



Naver Commerce SDK 제작기

Naver Commerce API를 위한 Python SDK

이승헌

PyThing 2026.02.07

배경 및 동기

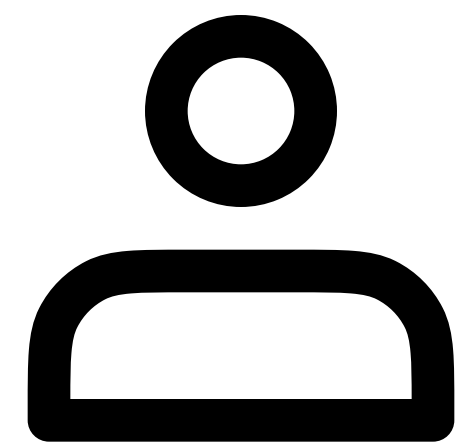
오직 한 사람만을 위한 셀프 투어 가이드

Travolo



잘 팔기 위해 선택한 네이버 스마트스토어

배경 및 동기



구매 발생



Webhook



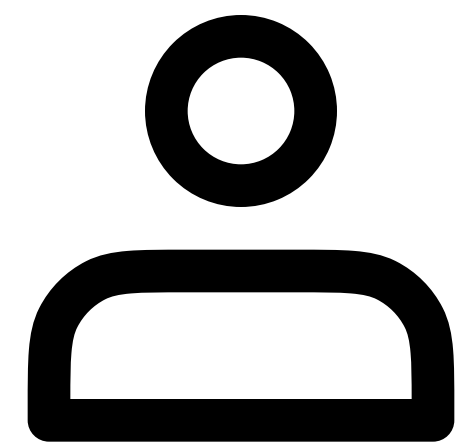
오직 한 사람만을 위한 셀프 투어 가이드



코드 발송 (mail, kakao notification)



배경 및 동기



구매 발생



Webhook



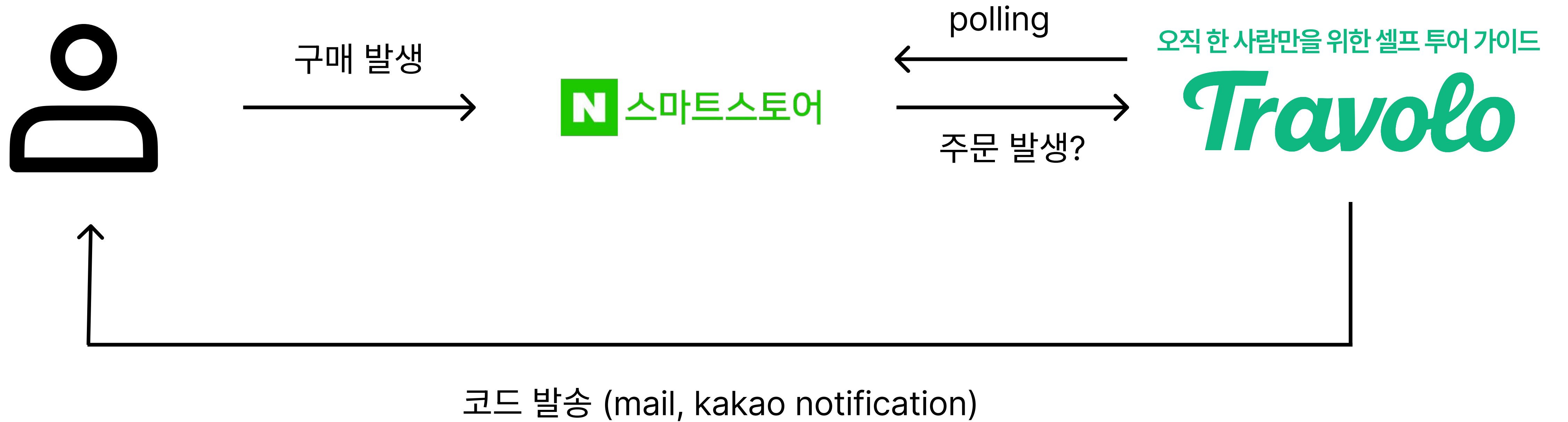
오직 한 사람만을 위한 셀프 투어 가이드



코드 발송 (mail, kakao notification)



배경 및 동기




배경 및 동기

The screenshot shows the Naver Commerce API documentation page. The top navigation bar includes the Naver logo, '커머스API docs', and links for '소개', '커머스API', '변경 이력', and '솔루션 가이드'. A search bar and version information '2.71.0 버전 선택' are also present. The left sidebar lists categories: '인증', 'API데이터솔루션(통계)', '문의', '상품', '정산', '주문', '커머스솔루션', and '판매자정보'. The main content area is titled '커머스API' and lists '최신 커머스API 문서' with eight category cards: '인증 (1 item)', 'API데이터솔루션(통계) (5 items)', '문의 (3 items)', '상품 (21 items)', '정산 (2 items)', '주문 (5 items)', '커머스솔루션 (4 items)', and '판매자정보 (4 items)'. A 'Next OAuth 2.0' link is visible at the bottom right of the main content area. The URL at the bottom is 'https://apicenter.commerce.naver.com/docs/commerce-api/current/그룹상품'.

