

デマンドバスシステム連携API システム設計書



目次

1. 開発スコープ	- 3 -
1-1. 概要	- 3 -
1-2. システムを利用する業務全体像とシステム利用フロー	- 4 -
2. デマンドバス連携 API システム：機能要件（FN/SL/AL/CO/HW/IF/UI）	- 17 -
2-1. システム機能（FN）	- 17 -
2-1-1. システムアーキテクチャ	- 17 -
2-1-2. システム機能一覧	- 18 -
2-1-3. システム機能の詳細	- 20 -
2-1-4. ソフトウェア・ライブラリ（SL）の詳細	- 48 -
2-1-5. 数理モデル・アルゴリズム（AL）の詳細	- 53 -
2-2. システムコンポーネント（CO）	- 54 -
2-2-1. システムコンポーネント図	- 54 -
2-2-2. システムコンポーネント一覧	- 55 -
2-3. ハードウェア（HW）	- 58 -
2-3-1. ハードウェアアーキテクチャ	- 58 -
2-3-2. ハードウェア一覧	- 58 -
2-3-3. ハードウェアの詳細	- 58 -
2-4. データインターフェース（IF）	- 59 -
2-4-1. データアーキテクチャ	- 59 -
2-4-2. データインターフェース一覧	- 60 -
2-4-3. データインターフェースの詳細	- 61 -
2-5. ユーザーインターフェース（UI）	- 265 -
2-5-1. 画面遷移図	- 265 -
2-5-2. ユーザーインターフェース一覧	- 266 -
2-5-3. ユーザーインターフェースの詳細	- 268 -
3. デマンドバス連携 API システム：非機能要件（NF）	- 278 -
3-1. 非機能要件一覧	- 278 -
3-2. 非機能要件の詳細	- 278 -
4. 実証調査に利用するデータ（DT）	- 279 -
4-1. 実証調査に利用するデータ一覧	- 279 -
4-2. 実証調査に利用するデータの詳細	- 279 -
5. 用語集	- 280 -

1. 開発スコープ

1-1. 概要

デマンドバスシステムとサードパーティが提供する MaaS アプリ等のユーザー向けアプリとの相互運用性（インターオペラビリティ）を確保するため、両システムが連携するための連携インターフェースの標準仕様を開発する。また、その有用性等を検証するため、実際のデマンドバスシステムと MaaS アプリを用い、標準的なインターフェースに準拠して両システムを改修する。

標準的なインターフェースとは、デマンドバスを予約するために必要な日時等のデータをインプットとし、デマンドバスの予約結果をアウトプットとすることを基本とするもので、外部サービスからデマンドバスシステムを利用できるようにするための API とする。

検証用システムは MaaS アプリである my route と、デマンドバスシステムである MONET Move の検証環境とし、以下のような構成とする。

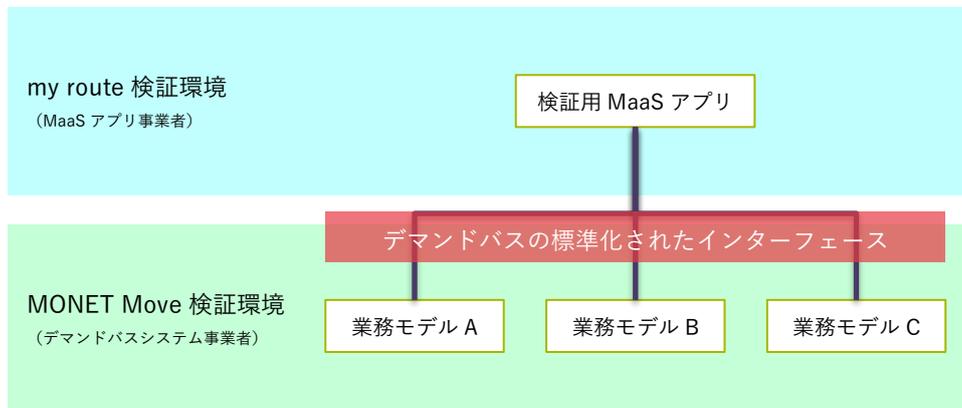


図 1-1 システム構成概要

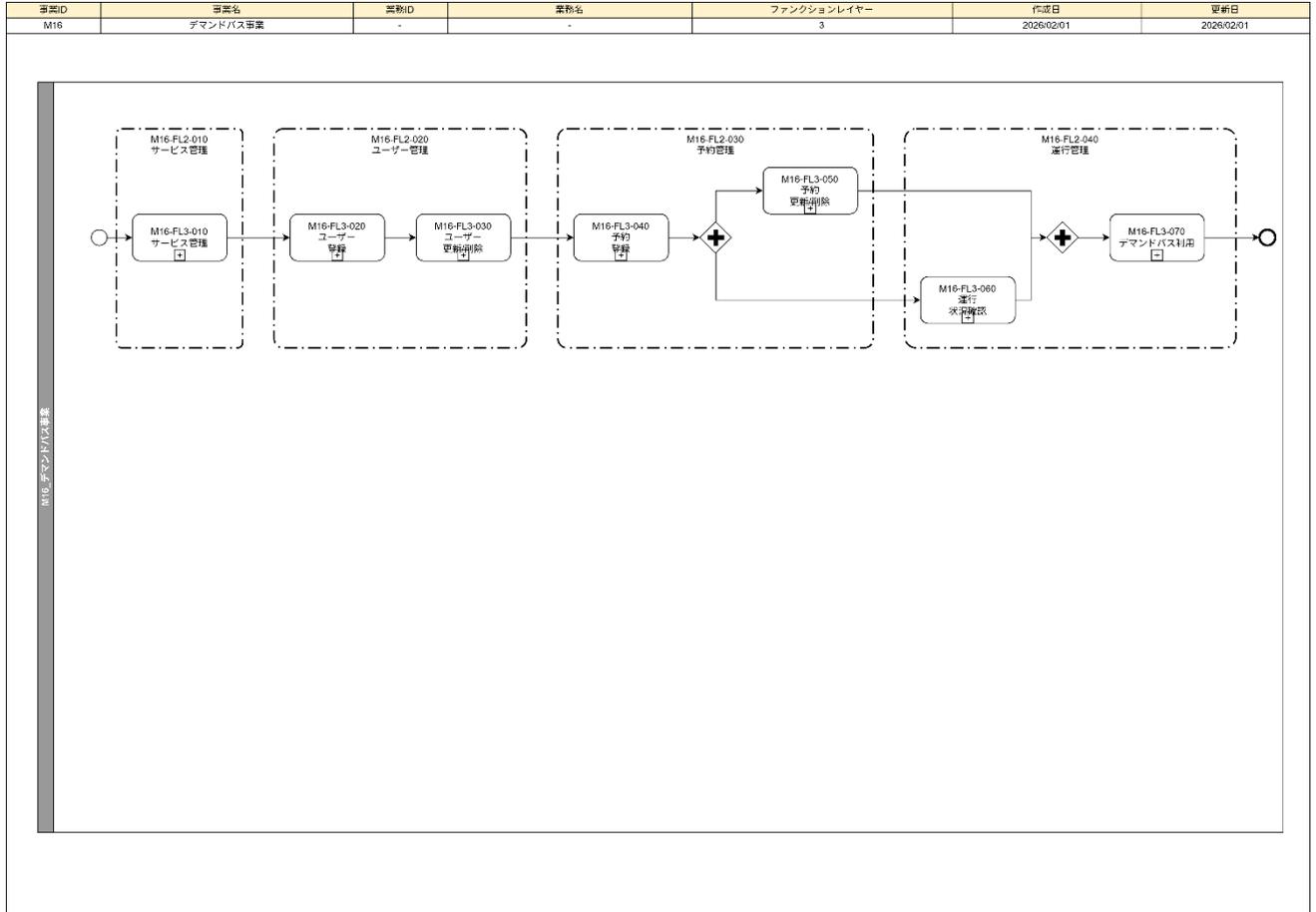
本システムはインターオペラビリティを改善する標準化されたインターフェースを持つ API が、外部サービスからデマンドバスの予約といった基本的な業務を実現できる機能を検証できることが目的である。

MaaS アプリ・デマンドバスシステム自体の改修は必要最小限とし、サービス間連携を実現する API の検証に焦点をあてた開発スコープとする。

1-2. システムを利用する業務全体像とシステム利用フロー

1. 業務フロー

調査結果に基づく、デマンドバスと外部サービス（今回は MaaS アプリを想定）のサービス間連携を実現する標準的な業務フローは以下のとおり。



デマンドバスシステム連携 API システム設計書

図 1-2 の業務フローについて、業務を行うアクター（ユーザー、MaaS サービス等）別に分解・詳細化した業務フローは以下のとおり。

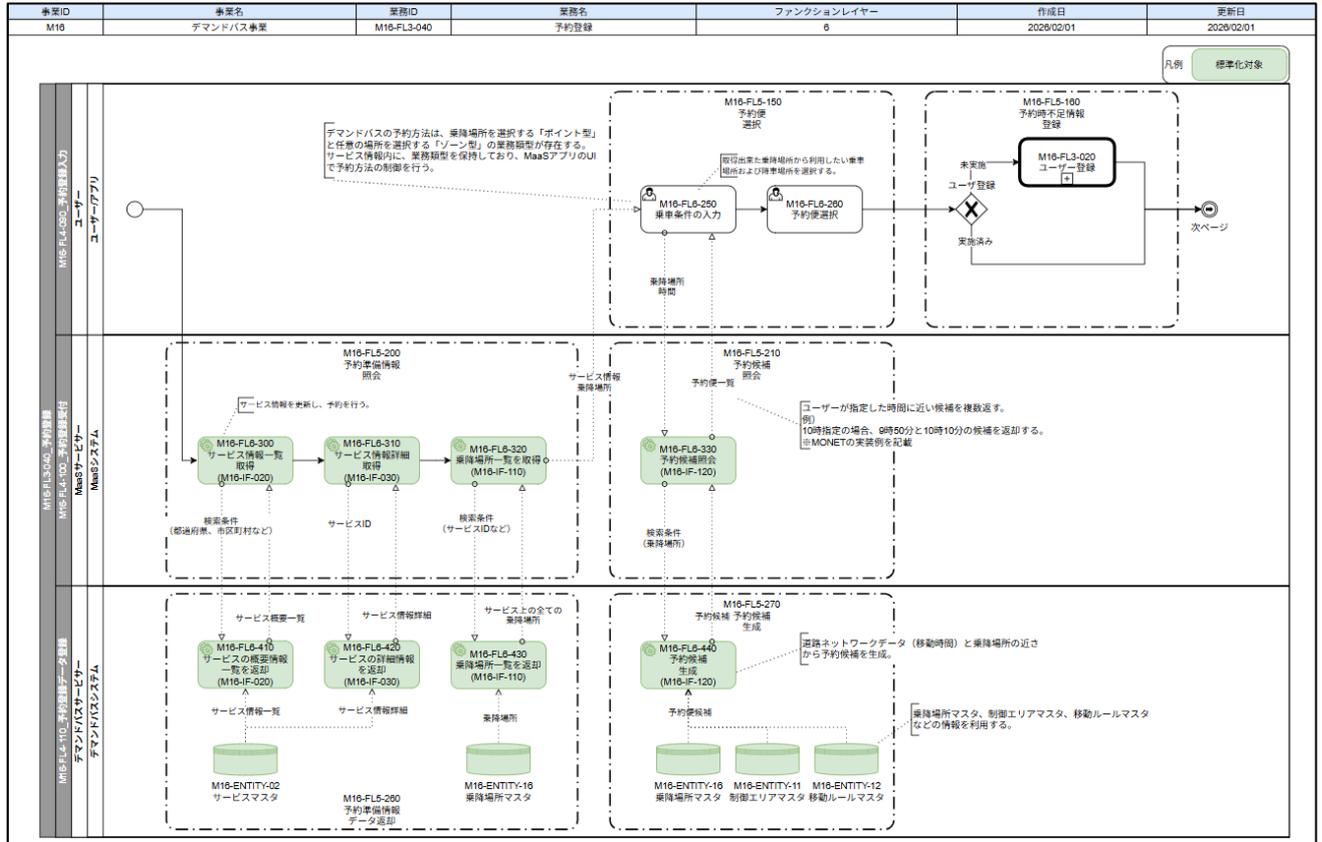


図 1-3 業務フロー詳細（予約登録①）

デマンドバスシステム連携 API システム設計書

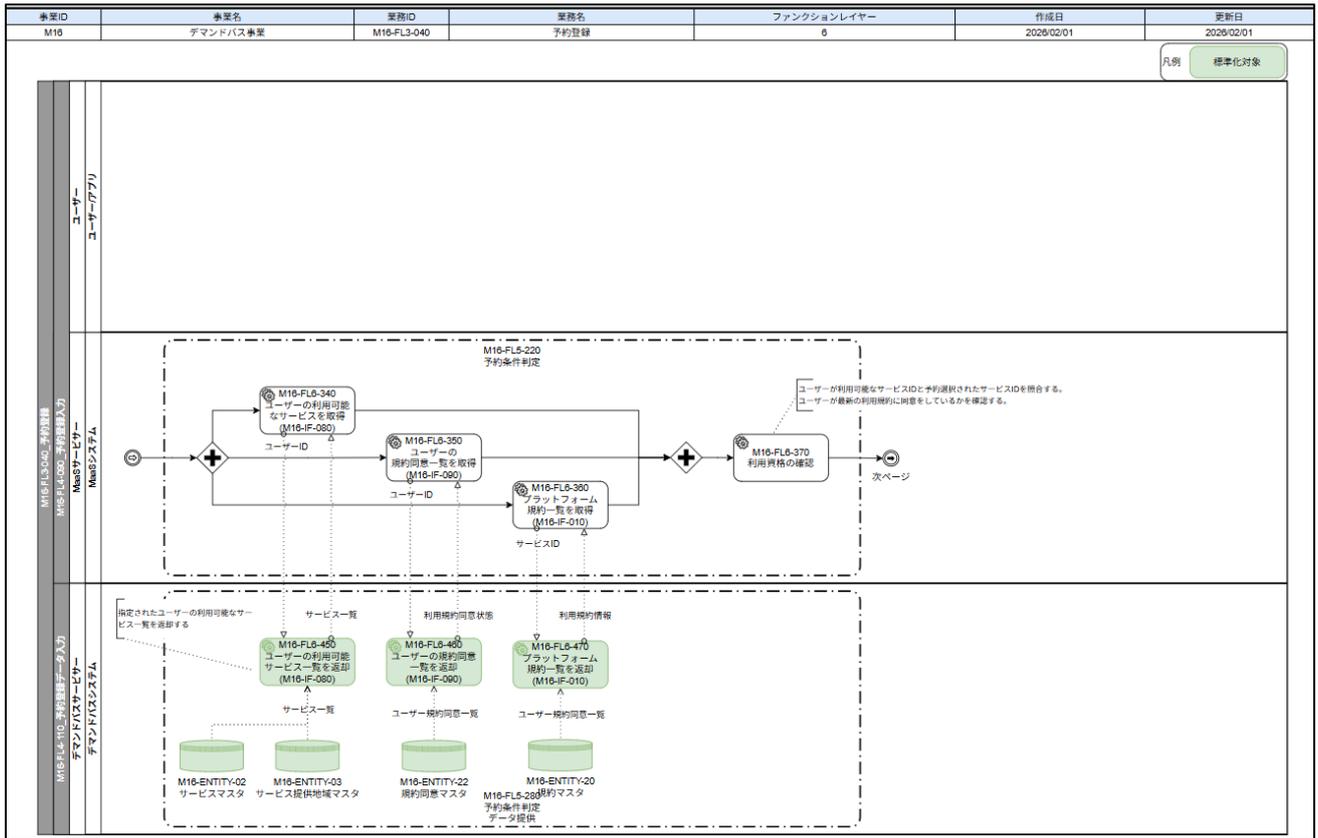


図 1-4 業務フロー詳細 (予約登録②)

デマンドバスシステム連携 API システム設計書

事業ID	事業名	業務ID	業務名	ファンクションレイヤー	作成日	更新日
M16	デマンドバス事業	M16-FL3-040	予約登録	6	2026/02/01	2026/02/01

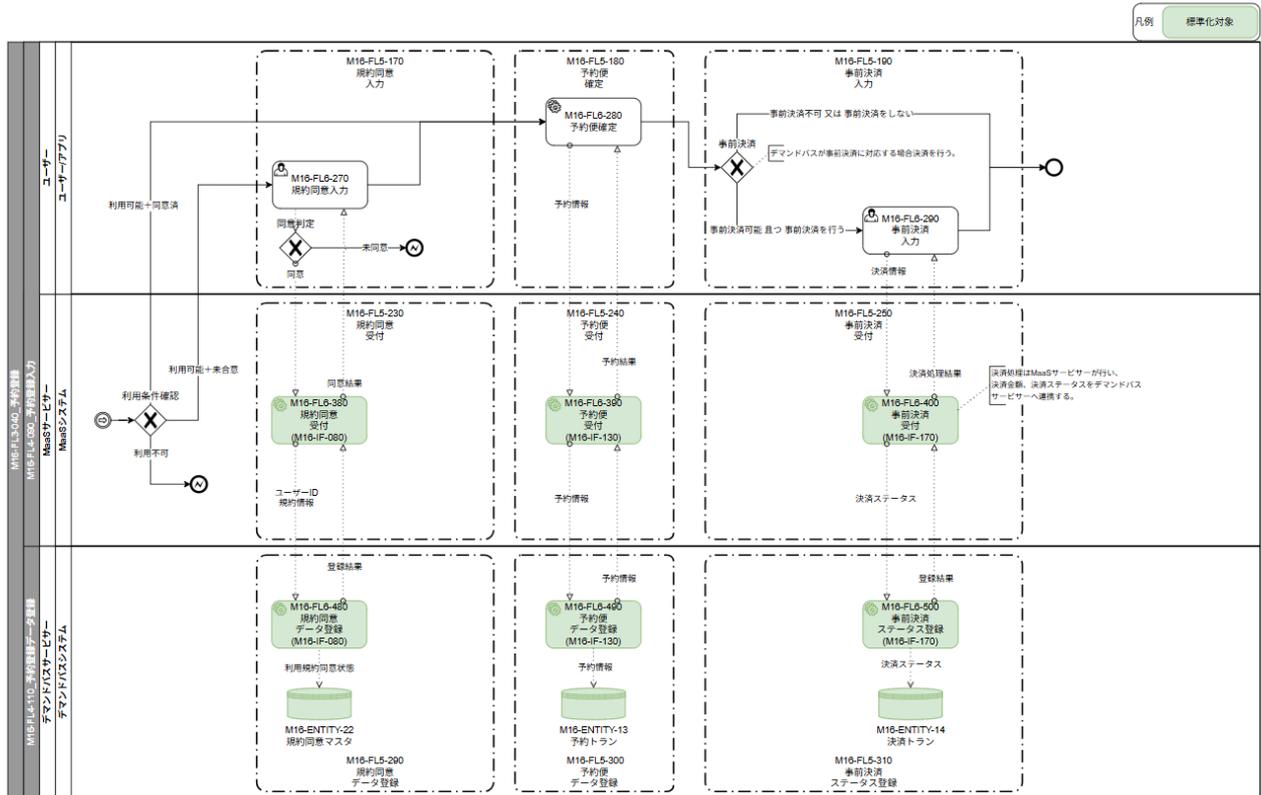


図 1-5 業務フロー詳細 (予約登録③)

デマンドバスシステム連携 API システム設計書

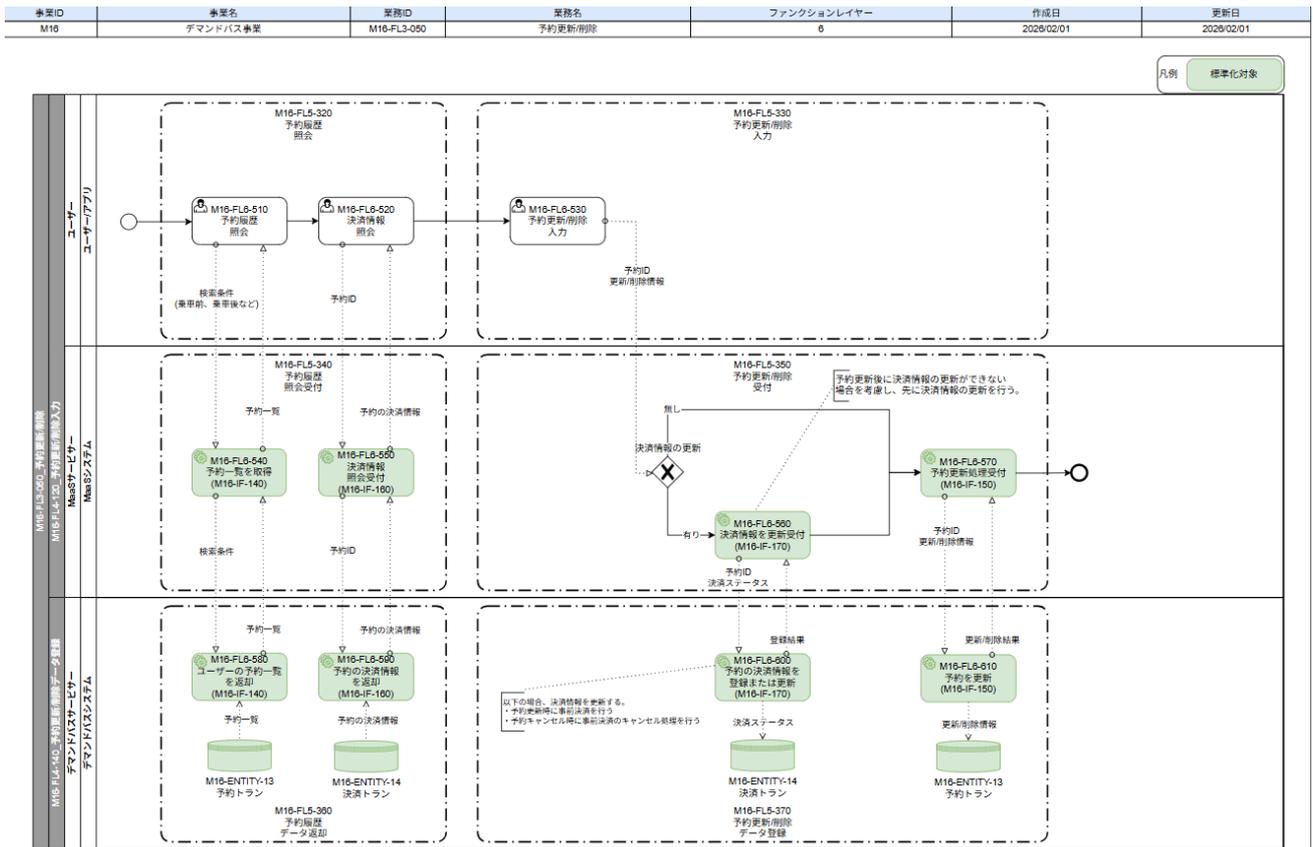


図 1-6 業務フロー詳細 (予約更新/削除)

デマンドバスシステム連携 API システム設計書

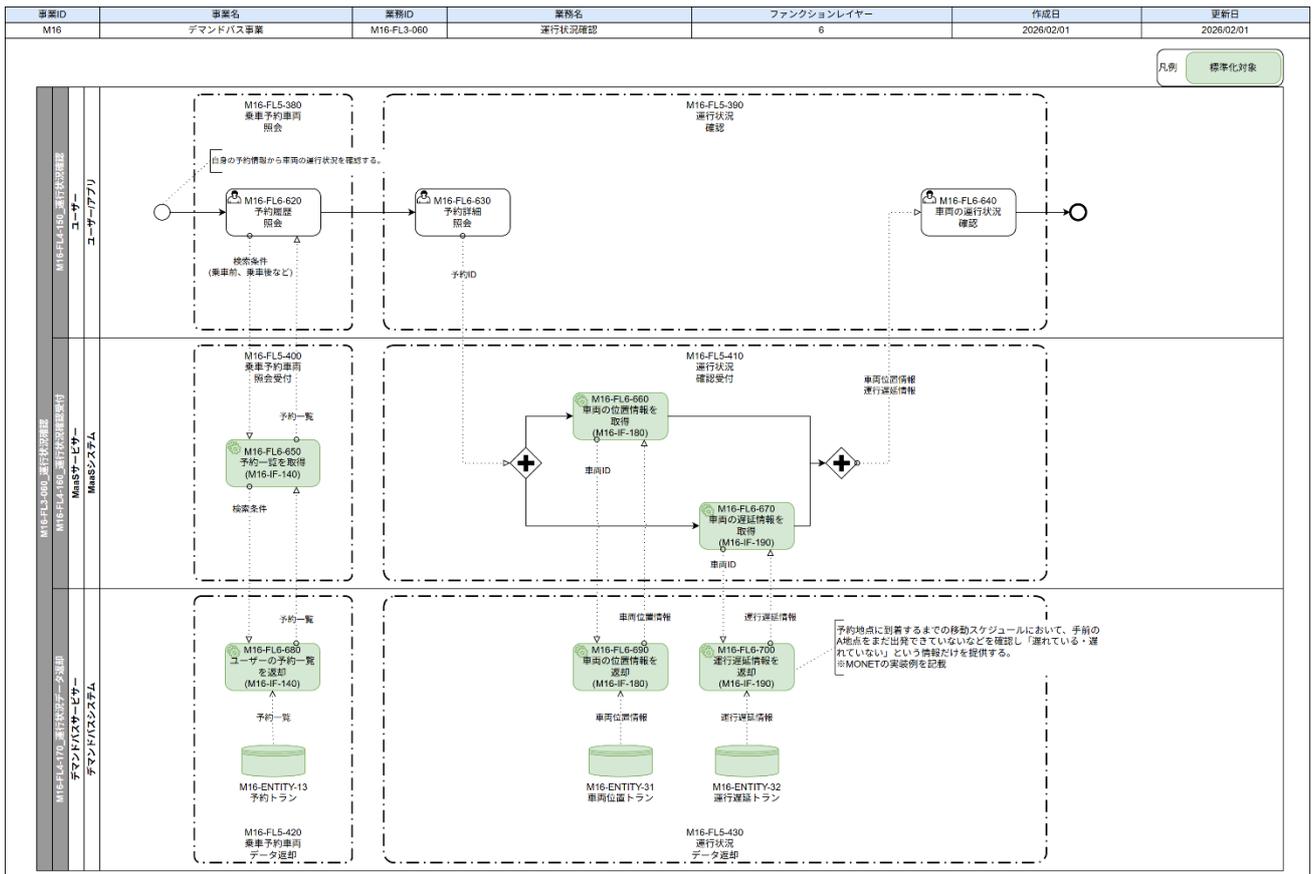


図 1-7 業務フロー詳細（乗車予約車両紹介、運行状況確認）

デマンドバスシステム連携 API システム設計書

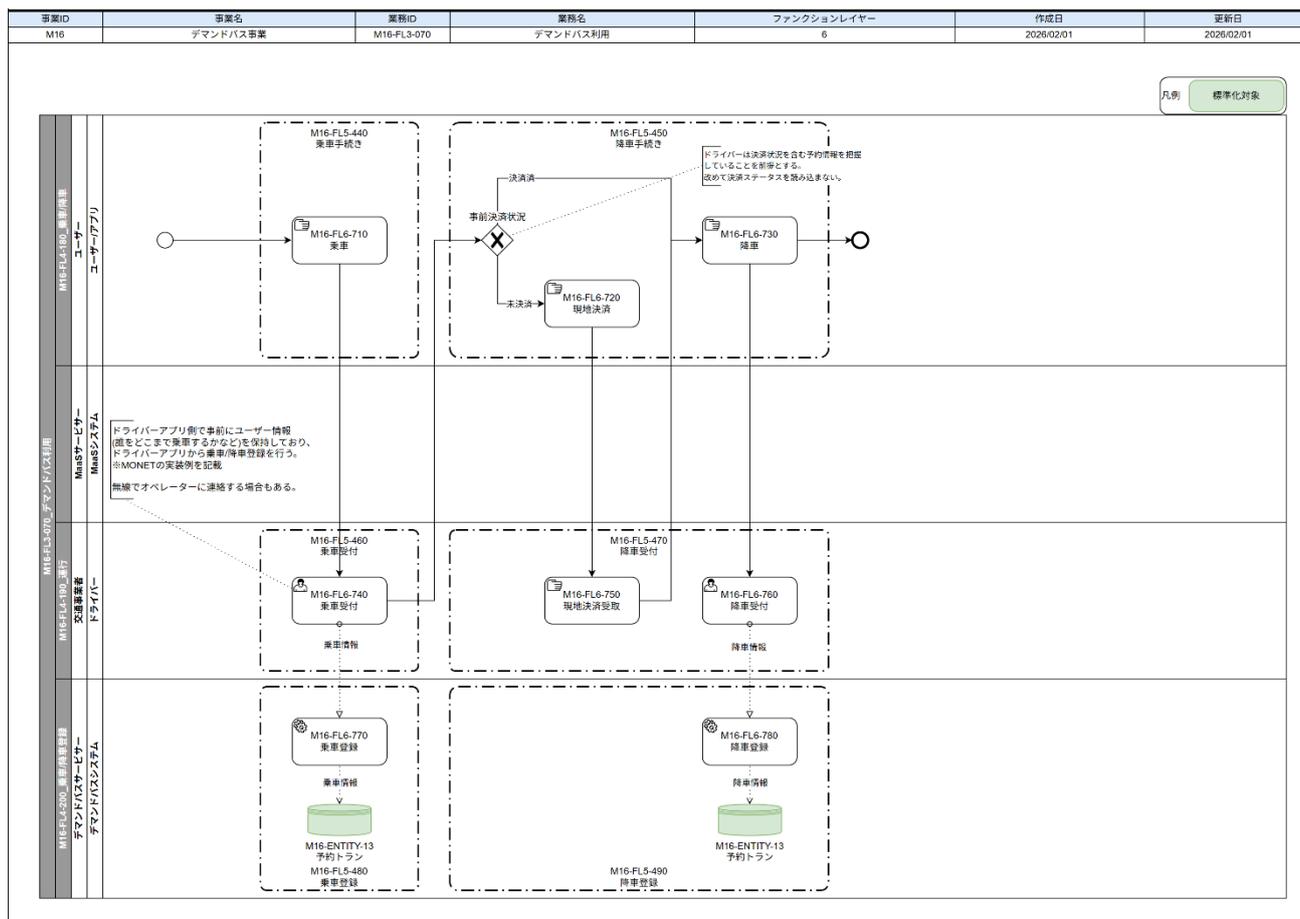


図 1-8 業務フロー詳細 (デマンドバス利用)

