバイスIDの配列
devices[]	string	Yes	デバイスID

【例】

```
{
  "devices": [
    "1", "3", "4", "5"
  ]
}
```

6.2 drEvents

DRイベントを扱う。クライアントはサーバに対して、DRイベントを登録したり、登録したDRイベントに関する応諾判断結果を取得できる。サーバが応諾可能か判断するために一定の処理時間を要することも想定し、DR応諾結果は、クライアント側から別途DR応諾結果取得リクエストを送信することにより取得するモデルとなっている。サーバは応諾可能な場合、DRイベント内にて指定された開始時刻になると、指定区間・制御量に応じて対象DR対象エネルギーリソースへのDR制御を順次開始する。

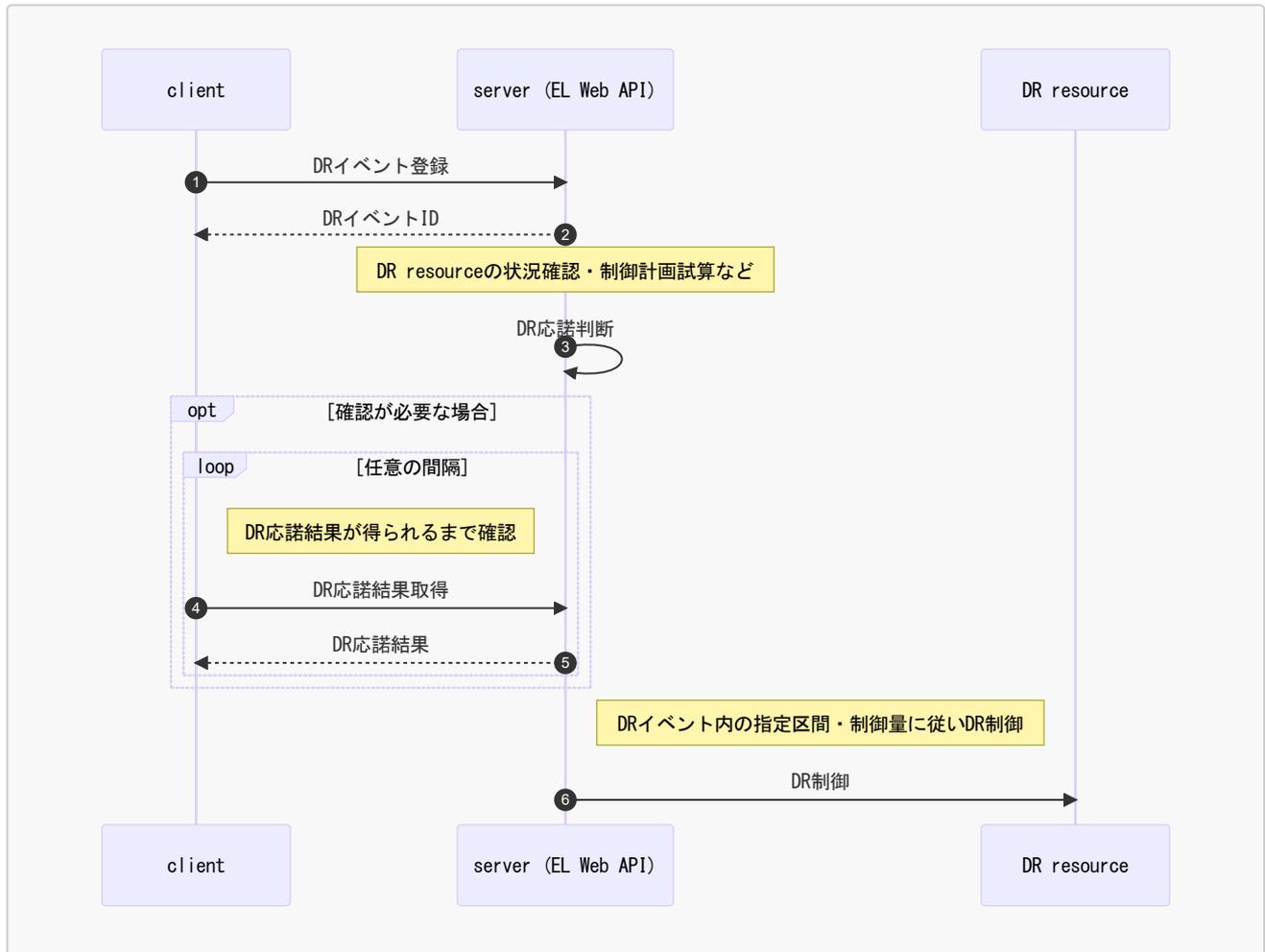


図6-2 DRイベントの操作

6.2.1 DRイベント登録 POST /elapi/v1/drEvents

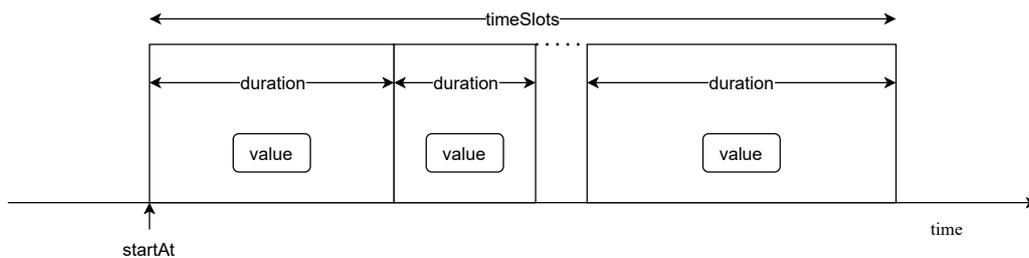


図6-3 DRイベントの登録内容

DR対象エネルギーリソースID (drResourceId) に対するDRイベントを登録する。クライアントはサーバに対して、DR開始時刻 (startAt) からタイムスロット (timeSlots) で列挙される区間 (duration) ・制御量 (value) の組に従い、DR対象エネルギーリソースへのDR制御を実施するよう、同内容を内包するDRイベントの登録を依頼する。DRイベントでは、イベントタイプ (eventType) によって所望のDR制御方法を指定することができる (詳しくは、各DRイベントタイプの説明にて記載)。サーバ上にてDRイベントの登録に成功すると、DRイベントIDがクライアントへ返却される。なお、DRイベントが実際にサーバ側で応諾されるか否かは、この時点で確定している必要はない (サーバがこのDRイベント内容に対して応諾可能か判断を終えるまでにある程度の処理時間がかかることを想定している)。登録したDRイベントの内容は、クライアント側から任意の項目について後述のPATCHメソッドにより更新可能である。ただし、DRイベント登録時はリビジョン番号に0以上の整数を指定し、更新時にはリビジョン番号を1増加した値を指定する必要がある。また、DRイベント登録時のオプション項目として、クライアント側にてDRイベントの配信決定時刻 (distributedAt) を指定することができる。この時刻以降、DRイベントの内容変更を行わないといった意向を伝えることに利用できる。DRイベントIDに対して、後述するgetOptsアクションをPOSTすることで、サーバのDR応諾判断結果を取得することができる (オプション)。DRイベント登録が成功すると、サーバはDRイベントのstatusを"activating" (応諾判断中) に設定する。このstatusは、後述の「DRイベント一覧取得」にて取得できる (オプション)。さらに、DRイベント登録時のオプション項目として、DR実施後にDR対象エネルギーリソースのDR実施前の状態に復帰させるか否かを示すフラグ (restoreMode) を指定することができる。デフォルト (省略時) は"false"である。"true"指定時には、サーバはDR実施前にDR対象エネルギーリソース内の各機器の状態を記憶し、DR実施後に同状態へ復帰させる処理を実施する。サーバが実施する具体的な状態記憶・復帰手順については省略する。

前述の通り、サーバはDRイベント登録の応答の時点で応諾判断が完了している必要はないが、応諾判断できる場合、適切なstatus codeと共にAPI仕様部[2]規定の"type", "message"によるエラーメッセージを返却しても良い。

DR対象エネルギーリソースタイプ (derType) によって選択可能なDRイベントタイプ (eventType) は下記の通りとなる。

DR対象エネルギーリソースタイプ (derType) がdemandGroupの場合 :

name	eventType	Unit	description
負荷増減制御 (Δ kWh制御)	deltaLoadControl	kWh	基準値ある場合、基準値との差分を指示。基準値ない場合、現在値との差分を指示。正值：負荷削減 (下げDR)。負値：負荷増加 (上げDR)。
直接負荷制御 (kW制御)	directLoadControl	kW	負荷 (電力) の絶対値を指示。正值のみ
直接負荷制御 (kWh制御)	directLoadControl	kWh	負荷 (電力量) の絶対値を指示。正值のみ

deltaLoadControlは負荷を調節するイベントのため、負荷を抑制する方向 (下げDR) は正值、負荷を増大する方向 (上げDR) は負値とする。

DR対象エネルギーリソースタイプ (derType) がstorageBatteryGroupの場合：

name	eventType	Unit	description
充放電制御 (一括)	chargeState	kW	蓄電池群に所属の蓄電池を一括充放電電力を指示する。正值：充電電力。 負値：放電電力
充放電制御 (一括)	chargeState	kWh	蓄電池群に所属の蓄電池を一括充放電の電量（現在の残量より差分）を指示する。正值：充電量。負値：放電量
充放電制御 (一括)	chargeState	%	蓄電池群に所属の蓄電池を一括SOC（全容量の割合）指示する、正值のみ 範囲：0 - 100

chargeStateは蓄電池の蓄電残量を調節するイベントのため、蓄電残量を増やす方向（充電）は正值、減らす方向（放電）は負値とする

REQUEST

REQUEST BODY - application/json

Property	Type	Required	Description
descriptions	object	Yes	DRイベント名称。DRイベントを特定するための名称
descriptions.ja	string	Yes	日本語記述
descriptions.en	string	Yes	英語記述
revision	number	Yes	リビジョン番号。初期値は0以上の整数を指定する。PATCHによる更新時には1増加した値を指定する。
distributedAt	string	No	DRイベントの配信決定時刻（RFC 3339（ISO 8601）形式）。
drResourceId	string	Yes	DR対象エネルギーリソースID

DRイベントタイプ。

【DR対象エネルギーリソースのderTypeが"demandGroup"の場合（下記のいずれか）】

- "deltaLoadControl" 負荷増減制御（ΔkW制御またはΔkWh制御）：単位kWまたはkWh,
- "directLoadControl" 直接負荷制御（kW制御またはkWh制御）：単位kWまたはkWh

【DR対象エネルギーリソースのderTypeが"storageBatteryGroup"の場合】

- "chargeState" 充放電制御（一括）：単位kWhまたはkW また%,

【他のderTypeについては現状未定義】

Property	Type	Required	Description
startAt	string	Yes	DRイベントの開始時刻（RFC 3339（ISO 8601）形式）。 timeSlots内の最初の区間における開始時刻に該当する。
durationUnit	string	Yes	timeSlots内要素である区間（duration）の時間単位。“hour” 時間, “minute” 分, “second” 秒のいずれか。
valueUnit	string	Yes	timeSlots内要素である制御量（value）の単位。“kW” kW, “kWh” kWh, “%” %のいずれか。
timeSlots	array	Yes	DRイベントの期間。複数の区間を指定できる。開始時刻（startAt）からの期間区切りにてスロットを記述する。複数区間を指定する場合、連続した区間となる点に注意。必ず1つ以上の区間を指定すること。
timeSlots[].duration	number	Yes	区間。正の整数値を指定すること。単位は、durationUnitにて指定される。
timeSlots[].value	number	Yes	制御量。ゼロを含む正負の整数値や実数値の指定が可能。単位は、valueUnitにて指定される。たとえば、eventTypeが“deltaLoadControl”の場合、正値は上げDR、負値は下げDRとなる。
restoreMode	boolean	No	復帰モード。“true”復帰する、“false”復帰しない。“true”指定時は、DR制御実施前にDR対象エネルギーリソースの状態を保存し、DR制御実施後にDR実施前と同等な状態へ復帰させる。省略時は“false”。

【例】

```
{
  "descriptions": {
    "ja": "下げDRイベント1",
    "en": "DownDR Event 1"
  },
  "revision": 0,
  "distributedAt": "2023-07-01T17:45:00+09:00",
  "drResourceId": "1",
  "eventType": "deltaLoadControl",
  "startAt": "2023-07-01T18:00:00+09:00",
  "durationUnit": "minute",
  "valueUnit": "kW",
  "timeSlots": [
    {
      "duration": 120,
      "value": 100
    },
    {
      "duration": 60,
      "value": 50
    }
  ]
}
```

```
"restoreMode": "true"
}
```