배경 및 동기

⏪ [🏠](#) > [커머스API](#) > [주문](#) > [주문 조회](#) > [조건형 상품 주문 상세 내역 조회](#)

조건형 상품 주문 상세 내역 조회

GET /v1/pay-order/seller/product-orders 

조건에 맞는 상품 주문에 대한 상세 내역을 조회합니다.

Request

▼ QUERY PARAMETERS

- from** string<date-time> **REQUIRED**
조회 기준의 시작 일시(inclusive)
Example: 2024-06-07T19:00:00.000+09:00
- to** string<date-time>
조회 기준의 종료 일시(inclusive). 생략 시 from으로부터 24시간 후로 자동 지정됩니다.
Example: 2024-06-08T19:00:00.000+09:00
- rangeType** rangeType.pay-order-seller (string)
조회 기준 유형. 생략 시 PAYED_DATETIME(결제 일시)으로 자동 지정됩니다. 250바이트 내외

배경 및 동기

오직 한 사람만을 위한 셀프 투어 가이드

Travolo

Jaewan Kim 🙌 오후 5:15
매출 20만원 & 찐 오가닉 유료 구매자 25명 달성 ~

image.png ▼

수량	판매 매출	수수료	정산 금액	구매자명	네이버 ID	수취인명
26	206,400	12,389	194,011	실제 사용자 37명 / 유료 구매자 25명		

👍 1 🗨️

너.. 왜 .. 팔려?

배경 및 동기

The screenshot shows an API documentation page for a shipping endpoint. The breadcrumb navigation is: < < Home > > 커머스API > > 주문 > > 발주/발송 처리 > > 발송 처리. The endpoint is a POST request to /v1/pay-order/seller/product-orders/dispatch. The content type is APPLICATION/JSON. The request body is required and contains a DispatchProductOrder object. The object has a dispatchProductOrders array of product orders and a deliveryMethod string. An example product order ID is 2022040521691281.

<< < Home > > 커머스API > > 주문 > > 발주/발송 처리 > > 발송 처리

발송 처리

POST /v1/pay-order/seller/product-orders/dispatch

단수 또는 복수 개 상품 주문을 발송 처리합니다. 요청 가능한 상품 주문 번호는 최대 30개입니다.

Request

APPLICATION/JSON

▼ BODY **REQUIRED**

DispatchProductOrder 배열

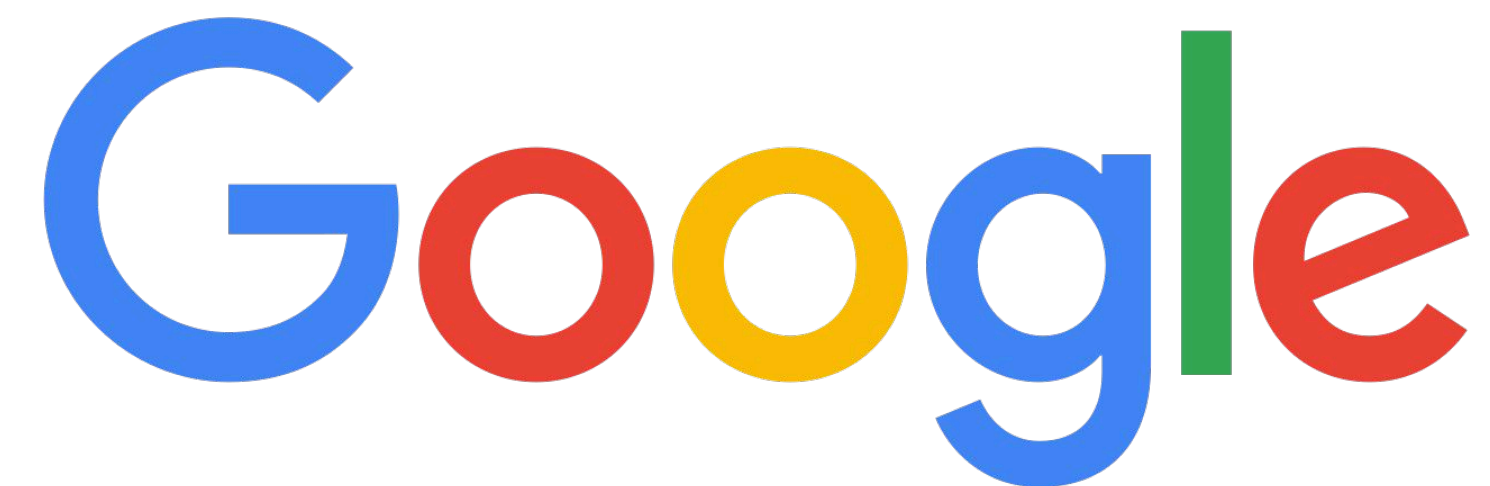
- ▼ **dispatchProductOrders** dispatchProductOrder.pay-order-seller (object [])
 - Array [
 - productId** string
상품 주문 번호
Example: 2022040521691281
 - deliveryMethod** deliveryMethod.pay-order-seller (string)
배송 방법 코드. 250바이트 내외

SDK를 한 번 만들어볼까?

