

# Hotelbeds Cache API – End-to-End Price Calculation Explained

## 0) Scope & Principles

NL: Dit gedeelte legt de basisprincipes uit: vendor-agnostisch (kan met elke leverancier), deterministisch en controleerbaar (altijd hetzelfde resultaat), scheiding van eligibility en berekening, afronden pas bij output, en altijd een check/reprice voor boeking. EN: This section explains the core principles: vendor-agnostic (works with any supplier), deterministic and auditable (same inputs = same outputs), separation of eligibility checks from math, rounding only at output boundaries, and always recheck/reprice before booking.

## 1) Inputs & Canonical Data Model

NL: Hier worden alle invoervelden beschreven zoals check-in, check-out, kamers, volwassenen, kinderen, valuta, promotiecodes. Daarnaast ook de genormaliseerde velden van de leverancier (hotelcode, kamer, prijs per nacht, belastingen, promoties, enz.), en wisselkoers- en marge-instellingen. EN: This part describes all request inputs such as check-in, check-out, rooms, adults, children, currency, promo codes. It also covers normalized vendor rate fields (hotel code, room, nightly base prices, taxes, promotions, etc.) and FX/margin models.

## 2) Chronological Pipeline

NL: Dit is de stap-voor-stap rekenpijplijn: validatie, ophalen tarieven, eligibility check, basisprijsvector bouwen, supplements, kortingen en promoties toepassen, gratis nachten verwerken, sub totaal berekenen, belastingen toevoegen, valutaomrekening, marge toepassen, coupon toepassen, afronden, totalen splitsen (merchant vs hotel), sanity checks en resultaat cachen. EN: This is the step-by-step pipeline: validate request, fetch rates, apply eligibility filters, build base vectors, apply supplements, discounts, and promotions, handle free nights, compute subtotals, add taxes, convert currency, apply margin, apply coupon, rounding, split totals (merchant vs hotel), sanity checks, and cache the result.

## 3) Eligibility Rules

NL: Regels voor verblijfsduur (LOS), gesloten aankomst/vertrek (CTA/CTD), allotments, resident-only tarieven, enz. EN: Rules for length of stay (LOS), closed-to-arrival/departure (CTA/CTD), allotments, resident-only rates, etc.

## 4) Taxes, Fees & Scopes

NL: Belastingen en toeslagen worden gescheiden in inbegrepen (bv. BTW) en uitgesloten (bv. city tax). Scopes bepalen hoe ze berekend worden (per persoon/nacht, per kamer/verblijf, enz.). EN: Taxes and fees are split into included (e.g., VAT) and excluded (e.g., city tax). Scopes define how they are applied (per person/night, per room/stay, etc.).

## 5) Promotions & Stacking Semantics

NL: Hoe promoties toegepast en gecombineerd worden: prioriteit, exclusiviteit, stackable ja/nee, vaste kortingen vóór percentages, gratis-nachten als speciale regel. EN: How promotions are applied and stacked: priority, exclusivity groups, stackable yes/no, fixed discounts before percentages, free nights as special operators.

## **6) Mathematics & Examples**

NL: Formules voor berekening en praktijkvoorbeelden (bv. lange-verblijfs korting, city tax per persoon, gratis nacht 3=2). EN: Mathematical formulas for calculations with worked examples (e.g., long stay discount, city tax per person, free night 3=2).

## **7) Implementation Blueprint**

NL: Pseudocode die stap-voor-stap laat zien hoe de berekening geprogrammeerd kan worden. EN: Pseudocode showing the exact implementation flow of the calculation logic.

## **8) QA Checklist**

NL: Testpunten zoals controle op upstream prijzen, kindregels, valuta, coupons, performance tests. EN: Checklist of tests like validating against upstream rates, child policies, currency accuracy, coupons, and load performance.

## **9) Glossary**

NL: Uitleg van termen zoals LOS, CTA/CTD, free night, margin vs commission, included vs excluded tax. EN: Explanation of terms like LOS, CTA/CTD, free night, margin vs commission, included vs excluded tax.