RESPONSE

STATUS CODE - 201:

RESPONSE MODEL - application/json

Property	Type	Required	Description
id	string	Yes	DRイベントID

【例】

```
{
  "id": "1"
}
```

6.2.2 DRイベント一覧取得 GET /elapi/v1/drEvents

DRイベントのID一覧を取得する。IDの他に、DRイベントのリビジョン番号、DRイベントの状態などが配列形式にて返却される。

REQUEST

No request parameters

RESPONSE

STATUS CODE - 200:

RESPONSE MODEL - application/json

Property	Type	Required	Description
registrationLimit	number	No	サーバ上でのDRイベント登録上限数
drEvents	array	Yes	DRイベント・リスト
drEvents[].id	string	Yes	DRイベントID
drEvents[].revision	number	Yes	DRイベントのリビジョン番号
drEvents[].status	string	Yes	DRイベントの状態。“activating” 応諾判断中, “activated” 応諾判断完（有効）, “aborted” 中断の いずれか。
drEvents[].descriptions	object	Yes	DRイベントの名称
drEvents[].descriptions.ja	string	Yes	日本語記述

Property	Type	Required	Description
drEvents[].descriptions.en	string	Yes	英語記述

【例】

```
{
  "registrationLimit": 100,
  "drEvents": [
    {
      "id": "1",
      "revision": 0,
      "status": "activating",
      "descriptions": {
        "ja": "下げDRイベント1",
        "en": "DownDR Event 1"
      }
    },
    {
      "id": "2",
      "revision": 0,
      "status": "activated",
      "descriptions": {
        "ja": "下げDRイベント2",
        "en": "DownDR Event 2"
      }
    }
  ]
}
```

6.2.3 DRイベントDescription取得 GET /elapi/v1/drEvents/{drEventId}

DRイベントIDを指定して、DRイベントDescriptionを取得する。

REQUEST

PATH PARAMETERS

Property	Type	Required	Description
drEventId	string	Yes	DR event ID

【例】

"1"

RESPONSE

STATUS CODE - 200:

RESPONSE MODEL - application/json

Property	Type	Required	Description
properties	object	Yes	「6.2.4 DRイベント内容取得」におけるプロパティリソースに関する定義
actions	object	Yes	「6.2.6 DRイベント応諾結果取得」、「6.2.7 DRイベント中止」、「6.2.8 DRイベント削除」におけるアクションリソースに関する定義

【例】

```

{
  "properties": {
    "descriptions": {
      "descriptions": {
        "ja": "DRイベント名称",
        "en": "name of a DR event"
      },
      "writable": true,
      "observable": false,
      "schema": {
        "type": "object",
        "properties": {
          "ja": {
            "type": "string"
          },
          "en": {
            "type": "string"
          }
        }
      }
    },
    "revision": {
      "descriptions": {
        "ja": "リビジョン番号",
        "en": "revision number"
      },
      "writable": true,
      "observable": false,
      "schema": {
        "type": "number",
        "minimum": 0,
        "multipleOf": 1
      }
    },
    "distributedAt": {
      "descriptions": {
        "ja": "DRイベントの配信決定時刻 (RFC 3339 (ISO 8601)形式)",
        "en": "delivery decision time for a DR event (RFC 3339 (ISO 8601) format)"
      },
      "writable": true,
      "observable": false,
      "schema": {
        "type": "string"
      }
    },
    "drResourceId": {
      "descriptions": {
        "ja": "DR対象エネルギーリソースID",

```

```
    "en": "DR target energy resource ID"
  },
  "writable": true,
  "observable": false,
  "schema": {
    "type": "string"
  }
},
"eventType": {
  "descriptions": {
    "ja": "DRイベントタイプ",
    "en": "type of DR event"
  },
  "writable": true,
  "observable": false,
  "schema": {
    "type": "string"
  }
},
"startAt": {
  "descriptions": {
    "ja": "DRイベントの開始時刻 (RFC 3339 (ISO 8601)形式)",
    "en": "start time of a DR event (RFC 3339 (ISO 8601) format)"
  },
  "writable": true,
  "observable": false,
  "schema": {
    "type": "string",
    "format": "date-time"
  }
},
"durationUnit": {
  "descriptions": {
    "ja": "timeSlots内要素である区間 (duration) の時間単位",
    "en": "unit of the time (duration) that is an element in timeSlots"
  },
  "writable": true,
  "observable": false,
  "schema": {
    "type": "string",
    "enum": [
      "hour",
      "minute",
      "second"
    ]
  }
},
"valueUnit": {
  "descriptions": {
    "ja": "timeSlots内要素である制御量 (value) の単位",
    "en": "unit of the control amount (value) that is an element in timeSlots"
  },
  "writable": true,
  "observable": false,
  "schema": {
    "type": "string",
    "enum": [
      "kW",
      "kWh",
      "%"
    ]
  }
}
```

```
    ]
  }
},
"timeSlots": {
  "descriptions": {
    "ja": "DRイベントの期間",
    "en": "time periods in a DR event"
  },
  "writable": true,
  "observable": false,
  "schema": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "duration": {
          "type": "number",
          "minimum": 0,
          "multipleOf": 1
        },
        "value": {
          "type": "number"
        }
      }
    }
  }
},
"restoreMode": {
  "descriptions": {
    "ja": "復帰モード",
    "en": "restore mode"
  },
  "writable": true,
  "observable": false,
  "schema": {
    "type": "boolean",
  }
},
"actions": {
  "getOpts": {
    "descriptions": {
      "ja": "DRイベント名称",
      "en": "name of a DR event"
    },
    "input": {
      "type": "object",
      "properties": {
        "revision": {
          "type": "number",
          "minimum": 0,
          "multipleOf": 1
        }
      }
    },
    "schema": {
      "type": "object",
      "properties": {
        "responseAt": {
          "type": "string",

```

```
    "format": "date-time"  
  },  
  "opts": {  
    "type": "string",  
    "enum": [  
      "optIn",  
      "optOut"  
    ]  
  }  
}  
}  
}  
}
```

6.2.4 DRイベント内容取得 GET /elapi/v1/drEvents/{drEventId}/properties

DRイベントIDを指定して、登録済みのDRイベント内容を取得する。

REQUEST

PATH PARAMETERS

Property	Type	Required	Description
drEventId	string	Yes	DR event ID

【例】

```
"1"
```

RESPONSE

STATUS CODE - 200:

RESPONSE MODEL - application/json

Property	Type	Required	Description
descriptions	object	Yes	DRイベント名称。DRイベントを特定するための名称
descriptions.ja	string	Yes	日本語記述
descriptions.en	string	Yes	英語記述
revision	number	Yes	リビジョン番号 (0以上の整数)
distributedAt	string	No	DRイベントの配信決定時刻 (RFC 3339 (ISO 8601)形式)。
drResourceId	string	Yes	DR対象エネルギーリソースID

Property	Type	Required	Description
eventType	string	Yes	<p>DRイベントタイプ。</p> <p>【DR対象エネルギーリソースのderTypeが“demandGroup”の場合（下記のいずれか）】</p> <ul style="list-style-type: none"> “deltaLoadControl” 負荷増減制御（ΔkW制御またはΔkWh制御）：単位kWまたはkWh, “directLoadControl” 直接負荷制御（kW制御またはkWh制御）：単位kWまたはkWh <p>【DR対象エネルギーリソースのderTypeが“storageBatteryGroup”の場合】</p> <ul style="list-style-type: none"> “chargeState” 充放電制御（一括）：単位kWhまたはkWまたは%. <p>【他のderTypeについては現状未定義】</p>
startAt	string	Yes	DRイベントの開始時刻（RFC 3339（ISO 8601）形式）。timeSlots内の最初の区間における開始時刻に該当する。
durationUnit	string	Yes	timeSlots内要素である区間（duration）の時間単位。“hour” 時間, “minute” 分, “second” 秒のいずれか。
valueUnit	string	Yes	timeSlots内要素である制御量（value）の単位。“kW” kW, “kWh” kWh, “%” %のいずれか。
timeSlots	array	Yes	DRイベントの期間。複数の区間を指定できる。開始時刻（startAt）からの期間区切りにてスロットを記述する。複数区間を指定する場合、連続した区間となる点に注意。必ず1つ以上の区間を指定すること。
timeSlots[].duration	number	Yes	区間。正の整数値を指定すること。単位は、durationUnitにて指定される。
timeSlots[].value	number	Yes	制御量。ゼロを含む正負の整数値や実数値の指定が可能。単位は、valueUnitにて指定される。たとえば、eventTypeが“deltaLoadControl”の場合、正値は上げDR、負値は下げDRとなる。
restoreMode	boolean	No	復帰モード。“true”復帰する、“false”復帰しない。“true”指定時は、DR制御実施前にDR対象エネルギーリソースの状態を保存し、DR制御実施後にDR実施前と同等な状態へ復帰させる。省略時は“false”。

【例】

```
{
  "descriptions": {
    "ja": "下げDRイベント1",
    "en": "DownDR Event 1"
  },
  "revision": 0,
  "distributedAt": "2023-07-01T17:45:00+09:00",
  "drResourceId": "1",
  "eventType": "deltaLoadControl",
  "startAt": "2023-07-01T18:00:00+09:00",
  "durationUnit": "minute",
  "valueUnit": "kW",
  "timeSlots": [
    {
      "duration": 120,
      "value": 100
    },
    {
      "duration": 60,
      "value": 50
    }
  ],
  "restoreMode": "true"
}
```

6.2.5 DRイベント内容変更 PATCH /elapi/v1/drEvents/{drEventId}/properties

DRイベントの内容を更新する。propertiesで取得できる任意のプロパティ（一部または全部）を変更可能だが、必ずリビジョン番号を指定し、1増加させた値とする。典型的には、startAtやtimeSlotsの内容を変更するケースが多いと考えられる。

REQUEST

PATH PARAMETERS

Property	Type	Required	Description
drEventId	string	Yes	DR event ID

【例】

```
"1"
```

REQUEST BODY – application/json

Property	Type	Required	Description
descriptions	object	No	DRイベント名称。DRイベントを特定するための名称
descriptions.ja	string	No	日本語記述
descriptions.en	string	No	英語記述
revision	number	Yes	リビジョン番号（0以上の整数）
distributedAt	string	No	DRイベントの配信決定時刻（RFC 3339（ISO 8601）形式）。
drResourceId	string	No	DR対象エネルギーリソースID
eventType	string	No	DRイベントタイプ。 【DR対象エネルギーリソースのderTypeが“demandGroup”の場合（下記のいずれか）】 <ul style="list-style-type: none">“deltaLoadControl” 負荷増減制御（ΔkW制御またはΔkWh制御）：単位kWまたはkWh,“directLoadControl” 直接負荷制御（kW制御またはkWh制御）：単位kWまたはkWh 【DR対象エネルギーリソースのderTypeが“storageBatteryGroup”の場合】 <ul style="list-style-type: none">“chargeState” 充放電制御（一括）：単位kWhまたはkWまたは%。 【他のderTypeについては現状未定義】
startAt	string	No	DRイベントの開始時刻（RFC 3339（ISO 8601）形式）。timeSlots内の最初の区間における開始時刻に該当する。
durationUnit	string	No	timeSlots内要素である区間（duration）の時間単位。“hour” 時間, “minute” 分, “second” 秒のいずれか。
valueUnit	string	No	timeSlots内要素である制御量（value）の単位。“kW” kW, “kWh” kWh, “%” %のいずれか。
timeSlots	array	No	DRイベントの期間。複数の区間を指定できる。開始時刻（startAt）からの期間区切りにてスロットを記述する。複数区間を指定する場合、連続した区間となる点に注意。必ず1つ以上の区間を指定すること。
timeSlots[].duration	number	No	区間。正の整数値を指定すること。単位は、durationUnitにて指定される。

Property	Type	Required	Description
timeSlots[].value	number	No	制御量。ゼロを含む正負の整数値や実数値の指定が可能。単位は、valueUnitにて指定される。たとえば、eventTypeが"deltaLoadControl"の場合、正値は上げDR、負値は下げDRとなる。

【例】

```
{
  "revision": 1,
  "timeSlots": [
    {
      "duration": 60,
      "value": 100
    },
    {
      "duration": 60,
      "value": 50
    },
    {
      "duration": 60,
      "value": 50
    }
  ]
}
```

RESPONSE

STATUS CODE - 200:

RESPONSE MODEL - application/json

Property	Type	Required	Description
descriptions	object	No	DRイベント名称。DRイベントを特定するための名称
descriptions.ja	string	No	日本語記述
descriptions.en	string	No	英語記述
revision	number	Yes	リビジョン番号 (0以上の整数)
distributedAt	string	No	DRイベントの配信決定時刻 (RFC 3339 (ISO 8601)形式)。
drResourceId	string	No	DR対象エネルギーリソースID

Property	Type	Required	Description
eventType	string	No	<p>DRイベントタイプ。</p> <p>【DR対象エネルギーリソースのderTypeが“demandGroup”の場合（下記のいずれか）】</p> <ul style="list-style-type: none"> “deltaLoadControl” 負荷増減制御（Δ kW制御またはΔ kWh制御）：単位kWまたはkWh, “directLoadControl” 直接負荷制御（kW制御またはkWh制御）：単位kWまたはkWh <p>【DR対象エネルギーリソースのderTypeが“storageBatteryGroup”の場合】</p> <ul style="list-style-type: none"> “chargeState” 充放電制御（一括）：単位kWhまたはkWまたは%. <p>【他のderTypeについては現状未定義】</p>
startAt	string	No	DRイベントの開始時刻（RFC 3339（ISO 8601）形式）。timeSlots内の最初の区間の開始時刻に該当する。
durationUnit	string	No	timeSlots内要素である区間（duration）の時間単位。“hour” 時間, “minute” 分, “second” 秒のいずれか。
valueUnit	string	No	timeSlots内要素である制御量（value）の単位。“kW” kW, “kWh” kWh, “%” %のいずれか。
timeSlots	array	No	DRイベントの期間。複数の区間を指定できる。開始時刻（startAt）からの期間区切りにてスロットを記述する。複数区間を指定する場合、連続した区間となる点に注意。必ず1つ以上の区間を指定すること。
timeSlots[].duration	number	No	区間。正の整数値を指定すること。単位は、durationUnitにて指定される。
timeSlots[].value	number	No	制御量。ゼロを含む正負の整数値や実数値の指定が可能。単位は、valueUnitにて指定される。たとえば、eventTypeが“deltaLoadControl”の場合、正値は上げDR、負値は下げDRとなる。

【例】

```

{
  "revision": 1,
  "timeSlots": [
    {
      "duration": 60,
      "value": 100
    },
    {

```

```

    "duration": 60,
    "value": 50
  },
  {
    "duration": 60,
    "value": 50
  }
]
}

```

6.2.6 DRイベント応諾結果取得 POST /elapi/v1/drEvents/{drEventId}/actions/getOpts

DRイベント（リビジョン番号指定）の登録または更新に対する応諾判断結果を取得する。具体的には、サーバの応諾時刻、区間毎の応諾結果を取得する。サーバにおいて応諾判断結果がまだ得られていない場合には、応答において responseAt が無く、opts が空のリスト（“[]”）となる。

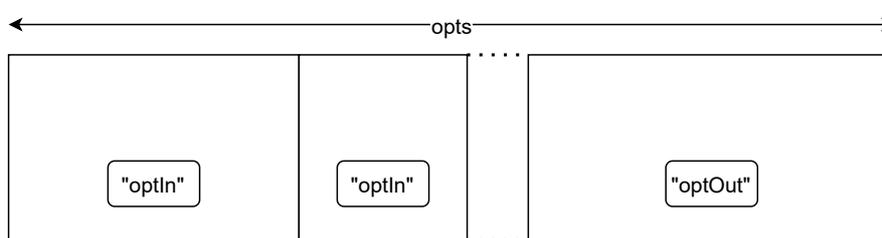


図6-4 区間毎の応諾結果例

REQUEST

PATH PARAMETERS

Property	Type	Required	Description
drEventId	string	Yes	DR event ID

【例】

```
"1"
```

REQUEST BODY – application/json

Property	Type	Required	Description
revision	number	Yes	リビジョン番号（0以上の整数）

【例】

```

{
  "revision": 0
}

```

RESPONSE

STATUS CODE - 201:

RESPONSE MODEL - application/json

Property	Type	Required	Description
responseAt	string	No	応答時刻 (RFC 3339 (ISO 8601)形式)。応諾判断中は無し。
opts	array	Yes	応諾結果の組。
opts[]	string	No	応諾結果。“optIn” 参加, “optOut” 不参加のいずれか。応諾判断中は空。)

【例】

```
{
  "responseAt": "2023-07-01T17:50:00+09:00",
  "opts": [
    "optIn",
    "optIn"
  ]
}
```

6.2.7 DRイベント中止 POST /elapi/v1/drEvents/{drEventId}/actions/abort

DRイベントIDに対応するDRイベントを中止する。成功すると、サーバはDRイベントのstatusを“aborted”に変更し、空のボディとstatus code 201が返却される。通常、statusが“activating”, “activated”の場合に呼び出されるが、すでに“aborted”の場合は失敗する。

REQUEST

PATH PARAMETERS

Property	Type	Required	Description
drEventId	string	Yes	DR event ID

【例】

```
"1"
```

RESPONSE

STATUS CODE - 201:

No response body

6.2.8 DRイベント削除 DELETE /elapi/v1/drEvents/{drEventId}

DRイベントIDに対応するDRイベントを削除する。成功すると、空のボディとstatus code 204が返却される。

REQUEST

PATH PARAMETERS

Property	Type	Required	Description
drEventId	string	Yes	DR event ID

【例】

```
"1"
```

RESPONSE

STATUS CODE - 204:

No response body

6.3 drReports

DRレポートは、DR対象エネルギーリソースに関する計測値・予測値・履歴値のいずれかを含む時系列データである。
drReportsを用いたAPIでは、計測値と予測値について扱う。drReportsを対象に、クライアントはサーバに対してDRレポート仕様を登録したり、その登録内容に基づきサーバ上へDRレポートを生成させ、これを取得することができる。履歴値については、大量のデータを扱う可能性があるため、非同期処理により取得するものとし、API仕様部[2]記載のresHistoriesを用いたAPIにて扱う。履歴値については、6.4にて説明する。

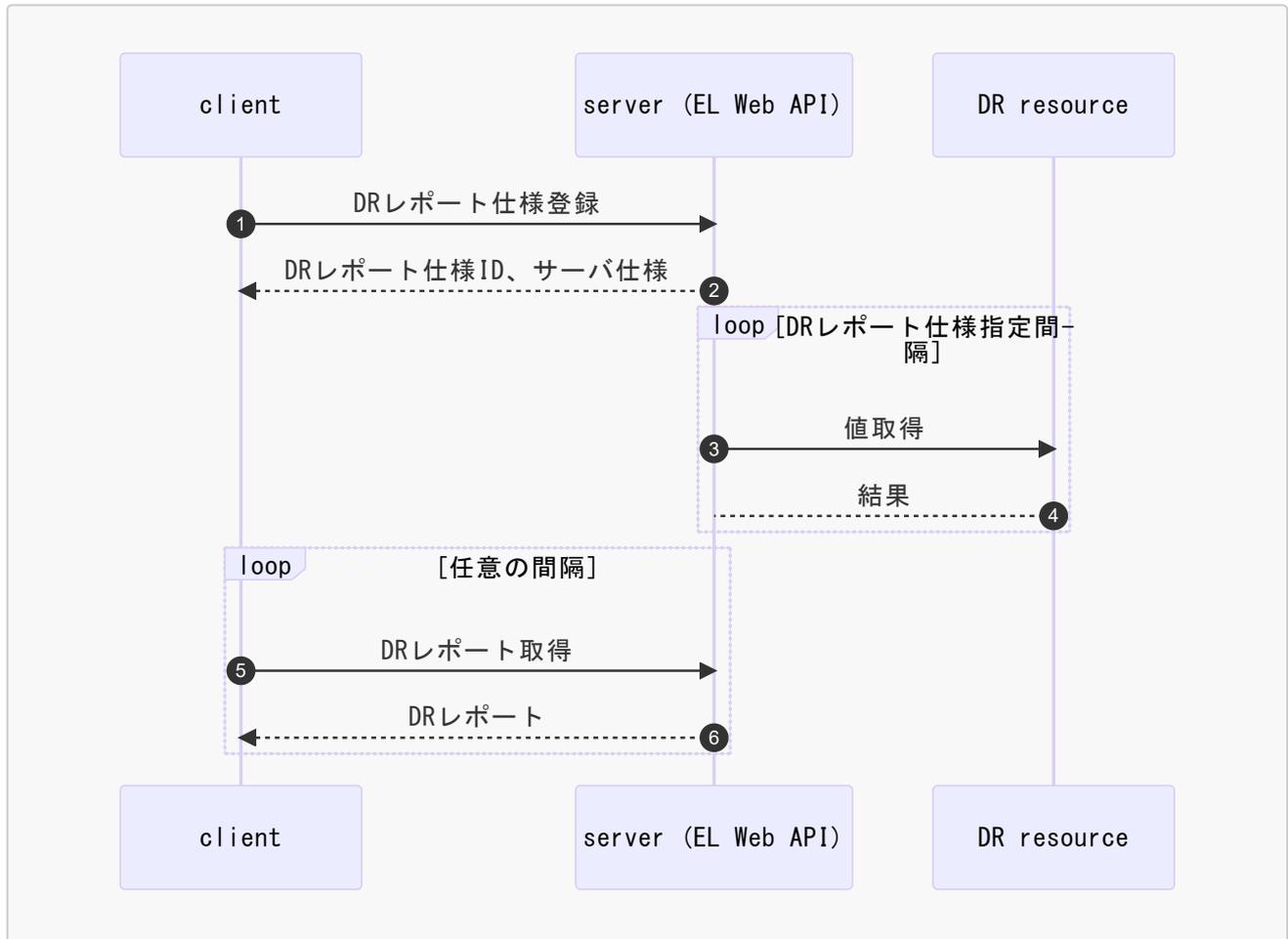


図6-5 DRレポートリソースの操作

6.3.1 DRレポート仕様登録 POST /elapi/v1/drReports

DR対象エネルギーリソースに対するDRレポートの仕様をサーバへ登録する。クライアントは、サーバに対してDR対象エネルギーリソースIDであるdrResourceIdに関するDRレポートのタイプ (type) (計測値もしくは予測値) や時間粒度 (granularity)、計測・予測する値の種類 (valueKind) を指定し、レポートを用意することを依頼・登録する。また、その際にクライアントは最大遅延時間 (maxDelayTime) も指定することができ、クライアントが別途レポート取得を要求した際にサーバからの応答がこの最大遅延時間以内に返却されることを要求できる (option)。指定したDRレポート仕様の登録に成功すると、DRレポート仕様ID (drReportId) を含む応答が返却される。この応答に含まれるstartAt以降のDRレポートに関して、同IDを用いた各種操作が可能となる。

計測値の扱い

図6-6、図6-7に計測値を扱う際の各値の関係について示す。いずれも、サーバが定期的にDR対象エネルギーリソースの各種計測値を取得し、クライアントから後述する`getValues`アクションを用いて（`t1`から`t3`、および、`t4`から`t6`を指定して）各々3つの計測値（DRレポート）を取得するケースを示している。各図では、サーバが定期的にDR対象エネルギーリソース（DR resource）に対して計測値を取得することを示す双方向の矢印を記載しているが、DR対象エネルギーリソース側からあらかじめ指定していた時刻・間隔にて値を通知する形式であっても良い（実装の形式は問わない）。

サーバは、本APIの呼び出しが成功すると、`startAt`を計測開始時刻として、`at (=tn)`の各時刻におけるDR対象エネルギーリソースの各種計測値`values`を記録する。`granularity`はDR対象エネルギーリソースの対象となる各値を計測する時間粒度（計測間隔）であり、`interval`はサーバのAPI層に同値が保存されてから次の値が保存されるまでの間隔となる。この`granularity`と`interval`が同一となる場合と、異なる場合のそれぞれについて図6-6、図6-7に例示している。

クライアントは、クライアントが`getValues`を発行してからDRレポートをサーバより受信するまでの許容可能な最大遅延時間`maxDelayTime`を指定することができる（Option）。サーバはクライアントの期待する`maxDelayTime`までの間に応答できない場合は、エラーを返す。また、クライアントによる`getValues`発行の間隔がサーバの想定する間隔よりも短い場合に応答を返せない可能性があるため、クライアントへ最短DRレポート取得間隔`minTransmissionInterval`を応答する。