2. システムシーケンス図

① サービス情報取得

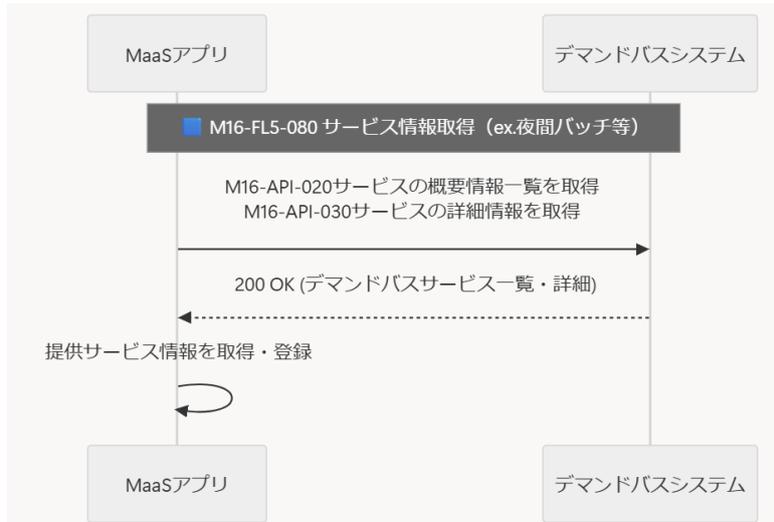


図 1-9 サービス情報取得のシーケンス図

② ユーザー登録

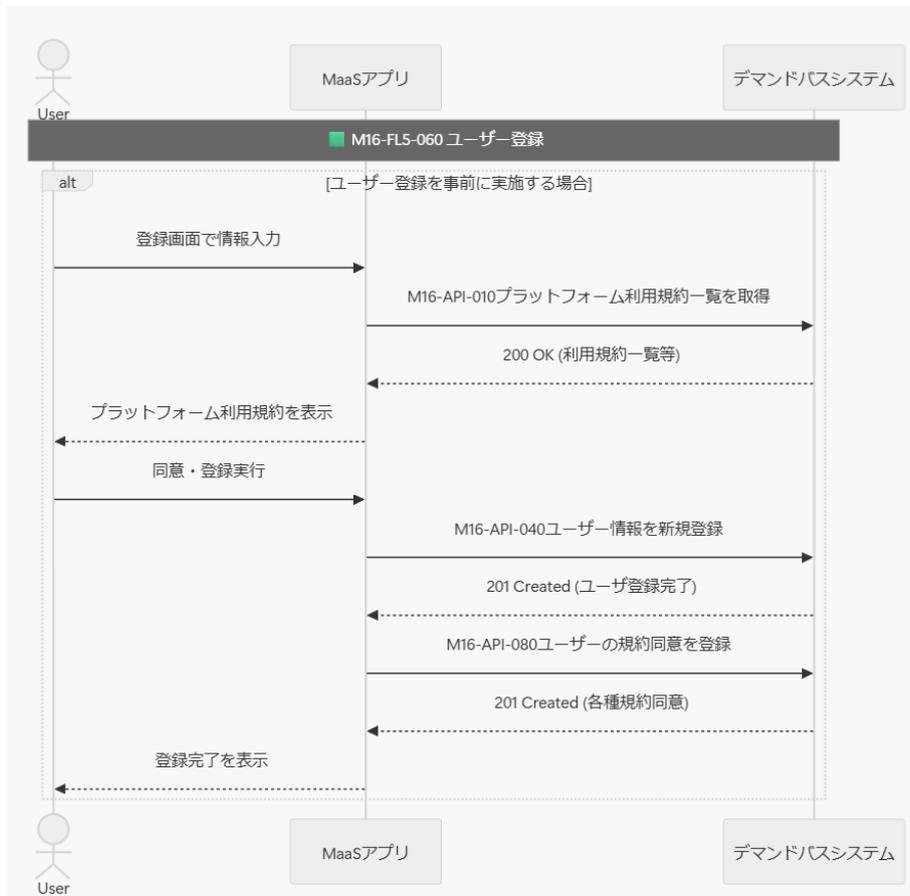


図 1-10 ユーザー登録のシーケンス図

デマンドバスシステム連携 API システム設計書

③ 予約登録

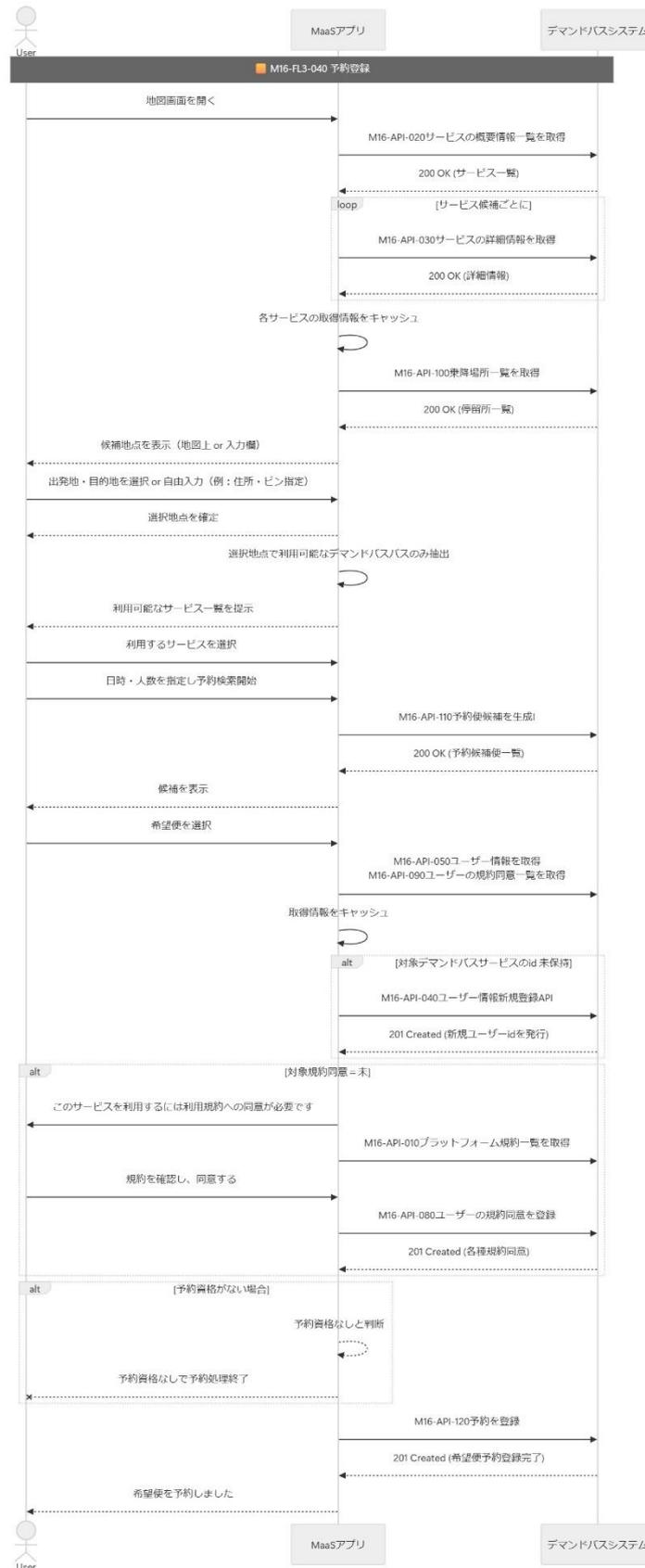


図 1-11 予約登録のシーケンス図

④ 決済

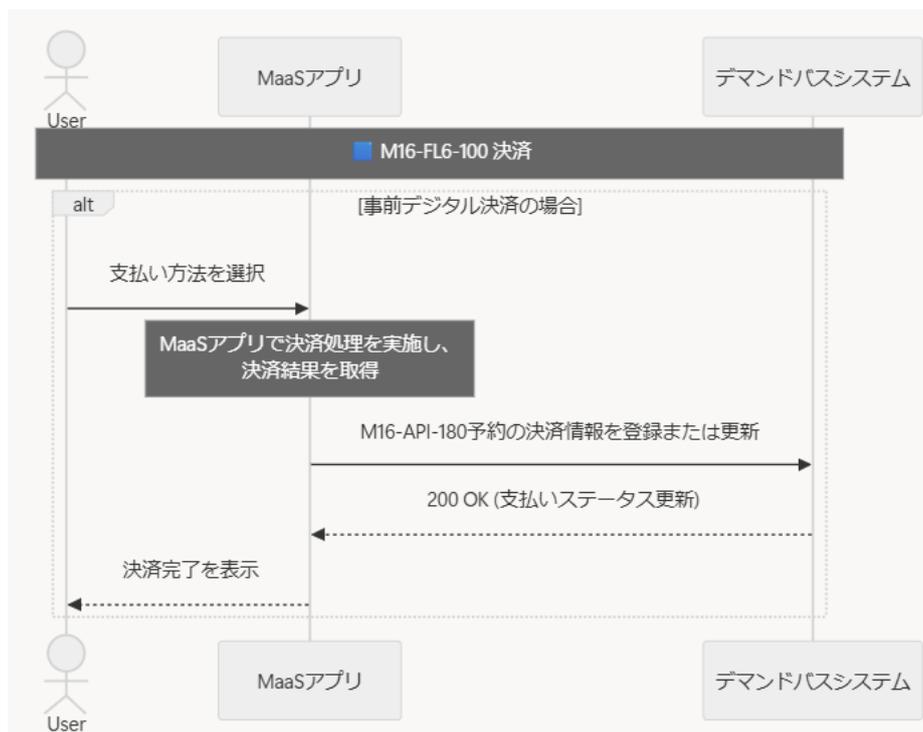


図 1-12 決済のシーケンス図

⑤ 予約更新/削除

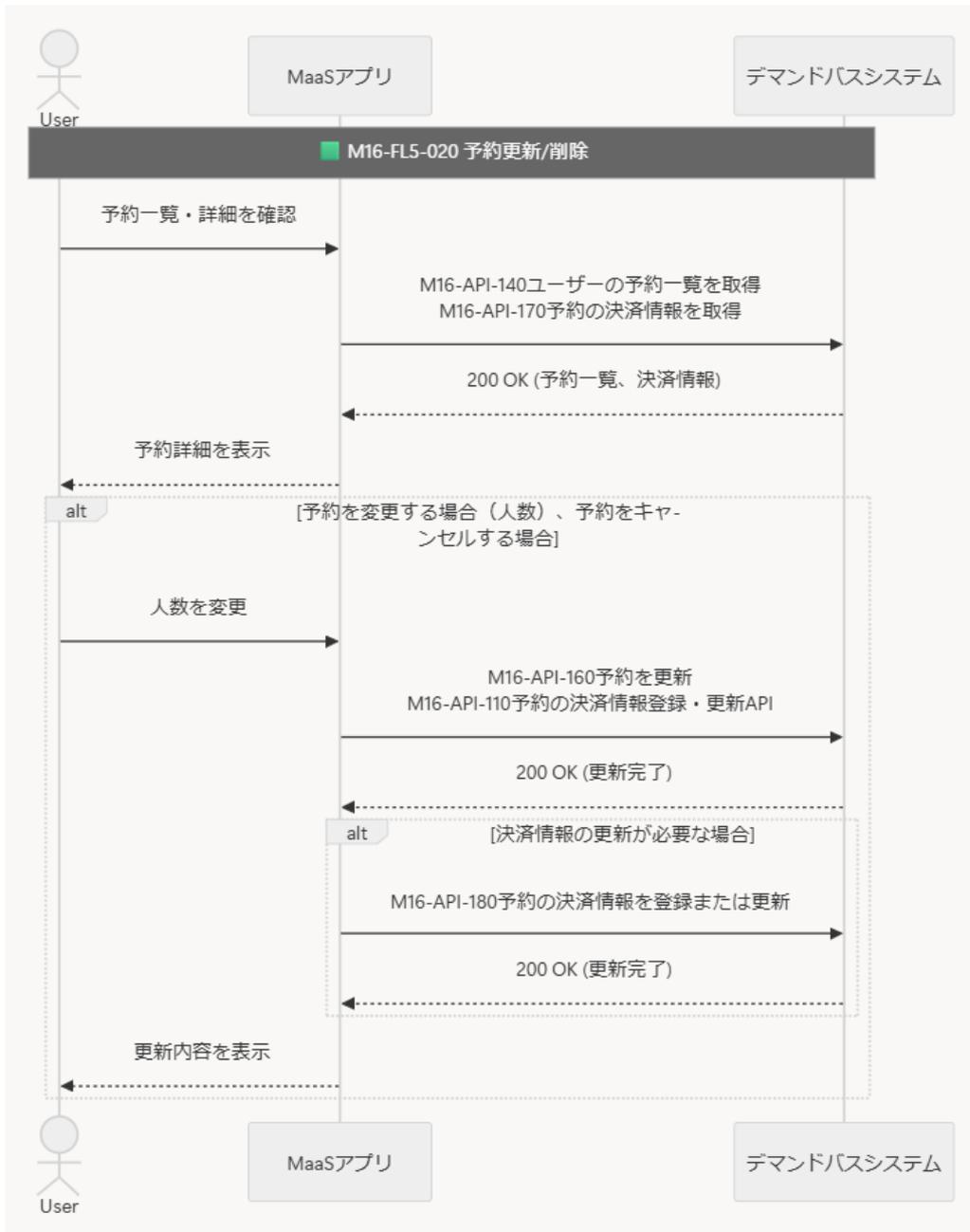


図 1-13 予約更新/削除のシーケンス図

⑥ 運行情報の確認・デマンドバスの利用

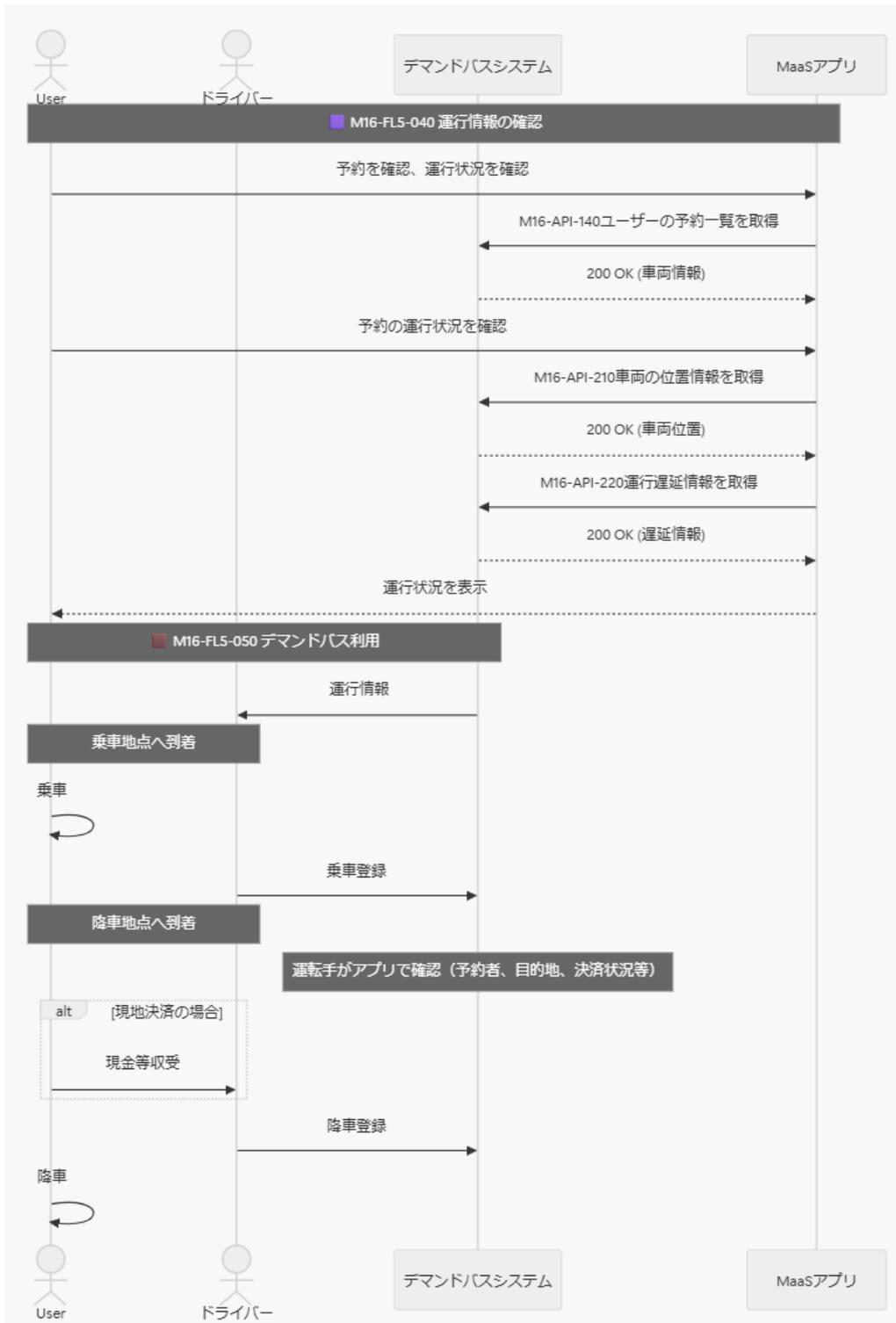


図 1-14 運行情報の確認・デマンドバスの利用のシーケンス図

⑦ 乗車予約履歴参照

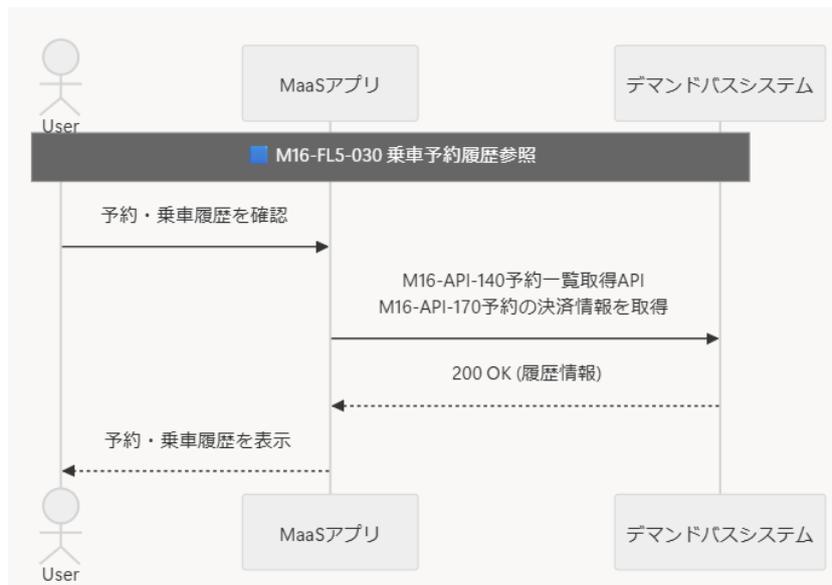


図 1-15 乗車予約履歴参照のシーケンス図

2. デマンドバス連携 API システム：機能要件 (FN/SL/AL/CO/HW/IF/UI)

2-1. システム機能 (FN)

2-1-1. システムアーキテクチャ

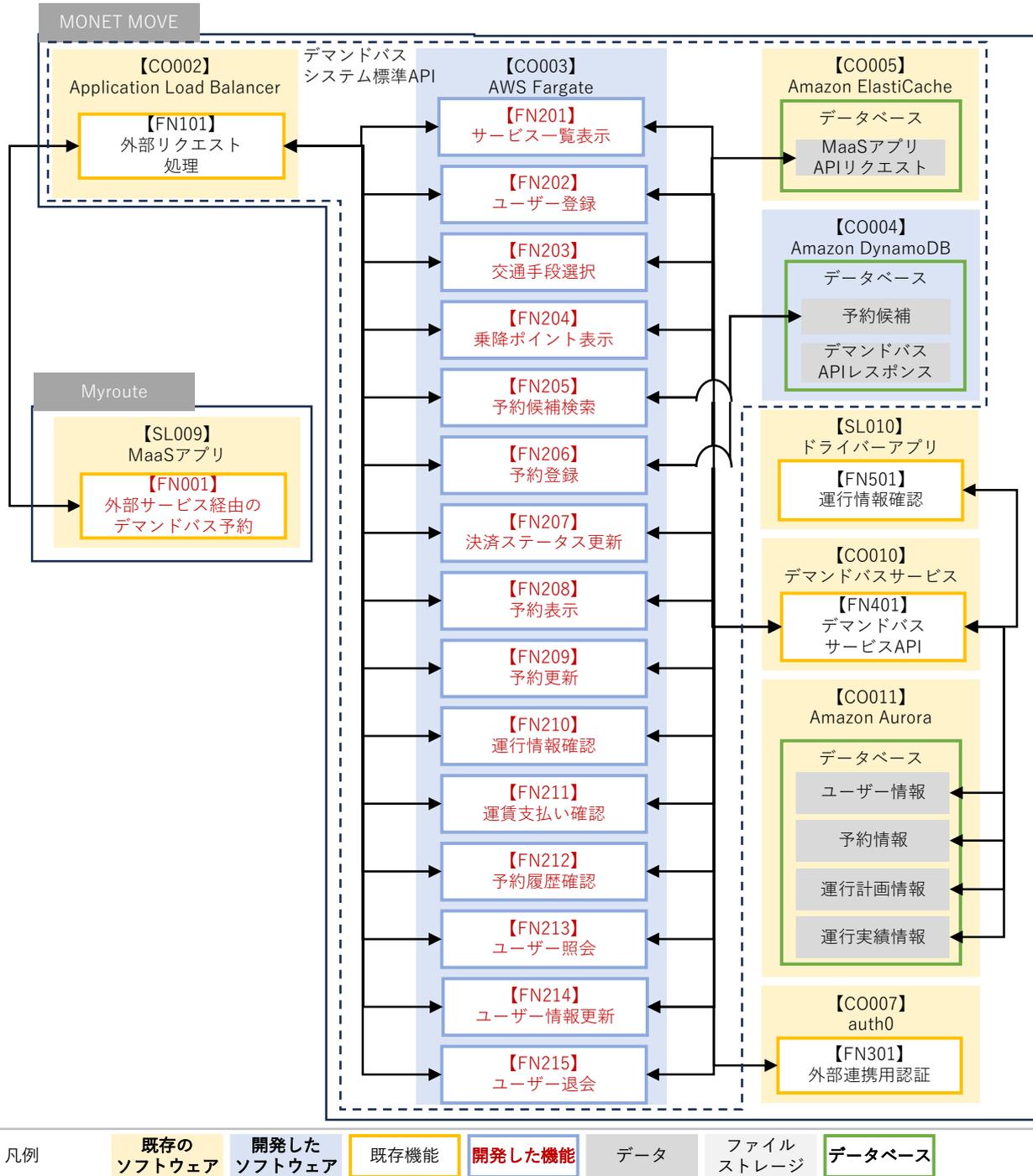


図 2-1 システムアーキテクチャ

2-1-2. システム機能一覧

表 2-1 my route システム機能一覧

※朱文字：新規開発・既存改修

ID	機能名	機能説明
FN001	外部サービス経由のデマンドバス予約	<ul style="list-style-type: none"> 外部連携 API にリクエストして、デマンドバス予約を実現 リクエストに応じて予約の表示・更新・取り消しを実現

表 2-2 MONET MOVE システム機能一覧

ID	機能名	機能説明
FN101	外部リクエスト処理	<ul style="list-style-type: none"> MaaS アプリからの API リクエストを適切なバックエンドのエンドポイントにルーティング実施
FN201	サービス一覧表示	<ul style="list-style-type: none"> MaaS アプリ初回起動時に、サービス情報を表示 サービスの概要情報一覧の取得標準 API を使用
FN202	ユーザー登録	<ul style="list-style-type: none"> MaaS アプリから入力した、個人の基本情報を基に、デマンドバスサービスにユーザーを作成 プラットフォーム規約一覧取得標準 API、ユーザー情報の新規登録標準 API、及びユーザー規約同意登録標準 API を使用
FN203	交通手段選択	<ul style="list-style-type: none"> MaaS アプリで利用可能な交通手段を表示 サービス詳細情報の取得標準 API を使用
FN204	乗降場所表示	<ul style="list-style-type: none"> 提供サービスエリア内の選択可能な乗降場所を表示 乗降場所一覧の取得標準 API を使用
FN205	予約候補検索	<ul style="list-style-type: none"> MaaS アプリから指定した検索条件を基に、予約候補を表示 予約候補の生成標準 API を使用
FN206	予約登録	<ul style="list-style-type: none"> MaaS アプリから選択された予約候補を基に、予約を登録 ユーザー規約同意登録標準 API、予約の登録標準 API、及びユーザーが利用可能なサービスの概要情報一覧の取得標準 API を使用
FN207	決済ステータス更新	<ul style="list-style-type: none"> MaaS アプリで、実施した決済ステータス（未収/受領済み/キャンセル等）を更新 予約の決済情報の登録または更新標準 API を使用
FN208	予約表示	<ul style="list-style-type: none"> MaaS アプリから予約した内容を表示 ユーザーの予約一覧の取得標準 API を使用
FN209	予約更新	<ul style="list-style-type: none"> MaaS アプリから予約した、予約のステータスと内容を更新 予約の更新標準 API を使用
FN210	運行情報確認	<ul style="list-style-type: none"> MaaS アプリから予約した、予約情報に紐づく運行情報を表示 車両位置情報取得標準 API 及び運行遅延情報取得標準 API を使用

デマンドバスシステム連携 API システム設計書

FN211	運賃支払い確認	<ul style="list-style-type: none"> ● MaaS アプリから予約した、予約の決済情報を表示 ● 予約の決済情報の取得標準 API の仕様
FN212	予約履歴確認	<ul style="list-style-type: none"> ● MaaS アプリから予約した、予約の履歴情報を表示 ● ユーザーの予約一覧の取得標準 API の使用
FN213	ユーザー照会	<ul style="list-style-type: none"> ● MaaS アプリから登録した、ユーザー情報を表示 ● ユーザー情報の取得標準 API 及びユーザーの規約同意一覧の取得標準 API を使用
FN214	ユーザー情報更新	<ul style="list-style-type: none"> ● MaaS アプリから登録した、ユーザー情報を更新 ● ユーザー情報の更新標準 API の使用
FN215	ユーザー退会	<ul style="list-style-type: none"> ● デマンドバスサービスのユーザー情報を削除 ● ユーザー情報の削除（同意情報含む）標準 API を使用
FN301	外部連携用認証	<ul style="list-style-type: none"> ● 外部連携 API サーバーからバックエンドのデマンドバスサービスへアクセスする際の JSON Web Token (JWT) の発行と検証
FN401	デマンドバス API	<ul style="list-style-type: none"> ● デマンドバスサービスを運営する事業者が公開している API
FN501	運行情報確認	<ul style="list-style-type: none"> ● デマンドバスサービスのスケジュールなど運行情報を確認

2-1-3. システム機能の詳細

以下に、システム機能の詳細を記す。なお、本業務においてを行うシステム機能は、機能名称を**朱文字**で示す。

【FN001】外部サービス経由のデマンドバス予約<新規開発>

- 本システム機能の概要
 - 外部連携 API にリクエストして、デマンドバスの予約に関する操作を行う。
 - リクエストに応じて予約の登録・表示・更新・取り消しを行う。
- 本システム機能の入力・処理・出力のフローチャート

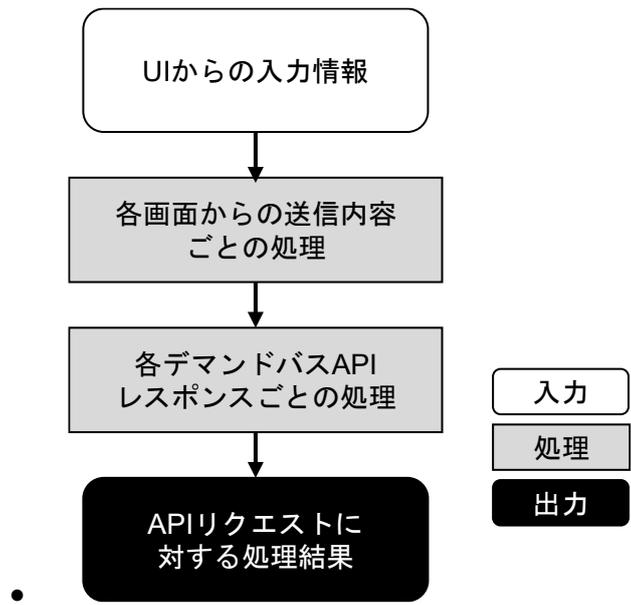


図 2-2 【FN001】のフローチャート

- 本システム機能の処理の詳細
 - 各画面からの送信内容ごとの処理
 - ◇ 処理内容
 - ユーザーが MaaS アプリに UI から入力した情報を基に、バックエンドの外部連携 API サーバーに、API リクエストを送れるようにデータを成形
 - ◇ 利用するライブラリ
 - 【SL009】 MaaS アプリ
 - ◇ 利用するアルゴリズム
 - -
 - 各デマンドバス API レスポンスごとの処理
 - ◇ 処理内容
 - バックエンドの外部連携 API サーバーに、API リクエストを送信
 - ◇ 利用するライブラリ
 - 【SL009】 MaaS アプリ
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様

➤ 入力

◇ UI からの入力情報

- データの内容
 - MaaS アプリ(my route)を利用するユーザーが UI から入力した情報
 - ◇ ユーザー登録情報（名前、連絡先、住所等）
 - ◇ 予約情報（乗降場所、乗車希望日時、乗車人数）
- データの形式
 - UI コンポーネント（ネイティブアプリ）
- 利用するデータインターフェース
 - 全 IF が通過
 - 【IF001】サービスの概要情報一覧の取得標準 API
 - 【IF002】サービス詳細情報の取得標準 API
 - 【IF003】プラットフォーム規約一覧取得 API
 - 【IF004】ユーザー情報の新規登録標準 API
 - 【IF005】ユーザー情報の取得標準 API
 - 【IF006】ユーザー情報の更新標準 API
 - 【IF007】ユーザー情報の削除（同意情報を含む）標準 API
 - 【IF008】ユーザー規約同意登録標準 API
 - 【IF009】ユーザーの規約同意一覧の取得標準 API
 - 【IF010】乗降場所一覧の取得標準 API
 - 【IF011】予約候補の生成標準 API
 - 【IF012】予約の登録標準 API
 - 【IF013】ユーザーが利用可能なサービスの概要情報一覧を取得標準 API
 - 【IF014】ユーザーの予約一覧の取得標準 API
 - 【IF015】予約の更新標準 API
 - 【IF016】予約の決済情報の取得標準 API
 - 【IF017】予約の決済情報の登録または更新標準 API
 - 【IF018】車両位置情報取得標準 API
 - 【IF019】運行遅延情報取得標準 API

➤ 出力

◇ API リクエストに対する処理結果

- データの内容
 - 外部連携 API リクエストからのレスポンスを基に表示する、処理結果画面
- データの形式
 - UI コンポーネント（ネイティブアプリ）
- 利用するデータインターフェース
 - -

【FN101】外部リクエスト処理

- 本システム機能の概要
 - MaaS アプリからの API リクエストを、適切なバックエンドのエンドポイントヘルパーティングす

る。

➤ 標準 API を通じて、デマンドバスサービスへのアクセスを中継する。

- 本システム機能の入力・処理・出力のフローチャート

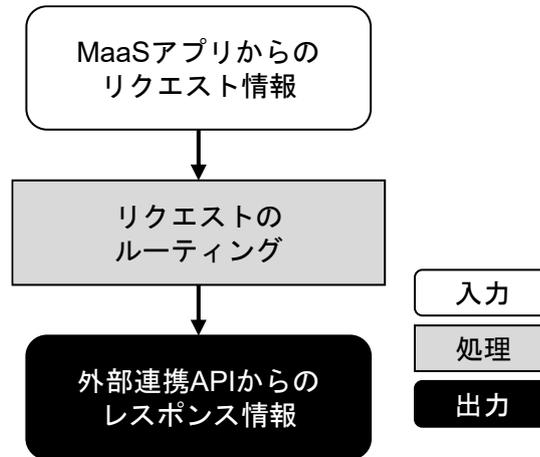


図 2-3 【FN101】のフローチャート

- 本システム機能の処理の詳細

➤ リクエストのルーティング

◇ 処理内容

- リクエスト情報を基に、バックエンドの API サーバーにルーティングする

◇ 利用するライブラリ

- -

◇ 利用するアルゴリズム

- -

- 本システム機能の入出力データの仕様

➤ 入力

◇ MaaS アプリからのリクエスト情報

- データの内容

➤ MaaS アプリ側(my route)から送信されてくる各リクエストに含まれる情報

- データの形式

➤ JSON 形式

- 利用するデータインターフェース

全 IF が通過

➤ 【IF001】サービスの概要情報一覧の取得標準 API

➤ 【IF002】サービス詳細情報の取得標準 API

➤ 【IF003】プラットフォーム規約一覧取得 API

➤ 【IF004】ユーザー情報の新規登録標準 API

➤ 【IF005】ユーザー情報の取得標準 API

➤ 【IF006】ユーザー情報の更新標準 API

➤ 【IF007】ユーザー情報の削除（同意情報を含む）標準 API

➤ 【IF008】ユーザー規約同意登録標準 API

➤ 【IF009】ユーザーの規約同意一覧の取得標準 API

➤ 【IF010】乗降場所一覧の取得標準 API

- 【IF011】 予約候補の生成標準 API
- 【IF012】 予約の登録標準 API
- 【IF013】 ユーザーが利用可能なサービスの概要情報一覧を取得標準 API
- 【IF014】 ユーザーの予約一覧の取得標準 API
- 【IF015】 予約の更新標準 API
- 【IF016】 予約の決済情報の取得標準 API
- 【IF017】 予約の決済情報の登録または更新標準 API
- 【IF018】 車両位置情報取得標準 API
- 【IF019】 運行遅延情報取得標準 API
- 出力
 - ◇ 標準 API からのレスポンス情報
 - データの内容
 - 標準 API で処理されたレスポンス情報
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 全 IF が通過
 - 【IF001】 サービスの概要情報一覧の取得標準 API
 - 【IF002】 サービス詳細情報の取得標準 API
 - 【IF003】 プラットフォーム規約一覧取得 API
 - 【IF004】 ユーザー情報の新規登録標準 API
 - 【IF005】 ユーザー情報の取得標準 API
 - 【IF006】 ユーザー情報の更新標準 API
 - 【IF007】 ユーザー情報の削除（同意情報を含む）標準 API
 - 【IF008】 ユーザー規約同意登録標準 API
 - 【IF009】 ユーザーの規約同意一覧の取得標準 API
 - 【IF010】 乗降場所一覧の取得標準 API
 - 【IF011】 予約候補の生成標準 API
 - 【IF012】 予約の登録標準 API
 - 【IF013】 ユーザーが利用可能なサービスの概要情報一覧を取得標準 API
 - 【IF014】 ユーザーの予約一覧の取得標準 API
 - 【IF015】 予約の更新標準 API
 - 【IF016】 予約の決済情報の取得標準 API
 - 【IF017】 予約の決済情報の登録または更新標準 API
 - 【IF018】 車両位置情報取得標準 API
 - 【IF019】 運行遅延情報取得標準 API

【FN201】 サービス一覧表示 < 新規開発 >

- 本システム機能の概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、提供対象のデマンドバスサービスの基本情報を取得・返却し、MaaS アプリ(my route)でのサービス表示や案内に活用する。

- 本システム機能の入力・処理・出力のフローチャート

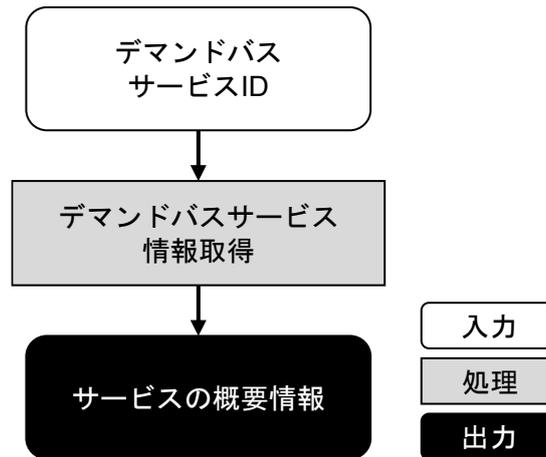


図 2-4 【FN201】のフローチャート

- 本システム機能の処理の詳細
 - デマンドバスサービス情報取得
 - ◇ 処理内容
 - リクエストのサービス ID を基に、デマンドバスサービスの API から、サービスの基本情報を取得する
 - ◇ 利用するライブラリ
 - 【SL001】 Go
 - 【SL002】 Gin
 - 【SL004】 dd-tracer-go
 - 【SL005】 go-jwt-middleware
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ デマンドバスサービス ID
 - データの内容
 - MaaS アプリ側で、管理するデマンドバスサービスの ID
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF001】 サービスの概要情報一覧の取得標準 API
 - 出力
 - ◇ デマンドバスサービスの概要情報
 - データの内容
 - デマンドバスサービスの概要情報
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF001】 サービスの概要情報一覧の取得標準 API

【FN202】 ユーザー登録<新規開発>

- 本システム機能の概要
 - MaaS アプリ等のサードパーティのユーザーが、各デマンドバスサービスを利用するために必要なユーザー情報を新規に登録する。
 - MaaS アプリ等のサードパーティから送られた基本情報と利用規約同意情報に基づき、デマンドバス API をリクエストして登録処理を行い、登録結果を返却する。
- 本システム機能の入力・処理・出力のフローチャート

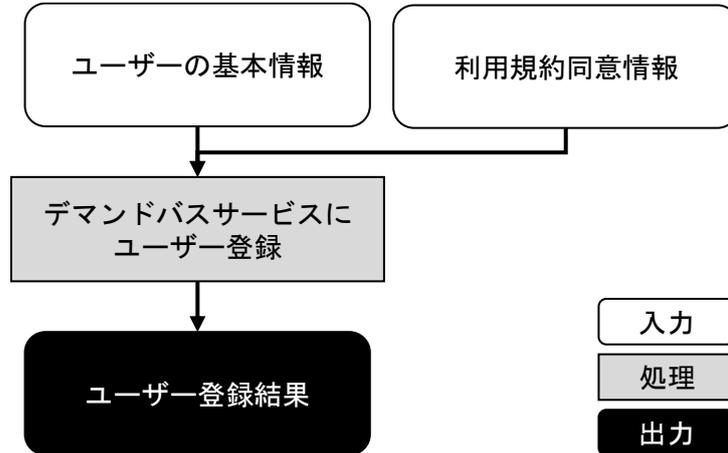


図 2-5 【FN202】のフローチャート

- 本システム機能の処理の詳細
 - デマンドバスサービスにユーザー情報を登録
 - ◇ 処理内容
 - MaaS アプリから送られてきた、ユーザー情報を基に、デマンドバス API をリクエストすることで、デマンドバスサービスに、新規にユーザーを登録
 - ◇ 利用するライブラリ
 - 【SL001】 Go
 - 【SL002】 Gin
 - 【SL004】 dd-tracer-go
 - 【SL005】 go-jwt-middleware
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ ユーザーの基本情報
 - データの内容
 - デマンドバスサービスに新規に登録する、ユーザーの基本情報
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF003】 プラットフォーム規約一覧取得 API
 - 【IF004】 ユーザー情報の新規登録標準 API
 - 【IF008】 ユーザー規約同意登録標準 API

- ◇ 利用規約同意情報
 - データの内容
 - システムを利用するにあたって必要な利用規約の同意情報
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF003】プラットフォーム規約一覧取得 API
 - 【IF004】ユーザー情報の新規登録標準 API
 - 【IF008】ユーザー規約同意登録標準 API
- 出力
 - ◇ ユーザー登録結果
 - データの内容
 - デマンドバスサービスに登録した、ユーザーの基本情報
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF003】プラットフォーム規約一覧取得 API
 - 【IF004】ユーザー情報の新規登録標準 API
 - 【IF008】ユーザー規約同意登録標準 API

【FN203】交通手段選択<新規開発>

- 本システム機能の概要
 - MaaS アプリ等のサードパーティが利用可能な交通手段の詳細情報を表示するため、リクエストのサービス ID を基に、デマンドバスサービスの API からサービス詳細情報を取得・返却し、交通手段の詳細を MaaS アプリで表示・案内する際に活用する。
- 本システム機能の入力・処理・出力のフローチャート

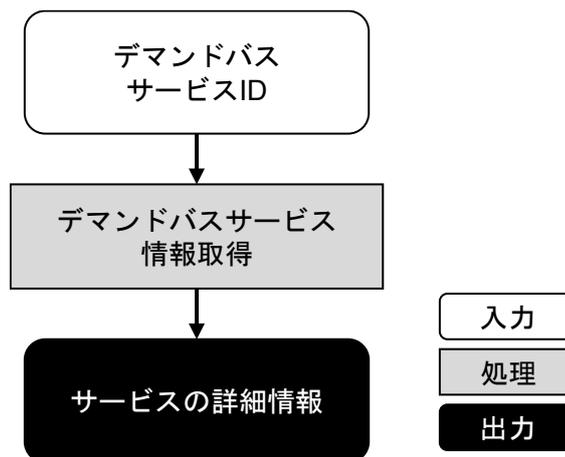


図 2-6 【FN203】のフローチャート

- 本システム機能の処理の詳細
 - デマンドバスサービス情報取得
 - ◇ 処理内容
 - リクエストのサービス ID を基に、デマンドバスサービスの API から、サービスの詳細情報

報を取得する

- ◇ 利用するライブラリ
 - 【SL001】 Go
 - 【SL002】 Gin
 - 【SL004】 dd-tracer-go
 - 【SL005】 go-jwt-middleware
- ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ デマンドバスサービス ID
 - データの内容
 - MaaS アプリ側で、管理するデマンドバスサービスの ID
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF002】 サービス詳細情報の取得標準 API
 - 出力
 - ◇ デマンドバスサービスの基本情報
 - データの内容
 - デマンドバスサービスの詳細情報
 - 【IF001】 で取得できる情報よりも詳細な情報
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF002】 サービス詳細情報の取得標準 API

【FN204】乗降場所表示<新規開発>

- 本システム機能の概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、提供サービスエリア内の選択可能な乗降場所（ポイント）情報を取得・返却し、MaaS アプリ(my route)での乗降ポイント表示やユーザーの出発地・目的地指定に活用する。
- 本システム機能の入力・処理・出力のフローチャート

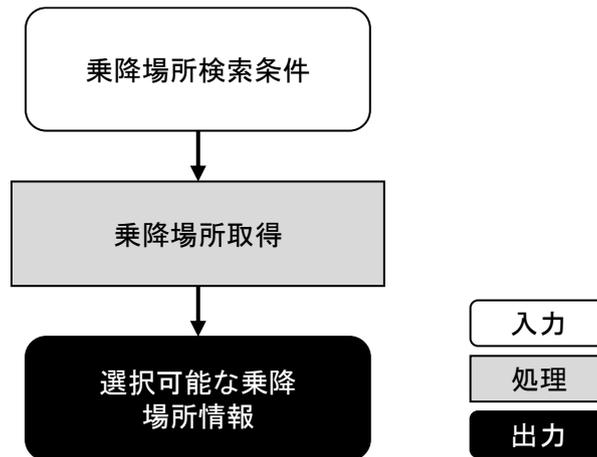


図 2-7 【FN204】のフローチャート

- 本システム機能の処理の詳細
 - 乗降ポイント情報取得
 - ◇ 処理内容
 - リクエストのサービス ID を基に、選択可能な乗降ポイントの情報をデマンドバスサービスから取得する
 - ◇ 利用するライブラリ
 - 【SL001】 Go
 - 【SL002】 Gin
 - 【SL004】 dd-tracer-go
 - 【SL005】 go-jwt-middleware
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ デマンドバスサービス ID
 - データの内容
 - MaaS アプリ側で、管理するデマンドバスサービスの ID
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF010】乗降場所一覧の取得標準 API
 - 出力
 - ◇ 選択可能な乗降ポイント情報
 - データの内容
 - 対象のデマンドバスサービスで、選択可能な固定の乗降ポイント情報
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF010】乗降場所一覧の取得標準 API

【FN205】 予約候補検索<新規開発>

- 本システム機能の概要
 - MaaS アプリ等のサードパーティのユーザーがリクエストした予約条件（出発地、目的地、時間など）を基に、予約可能な便候補を検索し、結果を返却し、ユーザーの予約便選択に活用する。
 - 検索結果はデマンドバスサービスから取得した後、Amazon DynamoDB にキャッシュされる。
- 本システム機能の入力・処理・出力のフローチャート

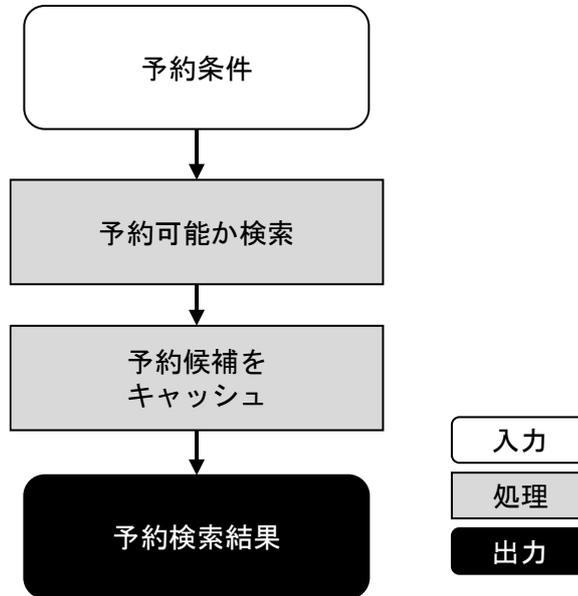


図 2-8 【FN205】 のフローチャート

- 本システム機能の処理の詳細
 - 予約可能か検索
 - ◇ 処理内容
 - 予約条件を基に、対象のデマンドバスサービスに対して、予約可能かどうかを検索するリクエストを送信
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
 - 予約候補をキャッシュ
 - ◇ 処理内容
 - デマンドバスサービスから取得した、予約候補を Amazon DynamoDB にキャッシュする
 - ◇ 利用するライブラリ
 - 【SL001】 Go
 - 【SL002】 Gin
 - 【SL003】 aws-sdk-go-v2
 - 【SL004】 dd-tracer-go
 - 【SL005】 go-jwt-middleware
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様

- 入力
 - ◇ 予約条件
 - データの内容
 - 出発地、目的地、時間など、予約したい条件
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF011】 予約候補の生成標準 API
- 出力
 - ◇ 予約検索結果
 - データの内容
 - 予約可能な便候補
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF011】 予約候補の生成標準 API

【FN206】 予約登録<新規開発>

- 本システム機能の概要
 - MaaS アプリ等のサードパーティから選択された予約便候補を基に、デマンドバスサービスに予約を登録し、結果を返却する。
 - リクエストの予約候補 ID を使って Amazon DynamoDB から候補情報を取得し、それに基づいてデマンドバスサービスへ予約登録を行う。
- 本システム機能の入力・処理・出力のフローチャート

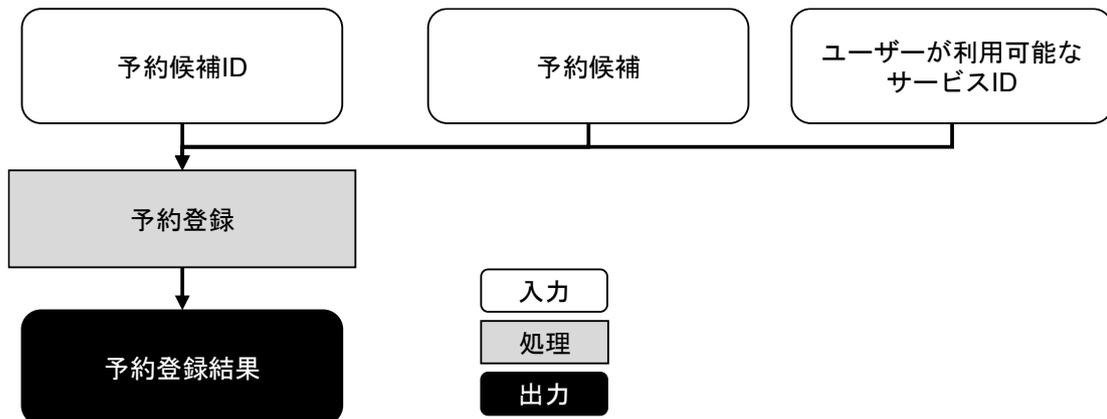


図 2-9 【FN206】 のフローチャート

- 本システム機能の処理の詳細
 - 予約登録
 - ◇ 処理内容
 - リクエストの予約候補 ID で、Amazon DynamoDB から予約候補を取得し、その予約候補を基に、デマンドバスサービスに予約を登録する
 - ◇ 利用するライブラリ
 - 【SL001】 Go

- 【SL002】 Gin
- 【SL003】 aws-sdk-go-v2
- 【SL004】 dd-tracer-go
- 【SL005】 go-jwt-middleware
- ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 予約候補 ID
 - データの内容
 - 予約候補に紐づく ID
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF012】 予約の登録標準 API
 - ◇ 予約候補情報
 - データの内容
 - 出発地・目的地・乗車人数など予約候補のステータス情報
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF012】 予約の登録標準 API
 - ◇ ユーザーが利用可能なサービス ID
 - データの内容
 - 予約をするユーザーが利用可能なサービス ID
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF013】 ユーザーが利用可能なサービスの概要情報一覧を取得標準 API
 - 出力
 - ◇ 予約登録結果
 - データの内容
 - 登録した、予約内容
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF012】 予約の登録標準 API

【FN207】 決済ステータス更新<新規開発>

- 本システム機能の概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、予約に紐づく決済ステータス（未収、

受領済み、キャンセルなど)を更新し、結果を返却する。

- MaaS アプリで実施した決済結果を、リクエストの予約 ID と決済ステータスを基にデマンドバスサービス側で更新し、予約の金銭的な状況を反映する。

- 本システム機能の入力・処理・出力のフローチャート

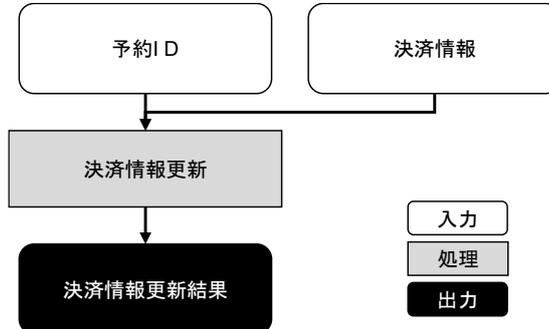


図 2-10 【FN207】のフローチャート

- 本システム機能の処理の詳細

- 決済ステータス更新

- ◇ 処理内容

- リクエストの予約 ID と決済ステータスを基に、決済ステータスを更新する

- ◇ 利用するライブラリ

- 【SL001】 Go
- 【SL002】 Gin
- 【SL004】 dd-tracer-go
- 【SL005】 go-jwt-middleware

- ◇ 利用するアルゴリズム

- -

- 本システム機能の入出力データの仕様

- 入力

- ◇ 予約 ID

- データの内容
 - MaaS アプリ側で、管理する予約の ID
- データの形式
 - JSON 形式
- 利用するデータインターフェース
 - 【IF015】 予約の更新標準 API

