
介護保険資格確認等WEBサービスとの連携における API仕様書

令和 8 年 8 月

【更新履歴】

No.	年月日	更新内容
1	令和8年8月	新規作成

目次

1. 本書の目的	4
2. 介護保険資格確認等WEBサービスについて	5
2.1 システム概要	5
2.2 システム利用準備	7
2.2.1 クライアント証明書	7
2.2.2 ユーザの払い出し	8
2.2.3 オンライン資格確認等システム（マイナ資格確認アプリ情報）の認証	8
2.3 画面遷移イメージ	9
2.4 サービス種類と職種による処理権限	10
3. 介護保険資格確認等WEBサービスのAPI	10
3.1 APIの処理概要（介護情報・資格確認）	11
3.1.1 データ連携のパターン	11
3.1.2 取得項目の留意事項	11
3.1.3 データ連携方法	12
3.2 APIの処理概要（ケアプラン情報連携）	12
3.2.1 データ連携のパターン	13
3.2.2 データ連携方法	13
3.2.3 IFバージョンの取り扱い	13
3.2.4 バージョン変換	13
3.3 API仕様（共通）	13
3.3.1 共通仕様	13
3.3.1.5 異常・障害時の対応	17
3.3.2 正常レスポンス時の警告メッセージの対応	18
3.3.3 APIの呼び出し順序の考慮	19
3.3.4 伝送使用上の留意事項	19
3.3.5 API連携実装のサンプルコード	21
3.3.6 責任分界点	26

別紙

- 別紙 1. APIと関連する画面.pdf
- 別紙 2-1. API仕様書（介護情報・資格確認）.pdf
- 別紙 2-2-1. API仕様書（ケアプランデータ連携）.pdf
- 別紙 2-2-2. 送信様式レイアウト.pdf
- 別紙 2-2-3. 受信様式レイアウト.pdf
- 別紙 3. API仕様書の読み方.pdf

- 別紙 4. サービス種類と職種の処理権限.xlsx
- 別紙 5-1. ケアプランデータ連携標準仕様.pdf
- 別紙 5-2. 【ケアプランデータ】APIフォーマット概要.pdf
- 別紙 5-3. 送受信レイアウトと様式との対応関係.pdf

1. 本書の目的

本書は介護事業所で利用される介護ソフトが、公益社団法人国民健康保険中央会（以下、「国保中央会」という。）で運用する介護保険資格確認等WEBサービスと情報連携して業務を実現するためのAPI仕様を示す。

2章では介護保険資格確認等WEBサービスの概要について記載する。介護事業所がブラウザ操作を行いシステム利用する場合の取り扱いを示す。介護API開発の際に参考されたい。

3章では介護ソフトが介護保険資格確認等WEBサービスのAPIを用いる際の仕様を示す。

2. 介護保険資格確認等WEBサービスについて

2.1 システム概要

介護保険資格確認等WEBサービスは、主に介護事業所職員がインターネットを通じてブラウザ操作により介護被保険者の情報を照会することで、これまで書面による情報照会・管理をしていたところ、電子的な情報連携を行うことで業務効率を向上することを目的としている。

証情報等の介護保険資格確認等WEBサービスで参照対象となる業務データは、介護保険資格確認等WEBサービスでは管理せず、国保中央会で運用する介護情報基盤で情報管理している。このため、介護保険資格確認等WEBサービスではログイン処理を行い、ブラウザ操作によって介護情報基盤に格納されている情報の参照を行う。

また、令和8年度下期以降、ケアプラン情報連携に関わる機能として、以下が利用可能となる。

- 1) 事業所間連携機能
- 2) 交付用登録機能
- 3) 同意・確認用登録機能

事業所間連携機能は、2つの介護事業所間でケアプラン情報を授受する機能である。本機能は、介護事業所間における電子的な情報連携を可能とすることで、業務効率を向上することを目的としている。介護保険資格確認等WEBサービスでは、事業所間連携用のケアプラン情報を一時的に保有する仕組みとしている。介護事業所は、国保中央会が作成する「ケアプランデータ連携標準仕様」5.0版（以下、「5.0版」という）で定義（別紙5-1～5-3、2-2-1～2-2-3参照）された連携仕様に基づき、介護ソフトからのAPI通信によってデータの送受信を行うことで、介護事業所間のデータ連携を実現する。なお、一定期間は、介護保険資格確認等WEBサービスのブラウザ操作による、「ケアプランデータ連携標準仕様」の改訂について（令和6年10月11日付け厚生労働省老健局高齢者支援課長及び認知症施策・地域介護推進課長通知）で示す標準様式（以下、「4.1版」という）で定義されたCSVファイルのアップロード・ダウンロードによるデータ連携も可能とする。5.0版、4.1版のバージョン間のデータ変換は、介護保険資格確認等WEBサービスで行う。

交付用登録機能は、5.0版で定義された連携仕様に基づき、原案に対して利用者から同意を得て確定したあとケアプラン情報（以下、「交付用ケアプラン情報」と示す。）を介護情報基盤へ登録する機能である。各介護事業所が、各利用者の過去のケアプラン情報を参照できるようになること等により、利便性を向上することを目的としている。

また、利用者からケアプランの同意・確認の記録を取得している業務については、将来的にマイナポータルを通じて、オンラインでもケアプランの同意・確認の記録を取得可能とする仕組みを検討している。同意・確認用登録機能は、5.0版で定義された連携仕様に基づき、マイナポータルから同意・確認を取得するケアプラン情報（以下「同意・確認用ケアプラン情報」と示す。）を登録する機能である。

交付用ケアプラン情報や同意・確認用ケアプラン情報は、介護保険資格確認等WEBサービスでは管理せず、国保中央会で運用する介護情報基盤で情報管理している。このため、介護保険資格確認等WEBサービスではログイン処理を行い、ブラウザ操作によって介護情報基盤に格納されている情報の参照を行う。