핵심 설계 결정



openai sdk



google genai sdk

핵심 설계 결정



Stainless

Best-in-class developer interfaces for your API

핵심 설계 결정

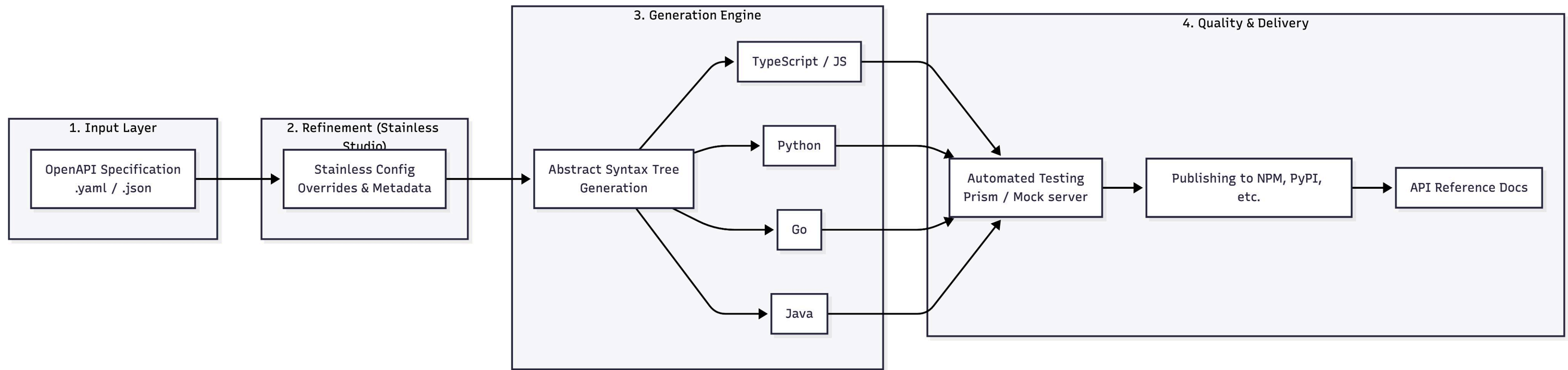
The screenshot shows the Stainless website landing page. At the top, there is a navigation bar with the Stainless logo, links for Products, Customers, Pricing, and Resources, and buttons for 'Ask an expert' and 'Sign in'. A 'New' badge highlights 'Stainless Docs Platform - Early Access'. The main headline reads 'Best-in-class developer interfaces for your API'. Below this, a paragraph states: 'A great developer experience fuels API adoption, reduces support burden, and helps integrators ship faster.' Another paragraph mentions: 'Built from decades of experience at API companies like Stripe, Heroku, Meta, Twilio, and more, Stainless helps you deliver robust and idiomatic SDKs, documentation that keeps up with your API, and token-efficient MCP servers.' A 'Get started' button is positioned below the text. On the right side, there is a code editor showing TypeScript code for interacting with an OpenAI API. The code includes imports for OpenAI, zodTextFormat, and zod, followed by the instantiation of OpenAI, the creation of a zodTextFormat object, and the use of the OpenAI client to parse a response. The code is as follows:

```
import OpenAI from "openai";
import { zodTextFormat } from "openai/helpers/zod";
import { z } from "zod";

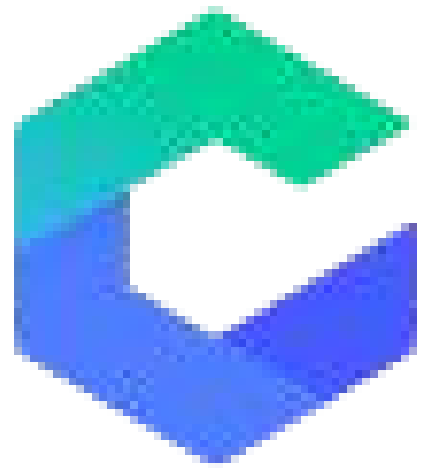
const openai = new OpenAI();
const format = zodTextFormat(
  z.object({ namesOfParticipants: z.array(z.string()) }),
  "namesOfParticipants"
);
const response = await openai.responses.parse({
  model: "gpt-5",
  instructions: "Extract the names of the participants.",
  input: "Alice and Bob are going to a science fair on Friday.",
  text: { format },
});
console.log(response.output_parsed);
```

Below the code editor, the text 'Trusted by top API companies' is followed by logos for ANTHROPIC, CLOUDFLARE, Google, and OpenAI. At the bottom, there is a row of logos for RONOME, MODERN TREASURY, MUX, ONA, perplexity, Profound, Replicate, RIPPLING, and rui. The final row of logos includes scorecard, SULLY.AI, supermemory, turbopuffer, Whop, airtop, and Anchor.