- ◇ 決済情報

- データの内容
 - 未収、受領済み、料金などの決済情報
- 形式
 - JSON 形式
- 利用するデータインターフェース
 - 【IF015】 予約の更新標準 API

- 出力

- ◇ 決済情報更新結果

- データの内容
 - 決済情報
- データの形式
 - JSON 形式
- 利用するデータインターフェース
 - 【IF015】予約の更新標準 API

【FN208】予約表示<新規開発>

- 本システム機能の概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、MaaS アプリから予約した内容を返却し、MaaS アプリ(my route)での予約内容の表示に活用する。
 - リクエストの予約IDを基に、デマンドバスサービスから予約情報を取得し、ユーザーに表示する。これにより、ユーザーは予約の詳細を確認できる。
- 本システム機能の入力・処理・出力のフローチャート

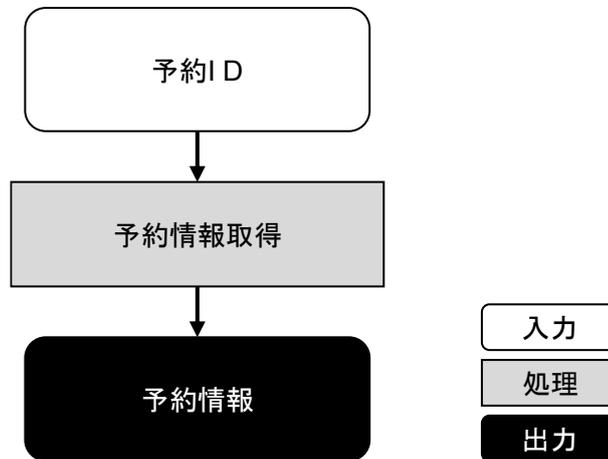


図 2-11 【FN208】のフローチャート

- 本システム機能の処理の詳細
 - 予約情報表示
 - ◇ 処理内容
 - リクエストの予約 ID を基に、デマンドバスサービスから予約を取得する
 - ◇ 利用するライブラリ
 - 【SL001】 Go
 - 【SL002】 Gin
 - 【SL004】 dd-tracer-go
 - 【SL005】 go-jwt-middleware
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 予約 ID
 - データの内容
 - MaaS アプリ側で、管理する予約の ID

- データの形式
 - JSON 形式
- 利用するデータインターフェース
 - 【IF014】ユーザーの予約一覧の取得標準 API
- 出力
 - ◇ 予約情報
 - データの内容
 - 予約 ID に紐づく予約情報
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF014】ユーザーの予約一覧の取得標準 API

【FN209】予約更新<新規開発>

- 本システム機能の概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、予約したデマンドバスの予約のステータスや内容を更新し、結果を返却する。
 - リクエストの予約 ID と更新したい予約情報（乗車人数やキャンセルなど）を基に、デマンドバスサービス側で処理を行い、予約の変更を反映させる。
- 本システム機能の入力・処理・出力のフローチャート

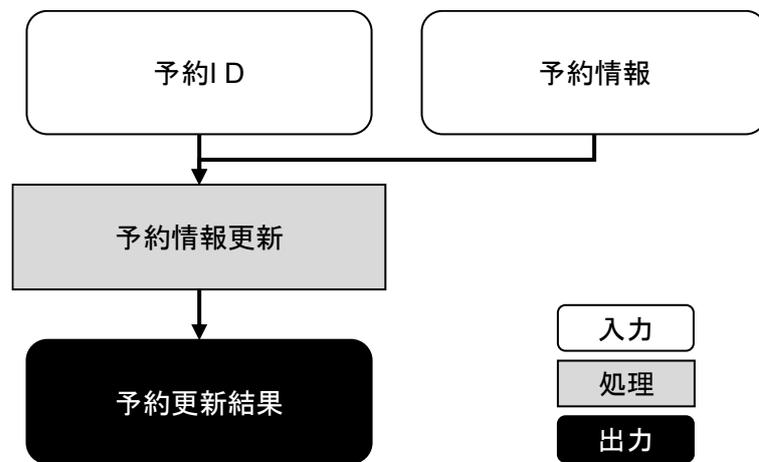


図 2-12 【FN209】のフローチャート

- 本システム機能の処理の詳細
 - 予約情報更新
 - ◇ 処理内容
 - リクエストの予約 ID と更新したい予約情報を基に、予約を更新する
 - ◇ 利用するライブラリ
 - 【SL001】 Go
 - 【SL002】 Gin
 - 【SL004】 dd-tracer-go
 - 【SL005】 go-jwt-middleware
 - ◇ 利用するアルゴリズム

- -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 予約 ID
 - データの内容
 - MaaS アプリ側で、管理する予約の ID
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF015】予約の更新標準 API
 - ◇ 予約情報
 - データの内容
 - 乗車人数など更新内容や、キャンセルなど予約のステータス情報
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF015】予約の更新標準 API
 - 出力
 - ◇ 予約更新結果
 - データの内容
 - 更新後の予約情報
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF015】予約の更新標準 API

【FN210】 運行情報確認 < 新規開発 >

- 本システム機能の概要
 - MaaS アプリから予約した予約情報に紐づく運行情報（車両位置、遅延状況など）を、MaaS アプリ等のサードパーティからのリクエストに応じて返却し、任意のタイミングで MaaS アプリ(my route)での表示に活用する。
 - 車両 ID を基にデマンドバスサービスから情報を取得し、ユーザーにリアルタイムな運行状況を提供する。
- 本システム機能の入力・処理・出力のフローチャート

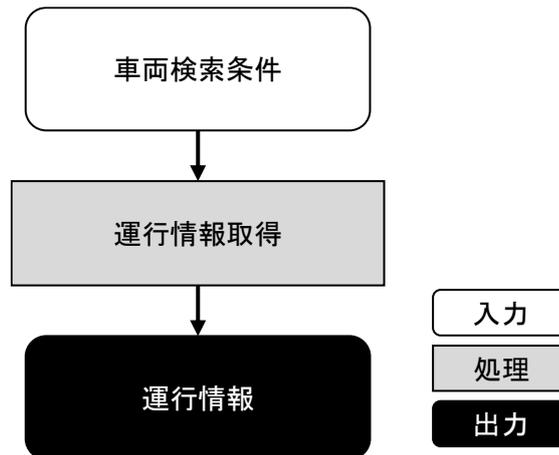


図 2-13 【FN210】のフローチャート

- 本システム機能の処理の詳細
 - 運行情報取得
 - ◇ 処理内容
 - リクエストの車両 ID を基に、運行情報を取得する
 - ◇ 利用するライブラリ
 - 【SL001】 Go
 - 【SL002】 Gin
 - 【SL004】 dd-tracer-go
 - 【SL005】 go-jwt-middleware
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 車両検索条件
 - データの内容
 - MaaS アプリ側で、管理する車両 ID やサービスの ID など
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF018】 車両位置情報取得標準 API
 - 【IF019】 運行遅延情報取得標準 API
 - 出力
 - ◇ 運行情報
 - データの内容
 - 車両の位置情報などの運行情報
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF018】 車両位置情報取得標準 API
 - 【IF019】 運行遅延情報取得標準 API

【FN211】 運賃支払い確認 <新規開発>

- 本システム機能の概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、予約単位で決済対象、決済状態、決済金額などの決済情報を取得・返却し、MaaS アプリでの表示に活用する。
 - リクエストの予約 ID を基に、デマンドバスサービスから必要な決済情報を取得し、ユーザーに表示する。
- 本システム機能の入力・処理・出力のフローチャート

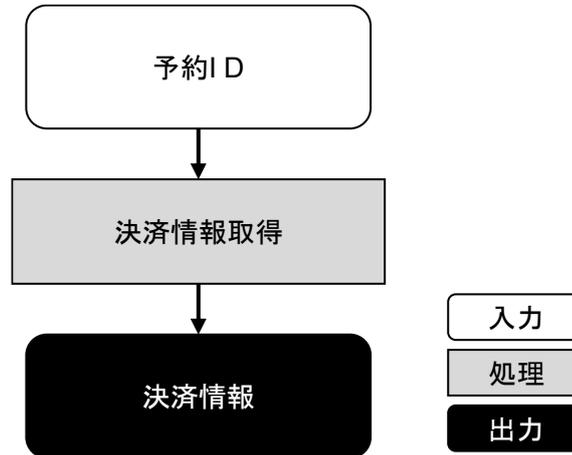


図 2-14 【FN211】 のフローチャート

- 本システム機能の処理の詳細
 - 決済情報取得
 - ◇ 処理内容
 - リクエストの予約 ID を基に、決済情報を取得する
 - ◇ 利用するライブラリ
 - 【SL001】 Go
 - 【SL002】 Gin
 - 【SL004】 dd-tracer-go
 - 【SL005】 go-jwt-middleware
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 予約 ID
 - データの内容
 - MaaS アプリ側で管理する予約の ID
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF016】 予約の決済情報の取得標準 API
 - 出力
 - ◇ 決済情報
 - データの内容

- 決済ステータスなどを含む車両情報
- データの形式
 - JSON 形式
- 利用するデータインターフェース
 - 【IF016】予約の決済情報の取得標準 API

【FN212】予約履歴確認<新規開発>

- 本システム機能の概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、指定された検索条件に基づいて、ユーザーの予約履歴情報を取得・返却し、MaaS アプリ(my route)での予約履歴表示に活用する。
 - リクエストのユーザーID を基に、デマンドバスサービス側で過去の予約情報を検索し、MaaS アプリに提供することで、履歴を参照可能にする。
- 本システム機能の入力・処理・出力のフローチャート

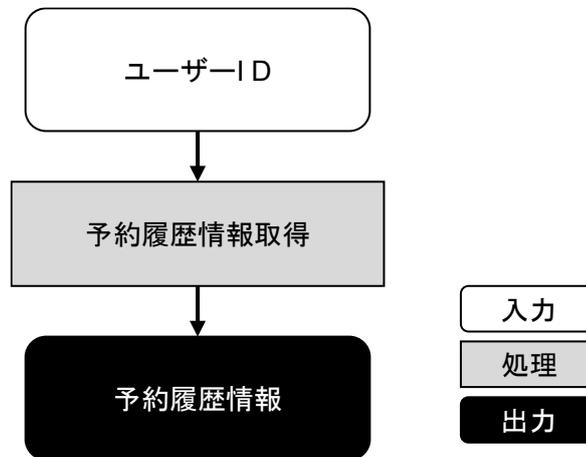


図 2-15 【FN212】のフローチャート

- 本システム機能の処理の詳細
 - 予約履歴情報取得
 - ◇ 処理内容
 - リクエストのユーザーID を基に、予約履歴情報を取得する
 - ◇ 利用するライブラリ
 - 【SL001】 Go
 - 【SL002】 Gin
 - 【SL004】 dd-tracer-go
 - 【SL005】 go-jwt-middleware
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ ユーザーID
 - データの内容
 - MaaS アプリ側で、管理するユーザーの ID
 - データの形式

- JSON 形式
- 利用するデータインターフェース
 - 【IF014】 ユーザーの予約一覧の取得標準 API
- 出力
 - ◇ 予約履歴情報
 - データの内容
 - 過去の予約情報
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF014】 ユーザーの予約一覧の取得標準 API

【FN213】 ユーザー照会<新規開発>

- 本システム機能の概要
 - MaaS アプリ等のサードパーティからのリクエストに応じデマンドバスサービスに登録済みのユーザー情報（基本情報や規約同意一覧）を取得・返却し、MaaS アプリ(my route)でのユーザー情報表示に活用する。
 - リクエストのユーザーID を基に照会処理を行う。
- 本システム機能の入力・処理・出力のフローチャート

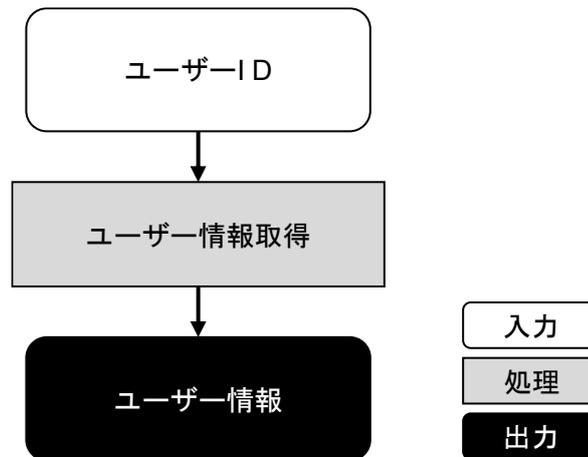


図 2-16 【FN213】 のフローチャート

- 本システム機能の処理の詳細
 - ユーザー情報取得
 - ◇ 処理内容
 - リクエストのユーザーID を基に、デマンドバスサービスに登録済みのユーザー情報を取得する
 - ◇ 利用するライブラリ
 - 【SL001】 Go
 - 【SL002】 Gin
 - 【SL004】 dd-tracer-go
 - 【SL005】 go-jwt-middleware
 - ◇ 利用するアルゴリズム

- -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ ユーザーID
 - データの内容
 - MaaS アプリ側で管理するユーザーの ID
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF005】ユーザー情報の取得標準 API
 - 【IF009】ユーザーの規約同意一覧の取得標準 API
 - 出力
 - ◇ ユーザー情報
 - データの内容
 - ユーザーの基本情報
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF005】ユーザー情報の取得標準 API
 - 【IF009】ユーザーの規約同意一覧の取得標準 API

【FN214】ユーザー情報更新<新規開発>

- 本システム機能の概要
 - MaaS アプリ等のサードパーティからのユーザー更新リクエストに応じ、デマンドバスサービスに登録済みのユーザー情報を更新し、結果を返却する。
 - リクエストのユーザーID と更新したい基本情報を基に、デマンドバスサービス側で処理を行い、ユーザー情報の変更を反映する。
- 本システム機能の入力・処理・出力のフローチャート

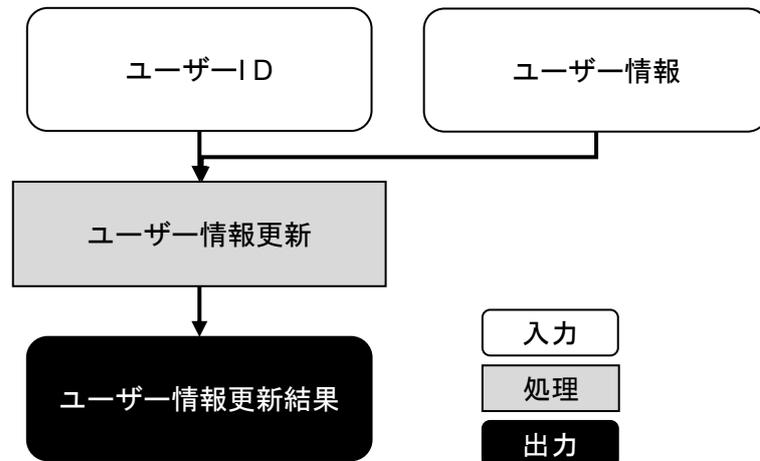


図 2-17 【FN214】のフローチャート

- 本システム機能の処理の詳細
 - ユーザー情報更新

- ◇ 処理内容
 - リクエストのユーザーID を基に、デマンドバスサービスに登録済みのユーザー情報を更新する
- ◇ 利用するライブラリ
 - 【SL001】 Go
 - 【SL002】 Gin
 - 【SL004】 dd-tracer-go
 - 【SL005】 go-jwt-middleware
- ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ ユーザーID
 - データの内容
 - MaaS アプリ側で、管理するユーザーの ID
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF006】ユーザー情報の更新標準 API
 - ◇ ユーザー情報
 - データの内容
 - 更新したいユーザーの基本情報
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF006】ユーザー情報の更新標準 API
 - 出力
 - ◇ ユーザー情報更新結果
 - データの内容
 - 更新後のユーザーの基本情報
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF006】ユーザー情報の更新標準 API

【FN215】ユーザー退会<新規開発>

- 本システム機能の概要
 - MaaS アプリ等のサードパーティからの退会リクエストに応じ、指定されたユーザーの退会処理（ユーザー情報の削除）を行い、結果を返却する。
 - リクエストのユーザーID を基に、デマンドバスサービス側で登録済みのユーザー情報を削除し、サービスからの退会を実現する。

- 本システム機能の入力・処理・出力のフローチャート

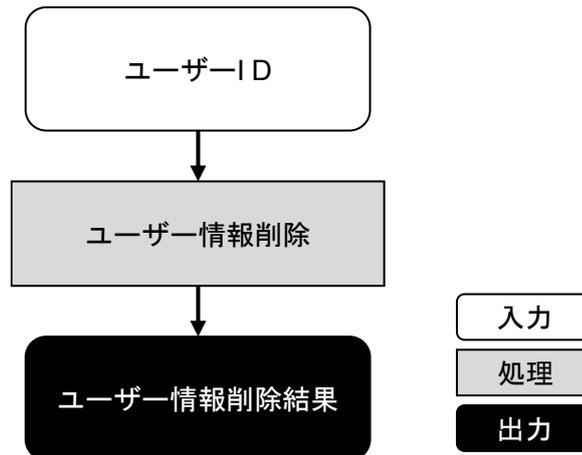


図 2-18 【FN215】のフローチャート

- 本システム機能の処理の詳細
 - ユーザー情報削除
 - ◇ 処理内容
 - リクエストのユーザーID を基に、デマンドバスサービスに登録済みのユーザー情報を削除する
 - ◇ 利用するライブラリ
 - 【SL001】 Go
 - 【SL002】 Gin
 - 【SL004】 dd-tracer-go
 - 【SL005】 go-jwt-middleware
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ ユーザーID
 - データの内容
 - MaaS アプリ側で、管理するユーザーの ID
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF007】ユーザー情報の削除（同意情報を含む）標準 API
 - 出力
 - ◇ ユーザー情報削除結果
 - データの内容
 - 過去の予約情報
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 【IF007】ユーザー情報の削除（同意情報を含む）標準 API

【FN301】 外部連携用認証

- 本システム機能の概要
 - 外部連携 API サーバーがバックエンドのデマンドバスサービスへアクセスする際に、認証のために JSON Web Token (JWT) を発行する。
 - MaaS アプリ側からの JWT 要求情報 (クライアント ID 等) に基づき JWT を発行する。
- 本システム機能の入力・処理・出力のフローチャート

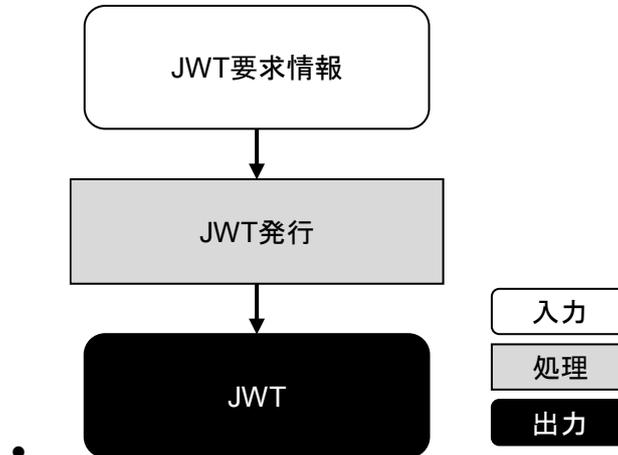


図 2-19 【FN301】 のフローチャート

- 本システム機能の処理の詳細
 - JWT 発行
 - ◇ 処理内容
 - JWT 要求情報を基に、JWT を発行する
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ JWT 要求情報
 - データの内容
 - MaaS アプリ側から送信されてくる JWT リクエスト (クライアント ID、クライアントシークレットなど)
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - -
 - 出力
 - ◇ JWT
 - データの内容
 - デマンドバス API をリクエストするための JSON Web Token
 - データの形式
 - JSON 形式

- 利用するデータインターフェース
 - -

【FN401】デマンドバス API

- 本システム機能の概要
 - デマンドバスサービスを運営する事業者が公開している API の機能。
 - 標準 API から送信されてくるリクエスト情報に基づき、既存システム内で API ごとに定義された処理を実行する。
- 本システム機能の入力・処理・出力のフローチャート

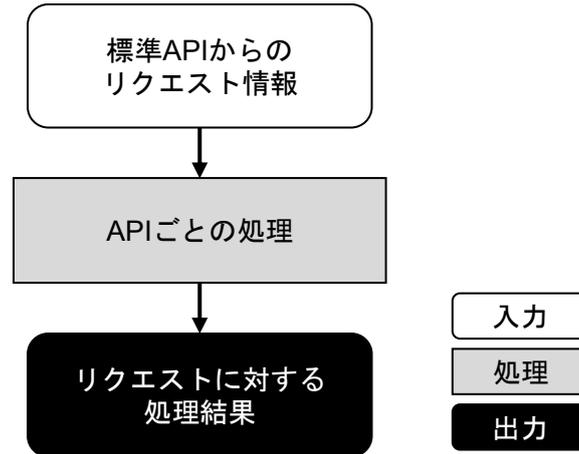


図 2-20 【FN401】のフローチャート

- 本システム機能の処理の詳細
 - API ごとの処理
 - ◇ 処理内容
 - API ごとに定義された処理を実行する
 - ◇ 利用するライブラリ
 - -
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ 標準 API からのリクエスト情報
 - データの内容
 - 標準 API から送信されてくるデマンドバス API の IF 定義に従ったリクエスト情報
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
 - 全 IF に対応し、既存システム内で IF に合わせた処理を実施
 - 【IF001】サービスの概要情報一覧の取得標準 API
 - 【IF002】サービス詳細情報の取得標準 API
 - 【IF003】プラットフォーム規約一覧取得 API
 - 【IF004】ユーザー情報の新規登録標準 API

- 【IF005】 ユーザー情報の取得標準 API
 - 【IF006】 ユーザー情報の更新標準 API
 - 【IF007】 ユーザー情報の削除（同意情報を含む）標準 API
 - 【IF008】 ユーザー規約同意登録標準 API
 - 【IF009】 ユーザーの規約同意一覧の取得標準 API
 - 【IF010】 乗降場所一覧の取得標準 API
 - 【IF011】 予約候補の生成標準 API
 - 【IF012】 予約の登録標準 API
 - 【IF013】 ユーザーが利用可能なサービスの概要情報一覧を取得標準 API
 - 【IF014】 ユーザーの予約一覧の取得標準 API
 - 【IF015】 予約の更新標準 API
 - 【IF016】 予約の決済情報の取得標準 API
 - 【IF017】 予約の決済情報の登録または更新標準 API
 - 【IF018】 車両位置情報取得標準 API
 - 【IF019】 運行遅延情報取得標準 API
- 出力
- ◇ リクエストに対する処理結果
 - データの内容
 - デマンドバス API リクエスト結果
 - データの形式
 - JSON 形式
 - 利用するデータインターフェース
- 全 IF に対応し、既存システム内で IF に合わせた処理を実施
- 【IF001】 サービスの概要情報一覧の取得標準 API
 - 【IF002】 サービス詳細情報の取得標準 API
 - 【IF003】 プラットフォーム規約一覧取得 API
 - 【IF004】 ユーザー情報の新規登録標準 API
 - 【IF005】 ユーザー情報の取得標準 API
 - 【IF006】 ユーザー情報の更新標準 API
 - 【IF007】 ユーザー情報の削除（同意情報を含む）標準 API
 - 【IF008】 ユーザー規約同意登録標準 API
 - 【IF009】 ユーザーの規約同意一覧の取得標準 API
 - 【IF010】 乗降場所一覧の取得標準 API
 - 【IF011】 予約候補の生成標準 API
 - 【IF012】 予約の登録標準 API
 - 【IF013】 ユーザーが利用可能なサービスの概要情報一覧を取得標準 API
 - 【IF014】 ユーザーの予約一覧の取得標準 API
 - 【IF015】 予約の更新標準 API
 - 【IF016】 予約の決済情報の取得標準 API
 - 【IF017】 予約の決済情報の登録または更新標準 API
 - 【IF018】 車両位置情報取得標準 API

➤ 【IF019】 運行遅延情報取得標準 API

【FN501】 運行情報確認

- 本システム機能の概要
 - デマンドバスサービスのスケジュールや運行情報をドライバーが確認する。
 - ドライバーアプリの UI からの入力や画面遷移を基に、バックエンドの API サーバーにリクエストを送信し、運行情報を取得・表示する。
- 本システム機能の入力・処理・出力のフローチャート

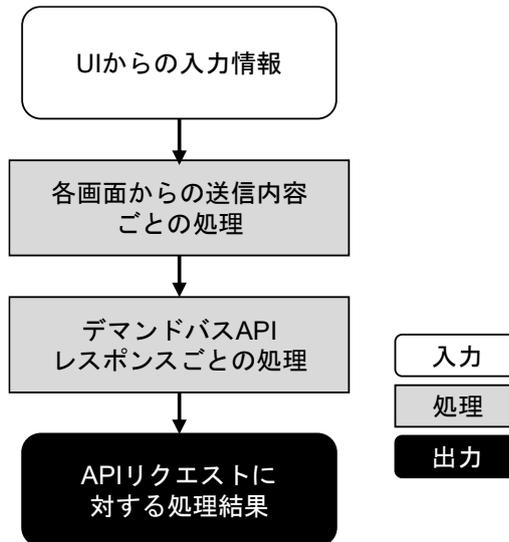


図 2-21 【FN501】 のフローチャート

- 本システム機能の処理の詳細
 - 各画面からの送信内容ごとの処理
 - ◇ 処理内容
 - ドライバーがドライバーアプリの UI から入力した情報や画面遷移を基に、バックエンドのデマンドバス API サーバーに、API リクエストを送れるようにデータを成形し送信
 - ◇ 利用するライブラリ
 - 【SL010】 ドライバーアプリ
 - ◇ 利用するアルゴリズム
 - -
 - デマンドバス API レスポンスごとの処理
 - ◇ 処理内容
 - バックエンドのデマンドバス API サーバーのレスポンスごとに、画面表示するために必要なデータの処理を実施
 - ◇ 利用するライブラリ
 - 【SL010】 ドライバーアプリ
 - ◇ 利用するアルゴリズム
 - -
- 本システム機能の入出力データの仕様
 - 入力
 - ◇ UI からの入力情報

デマンドバスシステム連携 API システム設計書

- データの内容
 - ドライバーアプリを利用するドライバーが UI から入力した情報
- データの形式
 - UI コンポーネント（ネイティブアプリ）
- 利用するデータインターフェース
 - -
- 出力
 - ◇ API リクエストに対する処理結果
 - データの内容
 - デマンドバス API にリクエストした内容の処理結果
 - データの形式
 - UI コンポーネント（ネイティブアプリ）
 - 利用するデータインターフェース
 - -

2-1-4. ソフトウェア・ライブラリ (SL) の詳細

表 2-3 ソフトウェア・ライブラリー一覧

※朱文字：新規開発・既存改修

ID	名称	バージョン	内容
SL001	Go	1.23.5	● バックエンド API の開発に使用しているプログラミング言語
SL002	Gin	1.10.1	● バックエンド開発に使用する Web フレームワーク
SL003	aws-sdk-go-v2	V1.36.6	● バックエンド API から、AWS リソースにアクセスする際に利用するライブラリ
SL004	dd-tracer-go	V2.1.0	● datadog にトレースを送信するために利用するライブラリ
SL005	go-jwt-middleware	V2.3.0	● JWT の発行・検証等で利用するミドルウェア
SL006	Kubernetes	1.32	● コンテナの実行・スケーリング基盤
SL007	Fluentbit	-	● サーバログのログルーターとして利用するソフトウェア
SL008	Datadog Agent	7.66.1	● APM (Application Performance Monitoring) 、ログ、メトリクス監視、障害検知・通知に利用するソフトウェア
SL009	my route	-	● 外部連携 API に接続し、デマンドバスの予約等を実施するアプリ
SL010	MONET ドライバ ーアプリ	-	● デマンドバスサービスを運行する際に、ドライバーが運行スケジュールを確認するためのアプリ

【SL001】 Go

- ベンダー
 - OSS
- 公式サイト
 - <https://go.dev/>
- 本ソフトウェア・ライブラリの概要
 - Google が開発したプログラミング言語で、シンプルな文法と高いパフォーマンスを両立
 - 並行処理を言語レベルで強力にサポートしており、多くの処理を効率的に実行するのが得意
 - その信頼性と速度から、Web サーバーやマイクロサービス、コマンドラインツールの開発で広く採用されている
- 開発するシステムにおいて利用する機能の詳細
 - API リクエストを処理

【SL002】 Gin

- ベンダー
 - OSS

デマンドバスシステム連携 API システム設計書

- 公式サイト
 - <https://gin-gonic.com/>
- 本ソフトウェア・ライブラリの概要
 - Gin (ジン) は、Go 言語で最も広く利用されている Web アプリケーションフレームワークの一つ
 - 非常に高速なパフォーマンスが最大の特徴で、API サーバーの構築に必要な機能をシンプルに提供
 - HTTP ルーティング、ミドルウェア、JSON の効率的な処理といった機能を備え、Web 開発を迅速に行える
- 開発するシステムにおいて利用する機能の詳細
 - API リクエストを処理

【SL003】 aws-sdk-go-v2

- ベンダー
 - AWS
- 公式サイト
 - <https://github.com/aws/aws-sdk-go-v2>
- 本ソフトウェア・ライブラリの概要
 - Go 言語アプリケーションから AWS サービスを操作するための公式ソフトウェア開発キット
 - V1 から設計が全面的に刷新され、サービスごとにパッケージをインポートするモジュラーアーキテクチャを採用し、パフォーマンスが向上
 - この SDK を使うことで、S3 へのファイルアップロードや EC2 インスタンスの起動といった AWS の操作を Go のコードで直接記述が可能
- 開発するシステムにおいて利用する機能の詳細
 - API サーバーから AWS へのアクセス：DynamoDB など各種コンポーネントへのアクセスが対象

【SL004】 dd-tracer-go

- ベンダー
 - Datadog
- 公式サイト
 - <https://github.com/DataDog/dd-trace-go>
- 本ソフトウェア・ライブラリの概要
 - dd-trace-go は、Datadog が提供する Go 言語アプリケーション向けの公式 APM (アプリケーションパフォーマンス監視) クライアント
 - アプリケーションに導入することで、リクエストの処理の流れを可視化 (分散トレーシング) し、パフォーマンスのボトルネック特定を支援
 - 主要な Web フレームワークやデータベース、gRPCなどを自動で計測 (インストルメンテーション) する機能を備えている
- 開発するシステムにおいて利用する機能の詳細
 - Datadog へ API リクエストのトレースを送信

【SL005】 go-jwt-middleware

- ベンダー

- Auth0
- 公式サイト
 - <https://github.com/auth0/go-jwt-middleware>
- 本ソフトウェア・ライブラリの概要
 - go-jwt-middleware は、Go 言語の Web アプリケーションで JWT (JSON Web Token) 認証を簡単に追加するためのミドルウェア
 - HTTP リクエストに含まれる JWT を自動で検証し、トークンが無効な場合はアクセスをブロックして API エンドポイントを保護
 - 少ないコードでセキュアな認証機能を実装するために利用
- 開発するシステムにおいて利用する機能の詳細
 - API リクエストに含まれる JWT の検証

【SL006】 Kubernetes

- ベンダー
 - OSS
- 公式サイト
 - <https://kubernetes.io/ja/>
- 本ソフトウェア・ライブラリの概要
 - コンテナ化されたアプリケーションのデプロイ、スケーリング、管理を自動化するためのオープンソースプラットフォーム
 - 複数の物理サーバーや仮想サーバーを一つのクラスターとして束ね、障害発生時にはコンテナを自動で復旧させる自己修復機能を持つ
 - 信頼性が高くスケーラブルなマイクロサービスアーキテクチャなどの運用を効率化する基盤として広く利用されている
- 開発するシステムにおいて利用する機能の詳細
 - コンテナ化済みの API サーバーの実行基盤

【SL007】 Fluentbit

- ベンダー
 - OSS
- 公式サイト
 - <https://fluentbit.io/>
- 本ソフトウェア・ライブラリの概要
 - 非常に軽量で高性能なオープンソースのログ収集・転送ツール
 - CPU やメモリの消費量が少ないため、コンテナ、IoT、エッジデバイスなどのリソースが限られた環境での利用に最適化されている
 - 様々な形式のログやメトリクスを収集・解析し、Elasticsearch や各種クラウドの監視サービスなど多様な宛先に転送可能
- 開発するシステムにおいて利用する機能の詳細
 - API サーバーのログを AWS や Datadog に送信

【SL008】 Datadog Agent

- ベンダー
 - Datadog
- 公式サイト
 - <https://github.com/DataDog/datadog-agent>
- 本ソフトウェア・ライブラリの概要
 - サーバーやコンテナにインストールして、パフォーマンスデータを収集し Datadog プラットフォームへ送信するソフトウェア
 - CPU やメモリなどのシステムメトリクス、ログファイル、アプリケーションのトレースといった多様なデータを一元的に収集
 - 豊富なインテグレーションにより、様々なミドルウェアやサービスの監視を可能にし、システム全体の可観測性 (Observability) を高める
- 開発するシステムにおいて利用する機能の詳細
 - API サーバーのパフォーマンスデータを収集

【SL009】 my route

- ベンダー
 - トヨタフィナンシャルサービス
- 公式サイト
 - <https://top.myroute.fun/>
- 本ソフトウェア・ライブラリの概要
 - トヨタが提供する MaaS (Mobility as a Service) プラットフォーム
 - 電車やバス、タクシー、シェアサイクルなど、様々な移動手段を横断して最適なルート検索が可能
 - 一部の交通サービスでは、アプリ内での予約から決済までをシームレスに完結させることも可能
- 開発するシステムにおいて利用する機能の詳細
 - API サーバーに、UI からリクエスト
- イメージ



図 2-22 MaaS アプリのトップ画面例

【SL010】 MONET ドライバーアプリ

- ベンダー
 - MONET Technologies
- 公式サイト
 - <https://www.monet-technologies.com/>
- 本ソフトウェア・ライブラリの概要
 - オンデマンドバスなどの運転手向けに提供される業務用ツール
 - 運行管理者から送られてくる当日の運行計画や、乗客の予約情報、最適な走行ルートなどをリアルタイムで確認できる
 - ナビゲーション機能のほか、乗客の乗降記録やチャット機能も備え、円滑な運行を支援
- 開発するシステムにおいて利用する機能の詳細
 - 運行状況などの確認
- イメージ



図 2-23 ドライバーアプリの表示画面例

2-1-5. 数理モデル・アルゴリズム (AL) の詳細

-

2-2. システムコンポーネント (CO)

2-2-1. システムコンポーネント図

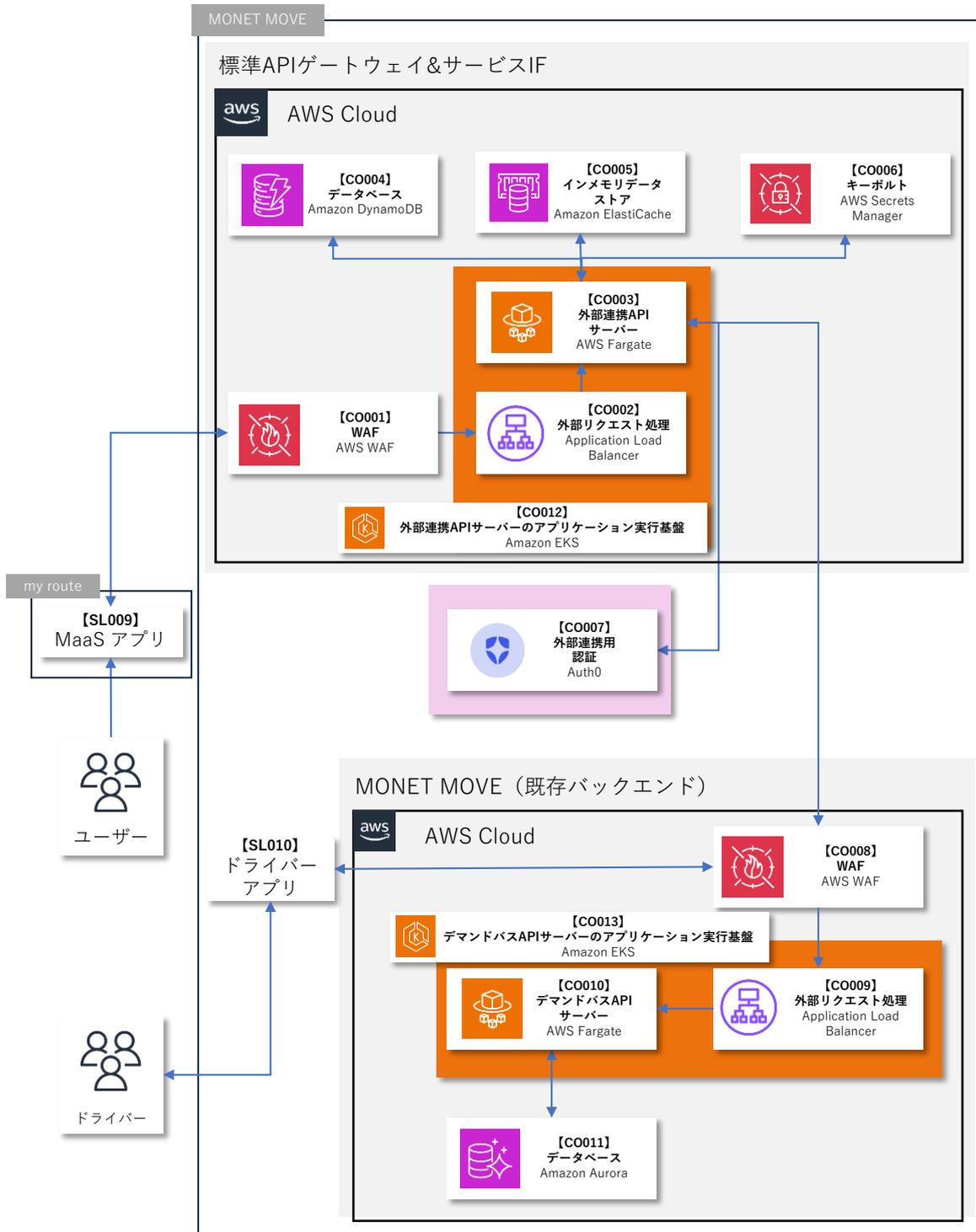


図 2-24 システムコンポーネント図

(Amazon Web Services および AWS の各サービスアイコンは、米国その他の諸国における、Amazon.com, Inc.またはその関連会社の商標です)

2-2-2. システムコンポーネント一覧

表 2-4 システムコンポーネント一覧

ID	種別	コンポーネント名	用途
CO001	WAF	AWS WAF	<ul style="list-style-type: none"> ● クライアントからの不正なリクエストをブロックするために利用。AWS WAF は、ウェブサイトをサイバー攻撃から守るためのサービスである。 ● SQL インジェクションのような不正なアクセスをルールに基づいて検知し、アプリケーションに到達する前にブロックする。 ● CloudFront や Application Load Balancer (ALB) などと連携させて、Web サイト全体のセキュリティを強化できる。
CO002	外部リクエスト処理	Application Load Balancer	<ul style="list-style-type: none"> ● MaaS アプリからの API リクエストを適切なバックエンドのエンドポイントにルーティングするために使用する。Application Load Balancer (ALB) は、リクエスト内容 (ホスト名・URL パス・HTTP ヘッダーなど) に基づいて複数のターゲットに振り分ける高度なレイヤー7 ロードバランサーで、モダンなマイクロサービス構成や複数ドメイン対応に特に適している。 ● TLS 終端 (HTTPS) 、SNI 対応、HTTP/2、WebSocket、gRPC など最新プロトコルをネイティブにサポートし、安全性と通信効率を両立する。
CO003	外部連携 API サーバ	AWS Fargate	<ul style="list-style-type: none"> ● MaaS アプリから受け取ったリクエスト情報を基に、バックエンドのデマンドバス API サーバーにリクエストを送信する。 ● AWS Fargate は、サーバー管理不要の サーバーレスコンテナ実行環境である。 ● 各タスクは専用の実行環境で高いセキュリティ分離がされ、自動スケーリングも組み込まれている。 ● 運用コストやプロビジョニング負担を大幅に削減でき、変動するワークロードに最適化できる。
CO004	データベース	Amazon DynamoDB	<ul style="list-style-type: none"> ● デマンドバス API のレスポンス結果をキャッシュする。Amazon DynamoDB は、AWS が提供する サーバーレスの NoSQL キー・バリュー/ドキュメント型データベースである。 ● クラスターのプロビジョニングやパッチ等の管理は不要で、オンデマンド課金+自動スケーリングにより未使用時はコストゼロで運用可能である。
CO005	インメモリデータスト	Amazon ElastiCache	<ul style="list-style-type: none"> ● Auth0 へのアクセス回数を減らすために、リクエスト情報の一部をキャッシュするために利用する。