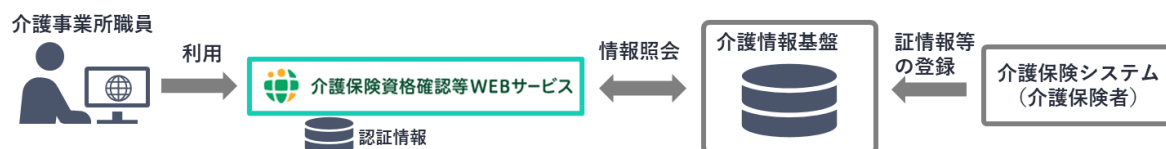


図 1.【介護保険資格確認等WEBサービスの処理フロー概要】介護情報・資格確認

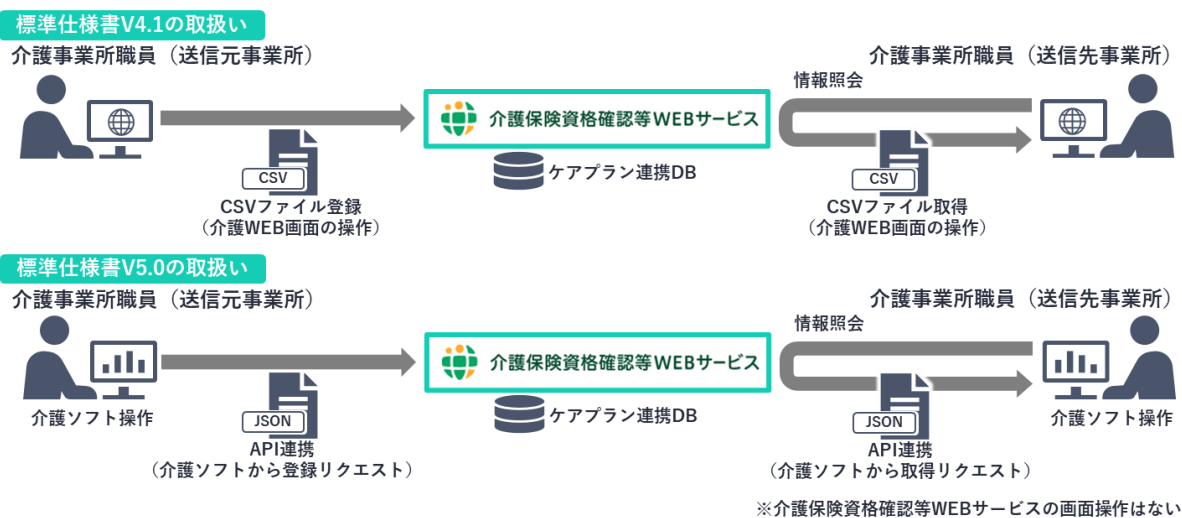


図 2.【介護保険資格確認等WEBサービスの処理フロー概要】介護事業所間のケアプラン情報連携

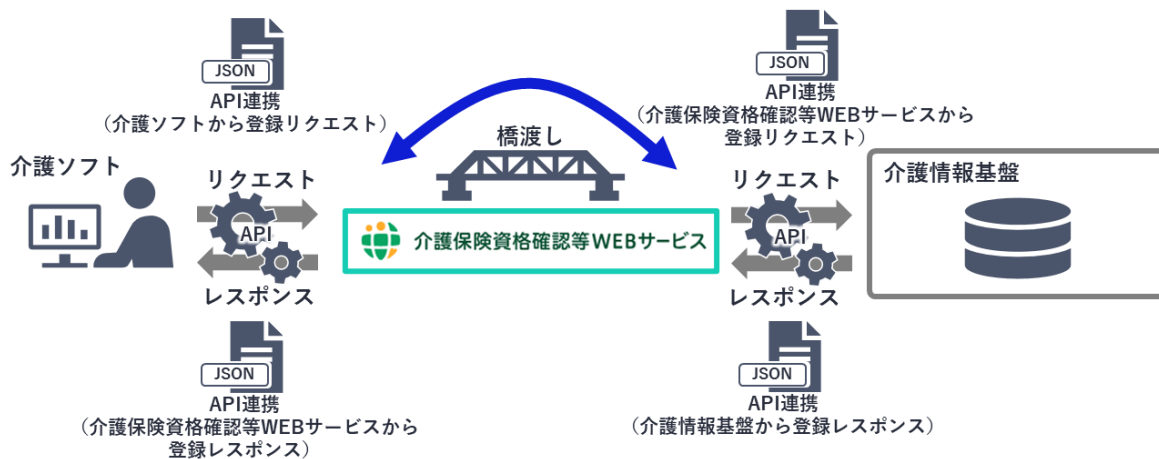


図 3.【介護保険資格確認等WEBサービスの処理フロー概要】ケアプラン情報連携の登録

2.2 システム利用準備

介護保険資格確認等WEBサービス利用を行う際のログイン時には、クライアント証明書・ユーザID・パスワードを用いる。また、マイナンバーカードを用いて、利用者情報を照会する際にはオンライン資格確認ユーザID、オンライン資格確認ユーザパスワード、端末固有IDを用いる。

2.2.1 クライアント証明書

クライアント証明書は国保中央会で運用する認証局から介護事業所単位に払い出される。介護事業所はこのクライアント証明書を介護事業所の端末にインストールし、介護保険資格確認等WEBサービスのログインを行う際に、ブラウザのダイアログからクライアント証明書を選択する。

介護事業所向けのクライアント証明書の種類は以下の2種類あり、いずれかを用いる。

表1 .証明書一覧

No.	証明書名称	説明
1	介護保険証明書	電子請求受付システムを用いて介護保険請求を行う事業所を対象に払い出す証明書。平成26年11月の電子請求稼働から払い出しを行っている。なお、代理人が請求を行う際の介護保険証明書（代理人用）は利用できない。
2	介護DX証明書	電子請求を行っていない事業所（介護保険証明書を保有していない事業所）を対象に払い出す証明書。令和7年10月から払い出しを開始する。なお、ケアプランデータ連携システムを利用、かつ、電子請求を行っていない事業所（介護保険証明書を保有していない事業所）を対象に払い出された請求委任事業所用ケアプラン証明書においては、有効期限までの間において、介護保険資格確認等WEBサービスのログイン時に委任事業所用ケアプラン証明書を使用する。また、請求委任事業所用ケアプラン証明書は介護DX証明書の払い出し開始とともに新規払い出しを停止し、有効期限が切れた事業所は介護DX証明書を取得する。

