

# API実習 2025 課題レポート（第4回）

API 実習課題： API提供者として、API設計・実装するためのAPI要件定義の策定

本課題（第4回レポート）は、**API提供者の立場で「API設計・実装」に必要な要件定義を策定すること**を目的とする

これは、\*\*第5回（最終課題）\*\*で実装する API の基礎文書という位置づけである。

これまでの「API利用者視点」を踏まえ、**自ら設計・実装可能なレベルの要件定義**を作成する。

- これまででは、主にAPIを理解して、APIを利用する視点が中心であった
- シラバスの学習目標として、API利用者視点に限らず、API提供者視点として、自らAPIを設計・実装出来ることが、  
単位獲得に不可欠である
- そこで、次回第5回課題レポート（最終課題）にて、自ら設計開発した、API設計・実装に必要な、**要件定義**を綿密に策定する

**出題範囲： API実習2025 第10回～第12回まで**

対象授業回：API実習 2025 第10回～第12回の内容

**提出期限： 2026/1/27(火) 17:00**

**期待するアウトカム**

- 業務背景からAPIの必要性を論理的に説明できる
- ユースケースに基づく**機能要件**と、運用・品質を支える**非機能要件**を具体化できる
- API資源（リソース）、エンドポイント、データモデル、認証・認可、エラーハンドリング、バージョニング等を一貫性ある形で記述できる
- 実装可能性（第5回への接続）を担保する文書として成立している

**課題： API提供者として、API設計・実装するためのAPI要件定義の策定**

**課題（提出物）**

以下の構成で\*\*要件定義書（Markdown）\*\*を作成すること。

図は **Mermaid** または別ファイル（PNG/SVG）添付可能。

## 1. 背景・業務領域の具体化

- 対象ドメイン（例：教育、EC、物流、医療、観光など）

- ・現状業務の流れ／課題の所在／制約条件（法規、組織、コスト、期間など）

## 2. 課題・目的・APIが必要な理由

- ・解決すべき業務課題
- ・APIの役割（社内・社外連携、モバイル／Webクライアントのバックエンド等）
- ・KPIまたは成功指標（例：処理時間短縮、エラー率低下、運用負荷削減）

## 3. 対象ユーザ／ステークホルダ

- ・主要利用者（ペルソナ）
- ・事業者、運用者、データ提供者／外部パートナー
- ・権限レベル（管理者／一般ユーザ／外部アプリなど）

## 4. 業務フロー・ユースケース図

- ・主要ユースケース（最低3件以上）
- ・フロー図／ユースケース図（Mermaid可）

## 5. 関連技術・先行事例

- ・選定予定の技術（REST/GraphQL、DB、認証基盤、APIゲートウェイなど）
- ・参考になる外部事例や標準（OpenAPI、HTTP標準、OAuth2/OIDC等）

## 6. 提案技術の比較

- ・候補技術の比較（例：REST vs GraphQL、RDB vs NoSQL、セッション方式等）
- ・選定理由（利点／欠点／トレードオフ）

## 7. 機能要件

- ・**データ（モデル）要件**：主要リソース、属性、整合性ルール、ER図
- ・**API機能リスト**：CRUD、検索、集計、通知、バッチ等
- ・**エンドポイント仕様**（HTTPメソッド、URI、リクエスト／レスポンス、ステータスコード、エラー仕様）
- ・認証・認可、レート制限、バージョニング、エラーハンドリング方針
- ・運用要件（監視、ログ、メトリクス、バックアップ、秘密情報管理）
- ・**外部サービス一覧**（例：IDaaS、メッセージング、支払い、ストレージ）
- ・**システム構成図**（クライアント／API／DB／外部連携）
- ・**画面要件（View/UI/UX）**：API前提の画面遷移・入力／出力項目（モック可）

## 8. 非機能要件

- ・動作環境（言語、フレームワーク、DB、クラウド／オンプレ）
- ・性能要件（目標スループット、レイテンシ、同時接続、ピーク時の設計）
- ・セキュリティ要件（認証・認可、暗号化、脆弱性対策、秘密情報取り扱い、監査ログ）
- ・可用性（SLA目標、冗長化、障害対応手順）
- ・拡張性（スケール戦略、水平／垂直、モジュール構成）
- ・保守性（テスト方針、API整合性ルール、デプロイ戦略、ドキュメント運用）

## 9. まとめ

- ・要件の妥当性、実装への影響、リスクと対応（最低3件）
- ・第5回最終課題での実装範囲（優先度P1/P2）

## 10. 付録（任意加点）

- ・OpenAPI（YAML/JSON）ドラフト

- 代表ユースケースのシーケンス図
- テスト観点一覧 (API契約テスト、負荷、セキュリティ)

## 提出形式

- 1ファイル : `reports-04.md` (回答用様式)
- 画像類 : `/figs/*.png` (図版はリンクで参照)
- 参考文献 : 文末に記載 (Web/書籍、標準仕様など)
- 提出方法 : Githubプライベートリポジトリにアップロード

## 採点基準 (100点)

- 背景・業務具体化 (10)
- 課題・目的・必要性 (10)
- ユーザ／ステークホルダ (8)
- 業務フロー・ユースケース (12)
- 関連技術・先行事例 (8)
- 技術比較／選定理由 (10)
- 機能要件 (**エンドポイントの明確さ・一貫性**) (18)
- 非機能要件 (**測定可能な指標・現実性**) (18)
- まとめ (4)
- 付録 (最大+2点の加点)

### 減点例 :

用語不整合、URI/メソッド設計とユースケースの齟齬、非機能要件が抽象的 (数値目標なし)、セキュリティ未記載、図がない／読めない、参考文献なし等。

## 記述チェックリスト (提出前)

- 背景・業務領域の具体化
- 課題／目的／APIが必要な理由
- 対象ユーザ・ステークホルダ
- 業務フロー／ユースケース図
- 関連技術／先行事例
- 提案技術の比較 (選定理由)
- 機能要件 (データ、機能、運用、外部サービス、構成図、画面)
- 非機能要件 (環境、性能、セキュリティ、可用性、拡張性、保守性)
- まとめ (リスク・優先度)

□ 参考文献の明記