	ア		<ul style="list-style-type: none"> ● AWS ElastiCache は、アプリケーションの応答速度を高速化するためのフルマネージド型インメモリキャッシュサービスである。 ● データベースへの問い合わせ結果や、頻繁に利用するデータを、ハードディスクよりも遥かに高速なメモリ上に一時的に保存（キャッシュ）することで、システム全体のパフォーマンスを劇的に向上させる。
CO006	キーボルト	AWS Secrets Manager	<ul style="list-style-type: none"> ● デマンドバス API にリクエストするための鍵を保存するために利用する。 ● AWS Secrets Manager は、データベースのパスワードや API キーといった機密情報（シークレット）を安全に保管し、管理するためのサービスである。 ● アプリケーションのソースコード内にパスワードなどをハードコーディングする必要がなくなり、セキュリティを大幅に向上できる。
CO007	外部連携用認証	Auth0	<ul style="list-style-type: none"> ● Auth0 は、アプリや API に認証（ログイン）・認可（アクセス制御）機能をすぐに組み込めるクラウド型プラットフォームで、パスワード・ソーシャルログイン・パスワードレス・MFA などを幅広くサポートしている。
CO008	WAF	AWS WAF	<ul style="list-style-type: none"> ● CO002 と同様。
CO009	外部リクエスト処理	Application Load Balancer	<ul style="list-style-type: none"> ● 標準からの API リクエストを適切なバックエンドのエンドポイントにルーティングするために使用する。 ● Application Load Balancer に関する詳細は、CO003 を参照。
CO010	デマンドバス API サーバー	AWS Fargate	<ul style="list-style-type: none"> ● デマンドバス運営事業者が公開している API。Fargate に関する詳細は、CO004 を参照。
CO011	データベース	Amazon Aurora	<ul style="list-style-type: none"> ● デマンドバス API のユーザー情報や予約情報を管理しているデータベース。 ● Amazon Aurora は、MySQL 及び PostgreSQL と互換性を持つクラウドネイティブなフルマネージド型リレーショナルデータベースエンジンであり、自動的に最大 128 TiB まで拡張する高速分散ストレージによって従来の MySQL 比で最大 5 倍、PostgreSQL 比で 3 倍のスループットを実現し、最大 15 台のリードレプリカとマルチ AZ 構成による高可用性を備えている。
CO012	アプリ実行基盤	Amazon EKS	<ul style="list-style-type: none"> ● 外部連携 API サーバーのアプリケーション実行基盤。Amazon EKS はコンテナ管理ツール「Kubernetes」のマネージドサービス。 ● Kubernetes の頭脳部分である「コントロールプレーン」の構築や運用の手間を省くことが可能になり、ユー

デマンドバスシステム連携 API システム設計書

			<p>ザーはインフラ管理の手間を減らし、アプリケーションの開発と実行に集中できる。</p>
CO013	アプリ実行 基盤	Amazon EKS	<ul style="list-style-type: none"> ● デマンドバス API サーバーのアプリケーション実行基盤。 ● EKS に関する説明は、CO012 を参照

2-3. ハードウェア (HW)

2-3-1. ハードウェアアーキテクチャ



図 2-25 ハードウェアアーキテクチャ

2-3-2. ハードウェア一覧

表 2-5 ハードウェア一覧

※朱文字：新規開発・既存改修

ID	種別	ベンダー	品番	用途
HW001	スマートフォン	Apple	iPhone シリーズ	● MaaS アプリの利用

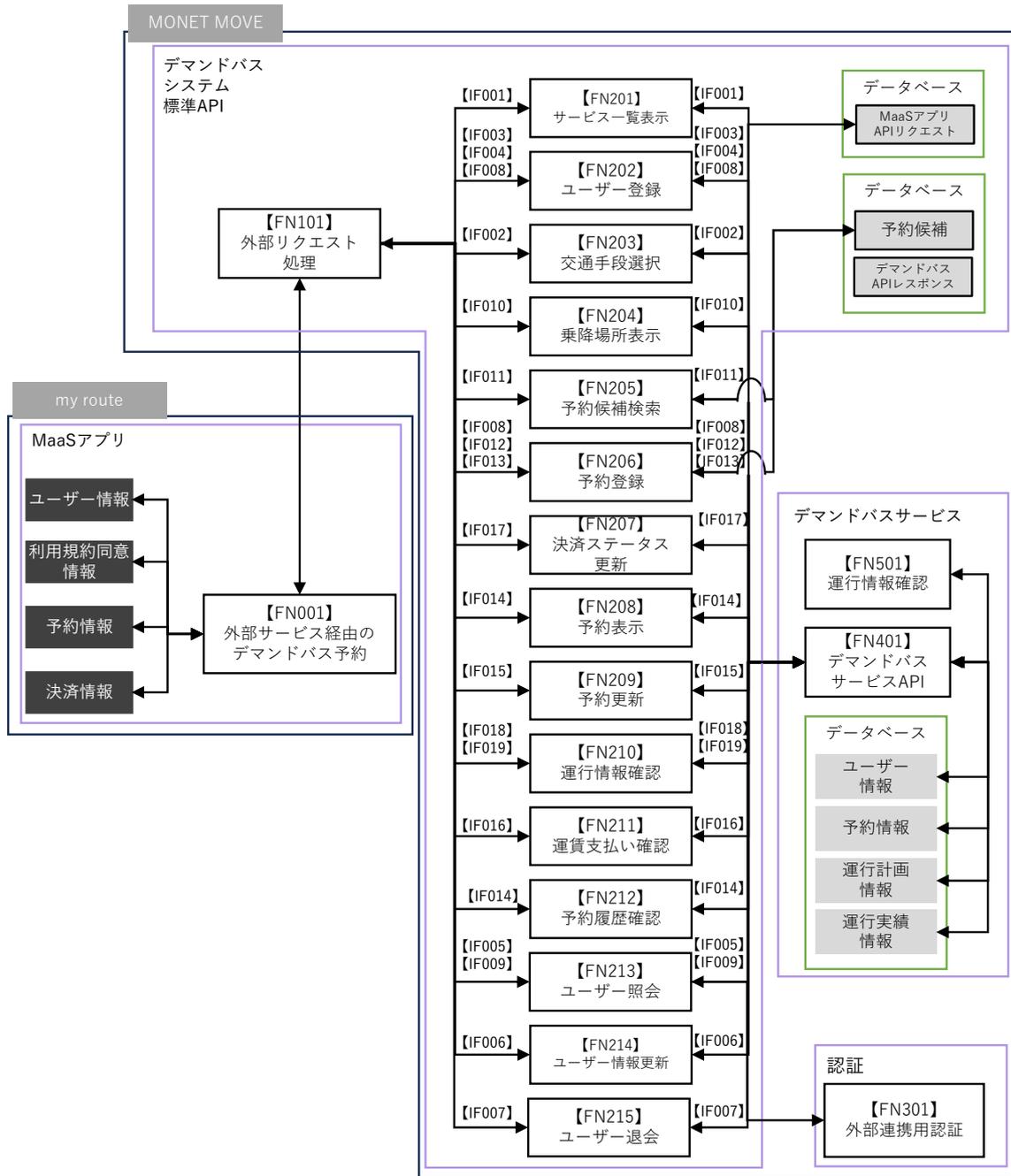
2-3-3. ハードウェアの詳細

【HW001】スマートフォン

- 本ハードウェアの概要
 - 検証用の MaaS アプリをインストールするスマートフォン
- 本ハードウェアを提供するベンダー
 - 検証用に iOS で準備
- 本ハードウェアの仕様・スペック
 - iOS の最新版をインストールし、検証用 MaaS アプリが動作すること

2-4. データインターフェース (IF)

2-4-1. データアーキテクチャ



凡例 UI入力 データ データベース ファイルストレージ ソフトウェア データ処理

図 2-26 データアーキテクチャ

2-4-2. データインターフェース一覧

表 2-6 データインターフェース一覧

※朱文字：新規開発・既存改修

ID	名称	出力側 ID	入力側 ID
IF001	サービスの概要情報一覧の取得標準 API	FN101	FN401
IF002	サービス詳細情報の取得標準 API	FN101	FN401
IF003	プラットフォーム規約一覧取得標準 API	FN101	FN401
IF004	ユーザー情報の新規登録標準 API	FN101	FN401
IF005	ユーザー情報の取得標準 API	FN101	FN401
IF006	ユーザー情報の更新標準 API	FN101	FN401
IF007	ユーザー情報の削除（同意情報含む）標準 API	FN101	FN401
IF008	ユーザー規約同意登録標準 API	FN101	FN401
IF009	ユーザーの規約同意一覧の取得標準 API	FN101	FN401
IF010	乗降場所一覧の取得標準 API	FN101	FN401
IF011	予約候補の生成標準 API	FN101	FN401
IF012	予約の登録標準 API	FN101	FN401
IF013	ユーザーが利用可能なサービスの概要情報一覧を取得標準 API	FN101	FN401
IF014	ユーザーの予約一覧の取得標準 API	FN101	FN401
IF015	予約の更新標準 API	FN101	FN401
IF016	予約の決済情報の取得標準 API	FN101	FN401
IF017	予約の決済情報の登録または更新標準 API	FN101	FN401
IF018	車両位置情報取得標準 API	FN101	FN401
IF019	運行遅延情報取得標準 API	FN101	FN401

2-4-3. データインターフェースの詳細

データインターフェースの詳細を記す。なお、本業務において開発（新規・改修）を行うデータインターフェースを**朱文字**で示す。

【IF001】サービスの概要情報一覧の取得標準 API

- 本インターフェースの概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、デマンドバスサービスの概要情報を一覧で取得・返却するための API
- 本インターフェースを利用する機能
 - **【FN201】サービス一覧表示**
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ GET
 - レスポンスデータ形式
 - ◇ JSON
 - URI
 - ◇ <https://example.com/services>
 - 使用例
 - ◇ [https://example.com/services?service_ids=1,2,3&name=港区
&prefecture_code=13&municipality_code=13103&offset=30&limit=10](https://example.com/services?service_ids=1,2,3&name=港区&prefecture_code=13&municipality_code=13103&offset=30&limit=10)
 - リクエストパラメータ

QUERY PARAMETERS

service_ids	<p>Array of strings</p> <p>Example: <code>service_ids=1,2,3</code></p> <p>取得対象のサービスIDをカンマ区切りで指定します。 指定されたいずれかのサービスIDに該当する対象を取得します (OR条件)。</p>
name	<p>string</p> <p>Example: <code>name=港区</code></p> <p>取得対象のサービスの名称 (部分一致) を指定します。</p>
prefecture_code	<p>string</p> <p>Example: <code>prefecture_code=13</code></p> <p>都道府県コードを指定します。</p>
municipality_code	<p>string</p> <p>Example: <code>municipality_code=13103</code></p> <p>市区町村コードを指定します。</p>
status	<p>Array of strings</p> <p>Default: <code>"upcoming,active"</code></p> <p>Items Enum: <code>"upcoming" "active" "ended"</code></p> <p>Example: <code>status=upcoming,active</code></p> <p>フィルター対象とするサービスのステータスを複数指定します。</p> <ul style="list-style-type: none"> <code>upcoming</code> 利用開始日時を迎える前のサービス <code>active</code> 現在利用できるサービス <code>ended</code> 利用終了日時を経過したサービス
offset	<p>integer <code>>= 0</code></p> <p>Default: <code>0</code></p> <p>Example: <code>offset=30</code></p> <p>取得する結果の開始位置を指定します。</p>
limit	<p>integer <code>[1..100]</code></p> <p>Default: <code>20</code></p> <p>Example: <code>limit=10</code></p> <p>取得する結果の最大件数を指定します。</p>

QUERY PARAMETERS

service_ids	<p>Array of strings</p> <p>Example: <code>service_ids=1,2,3</code></p> <p>取得対象のサービスIDをカンマ区切りで指定します。指定されたいずれかのサービスIDに該当する対象を取得します (OR条件)。</p>
name	<p>string</p> <p>Example: <code>name=港区</code></p> <p>取得対象のサービスの名称 (部分一致) を指定します。</p>
prefecture_code	<p>string</p> <p>Example: <code>prefecture_code=13</code></p> <p>都道府県コードを指定します。</p>
municipality_code	<p>string</p> <p>Example: <code>municipality_code=13103</code></p> <p>市区町村コードを指定します。</p>
status	<p>Array of strings</p> <p>Default: <code>["upcoming","active"]</code></p> <p>Items Enum: <code>"upcoming" "active" "ended"</code></p> <p>Example: <code>status=upcoming,active</code></p> <p>フィルター対象とするサービスのステータスを複数指定します。</p> <ul style="list-style-type: none"> <code>upcoming</code> 利用開始日時を迎える前のサービス <code>active</code> 現在利用できるサービス <code>ended</code> 利用終了日時を経過したサービス
offset	<p>integer <code>>= 0</code></p> <p>Default: <code>0</code></p> <p>Example: <code>offset=30</code></p> <p>取得する結果の開始位置を指定します。</p>
limit	<p>integer <code>[1 .. 100]</code></p> <p>Default: <code>20</code></p> <p>Example: <code>limit=10</code></p> <p>取得する結果の最大件数を指定します。</p>

➤ レスポンスパラメータ

◇ 200 OK

RESPONSE SCHEMA: application/json

services **required** Array of objects **>= 0 items**
サービス情報のリストです。

Array (**>= 0 items**) [

id required	string サービスの一意な識別子です。
name required	string サービスの名称です。
description	string サービスの説明です。
brand_image_url	string サービスのブランドイメージ画像のURLです。
operation_type required	string Enum: "on_demand" "slot_area_batch_confirmed" "fixed_time_fixed_route" サービスの運行方式の種類を示します。 <ul style="list-style-type: none">• on_demand 利用者の予約に応じて即時配車されるオンデマンド方式。• slot_area_batch_confirmed 時間帯ごとに運行エリアを切り替える方式で、各時間帯の締切時刻を設け、締切後に暫定予約をまとめて確定し、効率的なルートを生成して運行します。• fixed_time_fixed_route 決まったルートと時刻で運行される固定路線方式。

<p>allow_ridepooling required</p>	<p>boolean 相乗りが発生し得るか否かを示します。</p>
<p>start_datetime required</p>	<p>string <date-time> サービスの開始日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。</p>
<p>end_datetime</p>	<p>string <date-time> サービスの終了日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T20:59:59+09:00）です。この項目が指定されていない場合、サービスは「無期限（終了日未定）」として扱われます。</p>
<p>service_group_id</p>	<p>string サービスの上位グループの一意な識別子です。指定がない場合は、サービスが単独で存在することを示します。</p>
<p>cities required</p>	<p>Array of objects >= 0 items items サービスが提供される都道府県・市区町村のリストです。</p>

Array (>= 0 items) [

<p>prefecture_code required</p>	<p>string 都道府県コードです。</p>
<p>prefecture_name required</p>	<p>string 都道府県名です。</p>
<p>municipalities required</p>	<p>Array of objects >= 0 items 市区町村のリストです。</p>

Array (>= 0 items) [

<p>code required</p>	<p>string 市区町村コードです。</p>
<p>name required</p>	<p>string 市区町村名です。</p>

]

]

service_terms ▾
required

Array of objects `>= 0`
items
サービス利用に関する規約のリストです。

Array (`>= 0` items) [

<p>id required</p>	<p>string 規約の一意な識別子です。</p>
<p>category required</p>	<p>string Enum: "service" "privacy" "cancellation" "user" "third_party" 規約の分類カテゴリです。</p> <ul style="list-style-type: none"> • <code>service</code> サービス利用に関する規約。 • <code>privacy</code> プライバシーポリシーに関する規約。 • <code>cancellation</code> キャンセルポリシーに関する規約。 • <code>user</code> ユーザー向けの規約。 • <code>third_party</code> サードパーティとの関係に関する規約。
<p>name</p>	<p>string 規約の名称です。</p>
<p>url required</p>	<p>string 規約のURLです。</p>
<p>agreement_required required</p>	<p>boolean 規約に対してユーザーから同意を得ることが必須かどうかを示します。</p>
<p>created_at</p>	<p>string <date-time> 作成日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。</p>
<p>updated_at</p>	<p>string <date-time> 最終更新日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。</p>

]

available_passenger_types required Array of objects non-empty

サービスがサポートする乗客のタイプの情報です。
乗客のタイプは、例えば一般/未就学児/シニアなど、運賃が異なる乗客を示します。
order は昇順で表示して、未設定の場合は後方に配置します。

```

Array (non-empty) [
  passenger_type_id string required 乗客タイプの一意な識別子です。
  name string required 乗客タイプの名称です。
  order integer 表示順を示す番号です。
  description string 乗客タイプの説明です。
]
    
```

supported_accessibility_features required Array of strings >= 0 items

Items Enum:

- "wheelchair"
- "folding_wheelchair"
- "powered_wheelchair"
- "oversized_wheelchair"
- "scooter"
- "oversized_scooter"
- "walker"
- "oversized_walker"
- "only_car_walker"
- "only_car_rider"
- "extra_space_ambulance"
- "actual_door_to_door"
- "lift"
- "oversized_item"
- "stroller"
- "child_seat_baby"
- "child_seat_young"
- "bike"
- "white_cane"
- "service_animal"
- "oxygen_tank"
- "stretcher"

サービスがサポートする乗客補助機能のリストです。

- `wheelchair`
車椅子
- `folding_wheelchair`
折りたたみ式車椅子
- `powered_wheelchair`
電動車椅子
- `oversized_wheelchair`
大型車椅子
- `scooter`
スクーター
- `oversized_scooter`
大型スクーター
- `walker`
歩行器
- `oversized_walker`
大型歩行器
- `only_car_walker`
車両内のみ歩行可能な乗客向け
- `only_car_rider`
車両内のみ乗車可能な乗客向け
- `extra_space_ambulant`
追加スペースを必要とする歩行者向け
- `actual_door_to_door`
実際のドアからドアまでのサービス
- `lift`
リフト付き車両
- `oversized_item`
大型荷物
- `stroller`
ベビーカー
- `child_seat_baby`
ベビーシート
- `child_seat_young`
幼児用シート
- `bike`
自転車
- `white_cane`
白杖使用者向けサービス
- `service_animal`
介助動物同伴可能
- `oxygen_tank`
酸素ボンベ使用者向けサービス
- `stretcher`
ストレッチャー

operating_hours **required**

object
サービスの曜日および祝日ごとの運行時間帯を示します。
各曜日のプロパティは、当日中における運行時間帯のリストです。
該当の曜日が空配列 ([]) の場合は、当日は連休日とみなされます。
営業時間の制限がない場合は、00:00:00~23:59:59を返します。

monday **required**

Array of objects **>= 0 items**
月曜日の運行時間帯のリストです。

Array (**>= 0 items**) [

start_time_offset_sec **required** integer
運行開始時刻の当日
00:00からのオフセット
(秒単位) です。

end_time_offset_sec **required** integer
運行終了時刻の当日
00:00からのオフセット
(秒単位) です。

]

tuesday **required**

Array of objects **>= 0 items**
火曜日の運行時間帯のリストです。

Array (**>= 0 items**) [

start_time_offset_sec **required** integer
運行開始時刻の当日
00:00からのオフセット
(秒単位) です。

end_time_offset_sec **required** integer
運行終了時刻の当日
00:00からのオフセット
(秒単位) です。

]

→ wednesday ▾ Array of objects **>= 0 items**
required 水曜日の運行時間帯のリストです。

```
Array (>= 0 items) [
  | start_time_offset_sec integer
  | required              運行開始時刻の当日
  |                       00:00からのオフセット
  |                       (秒単位) です。
  |-----|
  | end_time_offset_sec   integer
  | required              運行終了時刻の当日
  |                       00:00からのオフセット
  |                       (秒単位) です。
  ]
```

→ thursday ▾ Array of objects **>= 0 items**
required 木曜日の運行時間帯のリストです。

```
Array (>= 0 items) [
  | start_time_offset_sec integer
  | required              運行開始時刻の当日
  |                       00:00からのオフセット
  |                       (秒単位) です。
  |-----|
  | end_time_offset_sec   integer
  | required              運行終了時刻の当日
  |                       00:00からのオフセット
  |                       (秒単位) です。
  ]
```

→ friday ▾ Array of objects **>= 0 items**
required 金曜日の運行時間帯のリストです。

```
Array (>= 0 items) [
  | start_time_offset_sec integer
  | required              運行開始時刻の当日
  |                       00:00からのオフセット
  |                       (秒単位) です。
  |-----|
  | end_time_offset_sec   integer
  | required              運行終了時刻の当日
  |                       00:00からのオフセット
  |                       (秒単位) です。
  ]
```

└─ saturday ▾ Array of objects **>= 0 items**
required 土曜日の運行時間帯のリストです。

```
Array (>= 0 items) [  
  └─ start_time_offset_sec integer  
     required 運行開始時刻の当日  
     00:00からのオフセット  
     (秒単位) です。  
  └─ end_time_offset_sec integer  
     required 運行終了時刻の当日  
     00:00からのオフセット  
     (秒単位) です。  
]
```

└─ sunday ▾ Array of objects **>= 0 items**
required 日曜日の運行時間帯のリストです。

```
Array (>= 0 items) [  
  └─ start_time_offset_sec integer  
     required 運行開始時刻の当日  
     00:00からのオフセット  
     (秒単位) です。  
  └─ end_time_offset_sec integer  
     required 運行終了時刻の当日  
     00:00からのオフセット  
     (秒単位) です。  
]
```

└─ holidays ▾ Array of objects **>= 0 items**
required 祝日の運行時間帯のリストです。

```
Array (>= 0 items) [  
  └─ start_time_offset_sec integer  
     required 運行開始時刻の当日  
     00:00からのオフセット  
     (秒単位) です。  
  └─ end_time_offset_sec integer  
     required 運行終了時刻の当日  
     00:00からのオフセット  
     (秒単位) です。  
]
```

special_operating_hours **required** Array of objects **>= 0 items**
特別な運行時間帯のリストです。

```
Array (>= 0 items) [  
  |  
  | date string <date>  
  | required 特別運行日の日付です。形式はRFC3339の  
  |           date形式文字列（例：2025-06-19）です。  
  |  
  | time_slots > Array of objects  
  | required 特別運行日の時間帯のリストです。  
  |  
  ]
```

announcements **required** Array of objects **>= 0 items**
サービスに関するお知らせのリストです。

```
Array (>= 0 items) [  
  |  
  | id string  
  | required お知らせの一意的識別子です。  
  |  
  | title string  
  | required お知らせのタイトルです。  
  |  
  | content string  
  | required お知らせの内容です。  
  |  
  | start_datetime string <date-time>  
  | required お知らせの開始日時です。形式は  
  |           RFC3339のdate-time形式文字列（例：  
  |           2025-06-19T07:00:00+09:00）です。  
  |  
  | end_datetime string <date-time>  
  | required お知らせの終了日時です。形式は  
  |           RFC3339のdate-time形式文字列（例：  
  |           2025-06-30T20:59:59+09:00）です。こ  
  |           の項目が指定されていない場合、お知ら  
  |           せは「無期限（終了日未定）」として扱  
  |           われます。  
  |  
  ]
```

<ul style="list-style-type: none"> └─ created_at └─ updated_at 	<p>string <date-time> 作成日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。</p> <hr/> <p>string <date-time> 最終更新日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。</p> <hr/>
<ul style="list-style-type: none"> └─ total required 	<p>integer 取得可能な総件数を示します。</p> <hr/>
<ul style="list-style-type: none"> └─ offset required 	<p>integer 取得する結果の開始位置を示します。デフォルトは0です。</p> <hr/>
<ul style="list-style-type: none"> └─ limit required 	<p>integer 取得する結果の最大件数を示します。デフォルトは20件です。</p> <hr/>

◇ 400 Bad Request

RESPONSE SCHEMA: application/json

<ul style="list-style-type: none"> └─ type required 	<p>string エラーの種類を識別するURIです。</p> <hr/>
<ul style="list-style-type: none"> └─ title required 	<p>string 人間が読めるエラーの簡素な概要です。</p> <hr/>
<ul style="list-style-type: none"> └─ status 	<p>integer HTTPステータスコードです。</p> <hr/>
<ul style="list-style-type: none"> └─ detail 	<p>string 人間が読めるエラーの詳細な説明です。</p> <hr/>
<ul style="list-style-type: none"> └─ instance 	<p>string このエラーが発生した特定のリクエストやリソースを識別するURIです。</p> <hr/>

◇ 401 Unauthorized

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 403 Forbidden

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 500 Internal Server Error

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

➤ レスポンスサンプル

◇ 200 OK

```
{
  "services": [
    {
      "id": "sv012345678",
      "name": "サンプルオンデマンドサービス",
      "description": "○×地区内で利用可能なオンデマンド交通サービスです。",
      "brand_image_url": "https://example.com/logo_image.png",
      "access_scope": "public",
      "operation_type": "on_demand",
      "allow_ridepooling": true,
      "start_datetime": "2025-06-19T00:00:00+09:00",
      "service_group_id": "contract123",
      "cities": [
        {
          "prefecture_code": "13",
          "prefecture_name": "東京都",
          "municipalities": [
            {
              "code": "13103",
              "name": "港区"
            },
            {
              "code": "13109",
              "name": "品川区"
            }
          ]
        },
        {
          "prefecture_code": "14",
          "prefecture_name": "神奈川県",
          "municipalities": [
            {
              "code": "14100",
              "name": "横浜市"
            }
          ]
        }
      ]
    },
    {
      "prefecture_code": "14",
      "prefecture_name": "神奈川県",
      "municipalities": [
        {
          "code": "14100",
          "name": "横浜市"
        }
      ]
    }
  ],
  "service_terms": [
```

```
{
  "id": "st012345678",
  "category": "service",
  "name": "サンプルオンデマンドサービスの利用規約",
  "url": "https://example.com/terms",
  "agreement_required": true,
  "created_at": "2025-06-19T12:00:00+09:00",
  "updated_at": "2025-06-19T12:00:00+09:00"
}
],
"available_passenger_types": [
  {
    "passenger_type_id": "pt012345678",
    "name": "一般",
    "order": 1,
    "description": "一般の大人の方"
  },
  {
    "passenger_type_id": "pt012345679",
    "name": "シニア",
    "order": 2,
    "description": "60 歳以上の方"
  }
],
"supported_accessibility_features": [
  "wheelchair",
  "stroller"
],
"operating_hours": {
  "monday": [
    {
      "start_time_offset_sec": 28800,
      "end_time_offset_sec": 64800
    }
  ],
  "tuesday": [
    {
      "start_time_offset_sec": 28800,
      "end_time_offset_sec": 64800
    }
  ],
}
```

```
"wednesday": [  
  {  
    "start_time_offset_sec": 28800,  
    "end_time_offset_sec": 64800  
  }  
],  
"thursday": [  
  {  
    "start_time_offset_sec": 28800,  
    "end_time_offset_sec": 64800  
  }  
],  
"friday": [  
  {  
    "start_time_offset_sec": 28800,  
    "end_time_offset_sec": 64800  
  }  
],  
"saturday": [  
  {  
    "start_time_offset_sec": 28800,  
    "end_time_offset_sec": 64800  
  }  
],  
"sunday": [],  
"holidays": [  
  {  
    "start_time_offset_sec": 28800,  
    "end_time_offset_sec": 43200  
  },  
  {  
    "start_time_offset_sec": 46800,  
    "end_time_offset_sec": 64800  
  }  
]  
},  
"special_operating_hours": [  
  {  
    "date": "2025-04-28",  
    "time_slots": [  
      {
```

```
        "start_time_offset_sec": 46800,
        "end_time_offset_sec": 64800
    }
]
},
{
    "date": "2025-04-29",
    "time_slots": [
        {
            "start_time_offset_sec": 28800,
            "end_time_offset_sec": 43200
        }
    ]
}
],
"announcements": [
    {
        "id": "an012345678",
        "title": "サービス開始のお知らせ",
        "content": "サンプルオンデマンドサービスが 2025 年 7 月 1 日(火)から開始されます。",
        "start_datetime": "2025-06-19T00:00:00+09:00",
        "end_datetime": "2025-12-31T23:59:59+09:00"
    }
],
"created_at": "2025-06-19T00:00:00+09:00",
"updated_at": "2025-06-19T12:00:00+09:00"
}
],
"total": 1,
"offset": 0,
"limit": 20
}
```

◇ 400 Bad Request

```
{
  "type": "string",
  "title": "string",
  "status": 400,
  "detail": "string",
  "instance": "string"
}
```

◇ 401 Unauthorized

```
{  
  "type": "string",  
  "title": "string",  
  "status": 401,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 403 Forbidden

```
{  
  "type": "string",  
  "title": "string",  
  "status": 403,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 500 Internal Server Error

```
{  
  "type": "string",  
  "title": "string",  
  "status": 500,  
  "detail": "string",  
  "instance": "string"  
}
```

【IF002】 サービス詳細情報の取得標準 API

- 本インターフェースの概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、指定されたサービス ID に対応する、デマンドバスサービスの詳細情報を取得・返却するための API
- 本インターフェースを利用する機能
 - **【FN203】** 交通手段選択
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ GET
 - レスポンスデータ形式
 - ◇ JSON
 - URI
 - ◇ <https://example.com/services/{id}>
 - 使用例
 - ◇ <https://example.com/services/{id}>

➤ リクエストパラメータ

PATH PARAMETERS

→ id string
required サービスIDを指定します。

➤ レスポンスパラメータ

◇ 200 OK

RESPONSE SCHEMA: application/json

→ id required	string サービスの一意な識別子です。
→ name required	string サービスの名称です。
→ description	string サービスの説明です。
→ brand_image_url	string サービスのブランドイメージ画像のURLです。
→ operation_type required	string Enum: "on_demand" "slot_area_batch_confirmed" "fixed_time_fixed_route" サービスの運行方式の種類を示します。 <ul style="list-style-type: none">• on_demand 利用者の予約に応じて即時配車されるオンデマンド方式。• slot_area_batch_confirmed 時間帯ごとに運行エリアを切り替える方式で、各時間帯の締切時刻を設け、締切後に暫定予約をまとめて確定し、効率的なルートを生成して運行します。• fixed_time_fixed_route 決まったルートと時刻で運行される固定路線方式。

デマンドバスシステム連携 API システム設計書

<code>allow_ridepooling</code> required	boolean 相乗りが発生し得るか否かを示します。
<code>start_datetime</code> required	string <date-time> サービスの開始日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。
<code>end_datetime</code>	string <date-time> サービスの終了日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T20:59:59+09:00）です。この項目が指定されていない場合、サービスは「無期限（終了日未定）」として扱われます。
<code>service_group_id</code>	string サービスの上位グループの一意的な識別子です。 指定がない場合は、サービスが単独で存在することを示します。
<code>cities</code> v required	Array of objects >= 0 items サービスが提供される都道府県・市区町村のリストです。

Array (>= 0 items) [

<code>prefecture_code</code> required	string 都道府県コードです。
<code>prefecture_name</code> required	string 都道府県名です。
<code>municipalities</code> v required	Array of objects >= 0 items 市区町村のリストです。

Array (>= 0 items) [

<code>code</code> required	string 市区町村コードです。
<code>name</code> required	string 市区町村名です。

]

]

service_terms v
required

Array of objects [>= 0 items](#)
サービス利用に関する規約のリストです。

Array (>= 0 items) [

id required	string 規約の一意な識別子です。
category required	string Enum: "service" "privacy" "cancellation" "user" "third_party" 規約の分類カテゴリです。 <ul style="list-style-type: none">service サービス利用に関する規約。privacy プライバシーポリシーに関する規約。cancellation キャンセルポリシーに関する規約。user ユーザー向けの規約。third_party サードパーティとの関係に関する規約。
name	string 規約の名称です。
url required	string 規約のURLです。
agreement_required required	boolean 規約に対してユーザーから同意を得ることが必須かどうかを示します。
created_at	string <date-time> 作成日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。
updated_at	string <date-time> 最終更新日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

]

available_passenger_types **required**

Array of objects **non-empty**
サービスがサポートする乗客のタイプの情報です。
乗客のタイプは、例えば一般/未就学児/シニアなど、運賃が異なる乗客を示します。
order は昇順で表示して、未設定の場合は後方に配置します。

Array (non-empty) [

passenger_type_id required	string	乗客タイプの一意な識別子です。
name required	string	乗客タイプの名称です。
order	integer	表示順を示す番号です。
description	string	乗客タイプの説明です。

]

<p>supported_accessibility_features required</p>	<p>Array of strings >= 0 items</p> <p>Items Enum: "wheelchair" "folding_wheelchair" "powered_wheelchair" "oversized_wheelchair" "scooter" "oversized_scooter" "walker" "oversized_walker" "only_car_walker" "only_car_rider" "extra_space_ambulant" "actual_door_to_door" "lift" "oversized_item" "stroller" "child_seat_baby" "child_seat_young" "bike" "white_cane" "service_animal" "oxygen_tank" "stretcher"</p> <p>サービスがサポートする乗客補助機能のリストです。</p> <ul style="list-style-type: none"> • wheelchair 車椅子 • folding_wheelchair 折りたたみ式車椅子 • powered_wheelchair 電動車椅子 • oversized_wheelchair 大型車椅子 • scooter スクーター • oversized_scooter 大型スクーター • walker 歩行器
--	--

- `oversized_walker`
大型歩行器
- `only_car_walker`
車両内のみ歩行可能な乗客向け
- `only_car_rider`
車両内のみ乗車可能な乗客向け
- `extra_space_ambulant`
追加スペースを必要とする歩行者向け
- `actual_door_to_door`
実際のドアからドアまでのサービス
- `lift`
リフト付き車両
- `oversized_item`
大型荷物
- `stroller`
ベビーカー
- `child_seat_baby`
ベビーシート
- `child_seat_young`
幼児用シート
- `bike`
自転車
- `white_cane`
白杖使用者向けサービス
- `service_animal`
介助動物同伴可能
- `oxygen_tank`
酸素ボンベ使用者向けサービス
- `stretcher`
ストレッチャー

operating_hours
 required

object
 サービスの曜日および祝日ごとの
 運行時間帯を示します。
 各曜日のプロパティは、当日中
 における運行時間帯のリストです。
 該当の曜日が空配列 ([]) の場合
 は、当日は運休日とみなされま
 す。
 営業時間の制限がない場合は、
 00:00:00~23:59:59 を返します。

monday
 required Array of objects >= 0 items
 月曜日の運行時間帯のリストです。

```

Array (>= 0 items) [
  | start_time_offset_sec integer
  | required              運行開始時刻の当日00:00からの
  |                      オフセット (秒単位) です。
  |
  | end_time_offset_sec   integer
  | required              運行終了時刻の当日00:00からの
  |                      オフセット (秒単位) です。
  |
]
    
```

tuesday ▼ Array of objects >= 0 items
 required 火曜日の運行時間帯のリストです。

```
Array (>= 0 items) [
  | start_time_offset_sec integer
  | required                運行開始時刻の当日00:00からの
  |                        オフセット（秒単位）です。
  |
  | end_time_offset_sec   integer
  | required                運行終了時刻の当日00:00からの
  |                        オフセット（秒単位）です。
  ]
```

wednesday ▼ Array of objects >= 0 items
 required 水曜日の運行時間帯のリストです。

```
Array (>= 0 items) [
  | start_time_offset_sec integer
  | required                運行開始時刻の当日00:00からの
  |                        オフセット（秒単位）です。
  |
  | end_time_offset_sec   integer
  | required                運行終了時刻の当日00:00からの
  |                        オフセット（秒単位）です。
  ]
```

thursday ▼ Array of objects >= 0 items
 required 木曜日の運行時間帯のリストです。

```
Array (>= 0 items) [
  | start_time_offset_sec integer
  | required                運行開始時刻の当日00:00からの
  |                        オフセット（秒単位）です。
  |
  | end_time_offset_sec   integer
  | required                運行終了時刻の当日00:00からの
  |                        オフセット（秒単位）です。
  ]
```

friday **required** Array of objects **>= 0 items**
金曜日の運行時間帯のリストです。

```
Array (>= 0 items) [  
  | start_time_offset_sec integer  
  | required              運行開始時刻の当日00:00からの  
  |                          オフセット（秒単位）です。  
  |-----  
  | end_time_offset_sec   integer  
  | required              運行終了時刻の当日00:00からの  
  |                          オフセット（秒単位）です。  
  |-----  
]
```

saturday **required** Array of objects **>= 0 items**
土曜日の運行時間帯のリストです。

```
Array (>= 0 items) [  
  | start_time_offset_sec integer  
  | required              運行開始時刻の当日00:00からの  
  |                          オフセット（秒単位）です。  
  |-----  
  | end_time_offset_sec   integer  
  | required              運行終了時刻の当日00:00からの  
  |                          オフセット（秒単位）です。  
  |-----  
]
```

sunday **required** Array of objects **>= 0 items**
日曜日の運行時間帯のリストです。

```
Array (>= 0 items) [  
  | start_time_offset_sec integer  
  | required              運行開始時刻の当日00:00からの  
  |                          オフセット（秒単位）です。  
  |-----  
  | end_time_offset_sec   integer  
  | required              運行終了時刻の当日00:00からの  
  |                          オフセット（秒単位）です。  
  |-----  
]
```

└─ holidays ▾ Array of objects **>= 0 items**
required 祝日の運行時間帯のリストです。

```
Array (>= 0 items) [  
  └─ start_time_offset_sec integer  
     required 運行開始時刻の当日00:00からの  
              オフセット（秒単位）です。  
  └─ end_time_offset_sec integer  
     required 運行終了時刻の当日00:00からの  
              オフセット（秒単位）です。  
]
```

└─ special_operating_hours ▾ Array of objects **>= 0 items**
required 特別な運行時間帯のリストです。

```
Array (>= 0 items) [  
  └─ date string <date>  
     required 特別運行日の日付です。形式はRFC3339のdate形  
              式文字列（例：2025-06-19）です。  
  └─ time_slots ▾ Array of objects  
     required 特別運行日の時間帯のリストです。
```

```
Array [  
  └─ start_time_offset_sec integer  
     required 運行開始時刻の当日00:00から  
              のオフセット（秒単位）です。  
  └─ end_time_offset_sec integer  
     required 運行終了時刻の当日00:00から  
              のオフセット（秒単位）です。  
]
```

]

→ announcements v required	Array of objects >= 0 items サービスに関するお知らせのリストです。
Array (>= 0 items) [
→ id required	string お知らせの一意的識別子です。
→ title required	string お知らせのタイトルです。
→ content required	string お知らせの内容です。
→ start_datetime required	string <date-time> お知らせの開始日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。
→ end_datetime required	string <date-time> お知らせの終了日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-30T20:59:59+09:00）です。この項目が指定されていない場合、お知らせは「無期限（終了日未定）」として扱われます。
]	

operation_settings required	object サービスの予約やキャンセルのルール、検索時に指定できる条件などをまとめた設定情報です。サービスの詳細取得API (GET /v1/services/{id}) でのみ提供され、一覧取得では含まれません。
time_specifiable required	object 出発時刻および到着時刻を予約時に指定することを許可するかどうか。
departure required	boolean 出発時刻の指定を許可するかどうか。
arrival required	boolean 到着時刻の指定を許可するかどうか。
max_advance_reservable_days required	integer 予約候補検索時に指定できる最大予約可能日数です。例えば、7 ならば、現在日から7日後までの便を検索できます。

<p>min_advance_reservable_minutes required</p>	<p>integer</p> <p>出発希望時刻に設定できる最も早い時間を、現在時刻からのオフセット（分単位）で示します。例えば、30 ならば、現在時刻の30分後以降の時刻を出発希望時刻に設定できます。</p>
<p>cancel_deadline_minutes required</p>	<p>integer</p> <p>予約便のキャンセルが可能な最終時刻を、出発予定時刻からのオフセット（分単位）で示します。例えば、60 ならば、出発予定時刻の60分前までキャンセル可能です。</p>
<p>search_passenger_limit required</p>	<p>integer</p> <p>予約便候補検索時に指定できる最大乗客数です。</p>
<p>search_accessibility_limit required</p>	<p>integer</p> <p>予約便候補検索時に指定できる最大の乗客補助機能数です。</p>
<p>reservable_location_types required</p>	<p>Array of strings</p> <p>Items Enum: "fixed_stop" "home" "custom_location"</p> <p>予約可能な出発・到着地点の区分です。</p> <ul style="list-style-type: none"> fixed_stop 固定された地点（例：駅、バス停など）からの予約。 home 自宅からの予約。 custom_location ユーザーが指定する任意の地点からの予約。

reservable_areas **required** Array of objects **>= 0 items**
サービスを利用できる運行エリア
のリストです。