2.2.2 ユーザの払い出し

2.2.2.1 ユーザの種類

介護保険資格確認等WEBサービスでは、以下の3種類のユーザを用いる。

表2. ユーザー一覧

No.	ユーザ名	説明
1	事業所ユーザ	事業所ユーザは電子請求受付システムにログインするために利用するアカウントと同一のアカウント。事業所ユーザ権限では、管理者ユーザおよび一般ユーザのアカウント管理ができる。 基本的に初回セットアップでのみ利用する想定。介護保険資格確認等WEBサービスの利用者が介護事業所であるかの確認を行う目的で電子請求受付システムに対して認証処理を行う。
2	管理者ユーザ	管理者ユーザは介護事業所の管理者に発行するユーザアカウント。管理者ユーザ権限では、一般ユーザのアカウント管理や、マイナ資格確認アプリ情報の登録、被保険者の介護情報・資格確認等の実務を行う。
3	一般ユーザ	一般ユーザは介護事業所の職員に発行するユーザアカウント。一般ユーザ権限では、被保険者の介護情報・資格確認等の実務を行う。

2.2.2.2 ワンタイムパスワード

ログイン処理時に、ユーザ払い出し作業で設定したメールアドレスもしくは携帯電話番号に対してワンタイムパスワードを通知する。介護事業所職員はワンタイムパスワードを用いてログイン処理を完了する。

2.2.3 オンライン資格確認等システム（マイナ資格確認アプリ情報）の認証

介護保険資格確認等WEBサービスで利用者情報を照会する際に、オンライン資格確認等システムへログインを行う。ログインに用いる情報は「オンライン資格確認ユーザID」、「オンライン資格確認ユーザパスワード」、「端末固有ID」の3種類であり、介護事業所職員が管理者ユーザでのログイン時に介護保険資格確認等WEBサービスへ初回のみ設定する。※端末の追加や変更等にてマイナ資格確認アプリの利用状況が変化する度に登録内容の修正が必要。

2.2.3.1 オンライン資格確認ユーザID、オンライン資格確認ユーザパスワード

介護保険資格確認等WEBサービスで利用者情報を参照する際に、マイナ資格確認アプリからマイナンバーカードを読み取る（PC端末でのカードリーダーによる読み取り、スマートフォンによる読み取り）。この際に、オンライン資格確認ユーザID、オンライン資格確認ユーザパスワードをマイナ資格確認アプリ及び介護保険資格確認等WEBサービスに設定する。

オンライン資格確認ユーザID、オンライン資格確認ユーザパスワードは介護事業所が国保中央会の介護情報基盤等コンタクトセンターに対して申請を行い取得する。

2.2.3.2 端末固有ID

PC端末またはスマートフォンでマイナ資格確認アプリの設定後に、マイナ資格確認アプリ上で読み取り端末単位で管理される端末固有IDが表示可能となる。

2.2.3.3 マイナ資格確認アプリの利用

介護事業所が介護保険資格確認等WEBサービスを利用する際、マイナンバーカードによる読み取り（PC端末でのカードリーダーによる読み取り、スマートフォンによる読み取り）により登録を行う。読み取りを行う際は、マイナ資格確認アプリを用いる。

マイナンバーカード利用しない場合や、マイナ健康保険証の登録を完了していない利用者については、「保険者番号」「被保険者番号」「生年月日」「性別コード」を用いて利用者を登録する。

2.3 画面遷移イメージ

介護保険資格確認等WEBサービスでは、以下の画面遷移で情報参照、情報登録を行う。

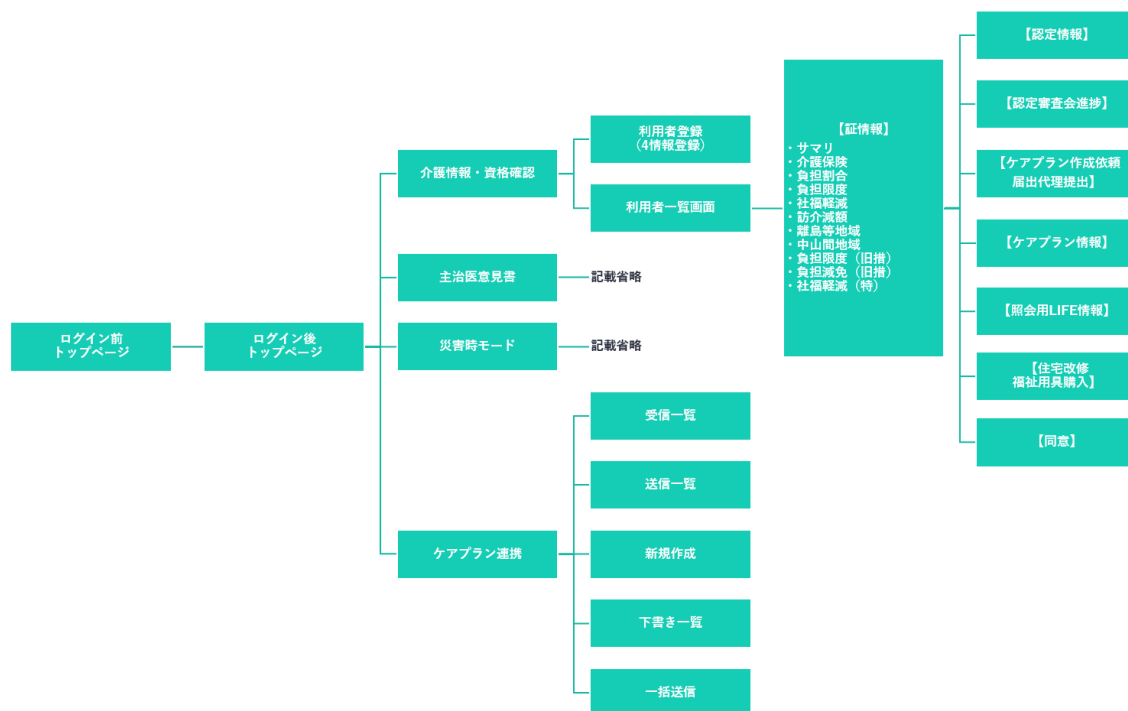


図4. 介護保険資格確認等WEBサービスにおける画面遷移

ログイン後トップページ以降の各画面についての概要は下表のとおり。

表3.画面一覧

No.	メニュー	画面	説明
1	介護情報・資格 確認	利用者登録 (4情報登 録)	マイナンバーカードを保持しない利用者に対して情報照会するために、「保険者番号」「被保険者番号」「生年月日」「性別コード」を用いて参照対象の利用者を登録する。
2		証情報	介護保険被保険者証等の情報を表示する。