図6-6は、`granularity`と`interval`が同一のケースを示している。実際には、DR対象エネルギーリソースから値を取得するまでの遅延時間があるため、時刻`t1`後にクライアントが`getValues`アクションを実行した場合でも、サーバのAPI層に値が保存されていない間は、空（`[]`）が返る可能性のあることに留意すること。

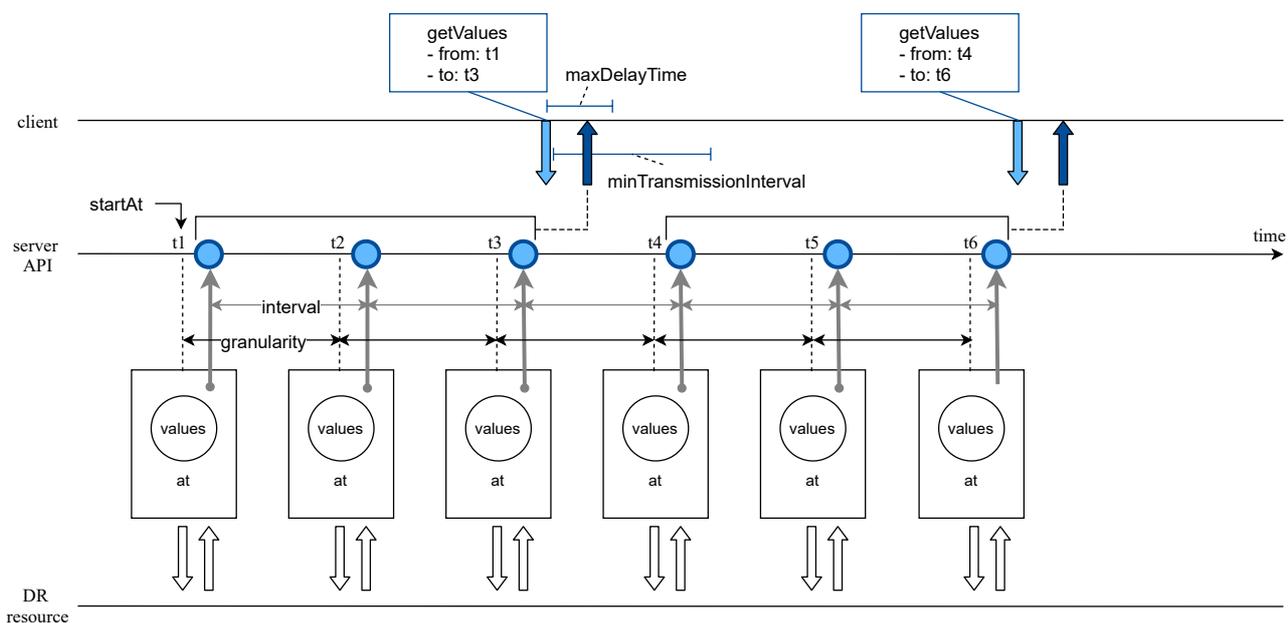


図6-6 計測値（type=measure）の例：granularityとintervalが同一ケース

図6-7は、3つの計測値をまとめてサーバのAPI層へ一斉に保存するケースを示している。この場合、`interval`は`granularity`とは一致せず、3つの計測値をサーバに保存し次の3つの計測値を保存するまでの間隔が`interval`となる。クライアントが`interval`期間内に2度`getValues`アクションを呼び出した場合、取得される値は変化しない可能性が高いことを留意した上で操作することが望ましい。

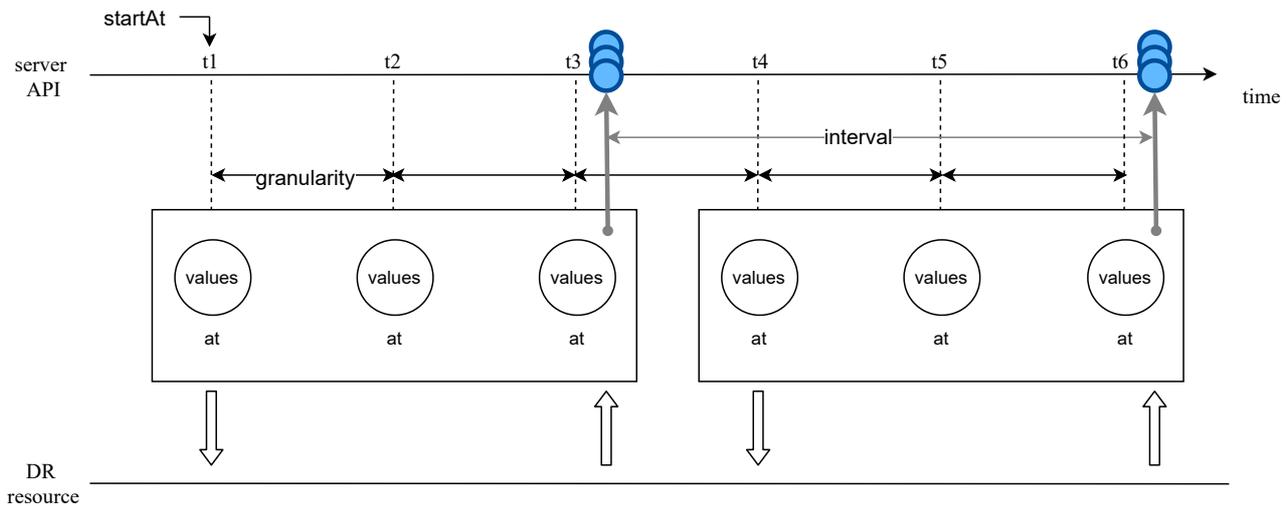


図6-7 計測値 (type=measure) の例 : granularityとintervalが異なるケース

図6-8は、計測値におけるdataCacheDurationについて示している。一度サーバに対してDRレポート仕様登録が成功すると、startAt時刻以降、計測値および予測値に対しても定期的にDRレポートが生成・保存されるが、これらの値は短期的に使用されるデータであるため、長期にわたってサーバ上に保存される必要はない。サーバはクライアントから要求されたDRレポート仕様登録に対して、サーバが規定するデータ保持期間dataCacheDurationをクライアントへ返却する。dataCacheDurationは現在時刻から遡った期間となる。計測値について長期に保存・検索用途にて使用したい場合は、後述のresHistoriesを利用することが望ましい。

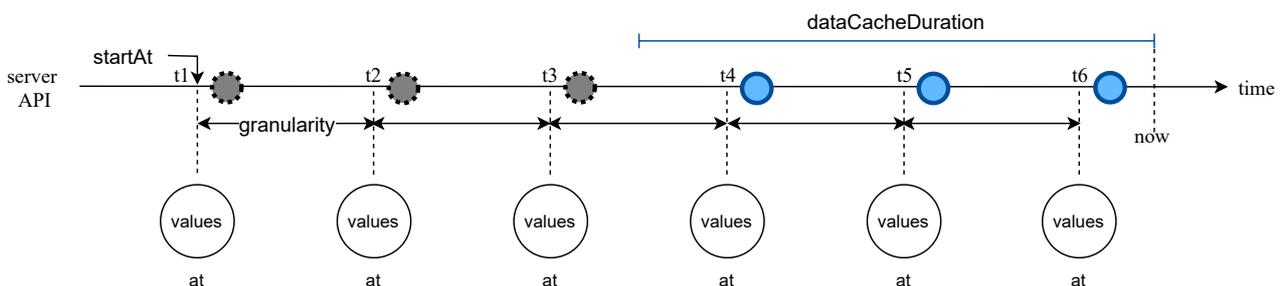


図6-8 計測値におけるdataCacheDurationの例

予測値の扱い

図6-9に予測値を扱う際の各値の関係について示す。クライアントがサーバへ予測値のDRレポート仕様登録をした後、クライアントから後述するgetValuesアクションを用いて (from、toを指定して) 2つの予測値のDRレポートを取得するケースを示している。なお、サーバにおける予測値の算出方法については言及しない。

予測値の場合は、クライアントがサーバに対して、一定時間後から予測値を作成するよう開始時刻までのオフセット時間futurePeriodを指示する。futurePeriodが省略された場合は0とみなす。サーバは、同指示を受けた時刻にfuturePeriodを加えた時刻以降の時刻を予測値の開始時刻startAtとして決定し、granularityで指定された時間粒度にてdataCacheDuration時間までの予測値を算出し、startAtまでこの値を保持する。startAtは「同指示を受けた時刻とfuturePeriodを加えた時刻 (txとする) 以降」としているが、tx以降でintervalを考慮して切りの良い時刻とすることが望ましい。maxDelayTimeやminTransmissionIntervalは、計測値と同じ扱いとなる。

図6-9では、DRレポート仕様登録時 t_0 から $startAt (=t_{11})$ までの間に、 $getValues$ アクションにより t_{11} から $t_{14} (=t_{11}+dataCacheDuration)$ までのDRレポート予測値が取得できる。 t_{11} から $t_{21} (=t_{11}+interval)$ の間は、 $getValues$ アクションにより t_{21} から $t_{24} (=t_{21}+dataCacheDuration)$ までのDRレポート予測値が取得できる。

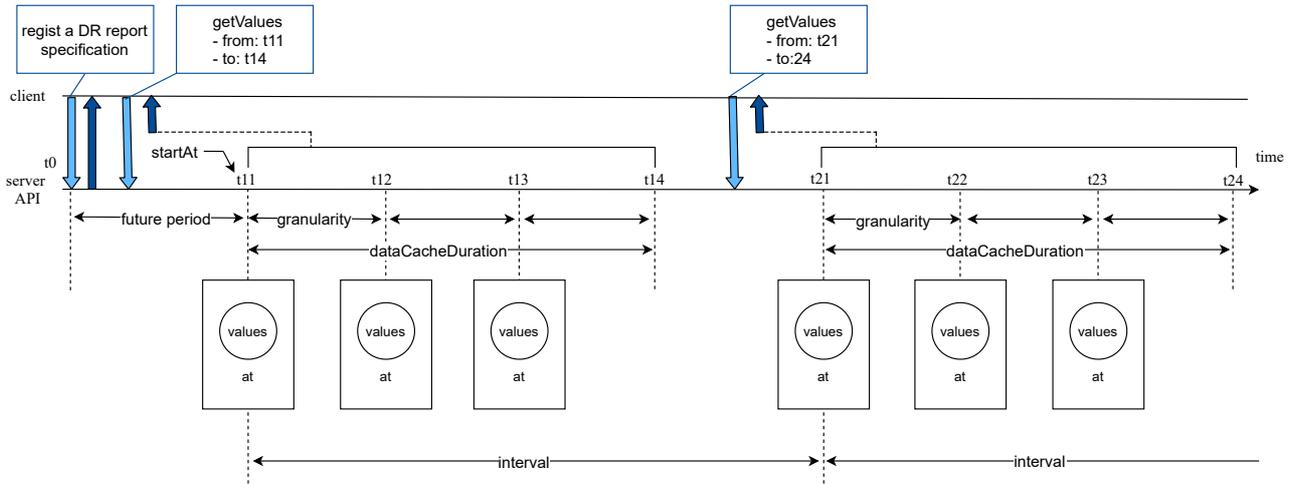


図6-9 予測値 (type=projected) の例 : dataCacheDurationがintervalより短い場合

図6-9では、 $dataCacheDuration$ が $interval$ より短い場合を示したが、図6-10では、 $dataCacheDuration$ が $interval$ より長い場合を示している (図中、クライアントからの操作は省略)。