Array (>= 0 items) [

id required	string 運行エリアの識別子です。
<hr/>	
name	string 運行エリアの名称です。
<hr/>	
description	string 運行エリアの説明です。
<hr/>	
range required	object 本スキーマはGeoJSON形式のうち、 FeatureCollection → Feature → Polygon の構成のみ をサポートしています。その他のGeoJSON構造 (Point, LineString, MultiPolygon など) は非対応で す。

type required	string Value: "FeatureCollection"
<hr/>	
features required	Array of objects

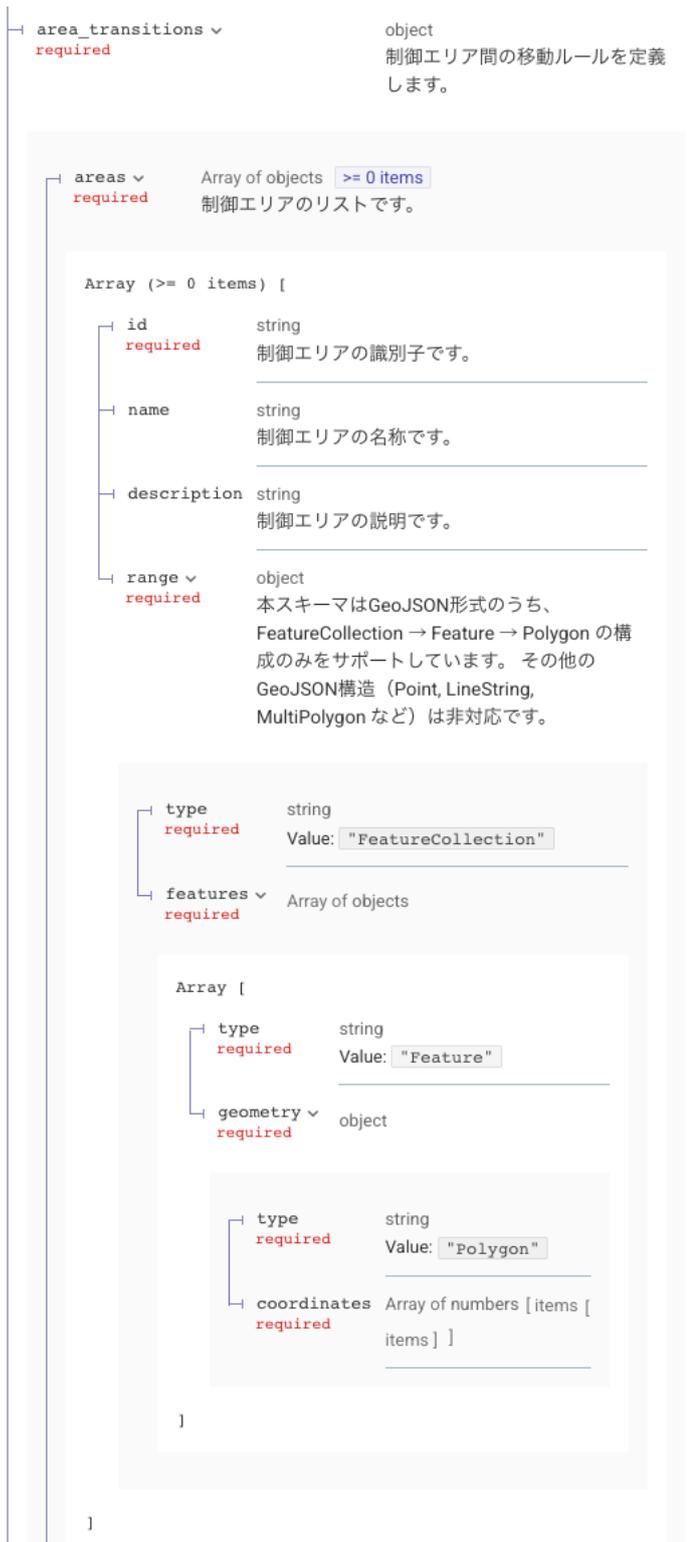
Array [

type required	string Value: "Feature"
<hr/>	
geometry required	object

type required	string Value: "Polygon"
<hr/>	
coordinates required	Array of numbers [items [items]]

]

]



rules required

Array of objects >= 0 items

制御エリア間の移動可否ルールです。

areas に1つ以上のエリアが定義されている場合は、すべての定義済みエリア間と、未定義エリアと定義済みエリア間、および未定義エリア同士の移動に関するルールが **rules** に漏れなく設定されます。いずれのエリアにも該当しない未定義エリアを **undefined_area** として扱います。

areas で定義したエリアに対し、**rules** で設定されるルールの例を以下に示します。エリアAおよびエリアBが定義されている場合に、定義済みエリア間の移動や未定義エリアからの移動について、どのようなルールを設定できるかを例示します。

移動元	移動先	移動可否
エリアA	エリアA	許可
エリアA	エリアB	許可
エリアA	未定義エリア	不許可
エリアB	エリアA	不許可
エリアB	エリアB	許可
エリアB	未定義エリア	不許可
未定義エリア	エリアA	不許可
未定義エリア	エリアB	許可
未定義エリア	未定義エリア	不許可

上記のルールが **rules** に設定される場合の例を Response samples の TestCase001 に示します。

```

Array (>= 0 items) [
  {
    from_area_id string
    required 出発エリアのIDです。
    to_area_id string
    required 到着エリアのIDです。
    is_allowed boolean
    required 出発エリアから到着エリアへの移動が許可されているかどうかを示します。
  }
]
    
```

created_at string <date-time>
作成日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

updated_at string <date-time>
最終更新日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

◇ 401 Unauthorized

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 403 Forbidden

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 404 Not Found

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 500 Internal Server Error

RESPONSE SCHEMA: application/json

type	string
required	エラーの種類を識別するURIです。
title	string
required	人間が読めるエラーの簡素な概要です。
status	integer
	HTTPステータスコードです。
detail	string
	人間が読めるエラーの詳細な説明です。
instance	string
	このエラーが発生した特定のリクエストやリソースを識別するURIです。

➤ レスポンスサンプル

◇ 200 OK

```
{
  "id": "sv012345678",
  "name": "サンプルオンデマンドサービス",
  "description": "○× 地区内で利用可能なオンデマンド交通サービスです。",
  "brand_image_url": "https://example.com/logo_image.png",
  "access_scope": "public",
  "operation_type": "on_demand",
  "allow_ridepooling": true,
  "start_datetime": "2025-06-19T00:00:00+09:00",
  "service_group_id": "contract123",
  "cities": [
    {
      "prefecture_code": "13",
      "prefecture_name": "東京都",
      "municipalities": [
        {
          "code": "13103",
          "name": "港区"
        },
        {
          "code": "13109",
          "name": "品川区"
        }
      ]
    }
  ],
  "prefecture_code": "14",
```

```
"prefecture_name": "神奈川県",
"municipalities": [
  {
    "code": "14100",
    "name": "横浜市"
  }
]
},
"service_terms": [
  {
    "id": "st012345678",
    "category": "service",
    "name": "サービス A B C の利用規約",
    "url": "https://example.com/terms",
    "agreement_required": true,
    "created_at": "2025-06-19T12:00:00+09:00",
    "updated_at": "2025-06-19T12:00:00+09:00"
  }
],
"available_passenger_types": [
  {
    "passenger_type_id": "pt012345678",
    "name": "一般",
    "order": 1,
    "description": "一般の大人の方"
  },
  {
    "passenger_type_id": "pt012345679",
    "name": "シニア",
    "order": 2,
    "description": "60 歳以上の方"
  }
],
"supported_accessibility_features": [
  "wheelchair",
  "stroller"
],
"operating_hours": {
  "monday": [
    {
```

```
"start_time_offset_sec": 28800,
"end_time_offset_sec": 64800
}
],
"tuesday": [
{
"start_time_offset_sec": 28800,
"end_time_offset_sec": 64800
}
],
"wednesday": [
{
"start_time_offset_sec": 28800,
"end_time_offset_sec": 64800
}
],
"thursday": [
{
"start_time_offset_sec": 28800,
"end_time_offset_sec": 64800
}
],
"friday": [
{
"start_time_offset_sec": 28800,
"end_time_offset_sec": 64800
}
],
"saturday": [
{
"start_time_offset_sec": 28800,
"end_time_offset_sec": 64800
}
],
"sunday": [],
"holidays": [
{
"start_time_offset_sec": 28800,
"end_time_offset_sec": 43200
},
{
```

```
    "start_time_offset_sec": 46800,
    "end_time_offset_sec": 64800
  }
]
},
"special_operating_hours": [
  {
    "date": "2025-04-28",
    "time_slots": [
      {
        "start_time_offset_sec": 46800,
        "end_time_offset_sec": 64800
      }
    ]
  },
  {
    "date": "2025-04-29",
    "time_slots": [
      {
        "start_time_offset_sec": 28800,
        "end_time_offset_sec": 43200
      }
    ]
  }
],
"announcements": [
  {
    "id": "an012345678",
    "title": "サービス開始のお知らせ",
    "content": "サンプルオンデマンドサービスが 2025 年 7 月 1 日(火)から開始されます。",
    "start_datetime": "2025-06-19T00:00:00+09:00",
    "end_datetime": "2025-12-31T23:59:59+09:00"
  }
],
"operation_settings": {
  "time_specifiable": {
    "departure": true,
    "arrival": true
  },
  "max_advance_reservable_days": 10,
  "min_advance_reservable_minutes": 30,
```

```
"cancel_deadline_minutes": 60,
"search_passenger_limit": 4,
"search_accessibility_limit": 1,
"reservable_location_types": [
  "fixed_stop",
  "home"
]
},
"reservable_areas": [
  {
    "id": "area012345678",
    "name": "○×地区",
    "description": "○×地区内の予約可能エリアです。",
    "range": {
      "type": "FeatureCollection",
      "features": [
        {
          "type": "Feature",
          "geometry": {
            "type": "Polygon",
            "coordinates": [
              [
                [
                  139.72351465041703,
                  35.65860559527819
                ],
                [
                  139.7182637631617,
                  35.63465766676801
                ],
                [
                  139.75257718716819,
                  35.633553804597994
                ],
                [
                  139.76375877681767,
                  35.65786994380646
                ],
                [
                  139.72351465041703,
                  35.65860559527819
                ]
              ]
            ]
          }
        }
      ]
    }
  }
]
```

```
    ]
  ]
]
}
}
],
"area_transitions": {
  "areas": [
    {
      "id": "at012345679",
      "name": "○×地区エリア A",
      "description": "○×地区内のエリア間移動を管理するエリア A です。",
      "range": {
        "type": "FeatureCollection",
        "features": [
          {
            "type": "Feature",
            "geometry": {
              "type": "Polygon",
              "coordinates": [
                [
                  [
                    139.72351465041703,
                    35.65860559527819
                  ],
                  [
                    139.72061703001884,
                    35.64539997517491
                  ],
                  [
                    139.7574661592795,
                    35.64429634273526
                  ],
                  [
                    139.76375877681767,
                    35.65786994380646
                  ],
                ]
              ]
            }
          }
        ]
      }
    }
  ]
}
```

```
        139.72351465041703,  
        35.65860559527819  
    ]  
  ]  
] }  
}  
]  
}  
,  
{  
  "id": "at012345680",  
  "name": "○×地区エリア B",  
  "description": "○×地区内のエリア間移動を管理するエリア B です。",  
  "range": {  
    "type": "FeatureCollection",  
    "features": [  
      {  
        "type": "Feature",  
        "geometry": {  
          "type": "Polygon",  
          "coordinates": [  
            [  
              [  
                139.72061703001884,  
                35.64539997517491  
              ],  
              [  
                139.7182637631617,  
                35.63465766676801  
              ],  
              [  
                139.75257718716819,  
                35.633553804597994  
              ],  
              [  
                139.7574661592795,  
                35.64429634273526  
              ],  
              [  
                139.72061703001884,
```

```
35.64539997517491
    ]
  ]
]
}
}
]
}
}
],
"rules": [
  {
    "from_area_id": "at012345679",
    "to_area_id": "at012345679",
    "is_allowed": true
  },
  {
    "from_area_id": "at012345679",
    "to_area_id": "at012345680",
    "is_allowed": true
  },
  {
    "from_area_id": "at012345679",
    "to_area_id": "undefined_area",
    "is_allowed": false
  },
  {
    "from_area_id": "at012345680",
    "to_area_id": "at012345679",
    "is_allowed": false
  },
  {
    "from_area_id": "at012345680",
    "to_area_id": "at012345680",
    "is_allowed": true
  },
  {
    "from_area_id": "at012345680",
    "to_area_id": "undefined_area",
    "is_allowed": false
  },
],
```

```
{
  "from_area_id": "undefined_area",
  "to_area_id": "at012345679",
  "is_allowed": false
},
{
  "from_area_id": "undefined_area",
  "to_area_id": "at012345680",
  "is_allowed": true
},
{
  "from_area_id": "undefined_area",
  "to_area_id": "undefined_area",
  "is_allowed": false
}
]
},
"created_at": "2025-06-19T00:00:00+09:00",
"updated_at": "2025-06-19T12:00:00+09:00"
}
```

◇ 401 Unauthorized

```
{
  "type": "string",
  "title": "string",
  "status": 401,
  "detail": "string",
  "instance": "string"
}
```

◇ 403 Forbidden

```
{
  "type": "string",
  "title": "string",
  "status": 403,
  "detail": "string",
  "instance": "string"
}
```

◇ 404 Not Found

```
{
  "type": "string",
  "title": "string",
  "status": 404,
```

```
"detail": "string",  
"instance": "string"  
}
```

◇ 500 Internal Server Error

```
{  
  "type": "string",  
  "title": "string",  
  "status": 500,  
  "detail": "string",  
  "instance": "string"  
}
```

【IF003】プラットフォーム規約一覧取得標準 API

- 本インターフェースの概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、プラットフォーム全体に適用される各種規約の最新一覧を取得・返却できる API
 - 内容が更新されるたびに新しい規約 ID が発行され、ユーザーはその URL から詳細を確認し、必要に応じて同意する
 - 各ユーザーの同意状況は、【IF009】ユーザーの規約同意一覧の取得標準 API から取得可能
 - この API から取得できる最新規約と照合し、再同意が必要か判断する
- 本インターフェースを利用する機能
 - 【FN202】ユーザー登録
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ GET
 - レスポンスデータ形式
 - ◇ JSON
 - URI
 - ◇ <https://example.com/terms>
 - 使用例
 - ◇ <https://example.com/terms?category=platform>
 - リクエストパラメータ

QUERY PARAMETERS

→ category

string

Enum: "platform" "privacy" "cancellation" "user" "provider" "third_party"

取得する利用規約のカテゴリを指定します。指定しない場合は全てのカテゴリの利用規約を取得します。

➤ レスポンスパラメータ

◇ 200 OK

→ terms ▾

Array of objects

プラットフォーム利用に関する規約のリストです。

Array [
id required	string 規約の一意な識別子です。
category required	string Enum: "platform" "privacy" "cancellation" "user" "provider" "third_party" 規約の分類カテゴリです。 <ul style="list-style-type: none">platform プラットフォーム全体に適用される規約。privacy プライバシーポリシーに関する規約。cancellation キャンセルポリシーに関する規約。user ユーザー向けの規約。provider サービスプロバイダー向けの規約。third_party サードパーティとの関係に関する規約。
name	string 規約の名称です。
url required	string 規約のURLです。
agreement_required required	boolean 規約に対してユーザーから同意を得ることが必須かどうかを示します。
created_at	string <date-time> 作成日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。
updated_at	string <date-time> 最終更新日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。
]	

◇ 400 Bad Request

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 401 Unauthorized

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 403 Forbidden

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 500 Internal Server Error

RESPONSE SCHEMA: application/json

type	string
required	エラーの種類を識別するURIです。
title	string
required	人間が読めるエラーの簡素な概要です。
status	integer
	HTTPステータスコードです。
detail	string
	人間が読めるエラーの詳細な説明です。
instance	string
	このエラーが発生した特定のリクエストやリソースを識別するURIです。

➤ レスポンスサンプル

◇ 200 OK

```
{
  "terms": [
    {
      "id": "pt012345678",
      "category": "platform",
      "name": "プラットフォーム利用規約",
      "url": "https://example.com/terms",
      "agreement_required": true,
      "created_at": "2025-06-01T10:00:00+09:00",
      "updated_at": "2025-06-01T10:00:00+09:00"
    },
    {
      "id": "pt012345679",
      "category": "privacy",
      "name": "プライバシーポリシー",
      "url": "https://example.com/privacy",
      "agreement_required": true,
      "created_at": "2025-06-02T09:00:00+09:00",
      "updated_at": "2025-06-02T09:00:00+09:00"
    },
    {
      "id": "pt012345680",
      "category": "cancellation",
      "name": "キャンセルポリシー",
      "url": "https://example.com/cancel",
      "agreement_required": true,
      "created_at": "2025-06-03T08:00:00+09:00",

```

```
"updated_at": "2025-06-19T12:00:00+09:00"  
}  
]  
}
```

◇ 400 Bad Request

```
{  
  "type": "string",  
  "title": "string",  
  "status": 400,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 401 Unauthorized

```
{  
  "type": "string",  
  "title": "string",  
  "status": 401,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 403 Forbidden

```
{  
  "type": "string",  
  "title": "string",  
  "status": 403,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 500 Internal Server Error

```
{  
  "type": "string",  
  "title": "string",  
  "status": 500,  
  "detail": "string",  
  "instance": "string"  
}
```

【IF004】 ユーザー情報の新規登録標準 API

- 本インターフェースの概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、デマンドバスサービスにユーザー情報を新規登録するための API
- 本インターフェースを利用する機能
 - **【FN202】 ユーザー登録**
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ POST
 - レスポンスデータ形式
 - ◇ JSON
 - URI
 - ◇ <https://example.com/passengers>
 - 使用例
 - ◇ <https://example.com/passengers>
 - リクエストパラメータ

デマンドバスシステム連携 API システム設計書

REQUEST BODY SCHEMA: application/json

first_name	string	ユーザーの氏名の名です。
last_name	string	ユーザーの氏名の姓です。
first_name_kana	string	ユーザーの氏名の名のフリガナです。
last_name_kana	string	ユーザーの氏名の姓のフリガナです。
gender	string	Enum: "male" "female" "other" "unspecified" ユーザーの性別です。 <ul style="list-style-type: none">male 男性female 女性other その他unspecified 指定なし
birthdate	string <date>	ユーザーの生年月日です。形式はRFC3339のfull-date形式文字列（例：2025-06-19）です。
phone_number	string	ユーザーの電話番号です。国際電話番号形式で、先頭に国コード（例：+81）を含めます。
email	string <email>	ユーザーのメールアドレスです。

デマンドバスシステム連携 API システム設計書

└─ home_address ▾ object
ユーザーの自宅住所を構成する位置情報オブジェクトです。
郵便番号・都道府県・市区町村・番地などの文字情報に加え、経度・緯度（WGS-84）による地理座標も含まれます。
自宅予約を行う場合は、`location` の指定が必要です。

└─ postal_code	string	郵便番号です。（ハイフンを含みません）
└─ state	string	都道府県名です。
└─ city	string	市区町村名です。
└─ street	string	丁目・番地などの住所の詳細部分です。
└─ location ▾	object	自宅位置を示す、経度・緯度（WGS-84）による地理座標です。 <code>location</code> を指定する場合は <code>type</code> 、 <code>coordinates</code> の両方の指定が必須です。

└─ type	string	Value: "Point" GeoJSONにおけるジオメトリタイプ（常に <code>Point</code> ）
└─ coordinates	Array of numbers <double> = 2 items [items <double >]	経度・緯度を含む配列です。 <code>[longitude, latitude]</code> の順で格納されます（GeoJSON仕様準拠）。

- レスポンスパラメータ
 - ◇ 201 Created

RESPONSE SCHEMA: application/json

id	string	ユーザーの一意的識別子です。
required		
first_name	string	ユーザーの氏名の名です。
last_name	string	ユーザーの氏名の姓です。
first_name_kana	string	ユーザーの氏名の名のフリガナです。
last_name_kana	string	ユーザーの氏名の姓のフリガナです。
gender	string	ユーザーの性別です。
	Enum: "male" "female" "other" "unspecified"	
	<ul style="list-style-type: none"> male 男性 female 女性 other その他 unspecified 指定なし 	
birthdate	string <date>	ユーザーの生年月日です。形式はRFC3339のfull-date形式文字列（例：2025-06-19）です。
phone_number	string	ユーザーの電話番号です。国際電話番号形式で、先頭に国コード（例：+81）を含めます。
email	string <email>	ユーザーのメールアドレスです。

デマンドバスシステム連携 API システム設計書

home_address	object	ユーザーの自宅住所を構成する位置情報オブジェクトです。 郵便番号・都道府県・市区町村・番地などの文字情報に加え、経度・緯度（WGS-84）による地理座標も含まれます。 自宅予約を行う場合は、 <code>location</code> の指定が必要です。
postal_code	string	郵便番号です。（ハイフンを含みません）
state	string	都道府県名です。
city	string	市区町村名です。
street	string	丁目・番地などの住所の詳細部分です。
location	object	自宅位置を示す、経度・緯度（WGS-84）による地理座標です。 <code>location</code> を指定する場合は <code>type</code> 、 <code>coordinates</code> の両方の指定が必須です。
type	string	Value: <code>"Point"</code> GeoJSONにおけるジオメトリタイプ（常に <code>Point</code> ）
coordinates	Array of numbers <double> = 2 items [items <double>]	経度・緯度を含む配列です。 <code>[longitude, latitude]</code> の順で格納されます（GeoJSON仕様準拠）。
created_at	string <date-time>	作成日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。
updated_at	string <date-time>	最終更新日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

◇ 400 Bad Request

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 401 Unauthorized

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 403 Forbidden

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 500 Internal Server Error

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

➤ レスポンスサンプル

◇ 201 Created

```
{
  "id": "ps012345678",
  "first_name": "太郎",
  "last_name": "山田",
  "first_name_kana": "タロウ",
  "last_name_kana": "ヤマダ",
  "gender": "male",
  "birthdate": "1990-01-01",
  "phone_number": "+819000000000",
  "email": "taro.yamada@example.com",
  "home_address": {
    "postal_code": "1234567",
    "state": "東京都",
    "city": "千代田区",
    "street": "千代田 1-1",
    "location": {
      "type": "Point",
      "coordinates": [
        139.766084,
        35.681382
      ]
    }
  }
},
  "created_at": "2025-06-19T12:00:00+09:00",
  "updated_at": "2025-06-19T12:00:00+09:00"
}
```

◇ 400 Bad Request

```
{  
  "type": "string",  
  "title": "string",  
  "status": 400,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 401 Unauthorized

```
{  
  "type": "string",  
  "title": "string",  
  "status": 401,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 403 Forbidden

```
{  
  "type": "string",  
  "title": "string",  
  "status": 403,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 500 Internal Server Error

```
{  
  "type": "string",  
  "title": "string",  
  "status": 500,  
  "detail": "string",  
  "instance": "string"  
}
```

【IF005】 ユーザー情報の取得標準 API

- 本インターフェースの概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、ユーザー情報を取得・返却するための API
- 本インターフェースを利用する機能
 - **【FN213】 ユーザー照会**
- 本 API の共通仕様
 - プロトコル

デマンドバスシステム連携 API システム設計書

- ◇ HTTPS
- メソッド
 - ◇ GET
- レスポンスデータ形式
 - ◇ JSON
- URI
 - ◇ `https://example.com/passengers/{id}`
- 使用例
 - ◇ `https://example.com/passengers/{id}`
- リクエストパラメータ

PATH PARAMETERS

<code>id</code>	<code>string</code>
<code>required</code>	ユーザーIDを指定します。

- レスポンスパラメータ
 - ◇ 200 OK

RESPONSE SCHEMA: application/json

id	string
required	ユーザーの一意的識別子です。
first_name	string ユーザーの氏名の名です。
last_name	string ユーザーの氏名の姓です。
first_name_kana	string ユーザーの氏名の名のフリガナです。
last_name_kana	string ユーザーの氏名の姓のフリガナです。
gender	string Enum: "male" "female" "other" "unspecified" ユーザーの性別です。 <ul style="list-style-type: none"> male 男性 female 女性 other その他 unspecified 指定なし
birthdate	string <date> ユーザーの生年月日です。形式はRFC3339のfull-date形式文字列（例：2025-06-19）です。
phone_number	string ユーザーの電話番号です。国際電話番号形式で、先頭に国コード（例：+81）を含めます。
email	string <email> ユーザーのメールアドレスです。

デマンドバスシステム連携 API システム設計書

home_address	object	ユーザーの自宅住所を構成する位置情報オブジェクトです。 郵便番号・都道府県・市区町村・番地などの文字情報に加え、経度・緯度（WGS-84）による地理座標も含まれます。 自宅予約を行う場合は、 <code>location</code> の指定が必要です。
postal_code	string	郵便番号です。（ハイフンを含みません）
state	string	都道府県名です。
city	string	市区町村名です。
street	string	丁目・番地などの住所の詳細部分です。
location	object	自宅位置を示す、経度・緯度（WGS-84）による地理座標です。 <code>location</code> を指定する場合は <code>type</code> 、 <code>coordinates</code> の両方の指定が必須です。
type	string	Value: "Point" GeoJSONにおけるジオメトリタイプ（常に <code>Point</code> ）
coordinates	Array of numbers <double> = 2 items [items <double >]	経度・緯度を含む配列です。 <code>[longitude, latitude]</code> の順で格納されます（GeoJSON仕様準拠）。
created_at	string <date-time>	作成日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。
updated_at	string <date-time>	最終更新日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

◇ 401 Unauthorized

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 403 Forbidden

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 404 Not Found

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 500 Internal Server Error

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

➤ レスポンスサンプル

◇ 200 OK

```
{
  "id": "ps012345678",
  "first_name": "太郎",
  "last_name": "山田",
  "first_name_kana": "タロウ",
  "last_name_kana": "ヤマダ",
  "gender": "male",
  "birthdate": "1990-01-01",
  "phone_number": "+819000000000",
  "email": "taro.yamada@example.com",
  "home_address": {
    "postal_code": "1234567",
    "state": "東京都",
    "city": "千代田区",
    "street": "千代田 1-1",
    "location": {
      "type": "Point",
      "coordinates": [
        139.766084,
        35.681382
      ]
    }
  }
},
  "created_at": "2025-06-19T12:00:00+09:00",
  "updated_at": "2025-06-19T12:00:00+09:00"
}
```

◇ 401 Unauthorized

```
{  
  "type": "string",  
  "title": "string",  
  "status": 401,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 403 Forbidden

```
{  
  "type": "string",  
  "title": "string",  
  "status": 403,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 404 Not Found

```
{  
  "type": "string",  
  "title": "string",  
  "status": 404,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 500 Internal Server Error

```
{  
  "type": "string",  
  "title": "string",  
  "status": 500,  
  "detail": "string",  
  "instance": "string"  
}
```

【IF006】 ユーザー情報の更新標準 API

- 本インターフェースの概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、指定されたユーザーID に対して、ユーザー情報を更新するための API
- 本インターフェースを利用する機能
 - **【FN214】 ユーザー情報更新**
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ PUT
 - レスポンスデータ形式
 - ◇ JSON
 - URI
 - ◇ `https://example.com/passengers/{id}`
 - 使用例
 - ◇ `https://example.com/passengers/{id}`
 - リクエストパラメータ

PATH PARAMETERS

→ id string
required ユーザーIDを指定します。

REQUEST BODY SCHEMA: application/json

→ first_name string
ユーザーの氏名の名です。

→ last_name string
ユーザーの氏名の姓です。

→ first_name_kana string
ユーザーの氏名の名のフリガナです。

→ last_name_kana string
ユーザーの氏名の姓のフリガナです。

→ gender string
Enum: "male" "female" "other" "unspecified"
ユーザーの性別です。

- male
男性
- female
女性
- other
その他
- unspecified
指定なし

デマンドバスシステム連携 API システム設計書

birthdate	string <date> ユーザーの生年月日です。形式はRFC3339のfull-date形式文字列（例：2025-06-19）です。
phone_number	string ユーザーの電話番号です。国際電話番号形式で、先頭に国コード（例：+81）を含めます。
email	string <email> ユーザーのメールアドレスです。
home_address	object ユーザーの自宅住所を構成する位置情報オブジェクトです。 郵便番号・都道府県・市区町村・番地などの文字情報に加え、経度・緯度（WGS-84）による地理座標も含まれます。 自宅予約を行う場合は、 <code>location</code> の指定が必要です。

postal_code	string 郵便番号です。（ハイフンを含みません）
state	string 都道府県名です。
city	string 市区町村名です。
street	string 丁目・番地などの住所の詳細部分です。
location	object 自宅位置を示す、経度・緯度（WGS-84）による地理座標です。 <code>location</code> を指定する場合は <code>type</code> 、 <code>coordinates</code> の両方の指定が必須です。

type	string Value: "Point" GeoJSONにおけるジオメトリタイプ（常に <code>Point</code> ）
coordinates	Array of numbers <double> [items <double >] 経度・緯度を含む配列です。 <code>[longitude, latitude]</code> の順で格納されます（GeoJSON仕様準拠）。

- レスポンスパラメータ
- ✧ 200 OK

デマンドバスシステム連携 API システム設計書

RESPONSE SCHEMA: application/json

→ id	string
required	ユーザーの一意な識別子です。
→ first_name	string
	ユーザーの氏名の名です。
→ last_name	string
	ユーザーの氏名の姓です。
→ first_name_kana	string
	ユーザーの氏名の名のフリガナです。
→ last_name_kana	string
	ユーザーの氏名の姓のフリガナです。
→ gender	string
	Enum: "male" "female" "other" "unspecified"
	ユーザーの性別です。
	<ul style="list-style-type: none">• male 男性• female 女性• other その他• unspecified 指定なし
→ birthdate	string <date>
	ユーザーの生年月日です。形式はRFC3339のfull-date形式文字列（例：2025-06-19）です。
→ phone_number	string
	ユーザーの電話番号です。国際電話番号形式で、先頭に国コード（例：+81）を含めます。

デマンドバスシステム連携 API システム設計書

email	string <email> ユーザーのメールアドレスです。
home_address	object ユーザーの自宅住所を構成する位置情報オブジェクトです。 郵便番号・都道府県・市区町村・番地などの文字情報に加え、経度・緯度（WGS-84）による地理座標も含まれます。 自宅予約を行う場合は、 <code>location</code> の指定が必要です。
postal_code	string 郵便番号です。（ハイフンを含みません）
state	string 都道府県名です。
city	string 市区町村名です。
street	string 丁目・番地などの住所の詳細部分です。
location	object 自宅位置を示す、経度・緯度（WGS-84）による地理座標です。 <code>location</code> を指定する場合は <code>type</code> 、 <code>coordinates</code> の両方の指定が必須です。
type	string Value: "Point" GeoJSONにおけるジオメトリタイプ（常に <code>Point</code> ）
coordinates	Array of numbers <double> = 2 items [items <double >] 経度・緯度を含む配列です。 <code>[longitude, latitude]</code> の順で格納されます（GeoJSON仕様準拠）。
created_at	string <date-time> 作成日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。
updated_at	string <date-time> 最終更新日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

◇ 400 Bad Request

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 401 Unauthorized

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 403 Forbidden

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 404 Not Found

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 500 Internal Server Error

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

➤ レスポンスサンプル

◇ 200 OK

```
{
  "id": "ps012345678",
  "first_name": "太郎",
  "last_name": "山田",
  "first_name_kana": "タロウ",
  "last_name_kana": "ヤマダ",
  "gender": "male",
  "birthdate": "1990-01-01",
  "phone_number": "+819000000000",
  "email": "taro.yamada@example.com",
  "home_address": {
    "postal_code": "1234567",
    "state": "東京都",
    "city": "千代田区",

```

```
"street": "千代田 1-1",  
"location": {  
  "type": "Point",  
  "coordinates": [  
    139.766084,  
    35.681382  
  ]  
}  
,  
"created_at": "2025-06-19T12:34:56+09:00",  
"updated_at": "2025-06-21T13:45:01+09:00"  
}
```

◇ 400 Bad Request

```
{  
  "type": "string",  
  "title": "string",  
  "status": 400,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 401 Unauthorized

```
{  
  "type": "string",  
  "title": "string",  
  "status": 401,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 403 Forbidden

```
{  
  "type": "string",  
  "title": "string",  
  "status": 403,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 404 Not Found

```
{  
  "type": "string",  
  "title": "string",  
  "status": 404,  
}
```

```
"detail": "string",  
"instance": "string"  
}
```

◇ 500 Internal Server Error

```
{  
  "type": "string",  
  "title": "string",  
  "status": 500,  
  "detail": "string",  
  "instance": "string"  
}
```

【IF007】ユーザー情報の削除（同意情報含む）標準 API

- 本インターフェースの概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、指定されたユーザーID に紐づくユーザー情報を削除するための API
- 本インターフェースを利用する機能
 - 【FN215】ユーザー退会
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ DELETE
 - レスポンスデータ形式
 - ◇ JSON
 - URI
 - ◇ `https://example.com/passengers/{id}`
 - 使用例
 - ◇ `https://example.com/passengers/{id}`
 - リクエストパラメータ

PATH PARAMETERS

→ id	string
required	ユーザーIDを指定します。

- レスポンスパラメータ
 - ◇ 204 No Content
(レスポンススキーマなし)
 - ◇ 401 Unauthorized

デマンドバスシステム連携 API システム設計書

RESPONSE SCHEMA: `application/json`

<code>type</code> <code>required</code>	<code>string</code> エラーの種類を識別するURIです。
<code>title</code> <code>required</code>	<code>string</code> 人間が読めるエラーの簡素な概要です。
<code>status</code>	<code>integer</code> HTTPステータスコードです。
<code>detail</code>	<code>string</code> 人間が読めるエラーの詳細な説明です。
<code>instance</code>	<code>string</code> このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 403 Forbidden

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 404 Not Found

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 500 Internal Server Error

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

➤ レスポンスサンプル

◇ 204 No Content

(レスポンススキーマがないため、サンプルもなし)

◇ 401 Unauthorized

```
{  
  "type": "string",  
  "title": "string",  
  "status": 401,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 403 Forbidden

```
{  
  "type": "string",  
  "title": "string",  
  "status": 403,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 404 Not Found

```
{  
  "type": "string",  
  "title": "string",  
  "status": 404,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 500 Internal Server Error

```
{  
  "type": "string",  
  "title": "string",  
  "status": 500,  
  "detail": "string",  
  "instance": "string"  
}
```

【IF008】 ユーザー規約同意登録標準 API

- 本インターフェースの概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、指定されたユーザーID に対して、利用規約 (terms_id) への新規同意結果を登録するための API
- 本インターフェースを利用する機能
 - 【FN202】 ユーザー登録
 - 【FN206】 予約登録
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ POST
 - レスポンスデータ形式
 - ◇ JSON
 - URI
 - ◇ https://example.com/passengers/{id}/agreements
 - 使用例
 - ◇ https://example.com/passengers/{id}/agreements
 - リクエストパラメータ

PATH PARAMETERS

→ id string
required ユーザーIDを指定します。

REQUEST BODY SCHEMA: application/json

→ agreements ✓ Array of objects non-empty
required

```
Array (non-empty) [  
  {  
    terms_id string  
      required 規約を識別する一意なIDです。  
    agreed_at string <date-time>  
      required ユーザーが規約に同意した日時です。形式はRFC3339の  
                date-time形式文字列（例：2025-06-  
                19T07:00:00+09:00）です。  
  }  
]
```

- レスポンスパラメータ
 - ◇ 200 OK

RESPONSE SCHEMA: application/json

→ **agreements** ▼ Array of objects non-empty
required

Array (non-empty) [

<div style="display: flex; align-items: flex-start;"> <div style="margin-right: 5px;">└─</div> <div style="margin-right: 5px;">terms_id</div> <div style="margin-left: 5px;">required</div> </div>	string	規約を識別する一意なIDです。
<hr/>		
<div style="display: flex; align-items: flex-start;"> <div style="margin-right: 5px;">└─</div> <div style="margin-right: 5px;">term_category</div> <div style="margin-left: 5px;">required</div> </div>	string	規約の種類です。 GET /terms および GET /services/{id} で取得できる規約の category を参照してください。
<hr/>		
<div style="display: flex; align-items: flex-start;"> <div style="margin-right: 5px;">└─</div> <div style="margin-right: 5px;">service_id</div> </div>	string	サービスIDを指定します。(サービスに関する規約の場合のみ設定されます)
<hr/>		
<div style="display: flex; align-items: flex-start;"> <div style="margin-right: 5px;">└─</div> <div style="margin-right: 5px;">agreed_at</div> <div style="margin-left: 5px;">required</div> </div>	string <date-time>	ユーザーが規約に同意した日時です。形式は RFC3339のdate-time形式文字列 (例: 2025-06-19T07:00:00+09:00) です。
<hr/>		
<div style="display: flex; align-items: flex-start;"> <div style="margin-right: 5px;">└─</div> <div style="margin-right: 5px;">status</div> <div style="margin-left: 5px;">required</div> </div>	string	Enum: "created" "already_agreed" 同意の状態を示します。 <ul style="list-style-type: none"> • created 新規に同意が登録されたことを示します。 • already_agreed すでに同意済みの規約が指定されたことを示します。

]

◇ 400 Bad Request

RESPONSE SCHEMA: `application/json`

<code>type</code> <code>required</code>	string エラーの種類を識別するURIです。
<code>title</code> <code>required</code>	string 人間が読めるエラーの簡素な概要です。
<code>status</code>	integer HTTPステータスコードです。
<code>detail</code>	string 人間が読めるエラーの詳細な説明です。
<code>instance</code>	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 401 Unauthorized

RESPONSE SCHEMA: `application/json`

<code>type</code> <code>required</code>	string エラーの種類を識別するURIです。
<code>title</code> <code>required</code>	string 人間が読めるエラーの簡素な概要です。
<code>status</code>	integer HTTPステータスコードです。
<code>detail</code>	string 人間が読めるエラーの詳細な説明です。
<code>instance</code>	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 403 Forbidden

RESPONSE SCHEMA: `application/json`

<code>type</code> <code>required</code>	string エラーの種類を識別するURIです。
<code>title</code> <code>required</code>	string 人間が読めるエラーの簡素な概要です。
<code>status</code>	integer HTTPステータスコードです。
<code>detail</code>	string 人間が読めるエラーの詳細な説明です。
<code>instance</code>	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 404 Not Found

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 500 Internal Server Error

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

➤ レスポンスサンプル

◇ 200 OK

```
{
  "agreements": [
    {
      "terms_id": "pt012345678",
      "term_category": "platform",
      "agreed_at": "2025-06-11T10:43:12+09:00",
      "status": "already_agreed"
    },
    {
      "terms_id": "st012345678",
      "term_category": "service",
      "service_id": "sv012345678",
      "agreed_at": "2025-06-25T12:23:35+09:00",
      "status": "created"
    }
  ]
}
```

```
}  
]  
}
```

◇ 400 Bad Request

```
{  
  "type": "string",  
  "title": "string",  
  "status": 400,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 401 Unauthorized

```
{  
  "type": "string",  
  "title": "string",  
  "status": 401,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 403 Forbidden

```
{  
  "type": "string",  
  "title": "string",  
  "status": 403,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 404 Not Found

```
{  
  "type": "string",  
  "title": "string",  
  "status": 404,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 500 Internal Server Error