핵심 설계 결정



핵심 설계 결정



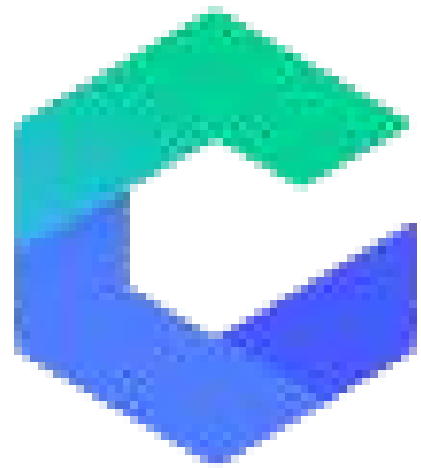
OAS 미비

1. 공식문서에서 OAS를 추출 → Stainless 사용

or

2. from scratch 개발

핵심 설계 결정



OAS 미비

1. 공식문서에서 OAS를 추출 → Stainless 사용

or

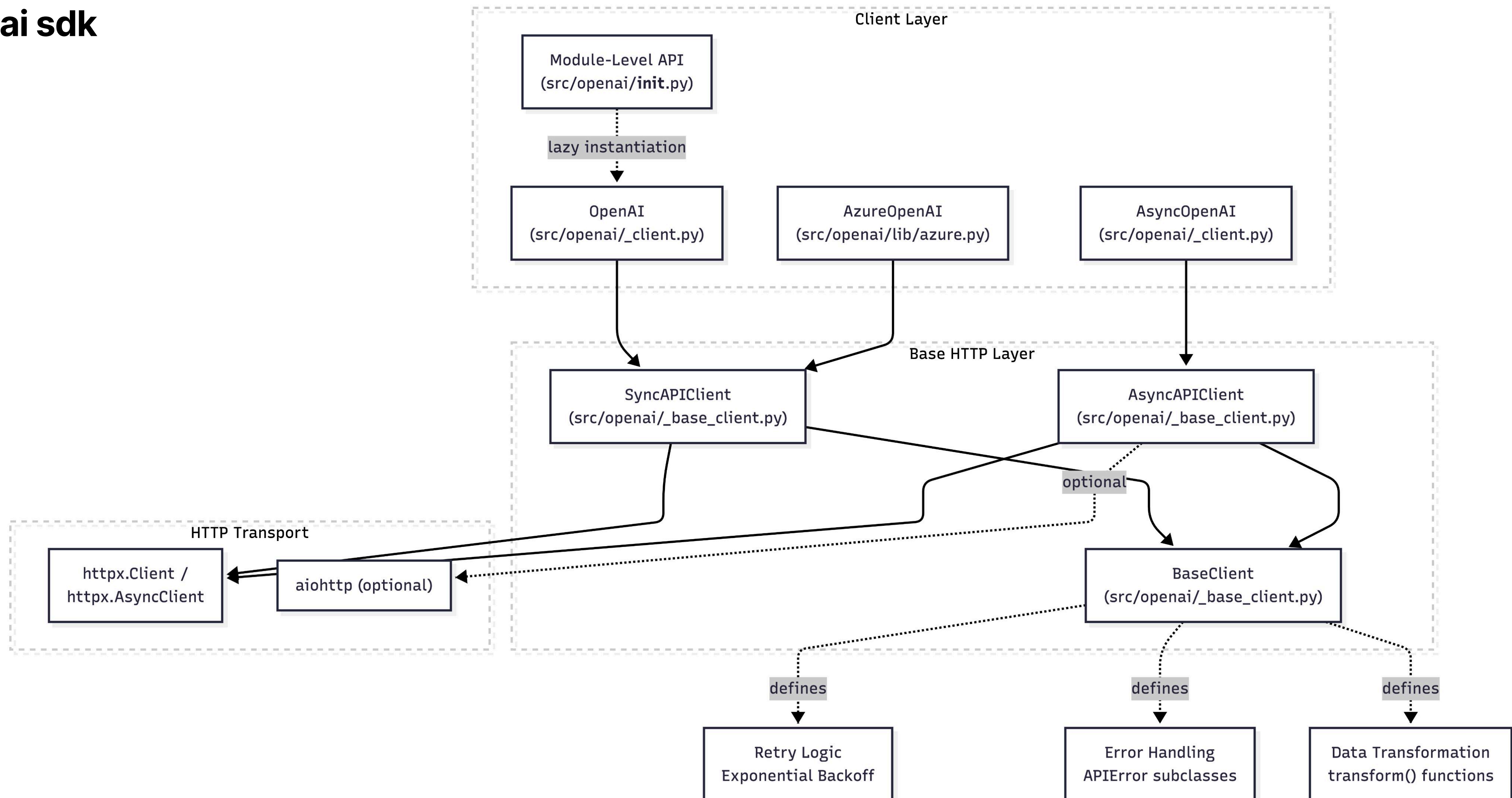
2. from scratch 개발

배경 및 동기



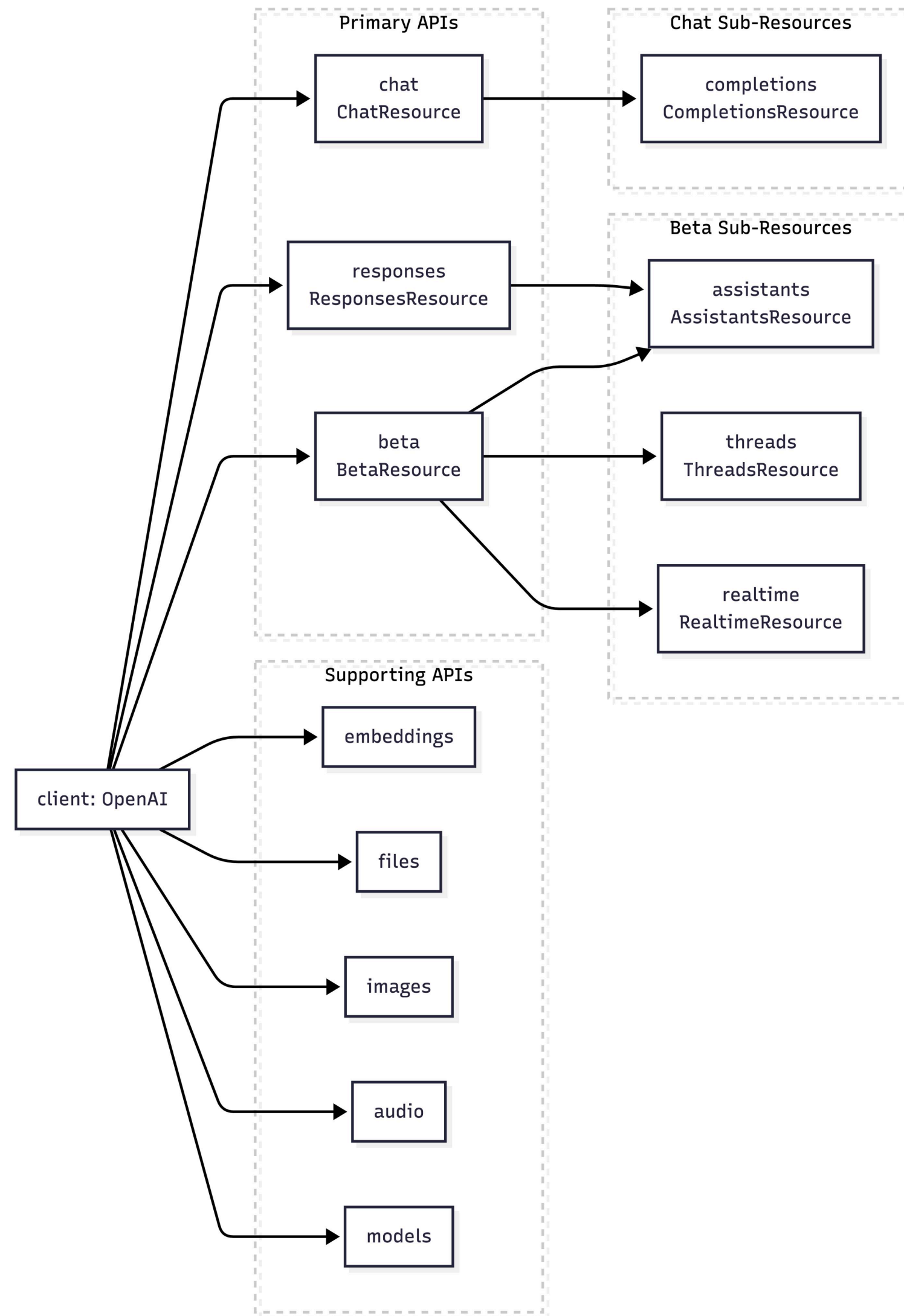
아키텍처 구조

openai sdk



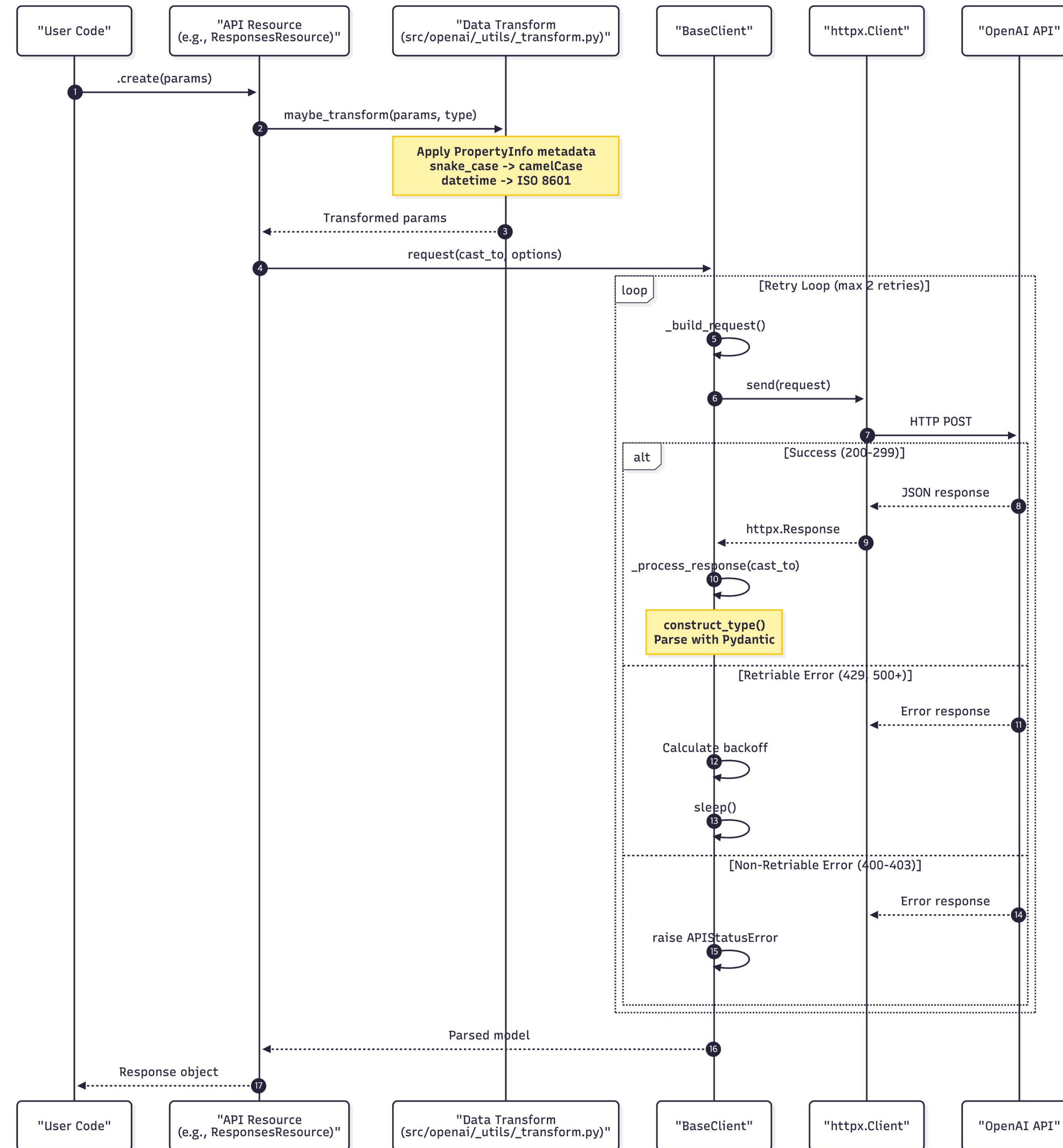
아키텍처 구조