3		認定情報	要介護認定の情報として「認定調査票」「主治医意見書」の情報を表示する。
4		認定審査会進捗	認定審査会の進捗状況を表示する。
5		ケアプラン作成 依頼届出 代理提出	居宅介護支援事業所・地域包括支援センターがケアプラン作成する際の届出を提出（システム登録）する。また、届出状況を表示する。
6		LIFE	LIFEシステムに登録された情報の一部を表示する。
7		ケアプラン	交付用ケアプラン情報を表示する。
8		住宅改修・福祉用具購入	住宅改修・福祉用具購入の給付可能額等を表示する。
9		同意	情報参照にかかる利用者の同意をシステム登録する。また、登録状況を表示する。
10	主治医意見書	—	医師等が主治医意見書・請求書の登録、保険者への提出を行う。
11	災害時モード	—	災害地域の事業所において介護情報・資格確認の範囲を拡大する。※特別運用時の機能。
12	ケアプラン情報連携（事業所間連携）	送信一覧	送信したケアプラン情報を一覧表示する。
13		受信一覧	受信したケアプラン情報を一覧表示する
14		新規作成	単一事業所に対してケアプラン情報を送信する。
15		一括送信	複数の宛先事業所を指定しケアプラン情報を一括送信する。
16		下書き一覧	送信前のケアプラン情報を一時保存する。

2.4 サービス種類と職種による処理権限

介護保険資格確認等WEBサービス種類は管理者ユーザと一般ユーザのユーザ払い出し時に「サービス種類」と「職種」を設定する。ログイン時にユーザIDに紐づく「サービス種類」と「職種」からメニューの表示・非表示の制御を行う。例えば、介護保険被保険者証情報の参照については「サービス種類」「職種」による制限がなく参照可能とするが、認定情報については「サービス種類」が「居宅介護支援」、「職種」が「ケアマネジャー」である場合に参照可能とする取り扱いとなる。

サービス種類の設定は当該事業所が指定を受けているサービス種類から、払い出しを行うユーザに紐づくサービス種類を設定する。例えば、同じ事業所番号で通所リハビリテーションと介護老人保健施設を併設・運営している事業所は、ユーザ払い出し時に通所リハビリテーションと介護老人保健施設が設定対象として画面表示され、払い出し対象のユーザの勤務に沿って設定する（複数選択可）。

職種は「ケアマネジャー」「医師」など払い出し対象のユーザの職種を設定する（複数選択可）。

3. 介護保険資格確認等WEBサービスのAPI

介護ソフトと介護保険資格確認等WEBサービスのAPIについて記載する。

なお、記載するAPIとしては、介護情報・資格確認に関するAPIと、ケアプラン情報連携に関するAPI（事業所間連携、交付用登録、同意・確認用登録）がある。

3.1 APIの処理概要（介護情報・資格確認）

介護保険資格確認等WEBサービスは、介護情報・資格確認等に係る証情報等データの管理、保管は行わず、介護情報基盤で管理する。このため、介護保険資格確認等WEBサービスは介護ソフトからのAPIリクエストを受付後にチェック処理を行い、介護情報基盤へ情報照会のリクエストを行う。介護情報基盤からのレスポンスを介護ソフトへ返却する。



図 1．介護保険資格確認等WEBサービスのAPIの概要：介護情報・資格確認

介護保険資格確認等WEBサービスのブラウザで作業する情報照会処理・情報登録処理のうち、介護ソフトのAPI対象となる処理は以下のとおりである。なお、参考としてAPI対象となる処理についての介護保険資格確認等WEBサービスのブラウザ画面のイメージについて「別紙 1．APIと関連する画面」に示す。

表 4.介護ソフト公開対象機能（介護情報・資格確認）

No.	メニュー	画面	対象・非対象
1	介護情報・資格確認	利用者登録（4情報登録）	非対象
2		証情報	対象
3		認定情報	対象
4		認定審査会進捗	対象
5		ケアプラン作成依頼届出代理提出	非対象
6		LIFE	非対象
7		ケアプラン	非対象
8		住宅改修・福祉用具購入	対象
9		同意	非対象
10	主治医意見書	—	非対象
11	災害時モード	—	非対象

3.1.1 データ連携のパターン

介護情報・資格確認に係る介護ソフトと介護保険資格確認等WEBサービスのAPIによる情報連携は、「情報参照（取得）」の1パターンであり、「情報登録」や「情報削除」は行わない。

3.1.2 取得項目の留意事項

介護情報基盤で管理する証情報等の項目数と介護保険資格確認等WEBサービスで画面表示する証情報等の項目数には差異がある。介護情報基盤の情報管理上、介護保険資格確認等WEBサービスから情報照会した項目には画面表示対象を超える項目が含まれるため、表示するときに取得結果から絞り込み、画面表示を行う。

介護ソフト向けAPIについても情報の取得単位は同じであるため、APIレスポンスには画面表示対象

を超える項目が含まれる。「別紙 2-1. API仕様書（介護情報・資格確認）」にAPI取得するそれぞれの項目に対して介護保険資格確認等WEBサービスで画面表示する対象項目を示す。

3.1.3 データ連携方法

介護保険資格確認等WEBサービスと介護ソフト間での情報連携は、インターネット回線を通じた伝送によって連携する。データ連携方法は、介護保険資格確認等WEBサービスが公開するリクエスト用のAPIを用いた、クライアントアプリケーションもしくはWEBアプリケーションによるJSON形式での情報送付とする。

3.2 APIの処理概要（ケアプラン情報連携）

事業所間連携用のケアプラン情報の一時保管は、介護保険資格確認等WEBサービスで行う。介護保険資格確認等WEBサービスは介護ソフトからのAPIリクエストを受付後にチェック処理を行い、介護ソフトへレスポンスを返却する。

交付用ケアプラン情報や同意・確認用ケアプラン情報の管理・保管は、介護保険資格確認等WEBサービスでは行わず、介護情報基盤で行う。このため、介護保険資格確認等WEBサービスは介護ソフトからのAPIリクエストを受付後にチェック処理を行い、介護情報基盤へ情報照会のリクエストを行う。介護情報基盤からのレスポンスを介護ソフトへ返却する。

