Ecco la sequenza completa di quello che abbiamo fatto, a partire da **AppConfig** e **StartISBN**, fino al refactoring finale con i due service e il client “snello”:

1. **Configurazione di base**
   * Creato AppConfig con un bean RestTemplate per le chiamate HTTP.
2. **Avvio automatico della popolazione ISBN**
   * Implementato StartISBN come ApplicationListener<ApplicationReadyEvent> (o ApplicationRunner) che, all’avvio dell’applicazione, chiama il servizio per popolare gli ISBN mancanti.
3. **Entity Libro**
   * Aggiunta della proprietà isbn (tipo String) e del LocalDate annoUscita mappato su colonna YEAR in database.
   * Evitato di toccare gli altri campi e generato getter/setter via Lombok.
4. **Repository JPA**
   * Creato LibroRepos con metodo custom List<Libro> findLibriSenzaIsbn(), che restituisce tutti i libri con isbn null o vuoto.
5. **Service di dominio**
   * LibroServiceImpl inietta LibroRepos e OpenLibraryClient (poi refattorizzato in OpenLibrarySearchService + OpenLibraryVolumesService), cicla sui libri senza ISBN e li aggiorna con il primo codice restituito.
6. **DTO per il Search API**
   * OpenLibrarySearchResponse con classe interna Doc, mappata su JSON di /search.json (campi title, author\_name, isbn, cover\_edition\_key, key).
7. **DTO per il Volumes Read API**
   * OpenLibraryReadVolumesResponse con Map<String, Record> records, e classi interne Record → RecordData → Identifiers (campi isbn\_10 e isbn\_13).
8. **Service di integrazione #1: Search API**
   * Interfaccia OpenLibrarySearchService con metodo SearchResult search(String title, String author).
   * Implementazione OpenLibrarySearchServiceImpl che chiama /search.json?title={title}&author={author}&limit=10, raccoglie in una SearchResult (liste di isbns e di olids).
9. **Service di integrazione #2: Volumes Read API**
   * Interfaccia OpenLibraryVolumesService con metodo Optional<List<String>> fetchIsbns(String olid).
   * Implementazione OpenLibraryVolumesServiceImpl che chiama /api/volumes/brief/olid/{olid}.json e restituisce gli ISBN trovati.
10. **Client snello di orchestrazione**
    * OpenLibraryClient (nel package integration), che inietta i due service sopra:

**Quando usare @Transient**

* **Se** vuoi soltanto mostrare la copertina nelle view a partire dall’ISBN, senza mai fare query basate sulla URL.
* **Se** preferisci mantenere la logica “stateless” e non appesantire il DB.
* **Se** non ti interessa salvare nulla lato server.

**Quando persistere in DB**

* **Se** in futuro devi fare report o filtri basati sulla presenza di una copertina.
* **Se** vuoi evitare di ricostruire la stringa anche a fronte di trafﬁco elevato (anche se l’impatto è trascurabile).