openai sdk



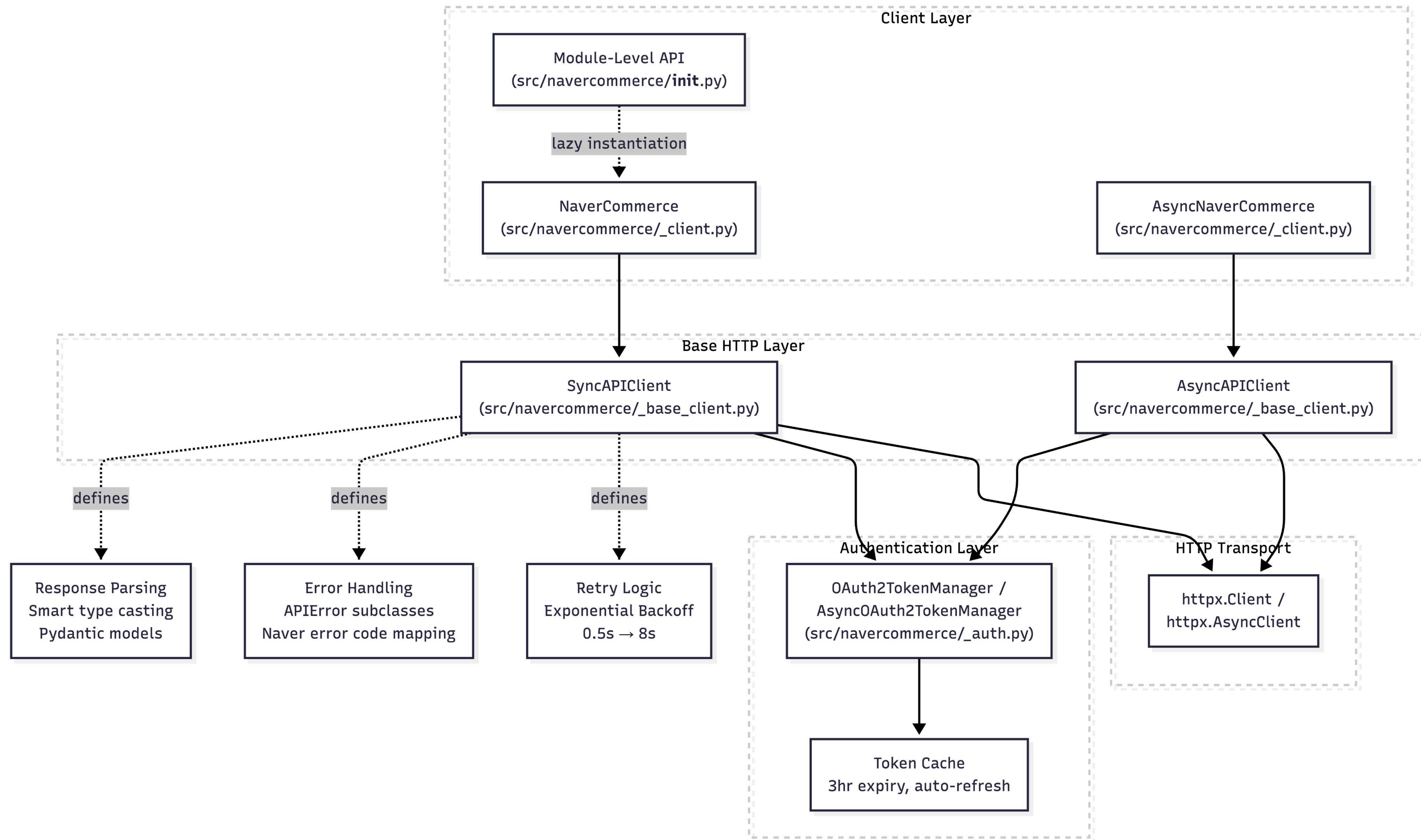
아키텍처 구조

openai sdk



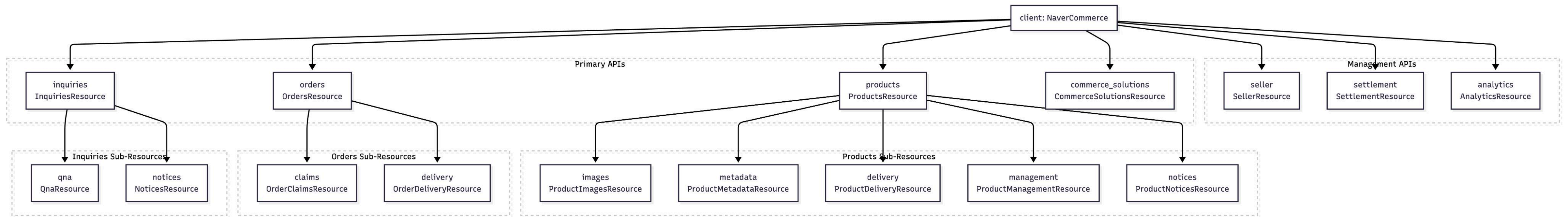
아키텍처 구조

navercommerce sdk



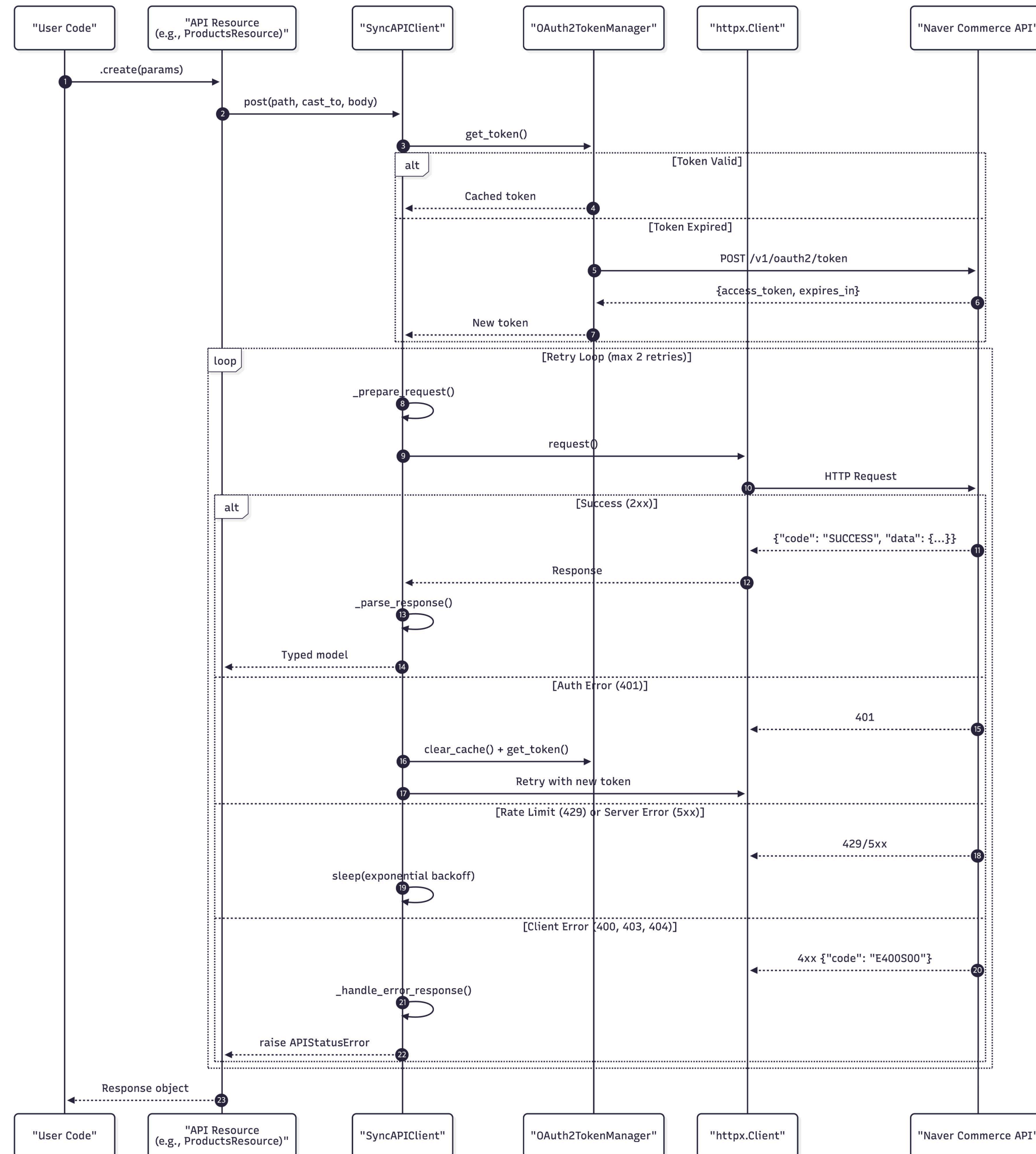
아키텍처 구조

navercommerce sdk



아키텍처 구조

navercommerce sdk



pydantic을 활용한 타입 안정성

```
1 class ShippingInfo(BaseModel):
2     """
3     Shipping information for an order.
4
5     Attributes:
6         shipping_company: Shipping company name
7         tracking_number: Tracking/invoice number
8         shipping_date: Date when item was shipped
9         delivery_method: Delivery method (DELIVERY, VISIT_RECEIPT, etc.)
10    """
11
12    shipping_company: str | None = Field(None, alias="shippingCompany")
13    tracking_number: str | None = Field(None, alias="trackingNumber")
14    shipping_date: str | None = Field(None, alias="shippingDate")
15    delivery_method: str | None = Field(None, alias="deliveryMethod")
16
17
18 class OrdererInfo(BaseModel):
19     """Information about the person who placed the order."""
20
21     name: str | None = None
22     tel: str | None = None
23     mobile: str | None = None
24     email: str | None = None
```