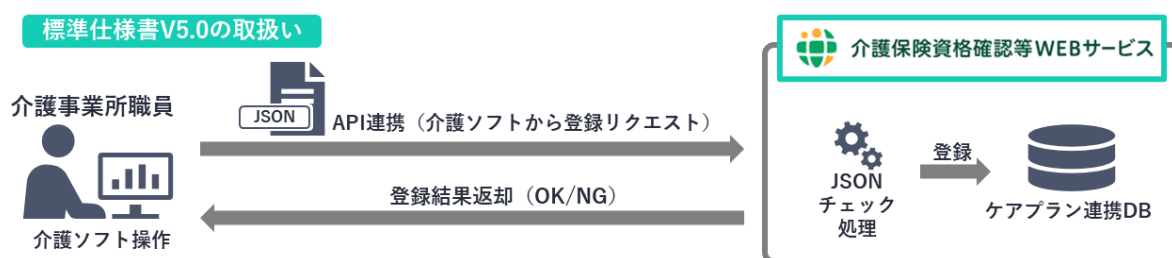


図 8. 介護保険資格確認等WEBサービスのAPIの概要：事業所間連携



図 2. 介護保険資格確認等WEBサービスのAPIの概要：交付用登録、同意・確認用登録

介護保険資格確認等WEBサービスのブラウザで作業する情報照会処理・情報登録処理のうち、介護ソフトのAPI対象となる処理は以下のとおりである。なお、参考としてAPI対象となる処理についての介護保険資格確認等WEBサービスのブラウザ画面のイメージについて「別紙 1. APIと関連する画面」に示す。

表 5. 介護ソフト公開対象機能（ケアプラン情報連携）

No.	メニュー	画面	対象・非対象
1	ケアプラン情報連	送信一覧	対象
2	携（事業所間連	受信一覧	対象

3	携)	新規作成	対象
4		一括送信	非対象 (複数宛先に送信する場合は、宛先ごとに送信APIを呼び出して送信する)
5		下書き	非対象

3.2.1 データ連携のパターン

ケアプラン情報連携に係る介護ソフトと介護保険資格確認等WEBサービスのAPIによる情報連携は、「情報参照（取得）」と「情報登録」がある。

3.2.2 データ連携方法

介護保険資格確認等WEBサービスと介護ソフト間での情報連携は、インターネット回線を通じた伝送によって連携する。データ連携方法は、介護保険資格確認等WEBサービスが公開するリクエスト用のAPIを用いた、クライアントアプリケーションもしくはWEBアプリケーションによるJSON形式での情報送付とする。

3.2.3 IFバージョンの取り扱い

呼び出し元の介護ソフトが対応する別紙5-1～5-3、2-2-1～2-2-3の版を識別し、版ごとに適切な処理を実施できるようにするため、本連携では「IFバージョン」を用いた方式を採用する。

介護保険資格確認等Webサービスでは、APIのURLにIFバージョンを含めることで、対向システムの介護ソフトがどの版に基づいて連携しているかを識別する。

IFバージョンは、別紙5-1「ケアプランデータ連携標準仕様 更新履歴」における「版数」欄で定義される値とする。また、各版で提供するAPIのURLは、当該「版数」を含む命名規則に従って定義する。

3.2.4 バージョン変換

事業所間連携機能では、送信元事業所と送信先事業所の介護ソフトが4.1版と5.0版（別紙5-1～5-3、2-2-1～2-2-3）で異なる場合、介護保険資格確認等WEBサービス上で4.1版（CSVファイル）と5.0版（JSON形式）間のバージョン変換を行う。4.1版から5.0版への変換は、5.0版介護ソフトからの受信APIリクエスト時に、5.0版から4.1版への変換は、介護保険資格確認等WEBサービス画面でのCSVファイルダウンロード操作時に、変換後のデータまたはファイルを返却する。

バージョン変換にあたっての留意点は、「別紙5-1. ケアプランデータ連携標準仕様」を参照すること。

3.3 API仕様（共通）

3.3.1 共通仕様

本APIは、REST(Representational State Transfer)形式で提供し、リクエスト・レスポンスともにJSON形式を採用する。

認証・認可ではクライアント証明書情報、ユーザID・パスワードを使用し、すべてのリクエストはHTTPS通信で行う。

ログインAPI実行時に、クライアント証明書認証とユーザID・パスワードによる多要素認証を行う。認証成功後、アクセストークンおよびリフレッシュトークンを発行する。なおクライアント証明書はTLSハンドシェ

イク時のサーバ側要求により連携する。なお認証失敗回数が介護保険資格確認等WEBサービスの定めるしきい値を越えた場合、一定時間アカウントがロックされる。

介護ソフトは、以降のAPI呼び出し時にアクセストークンをリクエストヘッダに設定する。アクセストークンの有効期限（1時間）が切れて認証エラー（401）となった場合はリフレッシュトークンによる認証を行い、アクセストークンの再発行を行う。その後、再度当該APIを実行する。

リフレッシュトークンの有効期限（8時間）が切れて認証エラー（401）となった場合は再度ログイン処理から再実行する。

レスポンスのステータスコードは標準的なHTTPステータスコードに準拠し、エラー発生時にはステータスコードに準拠したメッセージを返却する。エラー発生時に使用するHTTPステータスコードは、不正リクエスト（400）、認証エラー（401）、認可エラー（403）、接続エラー（404）、システムエラー（500、503）を使用し返却する。その他のインタフェース個別の処理結果に関するエラーにおいては、HTTPステータスコードに正常（200）を返却し、JSON項目の処理結果にエラーを設定して返却する。

また、正常時のレスポンスボディには介護情報基盤でのAPIの取得結果を設定する処理結果（processing_results）、処理結果詳細（processing_results_detail）を、エラー発生時のレスポンスボディには介護情報基盤に対してAPI実行する前に介護WEB内でエラーが起こった場合に設定するエラーコード（errorCode）、エラーメッセージ（errorMessage）を用意し、共通的に使用する。

3.3.1.1 文字コード

介護業務データが格納される介護情報基盤に準拠した以下を用いる。

- ・文字セット：「JISX 0213:2012」
- ・文字コード：「JISX 0221:2020」（日本語環境でのUnicode）
- ・符号化形式：「UTF-8」

3.3.1.2 フォーマット

介護保険資格確認等WEBサービスのAPIリクエスト、レスポンスのフォーマットは以下のとおり。具体的な仕様は「別紙2-1.API仕様書（介護情報・資格確認）」、「別紙2-2-1.API仕様書（ケアプランデータ連携）」を参照。

