OpenAPI形式で外部設計のAPI設計を納品するためには、以下のステップに従うとよいでしょう。

**1. API要件の整理**

* 提供するAPIの目的や機能、ユーザーがどのように利用するかを明確にします。  
  例: 認証が必要か、GET/POSTなどのメソッド、レスポンス形式、エラーハンドリングなど。

**2. OpenAPIの仕様に従う**

OpenAPI仕様（以前はSwaggerとして知られていた）は、APIの定義を標準的なフォーマットで記述するためのものです。  
OpenAPIの主な要素：

* **Paths**: APIのエンドポイントと、それに対応するHTTPメソッド（GET, POST, PUT, DELETEなど）
* **Components**: 再利用可能なデータモデル（schemas）、パラメータ、リクエスト/レスポンス定義
* **Security**: 認証に関する設定（例: OAuth2, APIキー）
* **Tags**: APIのエンドポイントを論理的にグループ化する
* **Info**: API全体の説明（バージョン、タイトル、ライセンスなど）

**3. API設計をOpenAPI形式で記述**

OpenAPIはYAMLまたはJSON形式で記述されます。これには手動で書くか、以下のようなツールを使うと便利です。

* **Swagger Editor**: ブラウザで動作し、リアルタイムでOpenAPI仕様を編集できる
* **Postman**: APIテストツールとしても使え、OpenAPI仕様の作成にも対応
* **Stoplight**: インタラクティブにOpenAPIドキュメントを作成するツール

**4. ドキュメントの検証**

書いたAPI設計がOpenAPIの仕様に適合しているか、検証ツールでチェックします。

* **Swagger Editor**: エラーがあればその場で指摘される
* **OpenAPI Generator**: OpenAPI仕様を基に、クライアントやサーバーのコードを自動生成

**5. APIドキュメントの生成**

OpenAPIの仕様書はそのままAPIドキュメントとして使用できます。これをクライアントやチームに共有します。

* **Swagger UI**: OpenAPIドキュメントから視覚的にわかりやすいインターフェースを自動生成してくれるツール
* **Redoc**: 美しくフォーマットされたAPIドキュメントを生成するツール

**6. 納品準備**

* 完成したOpenAPIファイル（.yamlまたは.json形式）
* 生成されたドキュメント（HTML形式などで出力可能）
* 必要に応じて、APIの利用方法やサンプルリクエスト/レスポンスの説明を追加

**サンプルの基本構造 (YAML形式)**

yaml

複製程式碼

openapi: 3.0.0

info:

title: Sample API

description: This is a sample API.

version: 1.0.0

servers:

- url: https://api.example.com/v1

paths:

/users:

get:

summary: Get a list of users

responses:

'200':

description: A JSON array of user objects

content:

application/json:

schema:

type: array

items:

$ref: '#/components/schemas/User'

components:

schemas:

User:

type: object

properties:

id:

type: integer

example: 1

name:

type: string

example: John Doe

**7. 納品形式**

依頼主の仕様に従い、以下のような形式で納品します。

* YAMLファイルまたはJSONファイル
* APIドキュメントとしてHTMLまたはPDF形式
* 必要に応じてコードやテスト環境も含める

これにより、APIの設計書を標準的な形式で納品することができます。