Sphinx를 통한 패키지 문서 생성

Naver Commerce SDK

Q Search

DOCUMENTATION:

- Getting Started
- User Guide
- Resource Guides
- Examples
- API Reference
- Advanced Topics

Naver Commerce SDK for Python

A production-grade Python SDK for the Naver Commerce API, providing type-safe, developer-friendly access to Naver's e-commerce platform with both synchronous and asynchronous support.

```
from navercommerce import NaverCommerce

client = NaverCommerce(
    client_id="your_client_id",
    client_secret="your_client_secret"
)

# Get seller information
account = client.seller.account()
print(f"Seller: {account.seller_name}")

# List products
products = client.products.list()
for product in products:
    print(f"{product.name}: {product.sale_price}원")
```

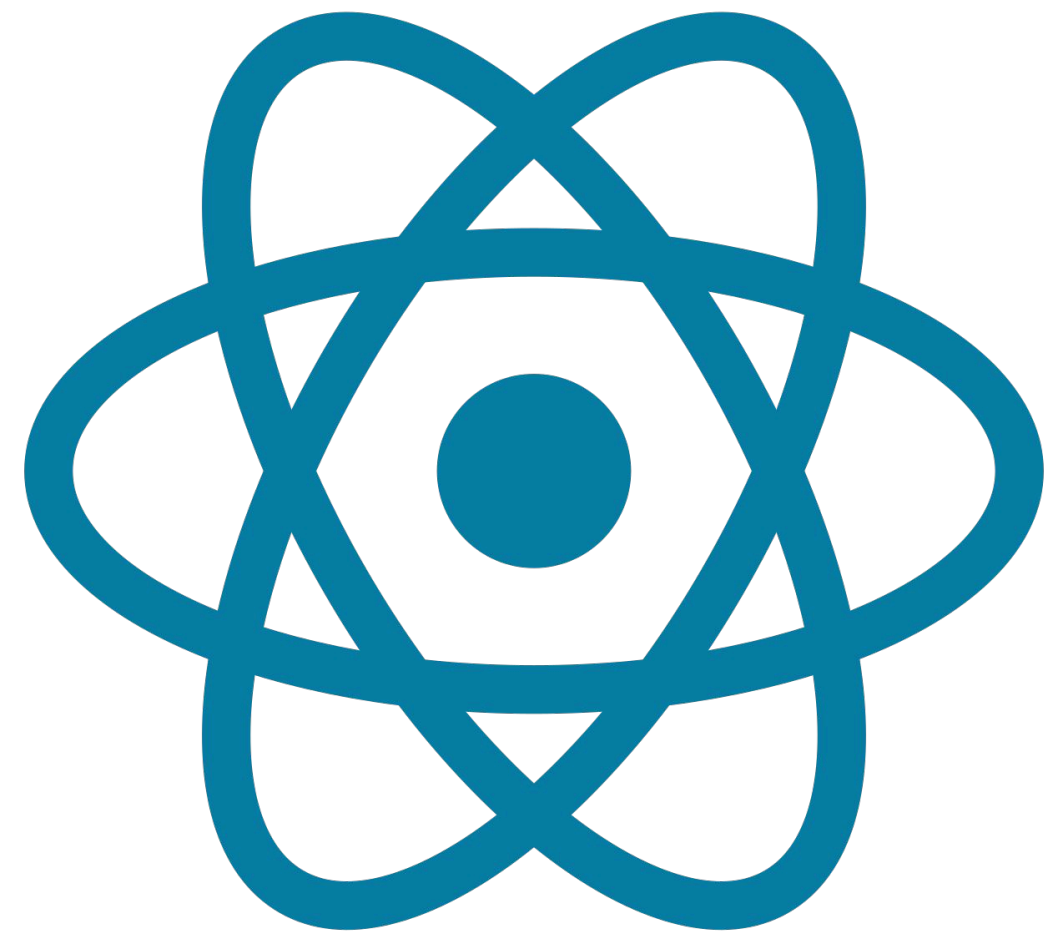
Features

- OAuth 2.0 Authentication**
Automatic token management with refresh
- Sync & Async Support**
Use synchronous or asynchronous clients based on your needs
- Type-Safe**
Full type hints and Pydantic models for IDE autocomplete and validation
- Auto-Retry**
Exponential backoff retry logic for transient failures
- Resource-Based**
Clean, intuitive API organized by resource types
- Comprehensive Error Handling**
Detailed exceptions mapped to Naver error codes

ON THIS PAGE

- Features
- Quick Links
- Documentation Contents
- Installation
- Quick Start
 - Synchronous Usage
 - Asynchronous Usage
- Available Resources
- Error Handling
- Configuration
 - Environment Variables
- Project Status
- Contributing
- License
- Links
- Indices and Tables

Next Step



<https://github.com/snghnl/navercommerce>

E.O.D