（1）APIリクエスト（介護ソフト用ログインAPI）

```
POST /api/v1/sample/samplerequest/bulk HTTP/1.1
Host: api.sample-host.jp
Accept: application/json; charset=UTF-8
Content-Type: application/json; charset=UTF-8

{
  "careFacilityId": "1234567890"
  "userId": "sampleUser"
  "password": "P@ssW0rd"
}
```

(2) APIリクエスト (介護ソフト用トークンリフレッシュAPI)

```
POST /api/v1/sample/samplerequest/bulk HTTP/1.1
Host: api.sample-host.jp
Accept: application/json; charset=UTF-8
Content-Type: application/json; charset=UTF-8

{
  "refresh_token": "YYYYYYYYYYY"
}
```

(3) APIリクエスト (認証時以外)

```
POST /api/v1/sample/samplerequest/bulk HTTP/1.1
Host: api.sample-host.jp
Accept: application/json; charset=UTF-8
Content-Type: application/json; charset=UTF-8Authorization: Bearer
XXXXXXXXXX

{
  "userIds": [
    "1234567890",
    "9876543210",
    "1122334455"
  ]
}
```

(4) APIレスポンス (異常時)

```
HTTP/1.1 401 NG
Content-Type: application/json; charset=UTF-8
AuthResult: 1
{
  "errorCode": "1",
  "errorMessage": "介護WEBのユーザーID/パスワードが誤っています。"
}
```

(5) APIレスポンス (正常時)

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
AuthResult: 0

{
  "id": "1234567890",
  "name": "sample"
}
```

}

3.3.1.3 呼び出し手順

APIを呼び出す際の基本的な手順は以下のとおりである。

(1)介護ソフトは、ログインAPIを実行し、クライアント証明書認証およびユーザID・パスワードによる多要素認証を行う。なお、クライアント証明書はTLSハンドシェイク時にサーバ側から要求されるものとし、ユーザID・パスワードは、事前に事業所職員がブラウザ操作により払い出したものを使用する。

(2)認証に成功した場合、介護WEBはアクセストークンおよびリフレッシュトークンを発行する。

(3)介護ソフトは、以降のAPI呼び出し時に、HTTPリクエストヘッダへアクセストークンを設定し、「別紙2-1.API仕様書（介護情報・資格確認）」および「別紙2-2-1.API仕様書（ケアプランデータ連携）」の記載に従い、適切なエンドポイントに対してHTTPリクエストを送信する。

(4)介護ソフトはレスポンスを受信し、HTTPステータスコード、レスポンスボディの処理結果・処理結果詳細（正常時）、エラーコード・エラーメッセージ（エラー発生時）を確認する。

(5)アクセストークンの有効期限切れ等により認証エラーが発生した場合は、リフレッシュトークンを使用してアクセストークンを再発行したうえで、再度当該APIを実行する。

(6)リフレッシュトークンの有効期限切れ等により再発行ができない場合は、ログインAPIから再実行する。

3.3.1.4 共通処理

介護保険資格確認等WEBサービスでは、以下の共通処理を全APIリクエストに対して適応する。

(1) 認証・認可処理

すべてのAPIリクエストに対して、HTTPS通信におけるTLSハンドシェイク時にクライアント証明書の有効性を検証する。

ログインAPIでは、クライアント証明書認証に加え、ユーザID・パスワードによる認証を行う。認証成功時には、アクセストークンおよびリフレッシュトークンを発行する。

ログイン後に実行される各APIでは、HTTPリクエストヘッダに設定されたアクセストークンの有効性を検証する。あわせて、アクセストークンに紐づくユーザIDおよび権限情報を基に、介護保険資格確認等WEBサービスに対して連携可能な情報であるかを検証する。

権限と対応可能な処理については「別紙 4. サービス種類と職種の処理権限」に示す。

(2) 入力値検証

リクエストパラメータの妥当性検証を行い、不正な値が含まれている場合はエラーオブジェクトを返却する。

(3) ロギング処理

セキュリティ監視およびトラブルシューティングのため、リクエスト・レスポンスの内容をログに記録する。

(4) エラーハンドリング

発生したエラーを適切に捕捉し、エラーに対応するメッセージを返却する。介護ソフトではエラーの内容に応じて、データの修正などの適切な対応を行うこと。エラーコードおよびエラーの内容については、「別紙 2 -1.API仕様書（介護情報・資格確認）」、「別紙 2 -2-1.API仕様書（ケアプランデータ連携）」に示す。

3.3.1.5 異常・障害時の対応

介護ソフトから介護保険資格確認等WEBサービスへ送付した情報にエラーが検出された場合、介護保険資格確認等WEBサービスからエラーメッセージを返却する。介護事業所職員はエラー内容を確認のうえ、必要な対応を行う。個別のエラー内容の詳細に関しては、「別紙 2 -1.API仕様書（介護情報・資格確認）」、「別紙 2 -2-1.API仕様書（ケアプランデータ連携）」を参照。

介護ソフトからのリクエストに対して介護保険資格確認等WEBサービスからのレスポンスがない場合は、再度時間を置いてリクエストを実施すること。

TLSハンドシェイク時に提示されたクライアント証明書は、介護保険資格確認等WEBサービスの基盤機能で検証する。検証に失敗した場合はアプリケーション側にリクエストは到達しないため業務エラーは返却しない。

表 6.介護ソフトと介護保険資格確認等WEBサービス間で検知するエラー及び障害

エラー/障害	ケース	エラー返却タイミング	対処内容
エラー	ユーザID/パスワード誤りのため認証ができない	介護保険資格確認等WEBサービスでJSON受信時	介護保険資格確認等WEBサービスから形式エラーを受信し、エラー内
	利用するAPIが認証ユーザ	介護保険資格確	

	の権限では実行できない	認等WEBサービスでJSON受信時	容に応じた対処を行う。
	オンライン資格確認ID/パスワード/端末固有IDが登録されていない	介護保険資格確認等WEBサービスでJSON受信時	
	バリデーションエラー（入力項目の不足）	介護保険資格確認等WEBサービスでJSON受信時	
	バリデーションエラー（型不一致）	介護保険資格確認等WEBサービスでJSON受信時	
	バリデーションエラー（桁数不一致）	介護保険資格確認等WEBサービスでJSON受信時	
	フォーマットエラー（API毎のタグ誤り）	介護保険資格確認等WEBサービスでJSON受信時	
	介護保険資格確認等WEBサービス内部処理エラー	介護保険資格確認等WEBサービスでJSON受信時	
障害	介護ソフトからデータを送付するも到着しない（介護WEBから処理結果が返却されない）	-	再度、APIの呼び出しを実施する。

表 7.介護保険資格確認等WEBサービスと介護情報基盤で検知するエラー及び障害

エラー/障害	ケース	エラー返却タイミング	対処内容
エラー	介護情報基盤でエラー	各API呼び出し時	再度、APIの呼び出しを実施する。

3.3.2 正常レスポンス時の警告メッセージの対応

正常レスポンス時に警告メッセージが含まれる場合がある。

その場合は、ステータスコード200、レスポンスボディの処理結果、処理結果詳細に値が入り返却する。具体的な内容は「別紙 2 -1.API仕様書（介護情報・資格確認）」、「別紙 2 -2-1.API仕様書（ケアプランデータ連携）」の「警告メッセージ一覧」を参照。

例として、利用者一覧情報の参照（WEB-IF_KST_01）で取得結果が0件の場合を以下に示す。

表 8 . 利用者一覧情報の参照（WEB-IF_KST_01）で取得結果が0件の場合のレスポンス

ステータスコード	200
処理結果	1（失敗）
処理結果詳細	テーブルにデータが存在しません

3.3.3 APIの呼び出し順序の考慮

事前に介護ソフト用ログインAPIを実行してアクセストークンを取得した後に、介護ソフト公開対象機能のAPIを呼び出しする。それ以外では介護ソフトがAPIを利用する際に順序に関して考慮すべきことはない。

3.3.4 伝送使用上の留意事項

介護保険資格確認等WEBサービスのAPIを安全に利用するために、介護ソフトの実装にあたり以下のセキュリティ上の留意事項を厳守すること。

(1) 認証情報の管理

- ・認証・認可はクライアント証明書・ユーザID・パスワード（「管理者ユーザ」もしくは「一般ユーザ」）を使用する。APIによる情報連携をする事前作業として、事業所単位でクライアント証明書を取得し、介護事業所職員によるブラウザ操作でユーザID・パスワードを職員個人単位で払い出す。
 - ・ユーザID・パスワードはソースコード内にハードコーディングせず、専用の構成ファイル等で管理する。
 - ・認証情報をログファイルに記録しない。
 - ・クライアント証明書情報をHTTPS通信におけるTLSハンドシェイク時に連携するように実装する。
 - ・ログインAPIでは、クライアント証明書認証およびユーザID・パスワード認証を行い、認証成功後にアクセストークンとリフレッシュトークンを発行する。
- 介護ソフトは以降のAPI呼び出し時にアクセストークンをヘッダへ設定する。
アクセストークンの有効期限切れ時はリフレッシュトークンにより再発行し、対象APIを再実行する。リフレッシュトークンも有効期限切れの場合は、再度ログイン処理を行う。
- ・アクセストークンの有効期限は60分、リフレッシュトークンの有効期限は8時間とする。

(2) 通信の保護

- ・介護保険資格確認等WEBサービスとの通信は必ずHTTPS（TLS1.3以上）を使用する。
- ・SSL証明書の検証を無効化（証明書のチェックの迂回）しない。

(3) 入出力データの検証

- ・APIで送信するデータは事前にサニタイズし、インジェクション攻撃を防止すること。
- ・ユーザ入力値をAPIリクエストに含める際は、桁数、入力値の必須、英数字のみ等の形式チェックを行い不正な値がないか検証する。

(4) 介護ソフトのエラー処理

- ・介護保険資格確認等WEBサービスが提供するAPIを使用して発生したエラーに対する介護ソ

フトの処理におけるスタックトレース等で、技術的詳細情報が外部に漏れないように注意する。

(5) 脆弱性対策

- ・定期的にセキュリティスキャンの実施や、潜在的な脆弱性を検出・修正する。

介護保険資格確認等WEBサービスでは、APIの提供にあたって以下の機能を担保する。

- ・認証基盤（クライアント証明書認証、ベーシック認証、アクセストークン）の提供
- ・API利用時のログ等、監査証跡の保管

介護ソフトでは、APIの利用にあたって以下を適切に管理する責を負うものとする。

- ・認証基盤で必要となるクライアント証明書、認証情報（ID/パスワード）アクセストークンの安全な保管
- ・利用ユーザに対する適切な権限付与と、離任時の権限失効の運用

3.3.5 API連携実装のサンプルコード

ここではAPI連携をJava17で最小構成の実装を行った場合のサンプルコードを示す。

3.3.5.1 ログインAPI実行用サンプル