```
{  
  "type": "string",  
  "title": "string",  
  "status": 500,  
}
```

```
"detail": "string",  
"instance": "string"  
}
```

【IF009】 ユーザーの規約同意一覧の取得標準 API

- 本インターフェースの概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、指定されたユーザーID が示すユーザーが、過去に同意した利用規約（terms_id 単位）の一覧を取得するための API
- 本インターフェースを利用する機能
 - 【FN213】 ユーザー照会
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ GET
 - レスポンスデータ形式
 - ◇ JSON
 - URI
 - ◇ <https://example.com/passengers/{id}/agreements>
 - 使用例
 - ◇ https://example.com/passengers/{id}/agreements?service_id={service_id}
 - リクエストパラメータ

PATH PARAMETERS

→ id	string
required	ユーザーIDを指定します。

QUERY PARAMETERS

→ service_id	string
	サービスIDを指定します。

- レスポンスパラメータ
 - ◇ 200 OK

RESPONSE SCHEMA: application/json

→ **agreements** ▼ Array of objects >= 0 items
required

Array (>= 0 items) [

→ terms_id required	string 規約を識別する一意なIDです。
→ term_category required	string 規約の種類です。 <code>GET /terms</code> および <code>GET /services/{id}</code> で取得できる規約の <code>category</code> を参照してください。
→ service_id	string サービスIDを指定します。(サービスに関する規約の場合のみ設定されます)
→ agreed_at required	string <date-time> ユーザーが規約に同意した日時です。形式はRFC3339のdate-time形式文字列(例: 2025-06-19T07:00:00+09:00)です。

]

◇ 400 Bad Request

RESPONSE SCHEMA: application/json

→ type required	string エラーの種類を識別するURIです。
→ title required	string 人間が読めるエラーの簡素な概要です。
→ status	integer HTTPステータスコードです。
→ detail	string 人間が読めるエラーの詳細な説明です。
→ instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 401 Unauthorized

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 403 Forbidden

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 404 Not Found

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 500 Internal Server Error

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

➤ レスポンスサンプル

◇ 200 OK

```
{
  "agreements": [
    {
      "terms_id": "pt012345678",
      "term_category": "platform",
      "agreed_at": "2025-06-25T12:23:35+09:00"
    },
    {
      "terms_id": "st123456789",
      "term_category": "service",
      "service_id": "sv012345678",
      "agreed_at": "2025-06-26T11:22:33+09:00"
    },
    {
      "terms_id": "cp123456789",
      "term_category": "cancellation",
      "service_id": "sv012345678",
      "agreed_at": "2025-06-26T11:25:45+09:00"
    }
  ]
}
```

◇ 400 Bad Request

```
{  
  "type": "string",  
  "title": "string",  
  "status": 400,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 401 Unauthorized

```
{  
  "type": "string",  
  "title": "string",  
  "status": 401,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 403 Forbidden

```
{  
  "type": "string",  
  "title": "string",  
  "status": 403,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 404 Not Found

```
{  
  "type": "string",  
  "title": "string",  
  "status": 404,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 500 Internal Server Error

```
{  
  "type": "string",  
  "title": "string",  
  "status": 500,  
  "detail": "string",  
  "instance": "string"  
}
```

【IF010】乗降場所一覧の取得標準 API

- 本インターフェースの概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、デマンドバスの固定の乗降場所情報を取得・返却するための API
- 本インターフェースを利用する機能
 - **【FN204】乗降場所表示**
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ GET
 - レスポンスデータ形式
 - ◇ JSON
 - URI
 - ◇ <https://example.com/stops>
 - 使用例
 - ◇ https://example.com/stops?service_ids=1,2,3&status=upcoming,active,ended&offset=0&limit=20
 - リクエストパラメータ

QUERY PARAMETERS

stop_ids	<p>Array of strings</p> <p>Example: <code>stop_ids=1,2,3</code></p> <p>取得対象の乗降場所IDをカンマ区切りで指定します。</p>
service_ids	<p>Array of strings</p> <p>Example: <code>service_ids=1,2,3</code></p> <p>取得対象のサービスIDをカンマ区切りで指定します。 指定されたいずれかのサービスIDに該当する対象を取得します (OR条件)。</p>
name	<p>string</p> <p>Example: <code>name=新橋</code></p> <p>取得対象の乗降場所の名称 (部分一致) を指定します。</p>
status	<p>Array of strings</p> <p>Default: <code>["upcoming", "active"]</code></p> <p>Items Enum: <code>"upcoming" "active" "ended"</code></p> <p>Example: <code>status=upcoming,active</code></p> <p>フィルター対象とする乗降場所のステータスを複数指定します。</p> <ul style="list-style-type: none"> <code>upcoming</code> 利用開始日時を迎える前の乗降場所 <code>active</code> 現在利用できる乗降場所 <code>ended</code> 利用終了日時を経過した乗降場所
location	<p>string^{^[+]?[0-9]*\.[+]?[0-9]*\.[+]?[0-9]*\.[+]?[0-9]*\$}</p> <p>Example: <code>location=139.767125,35.681236</code></p> <p>検索の中心座標を <code>longitude,latitude</code> のカンマ区切り形式で指定します。 例: <code>139.767125,35.681236</code> (経度・緯度の順に注意) <code>radius</code> と併せて指定することで、中心からの距離に基づく範囲検索が可能になります。</p>
radius	<p>integer ^{>= 1}</p> <p>Default: <code>500</code></p> <p>Example: <code>radius=500</code></p> <p><code>location</code> パラメータと併せて使用することで、中心座標からの検索半径 (メートル単位) を指定できます。 <code>location</code> が指定されており、この値が省略された場合は、デフォルトで <code>500</code> メートルが適用されます。 この値だけを指定した場合は無効です。</p>
offset	<p>integer ^{>= 0}</p> <p>Default: <code>0</code></p> <p>Example: <code>offset=30</code></p> <p>取得する結果の開始位置を指定します。</p>
limit	<p>integer ^[1..100]</p> <p>Default: <code>20</code></p> <p>Example: <code>limit=10</code></p> <p>取得する結果の最大件数を指定します。</p>

- レスポンスパラメータ
 - ◇ 200 OK

RESPONSE SCHEMA: application/json

stops ▼ required Array of objects >= 0 items
乗降場所情報のリストです。

Array (>= 0 items) [

- id required string
固定の乗降場所の一意的なIDです。
- service_ids required Array of strings
この乗降場所を利用可能なサービスのサービスID一覧です。
- location ▼ required object
乗降場所を示す、経度・緯度（WGS-84）による地理座標です。
 - type required string
Value: "Point"
GeoJSONにおけるジオメトリタイプ（常に Point）
 - coordinates required Array of numbers <double> = 2 items [items <double >]
経度・緯度を含む配列です。
[longitude, latitude] の順で格納されます（GeoJSON仕様準拠）。
- name required string
乗降場所の名称です。
- description string
乗降場所の説明文です。
- search_words Array of strings
検索用の補助語句のリストです。
このフィールドは **停留所名称の検索候補を拡張するために、クライアント側で使用することを想定しています。**
ふりがな・略称・英語表記・旧名称などが含まれることがあります。
例えば、インクリメンタル検索や予測変換時の候補マッチに活用できます。

start_datetime required	string <date-time> 乗降場所の利用開始日時です。形式は RFC3339 の date-time 形式文字列（例：2025-06-19T07:00:00+09:00）です。
end_datetime	string <date-time> 乗降場所の利用終了日時です。形式は RFC3339 の date-time 形式文字列（例：2025-06-19T20:59:59+09:00）です。 この項目が指定されていない場合、乗降場所は「無期限（終了日未定）」として扱われます。
is_boarding_available required	boolean 乗車可能かどうかを示します。
is_alighting_available required	boolean 降車可能かどうかを示します。
suspensions required	Array of objects >= 0 items 一時的な利用停止情報のリストです。

Array (>= 0 items) [

start_datetime required	string <date-time> 一時停止開始日時です。形式は RFC3339 の date-time 形式文字列（例：2025-06-19T07:00:00+09:00）です。
end_datetime required	string <date-time> 一時停止終了日時です。形式は RFC3339 の date-time 形式文字列（例：2025-06-19T20:59:59+09:00）です。
description	string 停止理由や補足説明です。

]

<ul style="list-style-type: none"> └ pictures required <ul style="list-style-type: none"> └ title string 画像のタイトルです。 └ url required string <uri> 画像のURLです。 └ description string 画像の説明文です。 	<p>Array of objects >= 0 items 乗降場所の画像のリストです。</p> <pre> Array (>= 0 items) [{ title: string url: string <uri> description: string }] </pre>
└ created_at	<p>string <date-time> 作成日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。</p>
└ updated_at	<p>string <date-time> 最終更新日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。</p>
└ total required	<p>integer 取得可能な総件数を示します。</p>
└ offset required	<p>integer 取得する結果の開始位置を示します。デフォルトは0です。</p>
└ limit required	<p>integer 取得する結果の最大件数を示します。デフォルトは20件です。</p>

◇ 400 Bad Request

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 401 Unauthorized

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 403 Forbidden

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 500 Internal Server Error

RESPONSE SCHEMA: application/json

type	string
required	エラーの種類を識別するURIです。
title	string
required	人間が読めるエラーの簡素な概要です。
status	integer
	HTTPステータスコードです。
detail	string
	人間が読めるエラーの詳細な説明です。
instance	string
	このエラーが発生した特定のリクエストやリソースを識別するURIです。

➤ レスポンスサンプル

◇ 200 OK

```
{
  "stops": [
    {
      "id": "st012345678",
      "service_ids": [
        "sv012345678"
      ],
      "location": {
        "type": "Point",
        "coordinates": [
          139.7649363,
          35.6864501
        ]
      },
      "name": "大手町パークビルディング",
      "description": "東京メトロ「大手町駅」C6a 出口直結、丸の内側のビル前に位置する乗降場所です。",
      "search_words": [
        "おおてまちぱーくびるでいんぐ",
        "大手町",
        "Otemachi"
      ],
      "start_datetime": "2024-04-01T00:00:00+09:00",
      "is_boarding_available": true,
      "is_alighting_available": true,
      "suspensions": [],
      "pictures": [
```

```
{
  "title": "ビル前乗降場所",
  "url": "https://example.com/images/otemachi.jpg",
  "description": "大手町パークビルディング前の乗降ポイントの写真"
},
{
  "created_at": "2025-06-19T00:00:00+09:00",
  "updated_at": "2025-06-19T12:00:00+09:00"
},
{
  "id": "st012345679",
  "service_ids": [
    "sv012345678"
  ],
  "location": {
    "type": "Point",
    "coordinates": [
      139.7600031,
      35.6639339
    ]
  },
  "name": "新橋駅前",
  "description": "JR 新橋駅の南改札を出てすぐ、SL 広場横の乗降ポイントです。",
  "search_words": [
    "しんばし",
    "SL 広場",
    "SL ひろば",
    "Shimbashi"
  ],
  "start_datetime": "2024-04-01T00:00:00+09:00",
  "is_boarding_available": true,
  "is_alighting_available": true,
  "suspensions": [
    {
      "start_datetime": "2025-06-19T00:00:00+09:00",
      "end_datetime": "2025-06-19T11:59:59+09:00",
      "description": "周辺イベント開催のため一時利用停止となります。"
    }
  ],
  "pictures": [
    {
```

```
"title": "SL 広場前乗降場所",
"url": "https://example.com/images/shimbashi.jpg",
"description": "新橋駅 SL 広場横の乗降ポイントの写真"
}
],
"created_at": "2025-06-19T00:00:00+09:00",
"updated_at": "2025-06-19T12:00:00+09:00"
}
],
"total": 2,
"offset": 0,
"limit": 20
}
```

◇ 400 Bad Request

```
{
  "type": "string",
  "title": "string",
  "status": 400,
  "detail": "string",
  "instance": "string"
}
```

◇ 401 Unauthorized

```
{
  "type": "string",
  "title": "string",
  "status": 401,
  "detail": "string",
  "instance": "string"
}
```

◇ 403 Forbidden

```
{
  "type": "string",
  "title": "string",
  "status": 403,
  "detail": "string",
  "instance": "string"
}
```

◇ 500 Internal Server Error

```
{
  "type": "string",
  "title": "string",
```

```
"status": 500,  
"detail": "string",  
"instance": "string"  
}
```

【IF011】 予約候補の生成標準 API

- 本インターフェースの概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、ユーザーが指定した条件（乗車場所・降車場所・希望時間・人数など）を基に、対象サービスの中から予約可能な便候補をリアルタイムに生成・返却するための API
- 本インターフェースを利用する機能
 - 【FN205】 予約候補検索
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ POST
 - レスポンスデータ形式
 - ◇ JSON
 - URI
 - ◇ <https://example.com/reservations/candidates>
 - 使用例
 - ◇ <https://example.com/reservations/candidates>
 - リクエストボディスキーマ

REQUEST BODY SCHEMA: application/json

<p>service_ids required</p>	<p>Array of strings サービスIDのリストです。 複数のサービスで利用できる乗車場所、 降車場所を用いる場合、それぞれのサー ビスに対して予約候補検索を行うこと が可能です。</p>
<p>preferred_time required</p>	<p>object 希望する乗車または降車の日時を指定し ます。</p>
<p>type required</p>	<p>string Enum: "pickup" "dropoff" 乗車時刻指定か降車時刻指定かを示します。</p> <ul style="list-style-type: none"> pickup 乗車時刻を指定する場合に選択します。 dropoff 降車時刻を指定する場合に選択します。
<p>datetime required</p>	<p>string <date-time> 希望する乗車または降車の日時です。形式はRFC3339の date-time形式文字列（例：2025-06- 19T07:00:00+09:00）です。</p>

pickup ▾
required

object or object or object
 固定の乗降場所・任意の地点・または自宅のいずれかを指定するための情報です。
`stop_id`、`location`、または
`passenger_id` のいずれか一つを指定します。

この構造は、予約リクエスト時の入力だけでなく、予約候補や予約情報のレスポンスにも使用されます。
 リクエストにおいては、対象サービスがサポートする以下の
`reservable_location_types` に応じて、指定できる項目が異なります：

- `fixed_stop` を含むサービスでは、`stop_id` を指定できます。
- `custom_location` を含むサービスでは、`location` を指定できます。
- `home` を含むサービスでは、`passenger_id` を指定することで自宅情報が使用されます。

One of object object object

type
required string
 Value: `"fixed_stop"`
 乗降場所の種類を示します。`fixed_stop` の場合、固定の乗降場所を示します。

stop_id
required string
 固定の乗降場所のIDです。

One of object object object

type
required string
 Value: `"custom_location"`
 乗降場所の種類を示します。`custom_location` の場合、任意の地点を示します。

location ▾
required object
 乗降場所を示す、経度・緯度（WGS-84）による地理座標です。

One of object object object

type string
required Value: "home"
乗降場所の種類を示します。home の場合、自宅を示します。

passenger_id string
required ユーザーの自宅位置を指定する場合のユーザーIDです。

dropoff ▼
 required

object or object or object
 固定の乗降場所・任意の地点・または自宅のいずれかを指定するための情報です。
`stop_id`、`location`、または
`passenger_id` のいずれか一つを指定します。

この構造は、予約リクエスト時の入力だけでなく、予約候補や予約情報のレスポンスにも使用されます。
 リクエストにおいては、対象サービスがサポートする以下の
`reservable_location_types` に応じて、指定できる項目が異なります：

- `fixed_stop` を含むサービスでは、`stop_id` を指定できます。
- `custom_location` を含むサービスでは、`location` を指定できます。
- `home` を含むサービスでは、`passenger_id` を指定することで自宅情報が使用されます。

One of object object object

type required string
 Value: "fixed_stop"
 乗降場所の種類を示します。`fixed_stop` の場合、固定の乗降場所を示します。

stop_id required string
 固定の乗降場所のIDです。

One of object object object

type required string
 Value: "custom_location"
 乗降場所の種類を示します。`custom_location` の場合、任意の地点を示します。

location required object
 乗降場所を示す、経度・緯度 (WGS-84) による地理座標です。

One of object object object

type string
 required Value: "home"
 乗降場所の種類を示します。home の場合、自宅を示します。

passenger_id string
 required ユーザーの自宅位置を指定する場合のユーザーIDです。

passenger_count Array of objects non-empty
 required 乗客の種類ごとの人数を示すリストです。
 乗客種別は、サービスの詳細情報 (GET /v1/services/{id}) で取得できる available_passenger_types を参照してください。
 合計人数が1人以上になるように指定してください。0人の種別は指定できません。

Array (non-empty) [

passenger_type_id string
 required 乗客の種別です。

count integer <int32> >= 1
 required この種別の乗客数です。

]

<p>accessibility_feature_count <small>required</small></p>	<p>Array of objects >= 0 items</p> <p>乗客全体のうち、補助機能を必要とする人数を種類ごとに指定します。各人数は <code>passenger_count</code> に含まれているものとして扱います（内数）。補助機能を利用する乗客が誰かは問わず、「何人分必要か」を表すための項目です。不要な場合は空配列 ([]) を指定してください。種類はサービス詳細 (GET /v1/services/{id}) の <code>supported_accessibility_features</code> を参照してください。</p>
<pre>Array (>= 0 items) [accessibility_feature_type string <small>required</small> 乗客補助機能の種類です。利用可能な種類はサービスの詳細情報 (GET /v1/services/{id}) で取得できる <code>supported_accessibility_featu</code> <code>res</code> を参照してください。 integer <int32> >= 1 この種類の乗客補助機能の必要数です。]</pre>	
<p>vehicle_id</p>	<p>string</p> <p>車両IDです。特定の車両を指定して便候補を検索したい場合には指定してください。</p>

- レスポンスパラメータ
- ✧ 200 OK

RESPONSE SCHEMA: application/json



pickup v
required object or object or object
 乗車場所と予定時刻の
 情報です。

One of object object object

type string
required Value: "fixed_stop"
 乗降場所の種類を示します。
fixed_stop の場合、固定の乗降場所
 を示します。

stop_id string
required 固定の乗降場所のIDです。 type が
fixed_stop の場合のみ設定されま
 ず。

location v object
required 乗降場所を示す、経度・緯度（WGS-
 84）による地理座標です。

type string
required Value: "Point"
 GeoJSONにおけるジオメトリタ
 イプ（常に Point）

coordinates Array of numbers <double> = 2
required items [items <double >]
 経度・緯度を含む配列です。
[longitude, latitude] の
 順で格納されます（GeoJSON仕
 様準拠）。

datetime string <date-time>
required 乗車または降車予定時刻です。形式は
 RFC3339のdate-time形式文字列（例：
 2025-06-19T07:00:00+09:00）です。

One of object object object

type **required** string
 Value: "custom_location"
 乗降場所の種類を示します。
 custom_location の場合、任意の地点を示します。

location **required** object
 乗降場所を示す、経度・緯度（WGS-84）による地理座標です。

type **required** string
 Value: "Point"
 GeoJSONにおけるジオメトリタイプ（常に Point）

coordinates **required** Array of numbers <double>
 items [items <double >]
 経度・緯度を含む配列です。
 [longitude, latitude] の
 順で格納されます（GeoJSON仕様準拠）。

datetime **required** string <date-time>
 乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

One of object object object

- type **required** string
 Value: "home"
 乗降場所の種類を示します。 home の場合、自宅を示します。
- passenger_id **required** string
 自宅の位置を指定する場合のユーザーIDです。 type が home の場合のみ設定されます。
- location **required** object
 乗降場所を示す、経度・緯度 (WGS-84) による地理座標です。

- type **required** string
 Value: "Point"
 GeoJSONにおけるジオメトリタイプ (常に Point)
 - coordinates **required** Array of numbers <double> = 2
 items [items <double >]
 経度・緯度を含む配列です。
 [longitude, latitude] の順で格納されます (GeoJSON仕様準拠)。
- datetime **required** string <date-time>
 乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列 (例: 2025-06-19T07:00:00+09:00) です。

dropoff **required** object or object or object
降車場所と予定時刻の
情報です。

One of **object** object object

type required string
Value: "fixed_stop"
乗降場所の種類を示します。
fixed_stop の場合、固定の乗降場所
を示します。

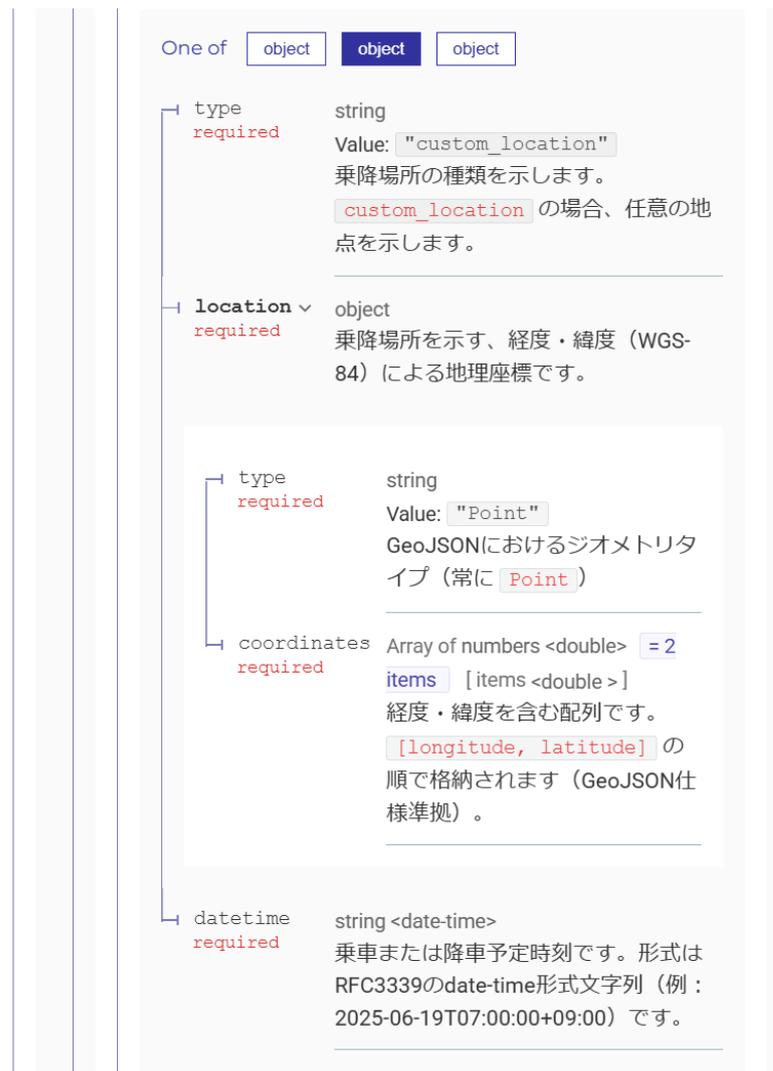
stop_id required string
固定の乗降場所のIDです。type が
fixed_stop の場合のみ設定されま
す。

location required object
乗降場所を示す、経度・緯度 (WGS-
84) による地理座標です。

type required string
Value: "Point"
GeoJSONにおけるジオメトリタ
イプ (常に Point)

coordinates required Array of numbers <double> = 2
items [items <double >]
経度・緯度を含む配列です。
[longitude, latitude] の
順で格納されます (GeoJSON仕
様準拠)。

datetime required string <date-time>
乗車または降車予定時刻です。形式は
RFC3339のdate-time形式文字列 (例:
2025-06-19T07:00:00+09:00) です。



One of object object object

- type **required** string
 Value: "home"
 乗降場所の種類を示します。 home の場合、自宅を示します。
- passenger_id **required** string
 自宅の位置を指定する場合のユーザーIDです。 type が home の場合のみ設定されます。
- location **required** object
 乗降場所を示す、経度・緯度（WGS-84）による地理座標です。

- type **required** string
 Value: "Point"
 GeoJSONにおけるジオメトリタイプ（常に Point）
 - coordinates **required** Array of numbers <double> = 2 items [items <double >]
 経度・緯度を含む配列です。
 [longitude, latitude] の順で格納されます（GeoJSON仕様準拠）。
- datetime **required** string <date-time>
 乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

`passenger_count` required Array of objects non-empty
 乗客の種別ごとの人数を示すリストです。

```

Array (non-empty) [
  | passenger_type_id string required 乗客の種別です。
  | count integer <int32> >= 1 required この種別の乗客数です。
]
    
```

`accessibility_feature_count` required Array of objects >= 0 items
 利用する乗客補助機能の種類と数です。

```

Array (>= 0 items) [
  | accessibility_feature_type string required 乗客補助機能の種類です。利用可能な種類はサービスの詳細情報（GET /v1/services/{id}）で取得できる supported_accessibility_features を参照してください。
  | count integer <int32> >= 1 required この種類の乗客補助機能の必要数です。
]
    
```

`vehicle_id` string
 車両IDです。



デマンドバスシステム連携 API システム設計書



pictures ▾	Array of objects >= 0 items 車両の画像のリストです。
<pre> Array (>= 0 items) [title string 画像のタイトルです。 ----- url string <uri> 画像のURLです。 ----- description string 画像の説明文です。 -----] </pre>	
created_at	string <date-time> 作成日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。
updated_at	string <date-time> 最終更新日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

◇ 400 Bad Request

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 401 Unauthorized

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 403 Forbidden

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 500 Internal Server Error

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

➤ レスポンスサンプル

◇ 200 OK

```
{
  "candidates": [
    {
      "reservation_request": {
        "candidate_id": "cd012345678",
        "service_id": "sv012345678",
        "pickup": {
          "type": "fixed_stop",
          "stop_id": "st012345678",
          "location": {
            "type": "Point",
            "coordinates": [
              139.751494,
              35.657774
            ]
          }
        },
        "datetime": "2025-07-01T09:30:00+09:00"
      },
      "dropoff": {
        "type": "fixed_stop",
        "stop_id": "st012345679",
        "location": {
          "type": "Point",
          "coordinates": [
            139.757875,
            35.666056
          ]
        },
        "datetime": "2025-07-01T09:55:00+09:00"
      },
      "passenger_count": [
        {
          "passenger_type_id": "pt012345678",
          "count": 2
        },
        {
          "passenger_type_id": "pt012345679",
          "count": 1
        }
      ]
    }
  ]
}
```

```
],
"accessibility_feature_count": [
  {
    "accessibility_feature_type": "wheelchair",
    "count": 1
  }
],
"vehicle_id": "vh012345678"
},
"display": {
  "pickup": {
    "display_name": "港区役所前"
  },
  "dropoff": {
    "display_name": "新橋駅前"
  },
  "fare": {
    "per_passenger_type": [
      {
        "passenger_type_id": "pt012345678",
        "subtotal": 1240
      },
      {
        "passenger_type_id": "pt012345679",
        "subtotal": 250
      }
    ],
    "total": 1490
  },
  "route_distance": 2654,
  "vehicle": {
    "id": "vh012345678",
    "name": "サンプル車両 1 号",
    "number_plate": "品川 999 ん 12-34",
    "capacity": {
      "seats": 4,
      "accessibility_features": [
        {
          "accessibility_feature_type": "wheelchair",
          "count": 1
        }
      ]
    }
  }
}
```

```
]
},
"pictures": [
  {
    "title": "サンプル車両 1 号の外観",
    "url": "https://example.com/images/vehicle-001.jpg",
    "description": "サンプル車両 1 号の外観写真"
  }
]
}
},
{
  "reservation_request": {
    "candidate_id": "cd012345679",
    "service_id": "sv012345679",
    "pickup": {
      "type": "fixed_stop",
      "stop_id": "st012345678",
      "location": {
        "type": "Point",
        "coordinates": [
          139.751494,
          35.657774
        ]
      }
    },
    "datetime": "2025-07-01T09:30:00+09:00"
  },
  "dropoff": {
    "type": "fixed_stop",
    "stop_id": "st012345679",
    "location": {
      "type": "Point",
      "coordinates": [
        139.757875,
        35.666056
      ]
    }
  },
  "datetime": "2025-07-01T09:55:00+09:00"
},
"passenger_count": [
```

```
{
  "passenger_type_id": "pt012345678",
  "count": 2
},
{
  "passenger_type_id": "pt012345679",
  "count": 1
}
],
"accessibility_feature_count": [
  {
    "accessibility_feature_type": "wheelchair",
    "count": 1
  }
],
"vehicle_id": "vh012345679"
},
"display": {
  "pickup": {
    "display_name": "任意地点"
  },
  "dropoff": {
    "display_name": "自宅"
  },
  "fare": {
    "per_passenger_type": []
  },
  "route_distance": 2654,
  "vehicle": {
    "id": "vh012345679",
    "name": "サンプル車両 3 号",
    "number_plate": "品川 999 ん 56-78",
    "capacity": {
      "seats": 5,
      "accessibility_features": [
        {
          "accessibility_feature_type": "wheelchair",
          "count": 1
        }
      ]
    }
  }
},
},
```

```
"pictures": [  
  {  
    "title": "サンプル車両 3 号の外観",  
    "url": "https://example.com/images/vh012345679.jpg",  
    "description": "サンプル車両 3 号の外観写真"  
  }  
]  
}  
}
```

◇ 400 Bad Request

```
{  
  "type": "string",  
  "title": "string",  
  "status": 400,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 401 Unauthorized

```
{  
  "type": "string",  
  "title": "string",  
  "status": 401,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 403 Forbidden

```
{  
  "type": "string",  
  "title": "string",  
  "status": 403,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 500 Internal Server Error

```
{  
  "type": "string",  
  "title": "string",  
  "status": 500,  
}
```

```
"detail": "string",  
"instance": "string"  
}
```

【IF012】 予約の登録標準 API

- 本インターフェースの概要
 - MaaS アプリ等のサードパーティからのリクエストに応じて事前に検索して取得された予約便候補を用いて、予約を登録するための API
- 本インターフェースを利用する機能
 - 【FN206】 予約登録
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ POST
 - レスポンスデータ形式
 - ◇ JSON
 - URI
 - ◇ <https://example.com/reservations>
 - 使用例
 - ◇ <https://example.com/reservations>
 - リクエストボディスキーマ

REQUEST BODY SCHEMA: application/json

passenger_id required	string	予約を登録するユーザーの一意な識別子です。
candidate_id required	string	予約候補の一意な識別子です。
service_id required	string	予約候補が属するサービスの一意な識別子です。
pickup > required	object or object or object	乗車場所と予定時刻の情報です。
dropoff > required	object or object or object	降車場所と予定時刻の情報です。
passenger_count v required	Array of objects non-empty	乗客の種別ごとの人数を示すリストです。
<pre> Array (non-empty) [passenger_type_id string required 乗客の種別です。 count integer <int32> >= 1 required この種別の乗客数です。] </pre>		
accessibility_feature_count > required	Array of objects >= 0 items	利用する乗客補助機能の種類と数です。
vehicle_id	string	車両IDです。

➤ レスポンスパラメータ

◇ 201 Created

RESPONSE SCHEMA: application/json

id required	string 登録済み予約の一意的識別子です。
passenger_id required	string 予約を登録したユーザーの一意的識別子です。
status required	string Enum: "tentative" "confirmed" "no_show" "in_transit" "completed" "cancelled" 予約のステータスです。 <ul style="list-style-type: none"> tentative 仮予約。乗車便が未確定で、締切前の状態です。確定処理が行われるまではこの状態となります。 confirmed 確定予約。確定予約。乗車便が成立した状態です。 in_transit 乗車中。車両が運行中で、乗客が乗車している状態です。 completed 降車済み。乗車が完了し、正常に終了した予約です。 no_show 未乗車。予約は確定していたものの、乗客が現れず乗車しなかった状態です。 cancelled キャンセル済み。利用者またはシステムにより予約が取り消された状態です。（仮予約・確定予約いずれからも遷移します）
service_id required	string 予約候補が属するサービスの一意的識別子です。

pickup **required** object or object or object
乗車場所と予定時刻の情報です。

One of **object** object object

- type required** string
Value: "fixed_stop"
乗降場所の種類を示します。fixed_stop の場合、固定の乗降場所を示します。
- stop_id required** string
固定の乗降場所のIDです。type が fixed_stop の場合のみ設定されます。
- display_name required** string
乗降場所の表示名です。type が fixed_stop の場合、乗降場所の名称が設定されます。
- location > required** object
乗降場所を示す、経度・緯度（WGS-84）による地理座標です。
- datetime required** string <date-time>
乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

One of object **object** object

- type required** string
Value: "custom_location"
乗降場所の種類を示します。custom_location の場合、任意の地点を示します。
- display_name required** string
乗降場所の表示名です。type が custom_location の場合「任意地点」が設定されます。
- location > required** object
乗降場所を示す、経度・緯度（WGS-84）による地理座標です。
- datetime required** string <date-time>
乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

One of object object object

- type **required** string
 Value: "home"
 乗降場所の種類を示します。home の場合、自宅を示します。
- passenger_id **required** string
 自宅の位置を指定する場合のユーザーIDです。
 type が home の場合のみ設定されます。
- display_name **required** string
 乗降場所の表示名です。type が home の場合「自宅」が設定されます。
- location > **required** object
 乗降場所を示す、経度・緯度（WGS-84）による地理座標です。
- datetime **required** string <date-time>
 乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

dropoff > **required** object or object or object
 降車場所と予定時刻の情報です。

One of object object object

- type **required** string
 Value: "fixed_stop"
 乗降場所の種類を示します。fixed_stop の場合、固定の乗降場所を示します。
- stop_id **required** string
 固定の乗降場所のIDです。type が fixed_stop の場合のみ設定されます。
- display_name **required** string
 乗降場所の表示名です。type が fixed_stop の場合、乗降場所の名称が設定されます。
- location > **required** object
 乗降場所を示す、経度・緯度（WGS-84）による地理座標です。
- datetime **required** string <date-time>
 乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

One of object object object

- type string
required Value: "custom_location"
乗降場所の種類を示します。custom_location の場合、任意の地点を示します。
- display_name string
required 乗降場所の表示名です。type が custom_location の場合「任意地点」が設定されます。
- location > object
required 乗降場所を示す、経度・緯度（WGS-84）による地理座標です。
- datetime string <date-time>
required 乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

One of object object object

- type string
required Value: "home"
乗降場所の種類を示します。home の場合、自宅を示します。
- passenger_id string
required 自宅の位置を指定する場合のユーザーIDです。type が home の場合のみ設定されます。
- display_name string
required 乗降場所の表示名です。type が home の場合「自宅」が設定されます。
- location > object
required 乗降場所を示す、経度・緯度（WGS-84）による地理座標です。
- datetime string <date-time>
required 乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

<p>passenger_count ▾ required</p>	<p>Array of objects non-empty 乗客の種別ごとの人数を示すリストです。</p>
<pre> Array (non-empty) [passenger_type_id string required 乗客の種別です。 ----- count required integer <int32> >= 1 この種別の乗客数です。] </pre>	
<p>accessibility_feature_count ▾ required</p>	<p>Array of objects >= 0 items 利用する乗客補助機能の種類と数です。</p>
<pre> Array (>= 0 items) [accessibility_feature_type string required 乗客補助機能の種類です。利用可能な種類はサービスの詳細情報 (GET /v1/services/{id}) で取得 できる supported_accessibility_features を参照してください。 ----- count required integer <int32> >= 1 この種類の乗客補助機能の必要数 です。] </pre>	
<p>fare ▾ required</p>	<p>object 運賃の情報です。</p>
<pre> per_passenger_type > required Array of objects >= 0 items 乗客の種別ごとの運賃小計を示すリストです。 ----- total integer <int32> >= 0 運賃の合計 (円) です。 </pre>	

route_distance	integer <int32> 乗車地点から降車地点までのルート探索に基づく推定距離（メートル）です。 直線距離ではなく、道路網上の最短ルートを基に計算された値です。				
vehicle	object この予約で利用する車両の情報です。 vehicle に値があるときは <code>id</code> と <code>name</code> は必ず値が設定されます。				
id	string 車両IDです。				
name	string 車両名です。				
number_plate	string 車両のナンバープレート番号です。				
capacity	object 車両の座席数と乗客補助機能の情報です。 各車両は、最大座席数と利用可能な乗客補助機能の種類とその最大数を持ちます。				
seats	integer <int32> <code>>= 1</code> 車両の最大座席数です。				
accessibility_features	Array of objects <code>>= 0 items</code> 車両に備わっている乗客補助機能の種類とそれぞれの最大数です。				
<p>Array (>= 0 items) [</p> <table border="1"> <tr> <td>accessibility_feature_type</td> <td>string 乗客補助機能の種類です。利用可能な種類はサービスの詳細情報（GET <code>/v1/services/{id}</code>）で取得できる <code>supported_accessibility_features</code> を参照してください。</td> </tr> <tr> <td>count</td> <td>integer <int32> <code>>= 1</code> この種類の乗客補助機能の必要数です。</td> </tr> </table> <p>]</p>		accessibility_feature_type	string 乗客補助機能の種類です。利用可能な種類はサービスの詳細情報（GET <code>/v1/services/{id}</code> ）で取得できる <code>supported_accessibility_features</code> を参照してください。	count	integer <int32> <code>>= 1</code> この種類の乗客補助機能の必要数です。
accessibility_feature_type	string 乗客補助機能の種類です。利用可能な種類はサービスの詳細情報（GET <code>/v1/services/{id}</code> ）で取得できる <code>supported_accessibility_features</code> を参照してください。				
count	integer <int32> <code>>= 1</code> この種類の乗客補助機能の必要数です。				

<code>pictures</code> ▾	Array of objects <code>>= 0 items</code> 車両の画像のリストです。
<pre>Array (>= 0 items) [title string 画像のタイトルです。 url string <uri> 画像のURLです。 description string 画像の説明文です。]</pre>	
<code>created_at</code>	string <date-time> 作成日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。
<code>updated_at</code>	string <date-time> 最終更新日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

◇ 400 Bad Request

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 401 Unauthorized

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 403 Forbidden

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 500 Internal Server Error

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

➤ レスポンスサンプル

◇ 201 Created

```
{
  "id": "rs012345678",
  "passenger_id": "ps012345678",
  "status": "confirmed",
  "service_id": "sv012345678",
  "pickup": {
    "type": "fixed_stop",
    "stop_id": "st012345678",
    "display_name": "港区役所前",
    "location": {
      "type": "Point",
      "coordinates": [
        139.751494,
        35.657774
      ]
    }
  },
  "datetime": "2025-07-01T09:30:00+09:00",
  "dropoff": {
    "type": "fixed_stop",
    "stop_id": "st012345679",
    "display_name": "新橋駅前",
    "location": {
      "type": "Point",
      "coordinates": [
        139.757875,
        35.666056
      ]
    }
  }
}
```

```
]
},
"datetime": "2025-07-01T09:55:00+09:00"
},
"passenger_count": [
  {
    "passenger_type_id": "pt012345678",
    "count": 2
  },
  {
    "passenger_type_id": "pt012345679",
    "count": 1
  }
],
"accessibility_feature_count": [
  {
    "accessibility_feature_type": "wheelchair",
    "count": 1
  }
],
"fare": {
  "per_passenger_type": [
    {
      "passenger_type_id": "pt012345678",
      "subtotal": 620
    },
    {
      "passenger_type_id": "pt012345679",
      "subtotal": 250
    }
  ],
  "total": 870
},
"route_distance": 2654,
"vehicle": {
  "id": "vh012345678",
  "name": "サンプル車両 1 号",
  "number_plate": "品川 999 ん 12-34",
  "capacity": {
    "seats": 4,
    "accessibility_features": [
```

```
{
  "accessibility_feature_type": "wheelchair",
  "count": 1
}
],
"pictures": [
  {
    "title": "サンプル車両 1 号の外観",
    "url": "https://example.com/images/vh012345678.jpg",
    "description": "サンプル車両 1 号の外観写真"
  }
]
},
"created_at": "2025-06-19T07:00:00+09:00",
"updated_at": "2025-06-19T07:00:00+09:00"
}
```

◇ 400 Bad Request

```
{
  "type": "string",
  "title": "string",
  "status": 400,
  "detail": "string",
  "instance": "string"
}
```

◇ 401 Unauthorized

```
{
  "type": "string",
  "title": "string",
  "status": 401,
  "detail": "string",
  "instance": "string"
}
```

◇ 403 Forbidden

```
{
  "type": "string",
  "title": "string",
  "status": 403,
  "detail": "string",
  "instance": "string"
}
```

◇ 500 Internal Server Error

```
{
  "type": "string",
  "title": "string",
  "status": 500,
  "detail": "string",
  "instance": "string"
}
```

【IF013】 ユーザーが利用可能なサービスの概要情報一覧を取得標準 API

- 本インターフェースの概要
 - MaaS アプリ等のサードパーティからのリクエストに応じて、指定したユーザーが利用可能なデマンドバスサービスを、複数件一覧で取得・返却するための API
- 本インターフェースを利用する機能
 - **【FN206】 予約登録**
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ GET
 - レスポンスデータ形式
 - ◇ JSON
 - URI
 - ◇ `https://example.com/passengers/{id}/services`
 - 使用例
 - ◇ `https://example.com/passengers/{id}/services?service_ids=1,2,3&name=港区&prefecture_code=13&municipality_code=13103&offset=30&limit=10`
 - リクエストパラメーター

PATH PARAMETERS

<code>id</code> <small>required</small>	string Example: <code>ps012345678</code> ユーザーIDを指定します。
--	--