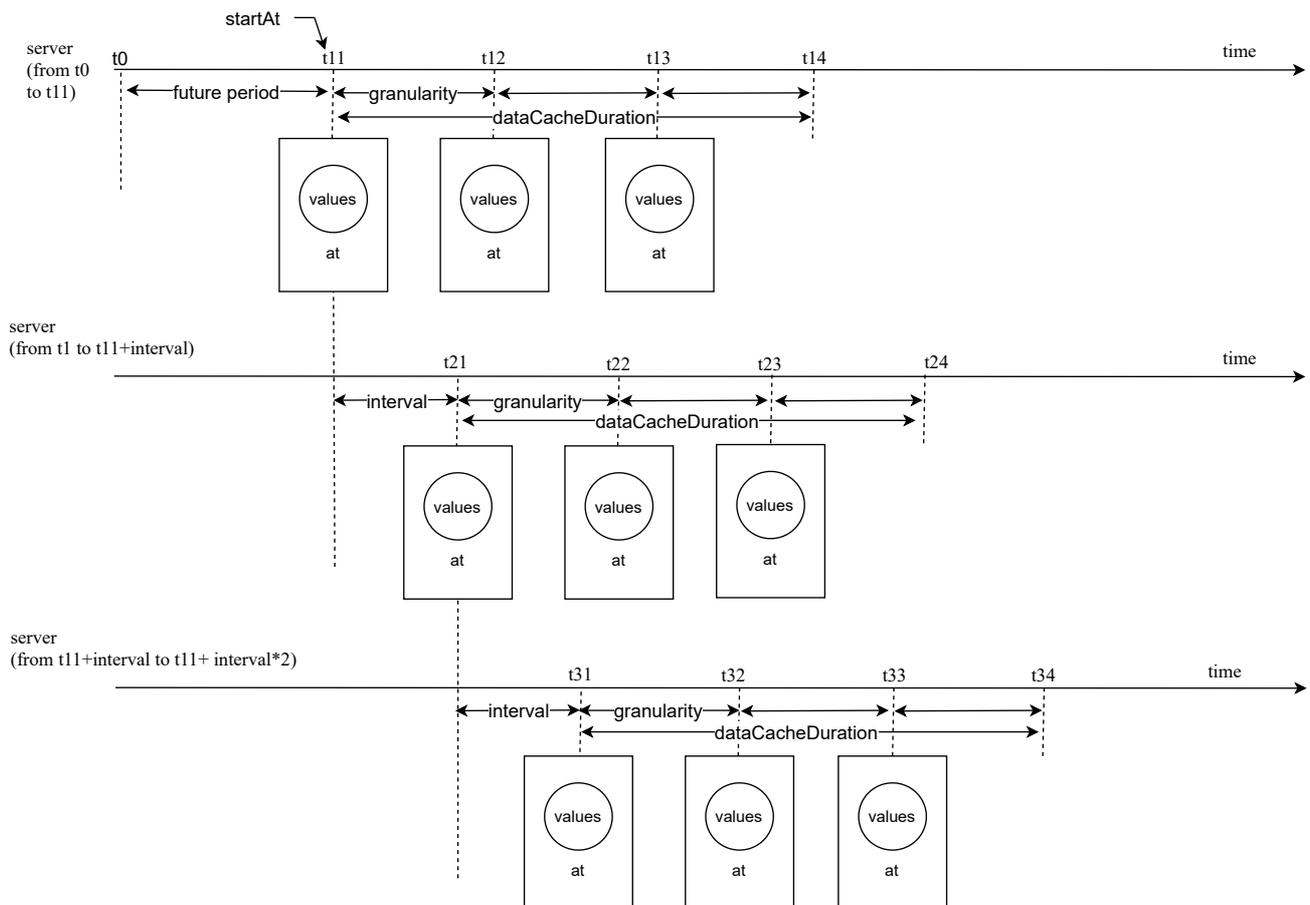


図6-10 予測値 (type=projected) の例 : dataCacheDurationがintervalより長い場合

各々の図では、t0からt11までの間に取得できる予測値（t11からt14（=t11+datCacheDuration）まで）、t11からt21までの間に取得できる予測値（t21からt24（=t21+dataCacheDuration）まで）、図6-10の場合はさらにt21からt31までの間に取得できる予測値（t31からt34（=t31+dataCacheDuration）まで）の例となる。

valueKindは、対象とするDR対象エネルギーリソースタイプ（derType）によって、取りうる値が異なる。以下に、現在想定する値について整理する。

DR対象エネルギーリソースタイプ（derType）がdemandGroupの場合：

name	valueKind	Unit	report type	description
制御可能量	drCapacity	kW	projected	正值：下げDR 負値：上げDR
基準値（事前予測方式）	reference	kW	projected	正值：潮流 負値：逆潮流
基準値（直前計測方式）	reference	kW	measure	正值：潮流 負値：逆潮流
実績電力値	electricPower	kW	measure	受電点の電力値（グループの合計値）。正值：潮流 負値：逆潮流
実績電力量値	electricEnergy	kWh	measure	受電点の電力消費量（グループの合計値）。正值：潮流 負値：逆潮流

DR対象エネルギーリソースタイプ（derType）がstorageBatteryGroupの場合：

name	valueKind	Unit	report type	description
状態	status	none	measure	蓄電池群の状態[ステータスコード]※別途定義（true：正常 false：異常）
充電電力	chargePower	kW	measure	蓄電池群の充電電力の合計
放電電力	dischargePower	kW	measure	蓄電池群の放電電力の合計
充電電力量	chargedEnergy	kWh	measure	蓄電池群の充電電力量の合計
放電電力量	dischargedEnergy	kWh	measure	蓄電池群の放電電力量の合計
充電可能量	chargeAvailable	kWh	measure	蓄電池群の充電可能量の合計
放電可能量	dischargeAvailable	kWh	measure	蓄電池群の放電可能量の合計

name	valueKind	Unit	report type	description
蓄電池 残量	storedEnergy	kWh	measure	蓄電池群の蓄電池残量の合計

REQUEST

REQUEST BODY - application/json

Property	Type	Required	Description
type	string	Yes	DRレポートタイプ。“measure” 計測値, “projected” 予測値のいずれか。
descriptions	object	Yes	DRレポート名称
descriptions.ja	string	Yes	日本語記述
descriptions.en	string	Yes	英語記述
drResourceId	string	Yes	DR対象エネルギーリソースID
granularity	number	Yes	計測値または予測値の時間粒度
granularityUnit	string	Yes	granularityの単位。“hour” 時間, “minute” 分, “second” 秒のいずれか
valueUnit	array	Yes	値の単位の組。valueKindの要素数分指定する。
valueUnit[]	string	No	値の単位。“kW” kW, “kWh” kWh, “none” 無しのいずれか。
valueKind	array	Yes	値の種類組。

Property	Type	Required	Description
			<p>値の種類。</p> <p>DR対象エネルギーリソースタイプ (derType) が“demandGroup”の場合、</p> <ul style="list-style-type: none"> “drCapacity” 制御可能量, “reference” 基準値 (事前予測方式または直前計測方式), “electricPower” 実績電力値, “electricEnergy” 実績電力量値 <p>のいずれかの組み合わせ。“reference” (直前計測方式), “electricPower”, “electricEnergy”はDRレポートタイプ (type) が“measure”の際に、“drCapacity”, “reference” (事前予測方式) はDRレポートタイプ (type) が“projected”の際に使用することができる。値の単位は、“electricEnergy”がkWhで、その他はkW。</p>
valueKind[]	string	No	<p>DR対象エネルギーリソースタイプ (derType) が“storageBatteryGroup”の場合、</p> <ul style="list-style-type: none"> “status” 状態, “chargePower” 充電電力, “dischargePower” 放電電力, “chargeEnergy” 充電電力量, “dischargeEnergy” 放電電力量, “chargeAvailable” 充電可能量, “dischargeAvailable” 放電可能量, “storedEnergy” 蓄電池残量 <p>のいずれかの組み合わせ。いずれもDRレポートタイプ (type) が“measure”の際に使用することができる。値の単位は、“status”が単位無し (“none”を指定。valueは、“online”オンライン、“offline”オフラインのいずれか (暫定仕様))、“chargePower”と“dischargePower”がkW、その他はkWh。</p>
maxDelayTime	number	No	DRレポート最大遅延時間。クライアントがgetValuesを発行してからDRレポートをサーバから受信するまでに 許容可能な最大遅延時間
maxDelayTimeUnit	string	No	DRレポート最大遅延時間の単位。“hour” 時間, “minute” 分, “second” 秒のいずれか。
futurePeriod	number	No	DRレポート生成時刻から予測データの最初の時刻までの時間。typeが“projected”の場合のみ有効であり、省略時は0。
futurePeriodUnit	string	No	futurePeriodの単位。“hour” 時間, “minute” 分, “second” 秒のいずれか。typeが“projected”の場合のみ有効。

【例1 : typeがmeasureの場合】

```
{
  "type": "measure",
  "descriptions": {
    "ja": "計測値レポート1",
    "en": "Actual value report1"
  },
  "drResourceId": "1",
  "granularity": 1,
  "granularityUnit": "minute",
  "valueUnit": [
    "kW",
    "kWh"
  ],
  "valueKind": [
    "electricPower",
    "electricEnergy"
  ],
  "maxDelayTime": 60,
  "maxDelayTimeUnit": "second"
}
```

【例2 : typeがprojectedの場合】

```
{
  "type": "projected",
  "descriptions": {
    "ja": "予測値レポート1",
    "en": "Prediction value report1"
  },
  "drResourceId": "1",
  "granularity": 30,
  "granularityUnit": "minute",
  "valueUnit": [
    "kW",
    "kWh"
  ],
  "valueKind": [
    "electricPower",
    "electricEnergy"
  ],
  "maxDelayTime": 60,
  "maxDelayTimeUnit": "second",
  "futurePeriod": 24,
  "futurePeriodUnit": "hour"
}
```

RESPONSE

STATUS CODE - 201:

RESPONSE MODEL - application/json

Property	Type	Required	Description
id	string	Yes	DRレポート仕様ID
startAt	string	Yes	DRレポートの起点時刻 (RFC 3339 (ISO 8601))。getValuesアクションの受付開始時刻。
minTransmissionInterval	number	Yes	サーバが許容可能な最短のDRレポート取得 (ポーリング) 間隔。
minTransmissionIntervalUnit	string	Yes	最短DRレポート取得間隔の時間単位。"hour" 時間, "minute" 分, "second" 秒のいずれか。
dataCacheDuration	number	Yes	データキャッシュの長さ。計測値の場合はデータがキャッシュされる時間、予測値の場合は予測の範囲を示す。
dataCacheDurationUnit	string	Yes	データキャッシュ期間の時間単位。"hour" 時間, "minute" 分, "second" 秒のいずれか。
interval	number	Yes	データ更新間隔
intervalUnit	string	Yes	インターバル時間単位。"hour" 時間, "minute" 分, "second" 秒

【例】

```
{
  "id": "1",
  "startAt": "2023-07-01T17:55:00+09:00",
  "minTransmissionInterval": 30,
  "minTransmissionIntervalUnit": "second",
  "dataCacheDuration": 15,
  "dataCacheDurationUnit": "minute",
  "interval": 1,
  "intervalUnit": "minute"
}
```

6.3.2 DRレポート一覧取得 GET /elapi/v1/drReports

DRレポートのID一覧を取得する。

REQUEST

No request parameters

RESPONSE

STATUS CODE - 200:

RESPONSE MODEL - application/json

Property	Type	Required	Description
registrationLimit	number	No	DRレポート要求仕様登録上限数
drReports	array	Yes	DRレポートIDリスト
drReports[].id	string	Yes	DRレポートID
drReports[].descriptions	object	Yes	名称記述
drReports[].descriptions.ja	string	Yes	日本語記述
drReports[].descriptions.en	string	Yes	英語記述

【例】

```
{
  "registrationLimit": 100,
  "drReports": [
    {
      "id": "1",
      "descriptions": {
        "ja": "計測レポート1",
        "en": "Performance report1"
      }
    },
    {
      "id": "2",
      "descriptions": {
        "ja": "計測レポート2",
        "en": "Performance report2"
      }
    }
  ]
}
```

6.3.3 DRレポートDescription取得 GET /elapi/v1/drReports/{drReportId}

DRレポートIDを指定して、DRレポートDescriptionを取得する。

REQUEST

PATH PARAMETERS

Property	Type	Required	Description
drReportId	string	Yes	DR report ID

【例】

"1"

RESPONSE

STATUS CODE - 200:

RESPONSE MODEL - application/json

Property	Type	Required	Description
properties	object	Yes	「6.3.4 DRレポート内容取得」におけるプロパティリソースに関する定義
actions	object	Yes	「6.3.5 DRレポート値取得」におけるアクションリソースに関する定義