```
import lombok.Getter;
import lombok.Setter;
import javax.net.ssl.KeyManagerFactory;
import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManagerFactory;
import java.io.InputStream;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;
import java.net.URI;
import java.nio.file.Files;
import java.nio.file.Path;
import java.security.KeyStore;
import com.fasterxml.jackson.databind.ObjectMapper;

public class LoginSample {
    public static void main(String[] args) throws Exception {
        // クライアント証明書のTLS設定
        KeyStore keyStore = KeyStore.getInstance("PKCS12");
        try (InputStream is = Files.newInputStream(Path.of("<<クライアント証明書保存パス>>"))) {
            keyStore.load(is, "<<クライアント証明書PASSWORD>>".toCharArray());
        }
        KeyManagerFactory kmf = KeyManagerFactory.getInstance(KeyManagerFactory.getDefaultAlgorithm());
        kmf.init(keyStore, "<<クライアント証明書PASSWORD>>".toCharArray());
        TrustManagerFactory tmf = TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());
        tmf.init((KeyStore) null);
        SSLContext sslContext = SSLContext.getInstance("TLSv1.3");
        sslContext.init(kmf.getKeyManagers(), tmf.getTrustManagers(), null);
        HttpClient client = HttpClient.newBuilder()
            .sslContext(sslContext).build();

        LoginRequest request = new LoginRequest();
        // 事業所番号・ユーザID・パスワードを設定
        request.setCareFacilityId("1234567890");
        request.setUserId("user001");
        request.setPassword("pass123");

        String json = new ObjectMapper().writeValueAsString(request);
        // 介護ソフト用ログインAPIにリクエスト
        HttpRequest httpRequest = HttpRequest.newBuilder()
            .uri(URI.create("https://dev.app.ltcip.jp/web-api/kaigoweb-v1/login"))
            .header("Content-Type", "application/json")
            .POST(HttpRequest.BodyPublishers.ofString(json))
            .build();
        // レスポンスを取得
        HttpResponse<String> response = client
            .send(httpRequest, HttpResponse.BodyHandlers.ofString());
        LoginResponse result = new ObjectMapper()
            .readValue(response.body(), LoginResponse.class);
        // 次のページへ続く
    }
}
```

