

# Uitleg in het Nederlands

Hotelbeds Cache API — Uitleg “End to End Price Calculation”

Dit document beschrijft stap voor stap hoe kamerprijzen correct en transparant moeten worden berekend, inclusief alle kortingen, promoties, toeslagen, belastingen en kosten. Het doel is een controleerbare totaalprijs te leveren die geschikt is voor boekingen.

## 1) Principes:

- Vendor-agnostisch: hetzelfde rekenmodel kan gebruikt worden voor Hotelbeds of andere leveranciers.
- Deterministisch & controleerbaar: dezelfde input moet altijd dezelfde output geven, met een audittrail.
- Scheiding van checks en berekeningen: beschikbaarheidsregels (LOS, CTA/CTD, stop-sell) los houden van de rekenstappen.
- Afronden: alleen bij de uiteindelijke weergave, niet tussentijds.
- Repricing: voor boeken altijd CheckRates uitvoeren om verschillen te detecteren.

## 2) Inputs & datamodel:

- Invoer: check-in/out, aantal nachten, aantal volwassenen/kinderen, markt, valuta, kanaal, device, kortingscodes, filters.
- Vendor data genormaliseerd naar een canoniek model: rate\_key, hotel/room/board, prijzen per nacht, policies, beschikbaarheid, promoties, belastingen.
- FX-tabellen en marges: valutakoersen + marge- of commissie-model.

## 3) Chronologisch proces:

- Validatie van aanvraag (nights  $\geq 1$ , leeftijden van kinderen, min. volwassenen).
- Opbouw van occupancy-hash.
- Ophalen en normaliseren van tarieven (cache of upstream).
- Eligibility checks: stop-sell, allotment, CTA/CTD, LOS, release\_days, residency.
- Basisprijsvector per nacht opbouwen.
- Toepassen van supplements (board, weekend, single use, kinderen, etc.).
- Toepassen van LOS-/seizoensaanpassingen.
- Promotie-engine: stacking, exclusiviteit, volgorde fixed  $\rightarrow$  percent.
- Gratis nacht (bv. 3=2): goedkoopste nacht op nul zetten.
- Subtotaal vóór belastingen berekenen.
- Belastingen en toeslagen toevoegen (inclusief vs exclusief).
- Valutaconversie naar display-currency.
- Merchant margin of commissie toepassen.
- Coupons pas na marge toepassen.
- Afronding alleen voor display, 1 cent correctieregel indien nodig.
- Eindbedrag splitsen in “Pay now” en “Pay at hotel”.
- Sanity checks: geen negatieve bedragen, marges gerespecteerd.
- Opslaan van resultaat met TTL in cache en audit trail.

## 4) Voorbeelden:

- Worked examples tonen hoe vaste en procentuele promoties samenkommen, hoe gratis nachten werken, en hoe belastingen worden berekend.

Kortom: dit playbook is een **“volledig recept”** voor ontwikkelaars om kamerprijzen correct en transparant te berekenen, zodat de gebruiker altijd een betrouwbare totaalprijs ziet.

# Explanation in English

## Hotelbeds Cache API — Explanation “End to End Price Calculation”

This document explains step by step how to correctly and transparently compute room prices, including all discounts, promotions, surcharges, taxes, and fees. The goal is to produce a verifiable total suitable for bookings.

### 1) Principles:

- Vendor agnostic: same pricing model works for Hotelbeds or other suppliers.
- Determinism & auditability: same inputs always produce same outputs, with an audit trail.
- Separation of concerns: eligibility rules (LOS, CTA/CTD, stop-sell) must be checked separately from math.
- Rounding: only at display/output, never during calculations.
- Repricing: always perform CheckRates before booking to detect changes.

### 2) Inputs & data model:

- Inputs: check-in/out, nights, adults/children, market, display currency, channel, device, promo codes, filters.
- Vendor data normalized to a canonical model: rate\_key, hotel/room/board, nightly prices, policies, availability, promotions, taxes.
- FX tables & margins: mid rates + markup, margin or commission models.

### 3) Chronological pipeline:

- Validate request (nights  $\geq 1$ , child ages, min adults).
- Build occupancy hash.
- Fetch and normalize rates (cache or upstream).
- Eligibility filters: stop-sell, allotment, CTA/CTD, LOS, release\_days, residency.
- Construct base nightly price vector.
- Apply supplements (board, weekend, occupancy, children).
- Apply LOS/season adjustments.
- Promotions engine: resolve stacking/exclusivity, apply fixed then percent.
- Free night promos (e.g., 3=2): zero cheapest night(s).
- Compute pre-tax subtotals.
- Add taxes & fees (included vs excluded).
- Convert to display currency.
- Apply merchant margin or commission model.
- Apply coupons after margin.
- Round only for display, add 1 cent adjustment if needed.
- Split totals into “Pay now” vs “Pay at hotel”.
- Sanity checks: no negatives, margins respected.
- Cache result with TTL and full audit trail.

### 4) Examples:

- Worked examples demonstrate combining % and fixed promos, applying free nights, and adding taxes correctly.

In summary: this playbook is a \*\*complete implementation recipe\*\* for developers to calculate room prices accurately and transparently, ensuring users always see a reliable total.