QUERY PARAMETERS

service_ids	<p>Array of strings</p> <p>Example: <code>service_ids=1,2,3</code></p> <p>取得対象のサービスIDをカンマ区切りで指定します。 指定されたいずれかのサービスIDに該当する対象を取得します (OR条件)。</p>
name	<p>string</p> <p>Example: <code>name=港区</code></p> <p>取得対象のサービスの名称 (部分一致) を指定します。</p>
prefecture_code	<p>string</p> <p>Example: <code>prefecture_code=13</code></p> <p>都道府県コードを指定します。</p>
municipality_code	<p>string</p> <p>Example: <code>municipality_code=13103</code></p> <p>市区町村コードを指定します。</p>
status	<p>Array of strings</p> <p>Default: <code>["upcoming", "active"]</code></p> <p>Items Enum: <code>"upcoming"</code> <code>"active"</code> <code>"ended"</code></p> <p>Example: <code>status=upcoming,active</code></p> <p>フィルター対象とするサービスのステータスを複数指定します。</p> <ul style="list-style-type: none"> <code>upcoming</code> 利用開始日時を迎える前のサービス <code>active</code> 現在利用できるサービス <code>ended</code> 利用終了日時を経過したサービス
offset	<p>integer <code>>= 0</code></p> <p>Default: <code>0</code></p> <p>Example: <code>offset=30</code></p> <p>取得する結果の開始位置を指定します。</p>
limit	<p>integer <code>[1..100]</code></p> <p>Default: <code>20</code></p> <p>Example: <code>limit=10</code></p> <p>取得する結果の最大件数を指定します。</p>

- レスポンスパラメータ
- ◇ 200 OK

デマンドバスシステム連携 API システム設計書

RESPONSE SCHEMA: application/json

services ▼ Array of objects >= 0 items
required サービス情報のリストです。

Array (>= 0 items) [

id required	string サービスの一意な識別子です。
name required	string サービスの名称です。
description	string サービスの説明です。
brand_image_url	string サービスのブランドイメージ画像のURLです。
access_scope required	string Enum: "public" "resident_only" "registered_user_only" "private" "other" サービスの公開範囲を示します。 <ul style="list-style-type: none">public 限定なし、一般公開されていて誰でも利用できるサービス。resident_only サービス提供エリアに在住の地域住民のみが利用可能なサービス。registered_user_only 事前登録をしたユーザーのみが利用可能なサービス。private 一般公開されていない、特定の内部ユーザーのみが利用可能なサービス。other その他。

デマンドバスシステム連携 API システム設計書

<code>service_terms</code> <code>required</code>	Array of objects <code>>= 0 items</code> サービス利用に関する規約のリストです。
Array (<code>>= 0 items</code>) [
<code>id</code> <code>required</code>	string 規約の一意な識別子です。
<code>category</code> <code>required</code>	string Enum: "service" "privacy" "cancellation" "user" "third_party" 規約の分類カテゴリです。 <ul style="list-style-type: none"><code>service</code> サービス利用に関する規約。<code>privacy</code> プライバシーポリシーに関する規約。<code>cancellation</code> キャンセルポリシーに関する規約。<code>user</code> ユーザー向けの規約。<code>third_party</code> サードパーティとの関係に関する規約。
<code>name</code>	string 規約の名称です。
<code>url</code> <code>required</code>	string 規約のURLです。
<code>agreement_required</code> <code>required</code>	boolean 規約に対してユーザーから同意を得ることが必須かどうかを示します。
<code>created_at</code>	string <date-time> 作成日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。
<code>updated_at</code>	string <date-time> 最終更新日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。
]	

デマンドバスシステム連携 API システム設計書

`available_passenger_types` required Array of objects non-empty
サービスがサポートする乗客のタイプの情報です。
乗客のタイプは、例えば一般/未就学児/シニアなど、運賃が異なる乗客を示します。
`order` は昇順で表示して、未設定の場合は後方に配置します。

```
Array (non-empty) [  
  passenger_type_id string  
  name string  
  order integer  
  description string  
]
```

`supported_accessibility_features` required Array of strings >= 0 items
Items Enum: "wheelchair" "folding_wheelchair" "powered_wheelchair"
"oversized_wheelchair" "scooter" "oversized_scooter" "walker"
"oversized_walker" "only_car_walker" "only_car_rider"
"extra_space_ambulant" "actual_door_to_door" "lift" "oversized_item"
"stroller" "child_seat_baby" "child_seat_young" "bike" "white_cane"
"service_animal" "oxygen_tank" "stretcher"
サービスがサポートする乗客補助機能のリストです。

- `wheelchair` 車椅子
- `folding_wheelchair` 折りたたみ式車椅子
- `powered_wheelchair` 電動車椅子
- `oversized_wheelchair` 大型車椅子
- `scooter` スクーター
- `oversized_scooter` 大型スクーター
- `walker` 歩行器
- `oversized_walker` 大型歩行器
- `only_car_walker` 車内のみ歩行可能な乗客向け
- `only_car_rider` 車内のみ乗車可能な乗客向け
- `extra_space_ambulant` 追加スペースを必要とする歩行者向け
- `actual_door_to_door` 実際のドアからドアまでのサービス
- `lift` リフト付き車両
- `oversized_item` 大型荷物
- `stroller` ベビーカー
- `child_seat_baby` ベビーシート
- `child_seat_young` 幼児用シート
- `bike` 自転車
- `white_cane` 白杖使用者向けサービス
- `service_animal` 介助動物同伴可能
- `oxygen_tank` 酸素ボンベ使用者向けサービス
- `stretcher` ストレッチャー

デマンドバスシステム連携 API システム設計書

operating_hours required object

サービスの曜日および祝日ごとの運行時間帯を示します。
各曜日のプロパティは、当日における運行時間帯のリストです。
該当の曜日が空配列 ([]) の場合は、当日は運休日とみなされます。
営業時間の制限がない場合は、00:00:00~23:59:59 を返します。

monday required Array of objects >= 0 items

月曜日の運行時間帯のリストです。

Array (>= 0 items) [

- start_time_offset_sec** required integer
運行開始時刻の当日00:00からのオフセット (秒単位) です。
- end_time_offset_sec** required integer
運行終了時刻の当日00:00からのオフセット (秒単位) です。

]

tuesday required Array of objects >= 0 items

火曜日の運行時間帯のリストです。

Array (>= 0 items) [

- start_time_offset_sec** required integer
運行開始時刻の当日00:00からのオフセット (秒単位) です。
- end_time_offset_sec** required integer
運行終了時刻の当日00:00からのオフセット (秒単位) です。

]

wednesday required Array of objects >= 0 items

水曜日の運行時間帯のリストです。

Array (>= 0 items) [

- start_time_offset_sec** required integer
運行開始時刻の当日00:00からのオフセット (秒単位) です。
- end_time_offset_sec** required integer
運行終了時刻の当日00:00からのオフセット (秒単位) です。

]

thursday required Array of objects >= 0 items

木曜日の運行時間帯のリストです。

Array (>= 0 items) [

- start_time_offset_sec** required integer
運行開始時刻の当日00:00からのオフセット (秒単位) です。
- end_time_offset_sec** required integer
運行終了時刻の当日00:00からのオフセット (秒単位) です。

]

friday required Array of objects >= 0 items

金曜日の運行時間帯のリストです。

Array (>= 0 items) [

- start_time_offset_sec** required integer
運行開始時刻の当日00:00からのオフセット (秒単位) です。
- end_time_offset_sec** required integer
運行終了時刻の当日00:00からのオフセット (秒単位) です。

]

デマンドバスシステム連携 API システム設計書

```
└─ saturday ▾ Array of objects >= 0 items  
  required  
  土曜日の運行時間帯のリストです。  
  
  Array (>= 0 items) [  
    └─ start_time_offset_sec integer  
      required  
      運行開始時刻の当日00:00からのオフセット（秒単位）です。  
    └─ end_time_offset_sec integer  
      required  
      運行終了時刻の当日00:00からのオフセット（秒単位）です。  
  ]  
  
└─ sunday ▾ Array of objects >= 0 items  
  required  
  日曜日の運行時間帯のリストです。  
  
  Array (>= 0 items) [  
    └─ start_time_offset_sec integer  
      required  
      運行開始時刻の当日00:00からのオフセット（秒単位）です。  
    └─ end_time_offset_sec integer  
      required  
      運行終了時刻の当日00:00からのオフセット（秒単位）です。  
  ]  
  
└─ holidays ▾ Array of objects >= 0 items  
  required  
  祝日の運行時間帯のリストです。  
  
  Array (>= 0 items) [  
    └─ start_time_offset_sec integer  
      required  
      運行開始時刻の当日00:00からのオフセット（秒単位）です。  
    └─ end_time_offset_sec integer  
      required  
      運行終了時刻の当日00:00からのオフセット（秒単位）です。  
  ]  
  
special_operating_hours ▾ Array of objects >= 0 items  
  required  
  特別な運行時間帯のリストです。  
  
  Array (>= 0 items) [  
    └─ date string <date>  
      required  
      特別運行日の日付です。形式はRFC3339のdate形式文字列（例：2025-06-19）です。  
    └─ time_slots ▾ Array of objects  
      required  
      特別運行日の時間帯のリストです。  
  
      Array [  
        └─ start_time_offset_sec integer  
          required  
          運行開始時刻の当日00:00からのオフセット（秒単位）です。  
        └─ end_time_offset_sec integer  
          required  
          運行終了時刻の当日00:00からのオフセット（秒単位）です。  
      ]  
  ]
```

デマンドバスシステム連携 API システム設計書

announcements <small>required</small>	Array of objects <small>>= 0 items</small> サービスに関するお知らせのリストです。
Array (<small>>= 0 items</small>) [
id <small>required</small>	string お知らせの一意識別子です。
title <small>required</small>	string お知らせのタイトルです。
content <small>required</small>	string お知らせの内容です。
start_datetime <small>required</small>	string <date-time> お知らせの開始日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。
end_datetime <small>required</small>	string <date-time> お知らせの終了日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-30T20:59:59+09:00）です。この項目が指定されていない場合、お知らせは「無期限（終了日未定）」として扱われます。
]	
created_at	string <date-time> 作成日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。
updated_at	string <date-time> 最終更新日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。
]	
total <small>required</small>	integer 取得可能な総件数を示します。
offset <small>required</small>	integer 取得する結果の開始位置を示します。デフォルトは0です。
limit <small>required</small>	integer 取得する結果の最大件数を示します。デフォルトは20件です。

◇ 400 Bad Request

RESPONSE SCHEMA: application/json

type <small>required</small>	string エラーの種類を識別するURIです。
title <small>required</small>	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 401 Unauthorized

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 403 Forbidden

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 404 Not Found

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 500 Internal Server Error

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

➤ レスポンスサンプル

◇ 200 OK

```
{
  "services": [
    {
      "id": "sv012345678",
      "name": "サンプル住民向けオンデマンドサービス",
      "description": "〇×地区に住所のある方のみが利用可能なオンデマンド交通サービスです。",
      "brand_image_url": "https://example.com/logo_image.png",
      "access_scope": "resident_only",
      "operation_type": "on_demand",
      "allow_ridepooling": true,
      "start_datetime": "2025-06-19T00:00:00+09:00",
      "service_group_id": "contract123",
      "cities": [
        {
          "prefecture_code": "13",
          "prefecture_name": "東京都",
          "municipalities": [
            {
              "code": "13103",
              "name": "港区"
            }
          ]
        }
      ],
      "service_terms": [
        {
          "id": "st012345678",
```

```
"category": "service",
"name": "サンプル住民向けオンデマンドサービスの利用規約",
"url": "https://example.com/terms",
"agreement_required": true,
"created_at": "2025-06-19T12:00:00+09:00",
"updated_at": "2025-06-19T12:00:00+09:00"
}
],
"available_passenger_types": [
{
"passenger_type_id": "pt012345678",
"name": "一般",
"order": 1,
"description": "一般の大人の方"
},
{
"passenger_type_id": "pt012345679",
"name": "シニア",
"order": 2,
"description": "60 歳以上の方"
}
],
"supported_accessibility_features": [
"wheelchair",
"stroller"
],
"operating_hours": {
"monday": [
{
"start_time_offset_sec": 28800,
"end_time_offset_sec": 64800
}
],
"tuesday": [
{
"start_time_offset_sec": 28800,
"end_time_offset_sec": 64800
}
],
"wednesday": [
{
```

```
    "start_time_offset_sec": 28800,
    "end_time_offset_sec": 64800
  }
],
"thursday": [
  {
    "start_time_offset_sec": 28800,
    "end_time_offset_sec": 64800
  }
],
"friday": [
  {
    "start_time_offset_sec": 28800,
    "end_time_offset_sec": 64800
  }
],
"saturday": [
  {
    "start_time_offset_sec": 28800,
    "end_time_offset_sec": 64800
  }
],
"sunday": [],
"holidays": [
  {
    "start_time_offset_sec": 28800,
    "end_time_offset_sec": 43200
  },
  {
    "start_time_offset_sec": 46800,
    "end_time_offset_sec": 64800
  }
]
},
"special_operating_hours": [
  {
    "date": "2025-04-28",
    "time_slots": [
      {
        "start_time_offset_sec": 46800,
        "end_time_offset_sec": 64800
      }
    ]
  }
]
```

```
    }
  ]
},
{
  "date": "2025-04-29",
  "time_slots": [
    {
      "start_time_offset_sec": 28800,
      "end_time_offset_sec": 43200
    }
  ]
},
],
"announcements": [
  {
    "id": "an012345678",
    "title": "サービス開始のお知らせ",
    "content": "サンプル住民向けオンデマンドサービスが 2025 年 10 月 1 日(水)から開始されます。",
    "start_datetime": "2025-09-20T00:00:00+09:00",
    "end_datetime": "2025-12-31T23:59:59+09:00"
  }
],
"created_at": "2025-09-19T00:00:00+09:00",
"updated_at": "2025-09-19T12:00:00+09:00"
}
],
"total": 1,
"offset": 0,
"limit": 20
}
```

◇ 400 Bad Request

```
{
  "type": "string",
  "title": "string",
  "status": 400,
  "detail": "string",
  "instance": "string"
}
```

◇ 401 Unauthorized

```
{
```

```
"type": "string",  
"title": "string",  
"status": 401,  
"detail": "string",  
"instance": "string"  
}
```

◇ 403 Forbidden

```
{  
  "type": "string",  
  "title": "string",  
  "status": 403,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 404 Not Found

```
{  
  "type": "string",  
  "title": "string",  
  "status": 404,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 500 Internal Server Error

```
{  
  "type": "string",  
  "title": "string",  
  "status": 500,  
  "detail": "string",  
  "instance": "string"  
}
```

【IF014】 ユーザーの予約一覧の取得標準 API

- 本インターフェースの概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、パスパラメータで指定したユーザーが登録した予約情報を、クエリパラメータの指定に応じて複数の条件で予約情報を絞り込んで、複数件の予約を一覧で取得・返却するための API
- 本インターフェースを利用する機能
 - **【FN208】 予約表示**
 - **【FN212】 予約履歴確認**

- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ GET
 - レスポンスデータ形式
 - ◇ JSON
 - URI
 - ◇ `https://example.com/passengers/{id}/reservations`
 - 使用例
 - ◇ `https://example.com/passengers/{id}/reservations?reservation_ids=1,2,3&status=tentative,confirmed&offset=30&limit=10`
 - リクエストパラメータ

PATH PARAMETERS

→ id
required

string

Example: `ps012345678`

予約を登録したユーザーIDを指定します。

QUERY PARAMETERS

pickup_date_from	string <date> Example: pickup_date_from=2025-07-01 乗車予定日のフィルター開始日です（この日を含みます）。形式はRFC3339のdate形式文字列（例：2025-06-19）です。 指定がない場合は制限されません。
pickup_date_to	string <date> Example: pickup_date_to=2025-07-05 乗車予定日のフィルター終了日です（この日を含みます）。形式はRFC3339のdate形式文字列（例：2025-06-19）です。 指定がない場合は制限されません。
reservation_ids	Array of strings Example: reservation_ids=1,2,3 取得対象の予約IDをカンマ区切りで指定します。
status	Array of strings Items Enum: "tentative" "confirmed" "no_show" "in_transit" "completed" "cancelled" Example: status=tentative,confirmed フィルター対象とする予約のステータスを複数指定します。 指定がない場合はすべてのステータスを含む結果が返されます。 <ul style="list-style-type: none"> tentative 仮予約。乗車便が未確定で、締切前の状態です。確定処理が行われるまではこの状態となります。 confirmed 確定予約。確定予約。乗車便が成立した状態です。 in_transit 乗車中。車両が運行中で、乗客が乗車している状態です。 completed 降車済み。乗車が完了し、正常に終了した予約です。 no_show 未乗車。予約は確定していたものの、乗客が現れず乗車しなかった状態です。 cancelled キャンセル済み。利用者またはシステムにより予約が取り消された状態です。（仮予約・確定予約いずれからも遷移します）
offset	integer >= 0 Default: 0 Example: offset=30 取得する結果の開始位置を指定します。
limit	integer [1..100] Default: 20 Example: limit=10 取得する結果の最大件数を指定します。

➤ レスポンスパラメータ

◇ 200 OK

reservations ▼ required Array of objects >= 0 items
 予約情報のリストです。

```

Array (>= 0 items) [
  | id required string
  | passenger_id required string
  | status required string
  | service_id required string

```

string
登録済み予約の一意な識別子です。

string
予約を登録したユーザーの一意な識別子です。

string
Enum: "tentative" "confirmed" "no_show" "in_transit" "completed" "cancelled"
予約のステータスです。

- tentative**
仮予約。乗車便が未確定で、締切前の状態です。確定処理が行われるまではこの状態となります。
- confirmed**
確定予約。確定予約。乗車便が成立した状態です。
- in_transit**
乗車中。車両が運行中で、乗客が乗車している状態です。
- completed**
降車済み。乗車が完了し、正常に終了した予約です。
- no_show**
未乗車。予約は確定していたものの、乗客が現れず乗車しなかった状態です。
- cancelled**
キャンセル済み。利用者またはシステムにより予約が取り消された状態です。(仮予約・確定予約いずれからも遷移します)

string
予約候補が属するサービスの一意な識別子です。

pickup ▼

required

object or object or object or object
乗車場所と予定時刻の情報です。

One of object object object object

<p>▶ type</p> <p style="color: red; font-weight: bold;">required</p>	<p>string</p> <p>Value: "fixed_stop"</p> <p>乗降場所の種類を示します。 fixed_stop の場合、固定の乗降場所を示します。</p>				
<p>▶ stop_id</p> <p style="color: red; font-weight: bold;">required</p>	<p>string</p> <p>固定の乗降場所のIDです。 type が fixed_stop の場合のみ設定されます。</p>				
<p>▶ display_name</p> <p style="color: red; font-weight: bold;">required</p>	<p>string</p> <p>乗降場所の表示名です。 type が fixed_stop の場合、乗降場所の名称が設定されます。</p>				
<p>▶ location ▼</p>	<p>object</p> <p>乗降場所を示す、経度・緯度（WGS-84）による地理座標です。</p>				
<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%; vertical-align: top; border-right: 1px solid #ccc; padding-right: 5px;"> <p>▶ type</p> <p style="color: red; font-weight: bold;">required</p> </td> <td style="padding: 5px 5px;"> <p>string</p> <p>Value: "Point"</p> <p>GeoJSONにおけるジオメトリタイプ（常に Point）</p> </td> </tr> <tr> <td style="border-right: 1px solid #ccc; padding-right: 5px;"> <p>▶ coordinates</p> <p style="color: red; font-weight: bold;">required</p> </td> <td style="padding: 5px 5px;"> <p>Array of numbers <double> = 2 items [items <double >]</p> <p>経度・緯度を含む配列です。</p> <p>[longitude, latitude] の順で格納されます（GeoJSON仕様準拠）。</p> </td> </tr> </table>	<p>▶ type</p> <p style="color: red; font-weight: bold;">required</p>	<p>string</p> <p>Value: "Point"</p> <p>GeoJSONにおけるジオメトリタイプ（常に Point）</p>	<p>▶ coordinates</p> <p style="color: red; font-weight: bold;">required</p>	<p>Array of numbers <double> = 2 items [items <double >]</p> <p>経度・緯度を含む配列です。</p> <p>[longitude, latitude] の順で格納されます（GeoJSON仕様準拠）。</p>	
<p>▶ type</p> <p style="color: red; font-weight: bold;">required</p>	<p>string</p> <p>Value: "Point"</p> <p>GeoJSONにおけるジオメトリタイプ（常に Point）</p>				
<p>▶ coordinates</p> <p style="color: red; font-weight: bold;">required</p>	<p>Array of numbers <double> = 2 items [items <double >]</p> <p>経度・緯度を含む配列です。</p> <p>[longitude, latitude] の順で格納されます（GeoJSON仕様準拠）。</p>				
<p>▶ datetime</p>	<p>string <date-time></p> <p>乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。</p>				

One of object object object object

<p>▶ type</p> <p style="color: red; font-weight: bold;">required</p>	<p>string</p> <p>Value: "custom_location"</p> <p>乗降場所の種類を示します。 custom_location の場合、任意の地点を示します。</p>				
<p>▶ display_name</p> <p style="color: red; font-weight: bold;">required</p>	<p>string</p> <p>乗降場所の表示名です。 type が custom_location の場合「任意地点」が設定されます。</p>				
<p>▶ location ▼</p>	<p>object</p> <p>乗降場所を示す、経度・緯度（WGS-84）による地理座標です。</p>				
<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%; vertical-align: top; border-right: 1px solid #ccc; padding-right: 5px;"> <p>▶ type</p> <p style="color: red; font-weight: bold;">required</p> </td> <td style="padding: 5px 5px;"> <p>string</p> <p>Value: "Point"</p> <p>GeoJSONにおけるジオメトリタイプ（常に Point）</p> </td> </tr> <tr> <td style="border-right: 1px solid #ccc; padding-right: 5px;"> <p>▶ coordinates</p> <p style="color: red; font-weight: bold;">required</p> </td> <td style="padding: 5px 5px;"> <p>Array of numbers <double> = 2 items [items <double >]</p> <p>経度・緯度を含む配列です。</p> <p>[longitude, latitude] の順で格納されます（GeoJSON仕様準拠）。</p> </td> </tr> </table>	<p>▶ type</p> <p style="color: red; font-weight: bold;">required</p>	<p>string</p> <p>Value: "Point"</p> <p>GeoJSONにおけるジオメトリタイプ（常に Point）</p>	<p>▶ coordinates</p> <p style="color: red; font-weight: bold;">required</p>	<p>Array of numbers <double> = 2 items [items <double >]</p> <p>経度・緯度を含む配列です。</p> <p>[longitude, latitude] の順で格納されます（GeoJSON仕様準拠）。</p>	
<p>▶ type</p> <p style="color: red; font-weight: bold;">required</p>	<p>string</p> <p>Value: "Point"</p> <p>GeoJSONにおけるジオメトリタイプ（常に Point）</p>				
<p>▶ coordinates</p> <p style="color: red; font-weight: bold;">required</p>	<p>Array of numbers <double> = 2 items [items <double >]</p> <p>経度・緯度を含む配列です。</p> <p>[longitude, latitude] の順で格納されます（GeoJSON仕様準拠）。</p>				
<p>▶ datetime</p>	<p>string <date-time></p> <p>乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。</p>				

One of object object object object

- type**
required

string
Value: "home"
乗降場所の種類を示します。home の場合、自宅を示します。
- passenger_id**
required

string
自宅の位置を指定する場合のユーザーIDです。type が home の場合のみ設定されます。
- display_name**
required

string
乗降場所の表示名です。type が home の場合「自宅」が設定されます。
- location** ▾

object
乗降場所を示す、経度・緯度（WGS-84）による地理座標です。

- type**
required

string
Value: "Point"
GeoJSONにおけるジオメトリタイプ（常に Point）
 - coordinates**
required

Array of numbers <double> = 2 items [items <double >]
経度・緯度を含む配列です。
[longitude, latitude] の順で格納されます（GeoJSON仕様準拠）。
- datetime**

string <date-time>
乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

One of object object object object

- type**
required

string
Value: "deleted"
乗降場所の種類を示します。deleted の場合、乗降場所の情報が削除済みであることを示します。
- display_name**
required

string
乗降場所の表示名です。type が deleted の場合「削除済み」が設定されます。

dropoff ▾
required

object or object or object or object
降車場所と予定時刻の情報です。

One of object object object object

<p>type required</p>	<p>string Value: "fixed_stop" 乗降場所の種類を示します。 <code>fixed_stop</code> の場合、固定の乗降場所を示します。</p>
<p>stop_id required</p>	<p>string 固定の乗降場所のIDです。 <code>type</code> が <code>fixed_stop</code> の場合のみ設定されます。</p>
<p>display_name required</p>	<p>string 乗降場所の表示名です。 <code>type</code> が <code>fixed_stop</code> の場合、乗降場所の名称が設定されます。</p>
<p>location ▾</p>	<p>object 乗降場所を示す、経度・緯度（WGS-84）による地理座標です。</p>

<p>type required</p>	<p>string Value: "Point" GeoJSONにおけるジオメトリタイプ（常に <code>Point</code>）</p>
<p>coordinates required</p>	<p>Array of numbers <double> = 2 items [items <double >] 経度・緯度を含む配列です。 <code>[longitude, latitude]</code> の順で格納されます（GeoJSON仕様準拠）。</p>

<p>datetime</p>	<p>string <date-time> 乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。</p>
-----------------	---

One of object object object object

<p>type required</p>	<p>string Value: "custom_location" 乗降場所の種類を示します。 <code>custom_location</code> の場合、任意の地点を示します。</p>
<p>display_name required</p>	<p>string 乗降場所の表示名です。 <code>type</code> が <code>custom_location</code> の場合「任意地点」が設定されます。</p>
<p>location ▾</p>	<p>object 乗降場所を示す、経度・緯度（WGS-84）による地理座標です。</p>

<p>type required</p>	<p>string Value: "Point" GeoJSONにおけるジオメトリタイプ（常に <code>Point</code>）</p>
<p>coordinates required</p>	<p>Array of numbers <double> = 2 items [items <double >] 経度・緯度を含む配列です。 <code>[longitude, latitude]</code> の順で格納されます（GeoJSON仕様準拠）。</p>

<p>datetime</p>	<p>string <date-time> 乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。</p>
-----------------	---

- 213 -

One of object object object object

type required	string Value: "home" 乗降場所の種類を示します。home の場合、自宅を示します。				
passenger_id required	string 自宅の位置を指定する場合のユーザーIDです。type が home の場合のみ設定されます。				
display_name required	string 乗降場所の表示名です。type が home の場合「自宅」が設定されます。				
location	object 乗降場所を示す、経度・緯度（WGS-84）による地理座標です。				
<table border="1"> <tr> <td>type required</td> <td>string Value: "Point" GeoJSONにおけるジオメトリタイプ（常に Point）</td> </tr> <tr> <td>coordinates required</td> <td>Array of numbers <double> = 2 items [items <double >] 経度・緯度を含む配列です。 [longitude, latitude] の順で格納されます（GeoJSON仕様準拠）。</td> </tr> </table>	type required	string Value: "Point" GeoJSONにおけるジオメトリタイプ（常に Point）	coordinates required	Array of numbers <double> = 2 items [items <double >] 経度・緯度を含む配列です。 [longitude, latitude] の順で格納されます（GeoJSON仕様準拠）。	
type required	string Value: "Point" GeoJSONにおけるジオメトリタイプ（常に Point）				
coordinates required	Array of numbers <double> = 2 items [items <double >] 経度・緯度を含む配列です。 [longitude, latitude] の順で格納されます（GeoJSON仕様準拠）。				
datetime	string <date-time> 乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。				

One of object object object object

type required	string Value: "deleted" 乗降場所の種類を示します。deleted の場合、乗降場所の情報が削除済みであることを示します。
display_name required	string 乗降場所の表示名です。type が deleted の場合「削除済み」が設定されます。

<p>passenger_count required</p>	<p>Array of objects non-empty 乗客の種別ごとの人数を示すリストです。</p>
<pre> Array (non-empty) [passenger_type_id string required 乗客の種別です。 count required integer <int32> >= 1 この種別の乗客数です。] </pre>	
<p>accessibility_feature_count required</p>	<p>Array of objects >= 0 items 利用する乗客補助機能の種類と数です。</p>
<pre> Array (>= 0 items) [accessibility_feature_type string required 乗客補助機能の種類です。利用可能な種類はサービスの詳細情報 (GET /v1/services/{id}) で取得できる supported_accessibility_features を参照してください。 count required integer <int32> >= 1 この種類の乗客補助機能の必要数です。] </pre>	
<p>fare required</p>	<p>object 運賃の情報です。</p>
<p>per_passenger_type required</p>	<p>Array of objects >= 0 items 乗客の種別ごとの運賃小計を示すリストです。</p>
<pre> Array (>= 0 items) [passenger_type_id string required 乗客の種別です。 subtotal required integer <int32> >= 0 この種別の運賃小計 (円) です。] </pre>	
<p>total</p>	<p>integer <int32> >= 0 運賃の合計 (円) です。</p>
<p>route_distance</p>	<p>integer <int32> 乗車地点から降車地点までのルート探索に基づく推定距離 (メートル) です。直線距離ではなく、道路網上の最短ルートを基に計算された値です。</p>

vehicle	object	この予約で利用する車両の情報です。 vehicle に値があるときは <code>id</code> と <code>name</code> は必ず値が設定されます。
id	string	車両IDです。
name	string	車両名です。
number_plate	string	車両のナンバープレート番号です。
capacity	object	車両の座席数と乗客補助機能の情報です。 各車両は、最大座席数と利用可能な乗客補助機能の種類とその最大数を持ちます。
seats	integer <int32> <code>>= 1</code>	車両の最大座席数です。
accessibility_features	Array of objects <code>>= 0 items</code>	車両に備わっている乗客補助機能の種類とそれぞれの最大数です。
Array (>= 0 items) [
accessibility_feature_type	string	乗客補助機能の種類です。利用可能な種類はサービスの詳細情報 (GET /v1/services/{id}) で取得できる <code>supported_accessibility_features</code> を参照してください。
count	integer <int32> <code>>= 1</code>	この種類の乗客補助機能の必要数です。
]		

デマンドバスシステム連携 API システム設計書

<code>pictures</code> ▾	Array of objects <code>>= 0 items</code> 車両の画像のリストです。
<pre>Array (>= 0 items) [title string 画像のタイトルです。 ----- url string <uri> 画像のURLです。 ----- description string 画像の説明文です。 -----]</pre>	
<code>created_at</code>	string <date-time> 作成日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。
<code>updated_at</code>	string <date-time> 最終更新日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。
<code>total required</code>	integer 取得可能な総件数を示します。
<code>offset required</code>	integer 取得する結果の開始位置を示します。デフォルトは0です。
<code>limit required</code>	integer 取得する結果の最大件数を示します。デフォルトは20件です。

◇ 400 Bad Request

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 401 Unauthorized

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 403 Forbidden

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 404 Not Found

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 500 Internal Server Error

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

➤ レスポンスサンプル

◇ 200 OK

```
{
  "reservations": [
    {
      "id": "rs012345678",
      "passenger_id": "ps012345678",
      "status": "confirmed",
      "service_id": "sv012345678",
      "pickup": {
        "type": "fixed_stop",
        "stop_id": "st012345678",
        "display_name": "港区役所前",
        "location": {
          "type": "Point",
          "coordinates": [
```

```
    139.751494,  
    35.657774  
  ]  
},  
"datetime": "2025-07-01T09:30:00+09:00"  
},  
"dropoff": {  
  "type": "fixed_stop",  
  "stop_id": "st012345679",  
  "display_name": "新橋駅前",  
  "location": {  
    "type": "Point",  
    "coordinates": [  
      139.757875,  
      35.666056  
    ]  
  },  
  "datetime": "2025-07-01T09:55:00+09:00"  
},  
"passenger_count": [  
  {  
    "passenger_type_id": "pt012345678",  
    "count": 2  
  },  
  {  
    "passenger_type_id": "pt012345679",  
    "count": 1  
  }  
],  
"accessibility_feature_count": [  
  {  
    "accessibility_feature_type": "wheelchair",  
    "count": 1  
  }  
],  
"fare": {  
  "per_passenger_type": [  
    {  
      "passenger_type_id": "pt012345678",  
      "subtotal": 1240  
    }  
  ],  
}
```

```
{
  "passenger_type_id": "pt012345679",
  "subtotal": 250
},
{
  "total": 1490
},
{
  "route_distance": 2654,
  "vehicle": {
    "id": "vh012345678",
    "name": "サンプル車両 1 号",
    "number_plate": "品川 999 ん 12-34",
    "capacity": {
      "seats": 4,
      "accessibility_features": [
        {
          "accessibility_feature_type": "wheelchair",
          "count": 1
        }
      ]
    }
  },
  "pictures": [
    {
      "title": "サンプル車両 1 号の外観",
      "url": "https://example.com/images/vh012345678.jpg",
      "description": "サンプル車両 1 号の外観写真"
    }
  ]
},
{
  "created_at": "2025-06-01T12:00:00+09:00",
  "updated_at": "2025-06-01T12:00:00+09:00"
},
{
  "id": "rs012345681",
  "passenger_id": "ps012345678",
  "status": "cancelled",
  "service_id": "sv012345679",
  "pickup": {
    "type": "custom_location",
    "display_name": "任意地点",
    "location": {
```

```
"type": "Point",
  "coordinates": [
    139.691712,
    35.689512
  ]
},
"datetime": "2025-07-01T10:00:00+09:00"
},
"dropoff": {
  "type": "custom_location",
  "display_name": "任意地点",
  "location": {
    "type": "Point",
    "coordinates": [
      139.691706,
      35.689487
    ]
  },
  "datetime": "2025-07-01T10:30:00+09:00"
},
"passenger_count": [
  {
    "passenger_type_id": "pt012345678",
    "count": 1
  },
  {
    "passenger_type_id": "pt012345679",
    "count": 1
  }
],
"accessibility_feature_count": [
  {
    "accessibility_feature_type": "wheelchair",
    "count": 1
  }
],
"fare": {
  "per_passenger_type": []
},
"route_distance": 1800,
"created_at": "2025-06-02T12:00:00+09:00",
```

```
"updated_at": "2025-06-20T08:00:00+09:00"
}
],
"total": 2,
"offset": 0,
"limit": 20
}
```

◇ 400 Bad Request

```
{
  "type": "string",
  "title": "string",
  "status": 400,
  "detail": "string",
  "instance": "string"
}
```

◇ 401 Unauthorized

```
{
  "type": "string",
  "title": "string",
  "status": 401,
  "detail": "string",
  "instance": "string"
}
```

◇ 403 Forbidden

```
{
  "type": "string",
  "title": "string",
  "status": 403,
  "detail": "string",
  "instance": "string"
}
```

◇ 404 Not Found

```
{
  "type": "string",
  "title": "string",
  "status": 404,
  "detail": "string",
  "instance": "string"
}
```

◇ 500 Internal Server Error

```
{
  "type": "string",
  "title": "string",
  "status": 500,
  "detail": "string",
  "instance": "string"
}
```

【IF015】 予約の更新標準 API

- 概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、登録済みの予約情報を予約 ID 指定で更新するための API
- 本インターフェースを利用する機能
 - 【FN209】 予約更新
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ PUT
 - レスポンスデータ形式
 - ◇ JSON
 - URI
 - ◇ `https://example.com/reservations/{id}`
 - 使用例
 - ◇ `https://example.com/reservations/{id}`
 - リクエストパラメータ

PATH PARAMETERS

<code>→ id</code> <code>required</code>	<code>string</code> Example: <code>1</code> 予約IDを指定します。
--	---

- リクエストボディスキーマ

REQUEST BODY SCHEMA: application/json

<p>passenger_id required</p>	<p>string 予約を登録したユーザーの一意的識別子です。</p>
<p>status required</p>	<p>string Enum: "tentative" "confirmed" "no_show" "in_transit" "completed" "cancelled" 予約のステータスです。</p> <ul style="list-style-type: none"> <p>tentative 仮予約。乗車便が未確定で、締切前の状態です。確定処理が行われるまではこの状態となります。</p> <p>confirmed 確定予約。確定予約。乗車便が成立した状態です。</p> <p>in_transit 乗車中。車両が運行中で、乗客が乗車している状態です。</p> <p>completed 降車済み。乗車が完了し、正常に終了した予約です。</p> <p>no_show 未乗車。予約は確定していたものの、乗客が現れず乗車しなかった状態です。</p> <p>cancelled キャンセル済み。利用者またはシステムにより予約が取り消された状態です。（仮予約・確定予約いずれからも遷移します）</p>
<p>service_id required</p>	<p>string 予約便候補が属するサービスの一意的識別子です。</p>

pickup ▼ required object or object or object
乗車場所と予定時刻の情報です。

One of object object object

- ▶ type required string

Value: "fixed_stop"

乗降場所の種類を示します。fixed_stop の場合、固定の乗降場所を示します。

- ▶ stop_id required string

固定の乗降場所のIDです。type が fixed_stop の場合のみ設定されます。

- ▶ display_name required string

乗降場所の表示名です。type が fixed_stop の場合、乗降場所の名称が設定されます。

- ▶ location ▼ required object

乗降場所を示す、経度・緯度（WGS-84）による地理座標です。

- ▶ type required string

Value: "Point"

GeoJSONにおけるジオメトリタイプ（常に Point）

 - ▶ coordinates required Array of numbers <double> = 2 items [items <double >]

経度・緯度を含む配列です。
[longitude, latitude] の順で格納されます（GeoJSON仕様準拠）。

- ▶ datetime required string <date-time>

乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

One of object object object

<code>type</code> <code>required</code>	string Value: "custom_location" 乗降場所の種類を示します。 <code>custom_location</code> の場合、任意の地点を示します。
<code>display_name</code> <code>required</code>	string 乗降場所の表示名です。 <code>type</code> が <code>custom_location</code> の場合「任意地点」が設定されます。
<code>location</code> > <code>required</code>	object 乗降場所を示す、経度・緯度（WGS-84）による地理座標です。
<code>datetime</code> <code>required</code>	string <date-time> 乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

One of object object object

<code>type</code> <code>required</code>	string Value: "home" 乗降場所の種類を示します。 <code>home</code> の場合、自宅を示します。
<code>passenger_id</code> <code>required</code>	string 自宅の位置を指定する場合のユーザーIDです。 <code>type</code> が <code>home</code> の場合のみ設定されます。
<code>display_name</code> <code>required</code>	string 乗降場所の表示名です。 <code>type</code> が <code>home</code> の場合「自宅」が設定されます。
<code>location</code> > <code>required</code>	object 乗降場所を示す、経度・緯度（WGS-84）による地理座標です。
<code>datetime</code> <code>required</code>	string <date-time> 乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

dropoff ▼
required object or object or object
降車場所と予定時刻の情報です。

One of object object object

type required	string Value: "fixed_stop" 乗降場所の種類を示します。 fixed_stop の場合、固定の乗降場所を示します。
stop_id required	string 固定の乗降場所のIDです。 type が fixed_stop の場合のみ設定されます。
display_name required	string 乗降場所の表示名です。 type が fixed_stop の場合、乗降場所の名称が設定されます。
location ▼ required	object 乗降場所を示す、経度・緯度（WGS-84）による地理座標です。

type required	string Value: "Point" GeoJSONにおけるジオメトリタイプ（常に Point ）
coordinates required	Array of numbers <double> = 2 items [items <double >] 経度・緯度を含む配列です。 [longitude, latitude] の順で格納されます（GeoJSON仕様準拠）。

datetime required	string <date-time> 乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。
--	---

One of object object object

type required	string Value: "custom_location" 乗降場所の種類を示します。 custom_location の場合、任意の地点を示します。
display_name required	string 乗降場所の表示名です。 type が custom_location の場合「任意地点」が設定されます。
location > required	object 乗降場所を示す、経度・緯度（WGS-84）による地理座標です。
datetime required	string <date-time> 乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

One of object object object

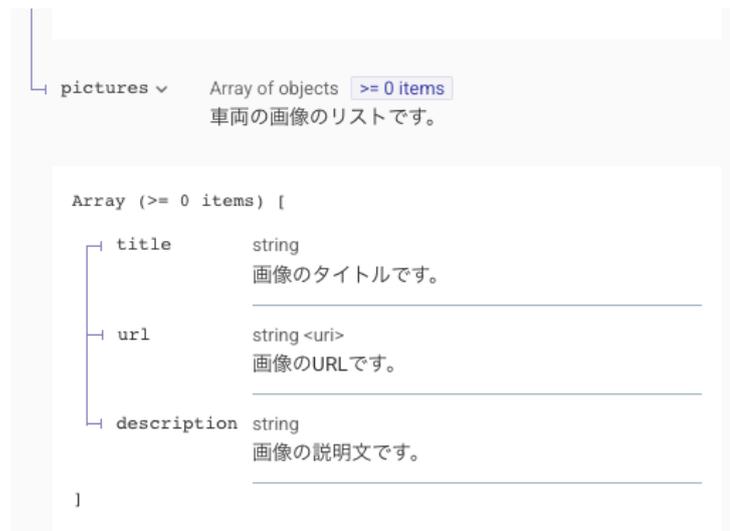
type required	string Value: "home" 乗降場所の種類を示します。 home の場合、自宅を示します。
passenger_id required	string 自宅の位置を指定する場合のユーザーIDです。 type が home の場合のみ設定されます。
display_name required	string 乗降場所の表示名です。 type が home の場合「自宅」が設定されます。
location > required	object 乗降場所を示す、経度・緯度（WGS-84）による地理座標です。
datetime required	string <date-time> 乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