```

// 前のページから続く
// レスポンス結果からHTTPステータスコードとレスポンスボディ部を取得
System.out.println("Status: " + response.statusCode());
System.out.println("Access Token: " + result.getAccess_token());
System.out.println("Refresh Token: " + result.getRefresh_token());
}

// リクエストボディ部用のクラス
@Getter
@Setter
public static class LoginRequest {
    private String careFacilityId;
    private String userId;
    private String password;
}

// レスポンスボディ部用のクラス
@Getter
@Setter
public static class LoginResponse {
    private String access_token;
    private String refresh_token;
    private String access_token_expires_in;
    private String refresh_token_expires_in;
}
}

```

3.3.5.2 介護ソフト公開対象機能API実行とトークンリフレッシュAPI実行用サンプル

介護ソフト公開対象機能API（例：「利用者一覧情報の参照API」）

```
import java.util.List;
import lombok.Getter;
import lombok.Setter;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;
import java.net.URI;
import com.fasterxml.jackson.databind.ObjectMapper;

public class UserListSample {

    public static void main(String[] args) throws Exception {
        // 3.3.5.1 ログインAPI実行用サンプルで取得したアクセストークンとリフレッシュトークン
        String accessToken = "access-token";
        final String REFRESH_TOKEN = "refresh-token";
        HttpClient client = HttpClient.newHttpClient();
        // 利用者一覧情報の参照APIにリクエスト
        HttpRequest userListRequest = HttpRequest.newBuilder()
            .uri(URI.create("https://dev.app.itcip.jp/web-api/kaigoweb-v1/userlist"))
            .header("Content-Type", "application/json")
            .header("Authorization", "Bearer " + accessToken) // アクセストークンをBearerでセット
            .header("Disaster-Mode-Target-Flag", "0") // 災害時モード対象フラグヘッダは必須
            .POST(HttpRequest.BodyPublishers.noBody()) // 本APIはリクエストボディ部が不要
            .build();
        // レスポンスを取得
        HttpResponse<String> response = client.send(userListRequest,
            HttpResponse.BodyHandlers.ofString());
        // 401（認証エラー）の場合はトークンリフレッシュを行って再リクエスト
        if (response.statusCode() == 401) {
            // アクセストークンの再発行（内容は、2）トークンリフレッシュAPIで解説）
            accessToken = TokenRefreshSample.refresh(REFRESH_TOKEN, client);
            userListRequest = HttpRequest.newBuilder()
                .uri(URI.create("https://dev.app.itcip.jp/web-api/kaigoweb-v1/userlist"))
                .header("Content-Type", "application/json")
                .header("Authorization", "Bearer " + accessToken) // 再発行したアクセストークンをセット
                .header("Disaster-Mode-Target-Flag", "0")
                .POST(HttpRequest.BodyPublishers.noBody())
                .build();
            // レスポンスを取得
            response = client.send(userListRequest, HttpResponse.BodyHandlers.ofString());
        }
        // レスポンス結果からHTTPステータスコードとレスポンスボディ部を取得
        UserListResponse result = new ObjectMapper()
            .readValue(response.body(), UserListResponse.class);
        System.out.println("Status: " + response.statusCode());
        System.out.println("Result: " + result.getProcessing_results());
        System.out.println("被保険者証記載の氏名 " + result.getUser_info_list().get(0).getName());
    }

    // 次のページへ続く
```

```
// 前のページから続く
// レスポンスボディ部用のクラス
@Getter
@Setter
static class UserListResponse {
    private String processing_results;
    private String processing_results_detail;
    private List<UserInfoList> user_info_list;
    // 以下、レスポンスの全項目のプロパティを設定
}

// レスポンスボディ.利用者情報一覧用のクラス
@Getter
@Setter
static class UserInfoList {
    private String care_insure_provider_number; // 証記載介護保険者番号
    private String care_insurer_number; // 介護保険被保険者番号
    private String name; // 被保険者証記載の氏名
    // 以下、レスポンス.利用者情報一覧の全項目のプロパティを設定
}
}
```

トークンリフレッシュAPI

```
import lombok.Getter;
import lombok.Setter;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;
import java.net.URI;
import com.fasterxml.jackson.databind.ObjectMapper;

public class TokenRefreshSample {

    static String refresh(String refreshToken, HttpClient client) throws Exception {
        Request req = new Request();
        req.setRefresh_token(refreshToken);
        // トークンリフレッシュAPIにリクエスト
        HttpRequest httpRequest = HttpRequest.newBuilder()
            .uri(URI.create("https://dev.app.ltcip.jp/web-api/kaigoweb-v1/token/refresh"))
            .header("Content-Type", "application/json")
            // Authorization ヘッダは付与しないこと
            .POST(HttpRequest.BodyPublishers.ofString(new ObjectMapper().writeValueAsString(req)))
            .build();
        // レスポンスを取得
        HttpResponse<String> response = client.send(httpRequest,
            HttpResponse.BodyHandlers.ofString());
        // リフレッシュトークンの有効期限切れ
        if (response.statusCode() != 200) {
            throw new RuntimeException("API failed. Status: " + response.statusCode());
        }
        // 再発行したアクセストークンを返却
        return new ObjectMapper().readValue(response.body(), Response.class).getAccess_token();
    }

    // リクエストボディ部用のクラス
    @Getter
    @Setter
    static class Request {
        private String refresh_token;
    }

    // レスポンスボディ部用のクラス
    @Getter
    @Setter
    static class Response {
        private String access_token;
        private String refresh_token;
    }
}
```

3.3.6 責任分界点

介護保険資格確認等WEBサービスでは、APIの提供にあたって以下の機能を担保する。

- ・ 認証基盤（クライアント証明書認証、ベーシック認証、アクセストークン）の提供
- ・ API利用時のログ等、監査証跡の保管

介護ソフトでは、APIの利用にあたって以下を適切に管理する責を負うものとする。

- ・ 認証基盤で必要となるクライアント証明書、認証情報（ID/パスワード）、アクセストークンの安全な保管
- ・ 利用ユーザに対する適切な権限付与と、離任時の権限失効の運用

以上