【例】

```
{
  "properties": {
    "type": {
      "descriptions": {
        "ja": "DRレポートタイプ",
        "en": "type of DR report"
      },
      "writable": false,
      "observable": false,
      "schema": {
        "type": "string",
        "enum": [
          "measure",
          "projected"
        ]
      }
    }
  },
  "descriptions": {
    "descriptions": {
      "ja": "DRレポート名称",
      "en": "name of a DR report"
    },
    "writable": false,
    "observable": false,
    "schema": {
      "type": "object",
      "properties": {
        "ja": {
          "type": "string"
        },
        "en": {
          "type": "string"
        }
      }
    }
  },
  "drResourceId": {
```

```
"descriptions": {
  "ja": "DR対象エネルギーリソースID",
  "en": "DR target energy resource ID"
},
"writable": false,
"observable": false,
"schema": {
  "type": "string"
}
},
"granularity": {
  "descriptions": {
    "ja": "計測値または予測値の時間粒度",
    "en": "time granularity of measured or projected values"
  },
  "writable": false,
  "observable": false,
  "schema": {
    "type": "number"
  }
},
"granularityUnit": {
  "descriptions": {
    "ja": "DRイベントタイプ",
    "en": "type of DR event"
  },
  "writable": false,
  "observable": false,
  "schema": {
    "type": "string",
    "enum": [
      "hour",
      "minute",
      "second"
    ]
  }
},
"valueUnit": {
  "descriptions": {
    "ja": "値の単位の組",
    "en": "array of value units"
  },
  "writable": false,
  "observable": false,
  "schema": {
    "type": "array",
    "items": {
      "type": "string",
      "enum": [
        "kW",
        "kWh",
        "%",
        "none"
      ]
    }
  }
},
"valueKind": {
  "descriptions": {
    "ja": "値の種類",

```

```
    "en": "array of value types"
  },
  "writable": false,
  "observable": false,
  "schema": {
    "type": "array",
    "items": {
      "type": "string",
      "enum": [
        "drCapacity",
        "reference",
        "electricPower",
        "status",
        "chargePower",
        "dischargePower",
        "chargeEnergy",
        "dischargeEnergy",
        "chargeAvailable",
        "dischargeAvailable",
        "storedEnergy"
      ]
    }
  }
},
"maxDelayTime": {
  "descriptions": {
    "ja": "DRレポート最大遅延時間",
    "en": "maximum delay time in receiving a DR report"
  },
  "writable": false,
  "observable": false,
  "schema": {
    "type": "number"
  }
},
"maxDelayTimeUnit": {
  "descriptions": {
    "ja": "DRレポート最大遅延時間の単位",
    "en": "unit of maximum delay time in receiving a DR report"
  },
  "writable": false,
  "observable": false,
  "schema": {
    "type": "string",
    "enum": [
      "hour",
      "minute",
      "second"
    ]
  }
},
"futurePeriod": {
  "descriptions": {
    "ja": "DRレポート生成時刻から予測データの最初の時刻までの時間",
    "en": "period from the DR report generation time to the first time of the projected
data"
  },
  "writable": false,
  "observable": false,
  "schema": {
```

```
    "type": "number"
  }
},
"futurePeriodUnit": {
  "descriptions": {
    "ja": "futurePeriodの単位",
    "en": "unit of the future period"
  },
  "writable": false,
  "observable": false,
  "schema": {
    "type": "string",
    "enum": [
      "hour",
      "minute",
      "second"
    ]
  }
}
},
"actions": {
  "getValues": {
    "descriptions": {
      "ja": "DRイベント名称",
      "en": "name of a DR event"
    },
    "input": {
      "type": "object",
      "properties": {
        "from": {
          "type": "string",
          "format": "date-time"
        },
        "to": {
          "type": "string",
          "format": "date-time"
        }
      }
    }
  },
  "schema": {
    "type": "object",
    "properties": {
      "values": {
        "type": "array",
        "items": {
          "type": "object",
          "properties": {
            "at": {
              "type": "string",
              "format": "date-time"
            }
          }
        },
        "values": {
          "type": "object",
          "properties": {
            "kind": {
              "type": "string",
              "enum": [
                "drCapacity",
                "reference",
                "electricPower",

```


Property	Type	Required	Description
type	string	Yes	DRレポートタイプ。“measure” 計測値, “projected” 予測値 のいずれか。
descriptions	object	Yes	DRレポート名称
descriptions.ja	string	Yes	日本語記述
descriptions.en	string	Yes	英語記述
drResourceId	string	Yes	DR対象エネルギーリソースID
granularity	number	Yes	計測値または予測値の時間粒度
granularityUnit	string	Yes	granularityの単位。“hour” 時間, “minute” 分, “second” 秒のいずれか
valueUnit	array	Yes	値の単位の組。valueKindの要素数分指定する。
valueUnit[]	string	No	値の単位。“kW” kW, “kWh” kWh, “none”無しのいずれか。
valueKind	array	Yes	値の種類組

Property	Type	Required	Description
			<p>値の種類。</p> <p>DR対象エネルギーリソースタイプ (derType) が“demandGroup”の場合、</p> <ul style="list-style-type: none"> “drCapacity” 制御可能量, “reference” 基準値 (事前予測方式または直前計測方式), “electricPower” 実績電力値, “electricEnergy” 実績電力量値 <p>のいずれかの組み合わせ。“reference” (直前計測方式), “electricPower”, “electricEnergy”はDRレポートタイプ (type) が“measure”の際に、“drCapacity”, “reference” (事前予測方式) はDRレポートタイプ (type) が“projected”の際に使用することができる。値の単位は、“electricEnergy”がkWhで、その他はkW。</p>
valueKind[]	string	No	<p>DR対象エネルギーリソースタイプ (derType) が“storageBatteryGroup”の場合、</p> <ul style="list-style-type: none"> “status” 状態, “chargePower” 充電電力, “dischargePower” 放電電力, “chargeEnergy” 充電電力量, “dischargeEnergy” 放電電力量, “chargeAvailable” 充電可能量, “dischargeAvailable” 放電可能量, “storedEnergy” 蓄電池残量 <p>のいずれかの組合せ。いずれもDRレポートタイプ (type) が“measure”の際に使用することができる。値の単位は、“status”が単位無し (“none”を指定)、“chargePower”と“dischargePower”がkW、その他はkWh。</p>
maxDelayTime	number	No	DRレポート最大遅延時間。クライアントがgetValuesを発行してからDRレポートをサーバから受信するまでに 許容可能な最大遅延時間
maxDelayTimeUnit	string	No	DRレポート最大遅延時間の単位。“hour” 時間, “minute” 分, “second” 秒のいずれか。
futurePeriod	number	No	DRレポート生成時刻から予測データの最初の時刻までの時間。type が“projected”の場合のみ有効であり、省略時は0。
futurePeriodUnit	string	No	futurePeriodの単位。“hour” 時間, “minute” 分, “second” 秒のいずれか。typeが“projected”の場合のみ有効。

【例1 : typeがmeasureの場合】

```
{
  "type": "measure",
  "descriptions": {
    "ja": "計測値レポート1",
    "en": "Actual value report1"
  },
  "drResourceId": "1",
  "granularity": 1,
  "granularityUnit": "minute",
  "valueUnit": [
    "kW",
    "kWh"
  ],
  "valueKind": [
    "electricPower",
    "electricEnergy"
  ],
  "maxDelayTime": 60,
  "maxDelayTimeUnit": "second"
}
```

【例2：typeがprojectedの場合】

```
{
  "type": "projected",
  "descriptions": {
    "ja": "予測値レポート1",
    "en": "Prediction value report1"
  },
  "drResourceId": "1",
  "granularity": 30,
  "granularityUnit": "minute",
  "valueUnit": [
    "kW",
    "kWh"
  ],
  "valueKind": [
    "electricPower",
    "electricEnergy"
  ],
  "maxDelayTime": 60,
  "maxDelayTimeUnit": "second",
  "futurePeriod": 24,
  "futurePeriodUnit": "hour"
}
```

6.3.5 DRレポート値取得 POST /elapi/v1/drReports/{drReportId}/actions/getValues

DRレポートの値を取得する。リクエストボディにて、fromとtoで時間範囲を指定することができる。

計測値においては、記録した時点での値を扱うため、from、toにて指定された時刻におけるDRレポート値も含む点に留意すること。なお、値の単位がkWhの場合、atで示される時刻における値は、同時刻から直前のgranularityまでの期間に

おける値となる点に留意すること。たとえば、atが9時における値は、granularityが30分の場合の値は、8時30分から9時までのkWh値となる。

予測値においては、at時点からgranularity分先までの値を予測する点に留意すること。toが指定された場合、toまでの予測値までを必須とし、toからto+granularity分の予測値を含むか否かは実装依存とする。

なお、計測値および予測値いずれの場合もfrom, toは共に省略可能で、from省略時はDRレポート値の計測（または予測）開始時より、to省略時は計測（または予測）最新時までが指定されたものとみなす。from, toで指定された時間範囲内にデータが存在しない場合は、空（{"values": []}）を返す。

type	at (=ti) における単位kW指定時のvalues	at (=ti) における単位kWh指定時のvalues	toの扱い
measure (計測値)	at時点での計測値	atから直前のgranularity期間までの計測値	toの時刻で計測された値含む
projected (予測値)	atからgranularity先の期間までの予測値	atからgranularity先の期間までの予測値	toまでの予測値は必須。toからto+granularity分の予測値を含むか否かは実装依存

REQUEST

PATH PARAMETERS

Property	Type	Required	Description
drReportId	string	Yes	DR report ID

【例】

```
"1"
```

REQUEST BODY – application/json

Property	Type	Required	Description
from	string	No	指定開始時刻 (RFC 3339 (ISO 8601) 形式)
to	string	No	指定終了時刻 (RFC 3339 (ISO 8601) 形式)

【例】

```
{
  "from": "2023-07-01T17:59:00+09:00",
  "to": "2023-07-01T18:00:00+09:00"
}
```

RESPONSE

STATUS CODE - 201:

RESPONSE MODEL - application/json

Property	Type	Required	Description
values array	Yes	DRレポート記録時刻・値の配列	
values[].at	string	Yes	DRレポート記録時刻 (RFC 3339 (ISO 8601)形式)
values[].values	array	Yes	DRレポート値 (値の種類と値の組) の配列

Property	Type	Required	Description
			<p>値の種類の組。</p> <p>DR対象エネルギーリソースタイプ (derType) が“demandGroup”の場合、</p> <ul style="list-style-type: none"> “drCapacity” 制御可能量, “reference” 基準値 (事前予測方式または直前計測方式), “electricPower” 実績電力値, “electricEnergy” 実績電力量値 <p>のいずれかの組み合わせ。“reference” (直前計測方式), “electricPower”, “electricEnergy”はDRレポートタイプ (type) が“measure”の際に、“drCapacity”, “reference” (事前予測方式)はDRレポートタイプ (type) が“projected”の際に使用することができる。値の単位は、“electricEnergy”がkWhで、その他はkW。</p>
values[].values[].kind	string	Yes	<p>DR対象エネルギーリソースタイプ (derType) が“storageBatteryGroup”の場合、</p> <ul style="list-style-type: none"> “status” 状態, “chargePower” 充電電力, “dischargePower” 放電電力, “chargeEnergy” 充電電力量, “dischargeEnergy” 放電電力量, “chargeAvailable” 充電可能量, “dischargeAvailable” 放電可能量, “storedEnergy” 蓄電池残量 <p>のいずれかの組み合わせ。いずれもDRレポートタイプ (type) が“measure”の際に使用することができる。値の単位は、“status”が単位無し (“none”を指定)、“chargePower”と“dischargePower”がkW、その他はkWh。</p>
values[].values[].value	number	Yes	<p>DRレポート値。kindが“status”の場合、数値の意味は別途定義。</p>

【例】

```

{
  "values": [
    {
      "at": "2023-07-01T17:59:00+09:00",
      "values": [
        {

```

```
    "kind": "electricPower",  
    "value": 40  
  },  
  {  
    "kind": "electricEnergy",  
    "value": 5  
  }  
]  
},  
{  
  "at": "2023-07-01T18:00:00+09:00",  
  "values": [  
    {  
      "kind": "electricPower",  
      "value": 10  
    },  
    {  
      "kind": "electricEnergy",  
      "value": 4  
    }  
  ]  
}  
]  
}
```

STATUS CODE - 202:

No response body

6.3.6 DRレポート削除 DELETE /elapi/v1/drReports/{drReportId}

DRレポート仕様IDを削除する。成功すると、空のボディとstatus code 204を返却する。

REQUEST

PATH PARAMETERS

Property	Type	Required	Description
drReportId	string	Yes	DR report ID

【例】

```
"1"
```

RESPONSE

STATUS CODE - 204: Success

No response body

6.4 resHistories (DR用指定例)

DRレポートの履歴値については、API仕様部[2]規定のresHistoriesを利用することで実現することができる。ここでは、使用例について示す。

6.4.1 GET /elapi/v1/resHistories

リソース履歴IDの一覧取得。

【レスポンス例】

```
{
  "resHistories": [
    {
      "id": "1",
      "descriptions": {
        "ja": "履歴レポート1",
        "en": "History report1"
      }
    },
    {
      "id": "2",
      "descriptions": {
        "ja": "履歴レポート2",
        "en": "History report2"
      }
    }
  ]
}
```

