

# Uitleg in het Nederlands

Hotelbeds Cache API — Uitleg “Cheapest From Price” (Goedkoopste Vanaf-prijs)

Dit document legt uit hoe de “vanaf-prijs” per hotel wordt berekend en gecached. Het gaat om de **\*\*laagste totale prijs\*\*** (inclusief kortingen, promoties, belastingen en toeslagen) voor **\*\*2 volwassenen (double occupancy)\*\*** met het **\*\*goedkoopste board type\*\*** (RO, BB, HB, enz.).

## 1) Zakelijke regels:

- Citytrip: minimaal verblijf 2 nachten.
- Andere reizen: minimaal verblijf 5 nachten.
- Indien een “gratis nacht” promotie (bijv. 3=2) bij minLOS+1 goedkoper is, mag die gebruikt worden.
- Altijd dubbel bezet (2 volwassenen), geen kinderen.
- Checkout-coupons worden niet meegerekend.

## 2) Input:

- Zoekperiode (start/einddatum), bestemming of hotelcodes, valuta, markt/point of sale, kanaal, taal.
- Optionele filters (refundable, pay-at-hotel, merken, ketens, etc.).

## 3) Normalisatie van data:

- Uniforme structuur met rateKey, hotel, kamer, board, beleid, beschikbaarheid, prijzen per nacht, promoties en belastingen.

## 4) Algoritme:

- Definieer LOS-set (minLOS en eventueel +1 met free night).
- Itereer over alle check-in datums en LOS-combinaties.
- Bereken de totale prijs per board, kies de laagste.
- Resultaat: goedkoopste totaal, check-in datum, LOS, board type, payNow, payAtHotel, refundable, of free night gebruikt is.

## 5) Prijsberekening stap voor stap:

- Controleer eligibility (stop-sell, allotment, CTA/CTD, min/max nachten).
- Bereken basisprijs per nacht, supplementen, promoties en kortingen.
- Pas free night regels toe indien aanwezig.
- Tel belastingen en fees op, converteer valuta indien nodig, voeg marge/commissie toe.
- Rond alleen af voor weergave.

## 6) Caching:

- Redis key schema's voor rate details en min-prijzen per hotel.
- TTL's: availability 3–10 min, rate 15 min, content 48u.
- SWR en rolling recompute zorgen voor performance.

## 7) Output:

Per hotel één record met totaalprijs, valuta, check-in datum, aantal nachten, board, refundable, payNow, payAtHotel en of free night is gebruikt.

## 8) Voorbeeld:

Citytrip (2 nachten). RO geeft totaal EUR 197. HB met free night (3 nachten) geeft EUR 298, dus niet goedkoper → resultaat is RO = EUR 197. Als HB met free night goedkoper was geweest, zou die getoond worden met badge “Free night”.

Kortom: dit document beschrijft een **\*\*stapsgewijze methode\*\*** om altijd de meest accurate en

goedkoopste “vanaf-prijs” per hotel te tonen, rekening houdend met alle regels en promoties.

## Explanation in English

### Hotelbeds Cache API — Explanation of “Cheapest From Price”

This document explains how to compute and cache the “from price” per hotel. It represents the **lowest final total** (including discounts, promotions, taxes, and fees) for **2 adults (double occupancy)** using the **cheapest board type** (RO, BB, HB, etc.).

#### 1) Business rules:

- City trip: minimum stay 2 nights.
- Other trips: minimum stay 5 nights.
- If a “free night” promotion (e.g., 3=2) at minLOS+1 is cheaper, it can be used.
- Always double occupancy, no children.
- Checkout coupons excluded.

#### 2) Inputs:

- Search period (start/end date), destination or hotel codes, display currency, market/POS, channel, language.
- Optional filters (refundable only, pay-at-hotel only, brand filters, etc.).

#### 3) Normalization of data:

- Standardized structure with rateKey, hotel, room, board, policy, availability, nightly prices, promotions, taxes.

#### 4) Algorithm:

- Define LOS set (minLOS and optionally +1 if free night applies).
- Iterate through check-in dates and LOS combinations.
- Compute total per board and choose the lowest.
- Result: cheapest total, check-in date, LOS, board, payNow, payAtHotel, refundable, and free night flag.

#### 5) Price computation steps:

- Eligibility checks (stop-sell, allotment, CTA/CTD, min/max nights).
- Compute base nightly price, supplements, promos, discounts.
- Apply free night operator if eligible.
- Add taxes and fees, convert currency if needed, apply margin/commission.
- Round only for display.

#### 6) Caching:

- Redis keys for rate details and min-prices per hotel.
- TTLs: availability 3–10 min, rate 15 min, content 48h.
- SWR and rolling recompute ensure high performance.

#### 7) Output:

One record per hotel with total price, currency, check-in date, nights, board, refundable, payNow, payAtHotel, and free night usage.

#### 8) Example:

City trip (2 nights). RO total EUR 197. HB with free night (3 nights) yields EUR 298, so RO wins. If HB had been cheaper, result would show HB with 3 nights, EUR 190, and a “Free night” badge.

In summary: this document provides a **\*\*step-by-step recipe\*\*** to always show the most accurate and cheapest “from price” per hotel, accounting for all promotions and rules.