<p>passenger_count ▼ required</p>	<p>Array of objects non-empty 乗客の種別ごとの人数を示すリストです。</p>
<pre> Array (non-empty) [passenger_type_id string required 乗客の種別です。 ----- count required integer <int32> >= 1 この種別の乗客数です。 -----] </pre>	
<p>accessibility_feature_count ▼ required</p>	<p>Array of objects >= 0 items 利用する乗客補助機能の種類と数です。</p>
<pre> Array (>= 0 items) [accessibility_feature_type string required 乗客補助機能の種類です。利用可能な種類はサービスの詳細情報（GET /v1/services/{id}）で取得できる supported_accessibility_featu res を参照してください。 ----- count required integer <int32> >= 1 この種類の乗客補助機能の必要数です。 -----] </pre>	

vehicle ▾

object
 この予約で利用する車両の情報です。
 vehicle に値があるときは `id` と `name` は必ず値が設定されます。

id	string	車両IDです。
name	string	車両名です。
number_plate	string	車両のナンバープレート番号です。
capacity ▾	object	車両の座席数と乗客補助機能の情報です。 各車両は、最大座席数と利用可能な乗客補助機能の種類とその最大数を持ちます。
seats	integer <int32> <code>>= 1</code>	車両の最大座席数です。
accessibility_features ▾	Array of objects <code>>= 0 items</code>	車両に備わっている乗客補助機能の種類とそれぞれの最大数です。
Array (>= 0 items) [
accessibility_feature_type	string	乗客補助機能の種類です。利用可能な種類はサービスの詳細情報 (GET /v1/services/{id}) で取得できる <code>supported_accessibility_features</code> を参照してください。
count	integer <int32> <code>>= 1</code>	この種類の乗客補助機能の必要数です。
]		



- レスポンスパラメータ
 - ◇ 200 OK

RESPONSE SCHEMA: application/json

<p>id required</p>	<p>string 登録済み予約の一意な識別子です。</p>
<p>passenger_id required</p>	<p>string 予約を登録したユーザーの一意な識別子です。</p>
<p>status required</p>	<p>string Enum: "tentative" "confirmed" "no_show" "in_transit" "completed" "cancelled" 予約のステータスです。</p> <ul style="list-style-type: none"> • tentative 仮予約。乗車便が未確定で、締切前の状態です。確定処理が行われるまではこの状態となります。 • confirmed 確定予約。確定予約。乗車便が成立した状態です。 • in_transit 乗車中。車両が運行中で、乗客が乗車している状態です。 • completed 降車済み。乗車が完了し、正常に終了した予約です。 • no_show 未乗車。予約は確定していたものの、乗客が現れず乗車しなかった状態です。 • cancelled キャンセル済み。利用者またはシステムにより予約が取り消された状態です。（仮予約・確定予約いずれからも遷移します）
<p>service_id required</p>	<p>string 予約候補が属するサービスの一意な識別子です。</p>

pickup ▾ object or object or object
required 乗車場所と予定時刻の情報です。

One of object object object

- type required string
 Value: "fixed_stop"
 乗降場所の種類を示します。fixed_stop の場合、固定の乗降場所を示します。
- stop_id required string
 固定の乗降場所のIDです。type が fixed_stop の場合のみ設定されます。
- display_name required string
 乗降場所の表示名です。type が fixed_stop の場合、乗降場所の名称が設定されます。
- location > required object
 乗降場所を示す、経度・緯度（WGS-84）による地理座標です。
- datetime required string <date-time>
 乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

One of object object object

- type required string
 Value: "custom_location"
 乗降場所の種類を示します。custom_location の場合、任意の地点を示します。
- display_name required string
 乗降場所の表示名です。type が custom_location の場合「任意地点」が設定されます。
- location > required object
 乗降場所を示す、経度・緯度（WGS-84）による地理座標です。
- datetime required string <date-time>
 乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

One of object object object

- type **required** string
 Value: "home"
 乗降場所の種類を示します。home の場合、自宅を示します。
- passenger_id **required** string
 自宅の位置を指定する場合のユーザーIDです。
 type が home の場合のみ設定されます。
- display_name **required** string
 乗降場所の表示名です。type が home の場合「自宅」が設定されます。
- location > **required** object
 乗降場所を示す、経度・緯度（WGS-84）による地理座標です。
- datetime **required** string <date-time>
 乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

dropoff > **required** object or object or object
 降車場所と予定時刻の情報です。

One of object object object

- type **required** string
 Value: "fixed_stop"
 乗降場所の種類を示します。fixed_stop の場合、固定の乗降場所を示します。
- stop_id **required** string
 固定の乗降場所のIDです。type が fixed_stop の場合のみ設定されます。
- display_name **required** string
 乗降場所の表示名です。type が fixed_stop の場合、乗降場所の名称が設定されます。
- location > **required** object
 乗降場所を示す、経度・緯度（WGS-84）による地理座標です。
- datetime **required** string <date-time>
 乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

One of object object object

- type **required** string
 Value: "custom_location"
 乗降場所の種類を示します。custom_location の場合、任意の地点を示します。
- display_name **required** string
 乗降場所の表示名です。type が custom_location の場合「任意地点」が設定されます。
- location > **required** object
 乗降場所を示す、経度・緯度（WGS-84）による地理座標です。
- datetime **required** string <date-time>
 乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

One of object object object

- type **required** string
 Value: "home"
 乗降場所の種類を示します。home の場合、自宅を示します。
- passenger_id **required** string
 自宅の位置を指定する場合のユーザーIDです。type が home の場合のみ設定されます。
- display_name **required** string
 乗降場所の表示名です。type が home の場合「自宅」が設定されます。
- location > **required** object
 乗降場所を示す、経度・緯度（WGS-84）による地理座標です。
- datetime **required** string <date-time>
 乗車または降車予定時刻です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

<p>passenger_count ▾ required</p>	<p>Array of objects non-empty 乗客の種別ごとの人数を示すリストです。</p>
<pre> Array (non-empty) [passenger_type_id string required 乗客の種別です。 ----- count required integer <int32> >= 1 この種別の乗客数です。] </pre>	
<p>accessibility_feature_count ▾ required</p>	<p>Array of objects >= 0 items 利用する乗客補助機能の種類と数です。</p>
<pre> Array (>= 0 items) [accessibility_feature_type string required 乗客補助機能の種類です。利用可能な種類はサービスの詳細情報 (GET /v1/services/{id}) で取得 できる supported_accessibility_features を参照してください。 ----- count required integer <int32> >= 1 この種類の乗客補助機能の必要数 です。] </pre>	
<p>fare ▾ required</p>	<p>object 運賃の情報です。</p>
<pre> per_passenger_type > required Array of objects >= 0 items 乗客の種別ごとの運賃小計を示すリスト です。 ----- total integer <int32> >= 0 運賃の合計（円）です。 </pre>	
<p>route_distance</p>	<p>integer <int32> 乗車地点から降車地点までのルート探索に基づく推定距離（メートル）です。 直線距離ではなく、道路網上の最短ルートを基に計算された値です。</p>



pictures ▾	Array of objects >= 0 items 車両の画像のリストです。
<pre> Array (>= 0 items) [title string 画像のタイトルです。 url string <uri> 画像のURLです。 description string 画像の説明文です。] </pre>	
created_at	string <date-time> 作成日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。
updated_at	string <date-time> 最終更新日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

◇ 400 Bad Request

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 401 Unauthorized

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 403 Forbidden

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 404 Not Found

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 500 Internal Server Error

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

➤ レスポンスサンプル

◇ 200 OK

```
{
  "id": "rs012345678",
  "passenger_id": "ps012345678",
  "status": "confirmed",
  "service_id": "sv012345678",
  "pickup": {
    "type": "fixed_stop",
    "stop_id": "st012345678",
    "display_name": "港区役所前",
    "location": {
      "type": "Point",
      "coordinates": [
        139.751494,
        35.657774
      ]
    }
  },
  "datetime": "2025-07-01T09:30:00+09:00",
  "dropoff": {
    "type": "fixed_stop",
    "stop_id": "st012345679",
    "display_name": "新橋駅前",
    "location": {
      "type": "Point",
      "coordinates": [
        139.757875,
        35.666056
      ]
    }
  }
}
```

```
]
},
"datetime": "2025-07-01T09:55:00+09:00"
},
"passenger_count": [
  {
    "passenger_type_id": "pt012345678",
    "count": 2
  },
  {
    "passenger_type_id": "pt012345679",
    "count": 1
  }
],
"accessibility_feature_count": [
  {
    "accessibility_feature_type": "wheelchair",
    "count": 1
  }
],
"fare": {
  "per_passenger_type": [
    {
      "passenger_type_id": "pt012345678",
      "subtotal": 620
    },
    {
      "passenger_type_id": "pt012345679",
      "subtotal": 250
    }
  ],
  "total": 870
},
"route_distance": 2654,
"vehicle": {
  "id": "vh012345678",
  "name": "サンプル車両 1 号",
  "number_plate": "品川 999 ん 12-34",
  "capacity": {
    "seats": 4,
    "accessibility_features": [
```

```
{
  "accessibility_feature_type": "wheelchair",
  "count": 1
}
],
"pictures": [
  {
    "title": "サンプル車両 1 号の外観",
    "url": "https://example.com/images/vh012345678.jpg",
    "description": "サンプル車両 1 号の外観写真"
  }
]
},
"created_at": "2025-06-19T07:15:00+09:00",
"updated_at": "2025-06-20T11:23:00+09:00"
}
```

◇ 400 Bad Request

```
{
  "type": "string",
  "title": "string",
  "status": 400,
  "detail": "string",
  "instance": "string"
}
```

◇ 401 Unauthorized

```
{
  "type": "string",
  "title": "string",
  "status": 401,
  "detail": "string",
  "instance": "string"
}
```

◇ 403 Forbidden

```
{
  "type": "string",
  "title": "string",
  "status": 403,
  "detail": "string",
  "instance": "string"
}
```

```
}
◇ 404 Not Found
{
  "type": "string",
  "title": "string",
  "status": 404,
  "detail": "string",
  "instance": "string"
}
◇ 500 Internal Server Error
{
  "type": "string",
  "title": "string",
  "status": 500,
  "detail": "string",
  "instance": "string"
}
```

【IF016】 予約の決済情報の取得標準 API

- 本インターフェースの概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、登録済みの予約に対する決済情報を取得・返却するための API
- 本インターフェースを利用する機能
 - **【FN211】 運賃支払い確認**
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ GET
 - レスポンスデータ形式
 - ◇ JSON
 - URI
 - ◇ `https://example.com/reservations/{id}/payment`
 - 使用例
 - ◇ `https://example.com/reservations/{id}/payment`
 - リクエストパラメータ

PATH PARAMETERS

→ id string
 required Example: 1
 予約IDを指定します。

➤ レスポンスパラメータ

◇ 200 OK

RESPONSE SCHEMA: application/json

id	string
required	決済IDです。
amount	integer <int32> <code>>= 0</code>
required	決済金額（円）を示します。
payment_status	string
required	Enum: <code>"uncollected"</code> <code>"received"</code> <code>"excluded"</code> <code>"cancelled"</code>
	決済の状態を示します。
	<ul style="list-style-type: none"><code>uncollected</code> 未収<code>received</code> 受領済み<code>excluded</code> 決済対象外<code>cancelled</code> キャンセル済み
created_at	string <date-time>
	作成日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。
updated_at	string <date-time>
	最終更新日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

◇ 401 Unauthorized

RESPONSE SCHEMA: application/json

type	string
required	エラーの種類を識別するURIです。
title	string
required	人間が読めるエラーの簡素な概要です。
status	integer
	HTTPステータスコードです。
detail	string
	人間が読めるエラーの詳細な説明です。
instance	string
	このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 403 Forbidden

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 404 Not Found

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 500 Internal Server Error

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

➤ レスポンスサンプル

◇ 200 OK

```
{
  "id": "rp012345678",
  "amount": 820,
  "payment_status": "received",
  "created_at": "2025-07-11T12:00:00+09:00",
  "updated_at": "2025-07-11T12:00:00+09:00"
}
```

◇ 401 Unauthorized

```
{
  "type": "string",
  "title": "string",
  "status": 401,
  "detail": "string",
  "instance": "string"
}
```

◇ 403 Forbidden

```
{
  "type": "string",
  "title": "string",
  "status": 403,
  "detail": "string",
  "instance": "string"
}
```

◇ 404 Not Found

```
{
  "type": "string",
  "title": "string",
  "status": 404,
  "detail": "string",
  "instance": "string"
}
```

◇ 500 Internal Server Error

```
{
  "type": "string",
  "title": "string",
  "status": 500,
  "detail": "string",
  "instance": "string"
}
```

【IF017】 予約の決済情報の登録または更新標準 API

- 本インターフェースの概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、指定された予約に対して、決済金額及び決済ステータスを登録または更新するための API
- 本インターフェースを利用する機能
 - 【FN207】 決済ステータス更新
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ PUT
 - レスポンスデータ形式
 - ◇ JSON
 - URI
 - ◇ `https://example.com/reservations/{reservation_id}/payment`
 - リクエストパラメータ

PATH PARAMETERS

→ id string
required
Example: 1
予約IDを指定します。

- リクエストボディスキーマ

REQUEST BODY SCHEMA: application/json

```
amount integer <int32> >= 0  
required 決済金額（円）を示します。  
payment_status string  
required Enum: "uncollected" "received" "excluded"  
"cancelled"  
決済の状態を示します。  
• uncollected  
  未収  
• received  
  受領済み  
• excluded  
  決済対象外  
• cancelled  
  キャンセル済み
```

- レスポンスパラメータ
 - ◇ 200 OK

RESPONSE SCHEMA: application/json

id	string
required	決済IDです。
amount	integer <int32> >= 0
required	決済金額（円）を示します。
payment_status	string
required	Enum: "uncollected" "received" "excluded" "cancelled" 決済の状態を示します。 <ul style="list-style-type: none">uncollected 未収received 受領済みexcluded 決済対象外cancelled キャンセル済み
created_at	string <date-time> 作成日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。
updated_at	string <date-time> 最終更新日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

◇ 400 Bad Request

RESPONSE SCHEMA: application/json

type	string
required	エラーの種類を識別するURIです。
title	string
required	人間が読めるエラーの簡素な概要です。
status	integer
	HTTPステータスコードです。
detail	string
	人間が読めるエラーの詳細な説明です。
instance	string
	このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 401 Unauthorized

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 403 Forbidden

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 404 Not Found

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 500 Internal Server Error

RESPONSE SCHEMA: application/json

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

➤ レスポンスサンプル

◇ 200 OK

```
{
  "id": "rp012345678",
  "amount": 820,
  "payment_status": "cancelled",
  "created_at": "2025-07-11T12:00:00+09:00",
  "updated_at": "2025-07-11T12:00:00+09:00"
}
```

◇ 400 Bad Request

```
{
  "type": "string",
  "title": "string",
  "status": 400,
  "detail": "string",
  "instance": "string"
}
```

◇ 401 Unauthorized

```
{
  "type": "string",
  "title": "string",
  "status": 401,
  "detail": "string",
  "instance": "string"
}
```

◇ 403 Forbidden

```
{
  "type": "string",
  "title": "string",
  "status": 403,
  "detail": "string",
  "instance": "string"
}
```

```
"status": 403,  
"detail": "string",  
"instance": "string"  
}
```

◇ 404 Not Found

```
{  
  "type": "string",  
  "title": "string",  
  "status": 404,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 500 Internal Server Error

```
{  
  "type": "string",  
  "title": "string",  
  "status": 500,  
  "detail": "string",  
  "instance": "string"  
}
```

【IF018】 車両位置情報取得標準 API

- 本インターフェースの概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、車両の現在位置情報を取得・返却するための API
- 本インターフェースを利用する機能
 - 【FN210】 運行情報確認
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ GET
 - レスポンスデータ形式
 - ◇ JSON
 - URI
 - ◇ <https://example.com/vehicle-locations>
 - 使用例
 - ◇ https://example.com/vehicle-locations?vehicle_ids=1,2&service_ids=1,2&offset=30&limit=10
 - リクエストパラメータ

QUERY PARAMETERS

vehicle_ids	Array of strings Example: <code>vehicle_ids=1,2,3</code> 取得対象の車両IDをカンマ区切りで指定します。指定されたいずれかの車両IDに該当する対象を取得します（OR条件）。
service_ids	Array of strings Example: <code>service_ids=1,2,3</code> 取得対象のサービスIDをカンマ区切りで指定します。指定されたいずれかのサービスIDに該当する対象を取得します（OR条件）。
offset	integer <code>>= 0</code> Default: <code>0</code> Example: <code>offset=30</code> 取得する結果の開始位置を指定します。
limit	integer <code>[1 .. 100]</code> Default: <code>20</code> Example: <code>limit=10</code> 取得する結果の最大件数を指定します。

デマンドバスシステム連携 API システム設計書

➤ レスポンスパラメータ

◇ 200 OK

RESPONSE SCHEMA: application/json

vehicle_locations <small>required</small>	Array of objects <small>>= 0 items</small> 車両の位置情報のリストです。
Array (>= 0 items) [
vehicle_id <small>required</small>	string 車両の一意な識別子です。
service_id <small>required</small>	string 車両が属するサービスのIDです。
location <small>required</small>	object 車両位置を示す、経度・緯度 (WGS-84) による地理座標です。
type <small>required</small>	string Value: "Point" GeoJSONにおけるジオメトリタイプ (常に Point)
coordinates <small>required</small>	Array of numbers <double> <small>= 2 items</small> [items <double >] 経度・緯度を含む配列です。 [longitude , latitude] の順で格納されます (GeoJSON仕様準拠)。
timestamp <small>required</small>	string <date-time> 車両位置情報の取得日時です。形式はRFC3339のdate-time形式文字列 (例: 2025-06-19T07:00:00+09:00) です。
]	
total <small>required</small>	integer 取得可能な総件数を示します。
offset <small>required</small>	integer 取得する結果の開始位置を示します。デフォルトは0です。
limit <small>required</small>	integer 取得する結果の最大件数を示します。デフォルトは20件です。

◇ 400 Bad Request

RESPONSE SCHEMA: application/json

type <small>required</small>	string エラーの種類を識別するURIです。
title <small>required</small>	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 401 Unauthorized

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 403 Forbidden

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 500 Internal Server Error

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

➤ レスポンスサンプル

◇ 200 OK

```
{
  "vehicle_locations": [
    {
      "vehicle_id": "vh012345678",
      "service_id": "sv012345678",
      "location": {
        "type": "Point",
        "coordinates": [
          139.745412,
          35.658612
        ]
      },
      "timestamp": "2025-06-19T07:00:00+09:00"
    },
    {
      "vehicle_id": "vh012345679",
      "service_id": "sv012345678",
      "location": {
        "type": "Point",
        "coordinates": [
          139.751494,
          35.657774
        ]
      },
      "timestamp": "2025-06-19T07:05:00+09:00"
    }
  ],
  "total": 2,
  "offset": 0,
  "limit": 10
}
```

◇ 400 Bad Request

```
{
  "type": "string",
  "title": "string",
  "status": 400,
  "detail": "string",
  "instance": "string"
}
```

◇ 401 Unauthorized

```
{
  "type": "string",
  "title": "string",
  "status": 401,
  "detail": "string",
  "instance": "string"
}
```

◇ 403 Forbidden

```
{
  "type": "string",
  "title": "string",
  "status": 403,
  "detail": "string",
  "instance": "string"
}
```

◇ 500 Internal Server Error

```
{
  "type": "string",
  "title": "string",
  "status": 500,
  "detail": "string",
  "instance": "string"
}
```

【IF019】 運行遅延情報取得標準 API

- 本インターフェースの概要
 - MaaS アプリ等のサードパーティからのリクエストに応じ、運行遅延情報を取得・返却する API
- 本インターフェースを利用する機能
 - **【FN210】** 運行情報確認
- 本 API の共通仕様
 - プロトコル
 - ◇ HTTPS
 - メソッド
 - ◇ GET
 - レスポンスデータ形式
 - ◇ JSON
 - URI
 - ◇ <https://example.com/operation-delays>

➤ 使用例

◇ https://example.com/operation-delays?vehicle_ids=1,2&service_ids=1,2&offset=30&limit=10

➤ リクエストパラメータ

QUERY PARAMETERS

status	Array of strings Default: ["active"] Items Enum: "active" "resolved" Example: status=active,resolved フィルター対象とする 運行遅延 のステータスを複数指定します。 <ul style="list-style-type: none">active 現在進行中の遅延resolved 解決済みの遅延
vehicle_ids	Array of strings Example: vehicle_ids=1,2,3 取得対象の車両IDをカンマ区切りで指定します。指定されたいずれかの車両IDに該当する対象を取得します（OR条件）。
service_ids	Array of strings Example: service_ids=1,2,3 取得対象のサービスIDをカンマ区切りで指定します。指定されたいずれかのサービスIDに該当する対象を取得します（OR条件）。
started_at_from	string <date-time> Example: started_at_from=2025-06-01T00:00:00+09:00 取得対象とする遅延情報の開始日時（以降）です。RFC3339のdate-time形式文字列（例：2025-07-01T00:00:00+09:00）で指定してください。
started_at_to	string <date-time> Example: started_at_to=2025-06-30T23:59:59+09:00 取得対象とする遅延情報の開始日時（以前）です。RFC3339のdate-time形式文字列（例：2025-07-03T23:59:59+09:00）で指定してください。
offset	integer >= 0 Default: 0 Example: offset=30 取得する結果の開始位置を指定します。
limit	integer [1 .. 100] Default: 20 Example: limit=10 取得する結果の最大件数を指定します。

デマンドバスシステム連携 API システム設計書

レスポンスパラメータ

◇ 200 OK

RESPONSE SCHEMA: application/json

operation_delays <small>required</small>	Array of objects <small>>= 0 items</small> 運行遅延情報のリストです。
Array (>= 0 items) [
id <small>required</small>	string 遅延情報の一意な識別子です。
status <small>required</small>	string Enum: "active" "resolved" 遅延の状態を示します。 <ul style="list-style-type: none">• active 遅延している• resolved 遅延解消済み
vehicle_id <small>required</small>	string 遅延している車両のIDです。
service_id	string 車両が属するサービスのIDです。
delay_minutes	integer <int32> <small>>= 0</small> 車両の運行遅延時間（分単位）です。
delay_reason <small>required</small>	string Enum: "traffic" "operational" "accident" "weather" "vehicle_issue" "passenger_issue" "other" 遅延の理由を表すコードです。 <ul style="list-style-type: none">• traffic 交通渋滞• accident 事故• weather 天候による影響• vehicle_issue 車両の問題• operational 運行上の問題• passenger_issue 乗客に関する問題• other その他

delay_reason_detail	string 遅延の理由に関する詳細な説明です。
started_at	string <date-time> 遅延が開始した日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。
closed_at	string <date-time> 遅延が終了した日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。 <code>status</code> が <code>resolved</code> の場合に限り当該項目が設定されることがあります。
closed_reason	string Enum: <code>"vehicle_moved"</code> <code>"resolved"</code> <code>"schedule_changed"</code> <code>"cancelled"</code> 遅延が終了した理由を表すコードです。 <code>status</code> が <code>resolved</code> の場合に限り当該項目が設定されることがあります。 <ul style="list-style-type: none"> <code>vehicle_moved</code> 車両が動いた <code>resolved</code> 遅延が解消された <code>schedule_changed</code> スケジュール変更された <code>cancelled</code> 運行をキャンセルした
closed_reason_detail	string 遅延終了の理由に関する詳細な説明です。 <code>status</code> が <code>resolved</code> の場合に限り当該項目が設定されることがあります。
effective_end required	string <date-time> 遅延情報の有効期限を示す日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。 ここで指定された日時まで、有効な遅延情報として扱われます。
created_at	string <date-time> 作成日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。
updated_at	string <date-time> 最終更新日時です。形式はRFC3339のdate-time形式文字列（例：2025-06-19T07:00:00+09:00）です。

1

total required	integer	取得可能な総件数を示します。
offset required	integer	取得する結果の開始位置を示します。デフォルトは0です。
limit required	integer	取得する結果の最大件数を示します。デフォルトは20件です。

◇ 400 Bad Request

RESPONSE SCHEMA: application/json

type required	string	エラーの種類を識別するURIです。
title required	string	人間が読めるエラーの簡素な概要です。
status	integer	HTTPステータスコードです。
detail	string	人間が読めるエラーの詳細な説明です。
instance	string	このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 401 Unauthorized

RESPONSE SCHEMA: application/json

type required	string	エラーの種類を識別するURIです。
title required	string	人間が読めるエラーの簡素な概要です。
status	integer	HTTPステータスコードです。
detail	string	人間が読めるエラーの詳細な説明です。
instance	string	このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 403 Forbidden

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

◇ 500 Internal Server Error

RESPONSE SCHEMA: **application/json**

type required	string エラーの種類を識別するURIです。
title required	string 人間が読めるエラーの簡素な概要です。
status	integer HTTPステータスコードです。
detail	string 人間が読めるエラーの詳細な説明です。
instance	string このエラーが発生した特定のリクエストやリソースを識別するURIです。

➤ レスポンスサンプル

◇ 200 OK

```
{
  "operation_delays": [
    {
      "id": "de012345678",
      "status": "resolved",
      "vehicle_id": "vh012345678",
      "service_id": "sv012345678",
      "delay_minutes": 5,
      "delay_reason": "traffic",
      "delay_reason_detail": "途中の経路上で渋滞が発生しているため、遅れが発生しています。",
      "started_at": "2025-06-19T07:00:00+09:00",
      "closed_at": "2025-06-19T07:00:00+09:00",
      "closed_reason": "resolved",
      "closed_reason_detail": "渋滞を抜けたため、遅延が解消されました。",
      "effective_end": "2025-06-19T23:59:59+09:00",
      "created_at": "2025-06-19T07:00:00+09:00",
      "updated_at": "2025-06-19T07:12:23+09:00"
    },
    {
      "id": "de012345679",
      "status": "active",
      "vehicle_id": "vh012345679",
      "service_id": "sv012345679",
      "delay_minutes": 10,
      "delay_reason": "vehicle_issue",
      "delay_reason_detail": "車両の故障により、運行が遅れています。",
      "started_at": "2025-06-19T07:00:00+09:00",
      "effective_end": "2025-06-19T23:59:59+09:00",
      "created_at": "2025-06-19T07:00:00+09:00",
      "updated_at": "2025-06-19T07:12:23+09:00"
    }
  ],
  "total": 2,
  "offset": 0,
  "limit": 10
}
```

◇ 400 Bad Request

```
{  
  "type": "string",  
  "title": "string",  
  "status": 400,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 401 Unauthorized

```
{  
  "type": "string",  
  "title": "string",  
  "status": 401,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 403 Forbidden

```
{  
  "type": "string",  
  "title": "string",  
  "status": 403,  
  "detail": "string",  
  "instance": "string"  
}
```

◇ 500 Internal Server Error

```
{  
  "type": "string",  
  "title": "string",  
  "status": 500,  
  "detail": "string",  
  "instance": "string"  
}
```

2-5. ユーザーインターフェース (UI)

2-5-1. 画面遷移図

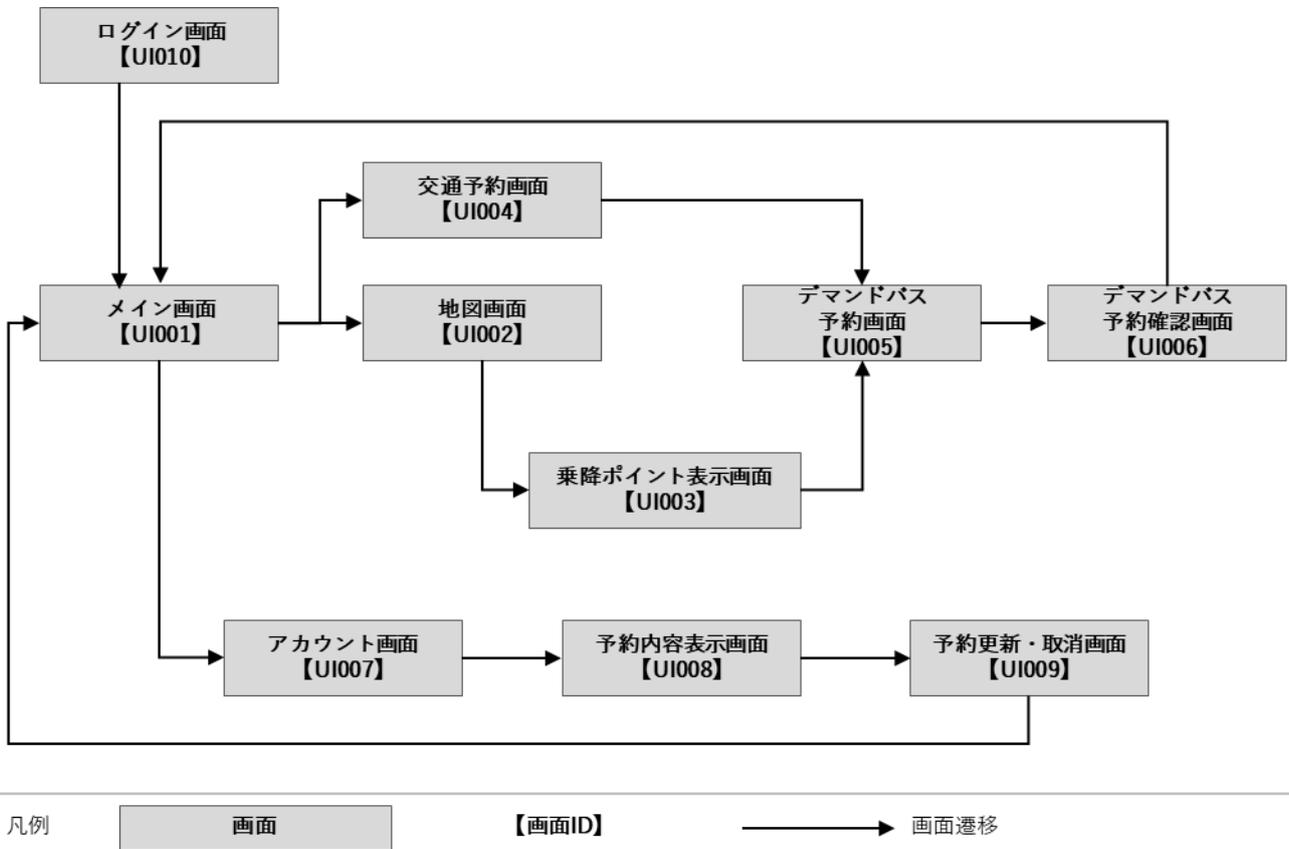


図 2-27 【HW001】 スマートフォンの MaaS アプリ画面遷移図

2-5-2. ユーザーインターフェース一覧

表 2-7 【HW001】 スマートフォンの MaaS アプリ画面一覧

※朱文字：新規開発・既存改修

ID	画面名	説明	画面を表示した機能 (ID)
UI001	メイン画面	● アプリインストール直後 or 起動直後のデフォルト画面	FN001
UI002	地図画面	● ユーザーが地図の上で、現在地やバス停、駅、施設などを確認する画面	FN001
UI003	乗降ポイント表示画面	● デマンドバスの乗降ポイント（設定有りの場合）の表示をする画面	FN001 FN101 FN201 FN204
UI004	交通予約画面	● 地図画面を経由せず、直接各地域の交通手段を予約する場合の入口画面	FN001 FN101 FN201 FN203
UI005	デマンドバス予約画面	● デマンドバスを予約する画面	FN001 FN101 FN201 FN205
UI006	デマンドバス予約確認画面	● デマンドバスの予約を確認する画面	FN001 FN101 FN202 FN206 FN207
UI007	アカウント画面	● my route のアカウントに関わる情報を表示する画面	FN101 FN208 FN212 FN213 FN214 FN215
UI008	予約内容表示画面	● 購入・予約済みの交通手段を表示する画面	FN101 FN208 FN210 FN211 FN212 FN213
UI009	予約更新・取消画面	● デマンドバスの予約詳細を表示し、予約の更新・予約の削除ができる画面	FN101 FN208

デマンドバスシステム連携 API システム設計書

			FN209 FN212
UI010	ログイン画面	● my route にログインする画面	FN001

2-5-3. ユーザーインターフェースの詳細

ユーザーインターフェース（画面）の詳細を記す。なお、本業務において開発（新規・改修）を行うユーザーインターフェース（画面）を**朱文字**で示す。

1) 【HW001】スマートフォンの MaaS アプリ画面
 【UI001】メイン画面



図 2-28 メイン画面

- 本画面の概要
 - アプリインストール直後 or 起動直後のデフォルト画面
 - 下のタブから「地図で探す」、あるいは「お得におでかけ」に遷移することが可能
- 本画面から利用する機能
 - 【FN001】外部サービス経由のデマンドバス予約

【UI002】地図画面



図 2-29 地図画面

- 本画面の概要
 - MaaS アプリで表示できる情報を地図上に表示する画面
 - 拡大・縮小（ピンチアウト・ピンチイン可能）、ポイント選択、関連情報の表示が可能
- 本画面から利用する機能
 - 【FN001】外部サービス経由のデマンドバス予約

【UI003】乗降ポイント表示画面



図 2-30 乗降ポイント表示画面

- 本画面の概要
 - バス停やデマンド交通の乗降ポイントを確認する地図画面
 - 拡大・縮小、ポイント選択可能
- 本画面から利用する機能
 - 【FN001】外部サービス経由のデマンドバス予約
 - 【FN101】外部リクエスト処理
 - 【FN201】サービス一覧表示
 - 【FN204】乗降場所表示

【UI004】交通予約画面

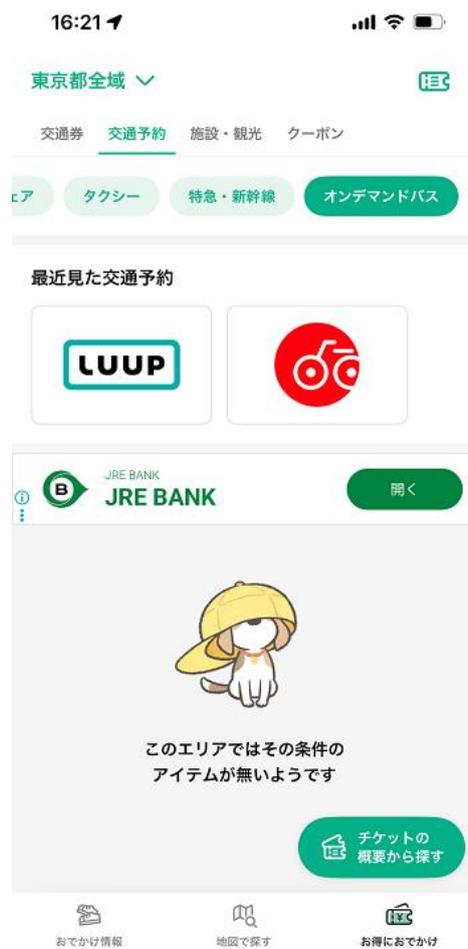


図 2-31 交通予約画面

- 本画面の概要
 - 各種交通手段の検索と予約をする交通予約画面
 - 地域と交通手段の切り替えが可能
- 本画面から利用する機能
 - 【FN001】外部サービス経由のデマンドバス予約
 - 【FN101】外部リクエスト処理
 - 【FN201】サービス一覧表示
 - 【FN203】交通手段選択

【UI005】 デマンドバス予約画面



<乗降場所選択>

<予約情報入力>

図 2-32 デマンドバス予約画面

- 本画面の概要
 - デマンドバスを予約する際の画面
 - 予約の詳細（乗降場所・乗降希望日時・乗車人数を最低限含む）などを表示が可能
- 本画面から利用する機能
 - 【FN001】 外部サービス経由のデマンドバス予約
 - 【FN101】 外部リクエスト処理
 - 【FN201】 サービス一覧表示
 - 【FN205】 予約候補検索

【UI006】デマンドバス予約確認画面



図 2-33 予約確認画面

- 本画面の概要
 - デマンド交通を予約する際の確認画面
 - 予約の詳細（乗降場所・乗降希望日時・乗車人数・決済状況を最低限含む）などを表示することができる
- 本画面から利用する機能
 - 【FN001】外部サービス経由のデマンドバス予約
 - 【FN101】外部リクエスト処理
 - 【FN202】ユーザー登録
 - 【FN206】予約登録
 - 【FN207】決済ステータス更新

【UI007】アカウント画面



図 2-34 アカウント画面

- 本画面の概要
 - my route のアカウントに関わる情報を表示する画面
- 本画面から利用する機能
 - 【FN101】外部リクエスト処理
 - 【FN208】予約表示
 - 【FN212】予約履歴確認
 - 【FN213】ユーザー照会
 - 【FN214】ユーザー情報更新
 - 【FN215】ユーザー退会

【UI008】予約内容表示画面



図 2-35 予約内容表示画面

- 本画面の概要
 - MaaS アプリで予約したデマンドバスの予約内容表示画面（乗車実績含む）
- 本画面から利用する機能
 - 【FN101】外部リクエスト処理
 - 【FN208】予約表示
 - 【FN210】運行情報確認
 - 【FN211】運賃支払い確認
 - 【FN212】予約履歴確認
 - 【FN213】ユーザー照会

【UI009】予約更新・取消画面



図 2-36 デマンドバス予約更新画面

- 本画面の概要
 - デマンドバスの予約詳細を表示し、予約の更新・予約の削除ができる画面
- 本画面から利用する機能
 - 【FN101】外部リクエスト処理
 - 【FN208】予約表示
 - 【FN209】予約更新
 - 【FN212】予約履歴確認

【UI010】 ログイン画面



図 2-37 ログイン画面

- 本画面の概要
 - MaaS アプリにログインするための画面
- 本画面から利用する機能
 - 【FN001】 外部サービス経由のデマンドバス予約

3. デマンドバス連携 API システム：非機能要件（NF）

3-1. 非機能要件一覧

表 3-1 非機能要件一覧

カテゴリ	ID	非機能項目	要件詳細
性能	NF001	レスポンス速度	今回開発する API については、レスポンス速度を「1s/req (p99) *」とする *レイテンシの 99 パーセンタイル値が 1 秒以下

3-2. 非機能要件の詳細

【NF001】レスポンス速度

- 本非機能要件の概要
 - 今回開発する API については、レスポンス速度を「1s/req (p99) *」とする
※レイテンシの 99 パーセンタイル値が 1 秒以下
- 設定理由
 - 本プロジェクトは、商用利用はせず、あくまでも検証レベルのため、厳密な要件は求めない
 - また、E2E のレスポンス速度は、標準 API が接続するデマンドバス API の性能要件に依存するため、あくまでも標準 API の処理時間のみを測定範囲とする

4. 実証調査に利用するデータ (DT)

4-1. 実証調査に利用するデータ一覧

表 4-1 実証調査に利用するダミーデータ一覧

※朱文字：本実証で変換・作成するデータ

ID	データ名称	データ形式	出所	データを利用する ID
-	-	-	-	-

4-2. 実証調査に利用するデータの詳細

-

5. 用語集

用語	定義・説明
Amazon EKS	Amazon EKS (Elastic Kubernetes Service) は、AWS (Amazon Web Services) 上で Kubernetes を簡単に実行できるマネージドサービスです。Kubernetes は、コンテナ化されたアプリケーションを自動的にデプロイ、スケーリング、管理するためのオープンソースのシステム。
インターオペラビリティ	異なるシステムやサービス間で連携・相互運用できる能力。
API	Application Programming Interface の略。システム間の機能やデータ連携を実現するためのインターフェース。
JWT	JSON Web Token。認証・認可情報をやり取りするためのトークン。
GeoJSON	地理空間データを表現するための JSON フォーマット。
my route	トヨタファイナンシャルサービス社が提供する MaaS アプリ。今回の検証でテストアプリを利用。
MONET Move	MONET Technologies 社が提供しているデマンドバスシステムの名称。



**デマンドバスシステム連携API システム設計書
Ver1.0**

発行日: 2026年3月

委託者: 国土交通省 総合政策局

モビリティサービス推進課

受託者: MONET Technologies株式会社