6.4.2 GET /elapi/v1/resHistories/{resHistoryId}

resHistoryIdで指定されたresource history descriptionを取得する。

【レスポンス例】

```
{
  "properties": {
    "descriptions": {
      "descriptions": {
        "ja": "resource historyの説明",
        "en": "explanation of resource history"
      },
      "writable": false,
      "observable": false,
      "schema": {
        "type": "object",
        "properties": {
          "ja": {
            "type": "string"
          },
          "en": {
            "type": "string"
          }
        }
      }
    }
  }
}
```

```
    }  
  }  
},  
"resourceType": {  
  "descriptions": {  
    "ja": "リソースタイプ",  
    "en": "resource type"  
  },  
  "writable": false,  
  "observable": false,  
  "schema": {  
    "type": "string",  
    "enum": [  
      "devices",  
      "drResources"  
    ]  
  }  
},  
"resourceId": {  
  "descriptions": {  
    "ja": "リソースID",  
    "en": "resource ID"  
  },  
  "writable": false,  
  "observable": false,  
  "schema": {  
    "type": "string"  
  }  
},  
"valueUnit": {  
  "descriptions": {  
    "ja": "値の単位の組",  
    "en": "array of value units"  
  },  
  "writable": false,  
  "observable": false,  
  "schema": {  
    "type": "array",  
    "items": {  
      "type": "string",  
      "enum": [  
        "kW",  
        "kWh",  
        "%",  
        "none"  
      ]  
    }  
  }  
},  
"valueKind": {  
  "descriptions": {  
    "ja": "値の種類組",  
    "en": "array of value types"  
  },  
  "writable": false,  
  "observable": false,  
  "schema": {  
    "type": "array",  
    "items": {
```

```
    "type": "string",
    "enum": [
      "drCapacity",
      "reference",
      "electricPower",
      "status",
      "chargePower",
      "dischargePower",
      "chargeEnergy",
      "dischargeEnergy",
      "chargeAvailable",
      "dischargeAvailable",
      "storedEnergy"
    ]
  }
}
},
"timing": {
  "descriptions": {
    "ja": "履歴データの取得タイミング",
    "en": "timing to record history data"
  },
  "writable": false,
  "observable": false,
  "schema": {
    "type": "object",
    "properties": {
      "timingType": {
        "type": "string",
        "enum": [
          "onChange",
          "interval"
        ]
      },
      "intervalValue": {
        "type": "number"
      },
      "intervalUnit": {
        "type": "string",
        "enum": [
          "second",
          "minute",
          "hour",
          "day",
          "month",
          "year"
        ]
      }
    }
  }
},
"first": {
  "descriptions": {
    "ja": "最初の記録時刻",
    "en": "time of the first record"
  },
  "writable": false,
  "observable": false,
  "schema": {
    "type": "string",
```

```
    "format": "date-time"
  }
},
"last": {
  "descriptions": {
    "ja": "最後の記録時刻",
    "en": "time of the last record"
  },
  "writable": false,
  "observable": false,
  "schema": {
    "type": "string",
    "format": "date-time"
  }
},
"total": {
  "descriptions": {
    "ja": "リソース履歴データの総個数",
    "en": "total count of the resource history data"
  },
  "writable": false,
  "observable": false,
  "schema": {
    "type": "number",
    "minimum": 0,
    "multipleOf": 1
  }
},
"actions": {
  "prepareRetrieveData": {
    "descriptions": {
      "ja": "取得用データの準備を指示する",
      "en": "Prepare data to retrieve."
    },
    "input": {
      "type": "object",
      "properties": {
        "from": {
          "type": "string",
          "format": "date-time"
        },
        "to": {
          "type": "string",
          "format": "date-time"
        }
      }
    },
    "schema": {
      "type": "object",
      "properties": {
        "dataId": {
          "type": "string"
        },
        "count": {
          "type": "number",
          "minimum": 0,
          "multipleOf": 1
        }
      },
      "countPerPage": {
```

```
        "type": "number",
        "minimum": 1,
        "multipleOf": 1
      }
    }
  },
  "retrieve": {
    "descriptions": {
      "ja": "履歴データを取得する",
      "en": "retrieve histories data."
    },
    "input": {
      "type": "object",
      "properties": {
        "dataId": {
          "type": "string"
        },
        "page": {
          "type": "number",
          "minimum": 1,
          "multipleOf": 1
        }
      }
    },
    "schema": {
      "type": "object",
      "properties": {
        "processStatus": {
          "type": "string",
          "enum": [
            "succeeded",
            "failed",
            "inProgress"
          ]
        }
      }
    },
    "values": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "at": {
            "type": "string",
            "format": "date-time"
          }
        }
      },
      "values": {
        "type": "object",
        "properties": {
          "kind": {
            "type": "string",
            "enum": [
              "drCapacity",
              "reference",
              "electricPower",
              "status",
              "chargePower",
              "dischargePower",
              "chargeEnergy",
              "dischargeEnergy",
              "chargeAvailable",
            ]
          }
        }
      }
    }
  }
}
```



```

    "timingType": "onChange"
  },
  "first": "2023-04-01T08:00:00+09:00",
  "last": "2023-04-24T22:00:00+09:00"
}

```

6.4.4 POST /elapi/v1/resHistories/{resHistoryId}/actions/prepareRetrieveData

クライアントは、resHistoryIdで指定される履歴データセットのうち、所望する履歴データの対象期間（開始時刻“from”、終了時刻“to”）を指定し、サーバに対して取得対象となる履歴データサブセットの確定を要求する。from, toは共に省略可能で、from省略時はDRレポート値の記録開始時より、to省略時は記録最新時までが指定されたものとみなす。

【リクエスト例】

```

{
  "from": "2023-04-10T00:00:00+09:00",
  "to": "2023-04-20T00:00:00+09:00"
}

```

【レスポンス例】

```

{
  "dataId": "0023",
  "count": 481,
  "countPerPage": 100
}

```

6.4.5 POST /elapi/v1/resHistories/{resHistoryId}/actions/retrieve

クライアントは、dataIdで指定される履歴データセットの取得を実行する。指定された時間範囲内にデータが存在しない場合は、空（“{“values”: []}”）を返す。

失敗時の応答は、API仕様部[2]を参照すること。

【リクエスト例】

```

{
  "dataId": "0023",
  "page": 2
}

```

【レスポンス例】

```

{
  "processStatus": "succeeded",
}

```

```
"values": [  
  {  
    "at": "2023-04-10T00:00:00+09:00",  
    "values": [  
      {  
        "kind": "electricPower",  
        "value": 0.5  
      },  
      {  
        "kind": "electricEnergy",  
        "value": 1.1  
      }  
    ]  
  },  
  {  
    "at": "2023-04-10T00:30:00+09:00",  
    "values": [  
      {  
        "kind": "electricPower",  
        "value": 0.6  
      },  
      {  
        "kind": "electricEnergy",  
        "value": 1.2  
      }  
    ]  
  }  
]
```

6.5 dsrmss

JEMA VPPガイドライン[1]にて規定されるDSR-MSモデルに従い、需要家側にある機器から制御対象となる機器を指定・登録するなど、リソース管理を扱う。

制御可能機器リストと制御対象機器指定リストの扱い

基本的に、JEMA VPPガイドライン[1]では、ある需要家が保有する制御可能な機器のうち、電力または電力量の制御が可能なネットワーク接続機器を列挙したものを「制御可能機器リスト」と定義している。また、制御可能機器リストのうち、実際にDR制御対象となる機器を列挙したものを「制御対象機器指定リスト」と定義している。本書では、前者のリストはサーバがあらかじめ何らかの手段を用いてサーバ内に登録するものと仮定し、具体的な登録手順については記載しない。後者のリストも同様にサーバにてあらかじめ登録する方法が考えられるものの、RAが制御可能機器リストの中から対象機器を選定の上、サーバへ登録・保持する方法についても必要に応じて選択可能とする。

DSR-MSは、制御対象機器指定リストを含む制御可能機器リストをひとまとまりのリソース単位として扱うモデルであり、本書では同リソースをDSR-MSリソースと呼ぶ。DSR-MSリソースは、単一の需要家が保持するリソースのみでなく、複数の需要家を跨って1つのDSR-MSリソースとして扱うこともできる。

DSR-MSにおけるDR対象エネルギーリソース

本書では、制御対象機器指定リストをDR対象エネルギーリソースと同一とみなす。DR対象エネルギーリソースIDを指定することで、前述のDRイベント、DRレポート操作が可能となる。

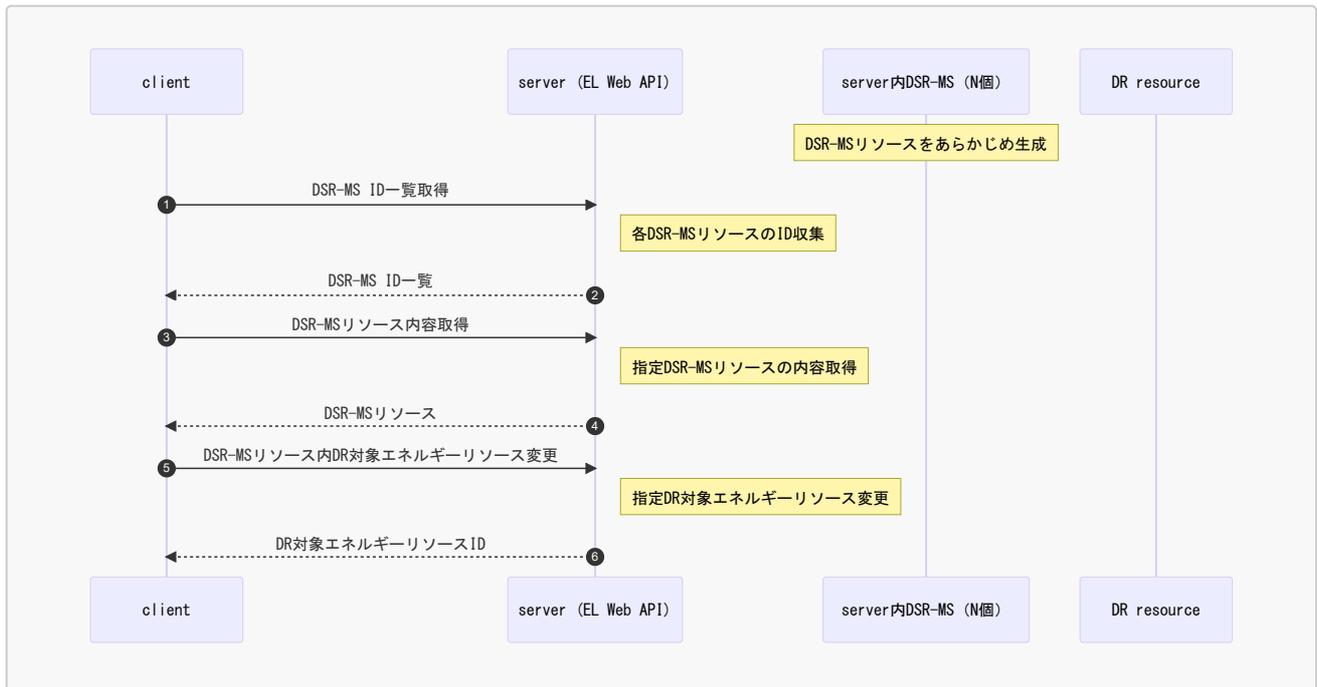


図6-11 DSR-MSリソースの操作

6.5.1 DSR-MS IDの一覧取得 GET /elapi/v1/dsrmss

DSR-MS IDの一覧を取得する。DSR-MSリソースはサーバにてあらかじめ何らかの手段を用いて登録が完了しており、同リソース毎にIDが付与されている。

REQUEST

No request parameters

RESPONSE

STATUS CODE – 200:

RESPONSE MODEL – application/json

Property	Type	Required	Description
dsrcms	array	Yes	DSR-MSのIDリスト。descriptionsを伴う。DSR-MSが登録されていない場合は空（“[]”）を返す。
dsrcms[].id	string	No	DSR-MS ID
dsrcms[].descriptions	object	No	DSR-MSリソースの名称
dsrcms[].descriptions.ja	string	No	日本語記述
dsrcms[].descriptions.en	string	No	英語記述

【例】

```
{
  "dsrcms": [
    {
      "id": "1",
      "descriptions": {
        "ja": "需要家A用 0001",
        "en": "resource group 0001 for demand group A"
      }
    },
    {
      "id": "2",
      "descriptions": {
        "ja": "需要家BおよびC用 0002",
        "en": "resource group 0002 for demand group B and C"
      }
    }
  ]
}
```

