

Uitleg in het Nederlands

Hotelbeds Cache API — Checklist (Samenvatting in het Nederlands)

Dit document geeft een uitgebreide checklist voor het ontwerp, de implementatie en het beheer van de Hotelbeds Cache API.

1) Scope & gebruik:

- De cache is de belangrijkste bron, live API kan gebruikt worden voor validatie of fallback.
- v1 = alleen validatie (cache gebruiken, batchgewijs verschillen loggen).
- v2 = optioneel live fallback bij afwijkingen.

2) Architectuur (hoog niveau):

- Ingest: data ophalen, uitpakken, parsen, laden in staging, upserten in kernschema.
- Precompute: bereken vooraf “vanaf € p.p.” prijzen en promoties om resultaten snel te serveren.
- API laag: Redis cache■aside patroon voor /search, /matrix, /static.

3) Database (MySQL):

- Tabellen voor hotels, kamers, rate plans, prijzen, beschikbaarheid, promoties, valutakoersen.
- Derived tables: cheapest_pp, search_index voor snelle queries.

4) Synchronisatieproces:

- Elke 60 min (korter in hoogseizoen).
- ZIP downloaden → verifiëren → parsen → staging → upsert.
- Delta detectie via hash of lastModified.

5) Redis caching:

- Duidelijke keystructuur (chp, mx, st) met TTL's en SWR (stale■while■revalidate).
- Cache■busting via versies of targeted deletes.

6) API's & filters:

- /search: lijst hotels met goedkoopste prijs per persoon.
- /hotels/{id}/matrix: detail met kamers, tarieven, promo's, policies.
- /hotels/static: statische info (beschrijving, foto's, voorzieningen).

7) Business/prijslagen:

- Stop■sell, allotment, CTA/CTD, min/max nachten.
- Promoties in vaste volgorde: gratis nachten → % korting → vaste korting → coupons.
- Valuta en policies correct toepassen.

8) Doel “Vanaf € p.p.”:

- Algoritme berekent goedkoopste periode en slaat op in cheapest_pp.

9) Fallback in zoekresultaten:

- Toon alternatief: andere kamer of volgende beschikbare datum.

10) Betrouwbaarheid & performance:

- Idempotentie, concurrency locks, timeouts, retries.
- Doel: snelle responsijden (/search ≤ 500ms met cache hit).

11) Beveiliging & GDPR:

- API key + IP■allowlist (v1), OAuth2 in v2.
- Geheimen via secret manager, rotatie elke 90 dagen.
- Logging en monitoring voor fouten en prestaties.

12) CI/CD & testen:

- OpenAPI als bron, automatische contracten en E2E-tests.
- Zero-downtime deploys, rollback plan.

13) Frontend compatibiliteit:

- API responses ontwerpen om eenvoudig in de UI te integreren.

Explanation in English

Hotelbeds Cache API — Checklist (Summary in English)

This document provides a comprehensive checklist for the design, implementation, and operation of the Hotelbeds Cache API.

1) Scope & use cases:

- Cache is the main source of truth, live API can be used for validation or fallback.
- v1 = validation-only (use cache, log batch deviations).
- v2 = optional live fallback if deviations are large.

2) Architecture (high-level):

- Ingest: download, unzip, parse, load into staging, upsert into core schema.
- Precompute: precalculate “From € p.p.” prices and promotions for fast results.
- API layer: Redis cache-aside pattern for /search, /matrix, /static.

3) Database (MySQL):

- Tables for hotels, rooms, rate plans, prices, availability, promotions, FX rates.
- Derived tables: cheapest_pp, search_index for optimized queries.

4) Sync process:

- Every 60 min (shorter in high season).
- Pipeline: download ZIP → verify → parse → staging → upsert.
- Delta detection via hash or lastModified.

5) Redis caching:

- Clear key design (chp, mx, st) with TTLs and SWR (stale-while-revalidate).
- Cache busting via version prefix or targeted deletes.

6) APIs & filters:

- /search: cheapest price per hotel/person.
- /hotels/{id}/matrix: detail view with rooms, rates, promos, policies.
- /hotels/static: static hotel data (descriptions, photos, amenities).

7) Business/pricing logic:

- Stop-sell, allotment, CTA/CTD, min/max nights.
- Promotions applied in fixed order: free nights → % discount → fixed discount → coupons.
- Proper handling of currency and policies.

8) Objective “From € p.p.”:

- Algorithm computes cheapest period and stores in cheapest_pp.

9) Search fallback logic:

- Show alternatives: other rooms or next available date.

10) Reliability & performance:

- Idempotency, concurrency locks, timeouts, retries.
- Targets: /search \leq 500ms (cache hit).

11) Security & GDPR:

- API key + IP allowlist (v1), OAuth2 in v2.
- Secrets via secret manager, rotate every 90 days.
- Logging and monitoring for errors and performance.

12) CI/CD & testing:

- OpenAPI as source of truth, automated contract and E2E tests.
- Zero-downtime deployments with rollback plan.

13) Frontend compatibility:

- API responses designed for easy integration in the UI.