Precompute Job for E.P.P Prices in Hotelbeds Cache API Integration

What is E.P.P?

E.P.P stands for End-user Payable Price. It is the final selling price to the traveler after base rates, mandatory fees, taxes, promotions/discounts, and currency conversion are applied. The goal is to precompute this price so that search results can display immediate, accurate, and final prices.

Why a Precompute Job?

A precompute job is a scheduled batch process (e.g., nightly or periodically) that calculates these E.P.P prices in advance. By doing so, you avoid live API calls during searches, ensuring fast responses, consistent prices, and reduced API costs.

What Needs to Be Done?

The precompute job should perform the following steps:

1. Define Scope

Determine the range of dates (e.g., next 365 days), lengths of stay, occupancies, and destinations to cover. Decide on output currency (EUR) and load FX tables if multi-currency is required.

2. Retrieve Raw Data

Fetch availability and net rates from Hotelbeds Cache, promotions data, and applicable policies such as meal plans, cancellation rules, fees, and city taxes.

3. Normalize & Check Eligibility

Deduplicate rate keys, consolidate room types, validate stop-sales, allotments, and min/max stay rules. Determine which promotions apply for each search scenario.

4. Apply Promotions

Execute the combinability matrix (promotion stacking, exclusive vs. cumulative discounts) and compute the resulting net price.

5. Add Fees & Taxes

Incorporate mandatory surcharges and city taxes. Ensure child policies and extra-bed fees are included where applicable.

6. Currency Conversion & Rounding

Convert all prices to EUR (and other currencies if needed). Apply rounding rules for display.

7. Select & Aggregate

For each hotel, check-in date, LOS, and occupancy, pick the cheapest sellable option as the list price (E.P.P). Retain other options (refundable vs. non-refundable, with/without breakfast) for the product detail page. Generate badges for promotions, free breakfast, refundable policies, etc.

8. Store in Cache/Index

Save compact records for quick search results. Each record should contain E.P.P price, board type, cancellation class, badges, and metadata like update timestamp and version. Use TTL and versioning for consistency.

9. Performance & Monitoring

Optimize for cache hit-rate (>70%). Parallelize processing, respect API rate limits, and implement retries. Monitor miss-rates, price deviations, error percentages, and batch durations.

10. Validation

Optionally, perform live validation checks against the Booking API. If deviations exceed policy thresholds (e.g., >2–3%), invalidate and recompute affected cache entries.

Expected Output

The job should produce: - A search index record per hotel/date/LOS/occupancy with minimum E.P.P and badges. - Detail snapshots for product detail pages with alternative rates. - An audit log showing which promotions, FX rates, and fee/tax breakdowns were applied.

Benefits

This approach ensures: - Extremely fast search results. - Accurate, consistent final prices. - Reduced API usage and predictable performance.