6.5.2 DSR-MS Description取得 GET /elapi/v1/dsrcms/{dsrcmsId}

DSR-MS IDを指定して、DSR-MS Descriptionを取得する。

REQUEST

PATH PARAMETERS

Property	Type	Required	Description
dsrcmsId	string	Yes	DSR-MS ID

【例】

```
"1"
```

RESPONSE

STATUS CODE - 200:

RESPONSE MODEL - application/json

Property	Type	Required	Description
properties	object	Yes	「6.5.3 DSR-MSリソース内容取得」におけるプロパティリソースに関する定義

【例】

```
{
  "properties": {
    "descriptions": {
      "descriptions": {
        "ja": "DSR-MSの説明",
        "en": "explanation of DSR-MS"
      },
      "writable": true,
      "observable": false,
      "schema": {
        "type": "object",
        "properties": {
          "ja": {
            "type": "string"
          },
          "en": {
            "type": "string"
          }
        }
      }
    },
    "drResources": {
      "descriptions": {
        "ja": "DR対象エネルギーリソースIDの配列（制御対象機器指定リスト）",
        "en": "array of energy resource IDs for DR (list of devices to be controlled)"
      },
      "writable": true,
      "observable": false,
      "schema": {
        "type": "array",
        "items": {
          "type": "string"
        }
      }
    }
  },
  "devices": {
```

```

    "descriptions": {
      "ja": "デバイスIDの配列",
      "en": "array of device IDs"
    },
    "writable": true,
    "observable": false,
    "schema": {
      "type": "array",
      "items": {
        "type": "string"
      }
    }
  }
}
}

```

6.5.3 DSR-MSリソース内容取得 GET /elapi/v1/dsrmss/{dsrmssId}/properties

DSR-MS IDを指定して、登録済みのDSR-MSリソース内容を取得する。

REQUEST

PATH PARAMETERS

Property	Type	Required	Description
dsrmssId	string	Yes	DSR-MS ID

【例】

```
"1"
```

RESPONSE

STATUS CODE - 200:

RESPONSE MODEL - application/json

Property	Type	Required	Description
descriptions	object	Yes	DSR-MSリソースの名称
descriptions.ja	string	Yes	日本語記述
descriptions.en	string	Yes	英語記述
drResources	array	Yes	DR対象エネルギーリソースIDの配列（制御対象機器指定リスト）
drResources[]	string	Yes	DR対象エネルギーリソースID
devices	array	Yes	デバイスIDの配列（制御可能機器リスト）
devices[]	string	Yes	デバイスID

【例】

```
{
  "descriptions": {
    "ja": "需要家A用 0001",
    "en": "resource group 0001 for demand group A"
  },
  "drResources": [
    "11", "13"
  ],
  "devices": [
    "0001", "0002", "0003", "0004", "0005"
  ]
}
```

6.5.4 DSR-MSリソース内DR対象エネルギーリソース変更 PUT

/elapi/v1/dsrms/{dsrmssId}/properties/drResources

DSR-MSリソース内のdrResources配列内にDR対象エネルギーリソースIDのエントリを追加するには、devices配列内にある任意のデバイスIDを複数指定したDR対象エネルギーリソースを6.1記載のAPIなどを用いて生成し、生成時に返却されるDR対象エネルギーリソースIDを加えたdrResources配列を用意して本APIによって上書きすることで実現することができる。

REQUEST

PATH PARAMETERS

Property	Type	Required	Description
drResources	array	Yes	DR対象エネルギーリソースIDの配列（制御対象機器指定リスト）
drResources[]	string	Yes	DR対象エネルギーリソースID

【例】

```
{
  "drResources": [
    "11", "13", "15"
  ]
}
```

RESPONSE

STATUS CODE - 200:

RESPONSE MODEL - application/json

Property	Type	Required	Description
drResources	array	Yes	DR対象エネルギーリソースIDの配列（制御対象機器指定リスト）

Property	Type	Required	Description
drResources[]	string	Yes	DR対象エネルギーリソースID

【例】

```

{
  "drResources": [
    "11", "13", "15"
  ]
}

```

6.5.5 (参考) DRに必要なDSR-MS情報の収集手順

上記DR用APIとECHONET Lite Web APIガイドラインにおける機器制御用基本APIやbulksなどの応用APIを用いて、DR制御に必要なDSR-MS情報を収集する手順例について示す。

概要

DR制御対象となる需要家の情報をDSR-MSから収集するにあたり、RAはDR用APIを搭載したDSR-MSより制御対象となるDSR-MSのIDを入手する(①)。さらに、得られたDSR-MS IDに対応するリソース内容を入手し、同リソース内の制御対象機器指定リストから、DR制御対象とするDR対象エネルギーリソースのIDを選定する(②)。DR用APIでは、基本的にこのDR対象エネルギーリソースIDを指定して、DR指示・制御やDR状況・結果取得などを実施する。

手順②で得られるリソース内には、制御対象機器指定リスト(DR対象エネルギーリソースIDリスト)の他、DSR-MSで制御候補となる制御可能機器リスト(機器IDリスト。コントローラーや機器など含む)などが含まれており、続いて、選定したDR対象エネルギーリソースIDに対応するリソース内容を入手することで(③)、同リソース内容から機器IDリスト内にある各機器の対応状態(離脱・異常発生状態)を確認できる(④)。また、各機器IDがどのような機器であるか確認するには、機器IDを指定してDevice Descriptionを取得し、deviceTypeの値を確認すれば良い(⑤)。なお、ECHONET Lite Web APIでは、Device Descriptionがプロパティマップ相当の機能をJSON形式にて表現している。

この機器IDを用いて、機器毎に詳細な情報・状態取得(各種プロパティ。運転モード設定や異常内容の詳細情報など含む)が可能となる(⑥)。PVや蓄電池、電気自動車充電器、燃料電池では系統連系状態も確認できる。スマートメーターでは、受電電力量、逆潮流電力量、瞬時電力計測値などが取得できる。複数の機器に関する複数プロパティ値の操作を一度の呼び出しで可能にするbulksと呼ばれる応用APIも活用することができる。

DR対象エネルギーリソースの内容には、リソースの属性情報としてDRサービス種(三次②下げDR、手動DRなど)、DER群の種類(受電点(需要家側リソース群)、機器点(現在蓄電池群のみ定義、将来EV群、PV群などサポート予定))などの情報も含まれる(③にて記載)。

DR対象エネルギーリソースIDを指定し、DRレポートAPIを呼び出すことで、受電点では制御可能量(上げDR/下げDR: kW)、機器点(蓄電池群)では充電/放電可能量(kWh)などの値が取得できる。こうした情報をRAが定期的に収集しDSR-MS内の状況を把握した上で、DRイベントによるDR制御の計画立案・指示を行うことになる。

具体例

- ① 需要家を特定するDSR-MSのID入手:

- RAは、DSR-MS IDの一覧取得API (GET /elapi/v1/dsrms) を用いてDSR-MSより返却されるID一覧から所望のDSR-MS IDを選択する、もしくは、事前にRAとDSR-MS管轄事業者との間で別手段により所望のDSR-MS IDを入手することで行う。前者にて複数のDSR-MS IDが返却される場合は、レスポンス内にてIDに付随する descriptions情報を参考に対象となるDSR-MS IDを選定する。
- ② 制御対象となるDR対象エネルギーリソースのID取得：
 - DSR-MSリソース内容取得API (GET /elapi/v1/dsrms/{dsrmssId}/properties) を用いてDSR-MSより返却されるリソース内に含まれる制御対象機器指定リスト (DR対象エネルギーリソースIDの配列) から所望のDR対象エネルギーリソースIDを選定することで行う。
- ③ 制御対象機器指定リストの取得・変更：
 - DSR-MSより返却されるDSR-MSリソースには、制御可能機器リスト (機器IDの配列) も含まれる。DR対象エネルギーリソースや制御可能機器リストは、DSR-MS内にて別途生成・管理されるが、RAにて制御対象機器指定リストを変更可能なAPIも提供されている (DSR-MSリソース内DR対象エネルギーリソース変更API (PUT /elapi/v1/dsrms/{dsrmssId}/properties/drResources))。制御対象機器指定リストは、同一目的のDRサービスに使用する機器の集合IDのリストとなる。機器IDを入手したい場合は、DR対象エネルギーリソース内容取得API (GET /elapi/v1/drResources/{drResourceId}/properties) を用いて、機器IDリストなどを取得すれば良い。同APIの応答には、機器IDリストの他、DRサービス種 (三次②下げDR、手動DRなど)、DER群の種類 (受電点 (需要家側リソース群)、機器点 (現在蓄電池群のみ定義、将来EV群、PV群などサポート予定))、機器毎の状態 (現在DR制御に利用可能/不可かを示す) などが含まれる。DR対象エネルギーリソース内にある機器IDリストの内容を変更したい場合は、DR対象エネルギーリソース内容変更API (PUT /elapi/v1/drResource/{drResourceId}/properties/{任意のプロパティリソース:ここでは devices}) を用いることができる。また、オプション扱いとなるが、RAがDR対象エネルギーリソース登録API (POST /elapi/v1/drResource) を用いてDR対象エネルギーリソースをDSR-MSへ登録することもできる。
- ④ DR対象エネルギーリソース内の各機器の状況確認：
 - 前述のDR対象エネルギーリソース内容取得APIにより取得したリソース内容には、機器IDリストに加え、各機器の対応状態リスト ("status") も含まれる。値の内容に基づき、利用可能・利用不可の状態を確認することができ、利用不可の場合は、実際の制御対象からは離脱していると判断できる。さらに、基本APIを用いて、異常内容プロパティリソース取得API (GET /elapi/v1/devices/{deviceId}/properties/faultDescription) を呼び出し、機器に起因するより詳細な離脱の要因 (異常時) について確認することができる (API仕様部V. 1.1.5では文字列型として定義)。なお、異常内容プロパティ (EPC=0x89) はECHONET Liteにてオプション扱いのため、機器が対応していない場合はfaultDescriptionリソースを取得できない。
- ⑤ 各機器の機器クラス確認：
 - DR対象エネルギーリソースや制御可能機器リストから得られる各機器IDに関して、Device Descriptionを取得し (GET /elapi/v1/devices/{deviceId} を呼び出す)、deviceTypeの値を確認することで、当該機器IDの機器クラスを把握できる。機器仕様部「5. 機器毎のDevice Description」にて、device typeの一覧が記載されており、たとえば、"storageBattery" (蓄電池)、"lvSmartElectricEnergyMeter" (低圧スマート電力量メータ)、"pvPowerGeneration" (住宅用太陽光発電)、"controller" (コントローラー)、"evChargerDischarger" (電気自動車充放電器) などが定義されている。
- ⑥ 各機器のプロパティ状態確認：
 - DR対象エネルギーリソースや制御可能機器リストから得られる各機器IDに関して、所望のプロパティリソースを指定して状態取得し (GET /elapi/v1/devices/{deviceId}/properties/{任意のプロパティリソース})、値を確認することで、当該機器のより詳細な情報や状態を把握できる。複数のプロパティリソ

スを一度の呼び出しで取得したい場合は、応用APIのbulksを活用することでRAとDSR-MS間の通信頻度を抑えることができる。複数の命令をリクエストボディに内包したbulkの作成し、DSR-MSへ登録後（POST /elapi/v1/bulks）、bulk実行の開始指示用API（POST /elapi/v1/bulks/{bulkId}/actions/execute）を呼び出し、bulk実行のレスポンス取得用API（POST /elapi/v1/bulks/{bulkId}/actions/getResults）にて複数の命令の実行結果をまとめて取得することが可能となる。

7. おわりに

本書では、DERを束ねたDR対象エネルギーリソースに対して各種DR操作を実現するためのAPI仕様について示した。

構成サービス種としては、DRイベント、DRレポートなどを定義し、従来のECHONET Lite Web APIガイドライン「API仕様部[2]」における規定スタイルに沿った形式にて提供している。DR制御対象となるDR対象エネルギーリソースの定義にバリエーションを持たせることで、さまざまなDR制御・DRシステム構成に対応できるAPIモデルとなっている。

今後、各種DR/VPP関連実証や実サービス展開が進むにつれ、新しい機能やサービスなどが導入される可能性がある。新しい課題や要件に関しても順次適用可能となるよう本API仕様の追加・拡充を進め、本API導入に係る各種事業者向けにより活用しやすいAPIとなるよう、仕様整備を進めていく。