

Calling Phillip's API: which fields are required, and which ones silently break the price.

Written from 14 live calls against staging on 17 Sep 2026, not from memory. Every example here is a real request and a real response.

estimate-api v1 · OpenAPI 3.1 · md5 14c078e9 · contracts/odoo/estimate-api.v1.json

1. **The spec requires exactly one field: `trip`** . But six fields, if missing, produce a wrong price with a 200 and no warning.
2. **Omit `diveFrom` and `diveTo` and all dive revenue disappears**. Measured: 10,000 pesos gone, response still 200 OK.
3. **`trip.guestType` is not the caller's role**. One decides the discount, the other decides what you are allowed to see. Do not conflate them.
4. **Unknown fields are dropped in silence**. Send `agencyName` and Odoo answers 200 and forgets it. Nothing tells you.
5. **Speed is not the problem**. Five calls in a row: 0.40 seconds. First call after idle: 7.5 seconds. Size the timeout for the cold call, not the average.

THE MAP

Five endpoints, one of them writes

<https://casa-escondida-03-staging-web-37790335.dev.odoo.com/estimate-api>

ENDPOINT	WHAT IT DOES	WRITES?	ANONYMOUS	MEASURED
GET /v1/health	Liveness only	no	yes	—
GET /v1/estimate/rates	The rate card for the caller's role	no	yes	705 B
GET /v1/estimate/rooms	The room pool, plus what is free if you pass dates	no	yes	736 B
POST /v1/estimate/compute	Price one trip	no	yes	4–39 KB
POST /v1/booking/submit	Creates a folio and draft quotations in Odoo	YES	untested	—

The first four have been called many times. The fifth has never been called, because it creates a real record in Phillip's staging. Tell Phillip before you try it.

THE MAIN QUESTION

Which fields are required

The answer has two tiers, and the second one is where the danger lives.

Tier 1: missing means a 422, and you see it

ENDPOINT	REQUIRED
compute	trip
submit	trip , contact.name , contact.email

That is the whole list. Every other field in Trip is optional according to the spec.

Tier 2: missing means a wrong price, and nothing tells you

FIELD	IF YOU OMIT IT	MEASURED
<code>trip.diveFrom</code> <code>trip.diveTo</code>	Odoo falls back to the demo trip's October 2026 window. The guest's dive days fall outside it, so every dive line disappears .	lost 10,000
<code>trip.transportType</code>	Defaults to <code>none</code> . A guest with <code>transport: true</code> still produces no van line.	van revenue lost
<code>trip.checkIn</code> <code>trip.checkOut</code>	Defaults to the demo trip's 5–9 Oct 2026. Room and board are billed for the wrong dates.	wrong dates
<code>guest.roomId</code>	A guest bound to no room gets no room line.	room revenue lost
<code>guest.days</code>	A guest with <code>diver: true</code> but no day entries dives on no day, and pays nothing for diving.	dive revenue lost
<code>trip.guestType</code>	Defaults to <code>retail</code> . An agent is billed retail, quoting 30% too high on rooms.	quoted too high

Proof: two requests, two lines apart

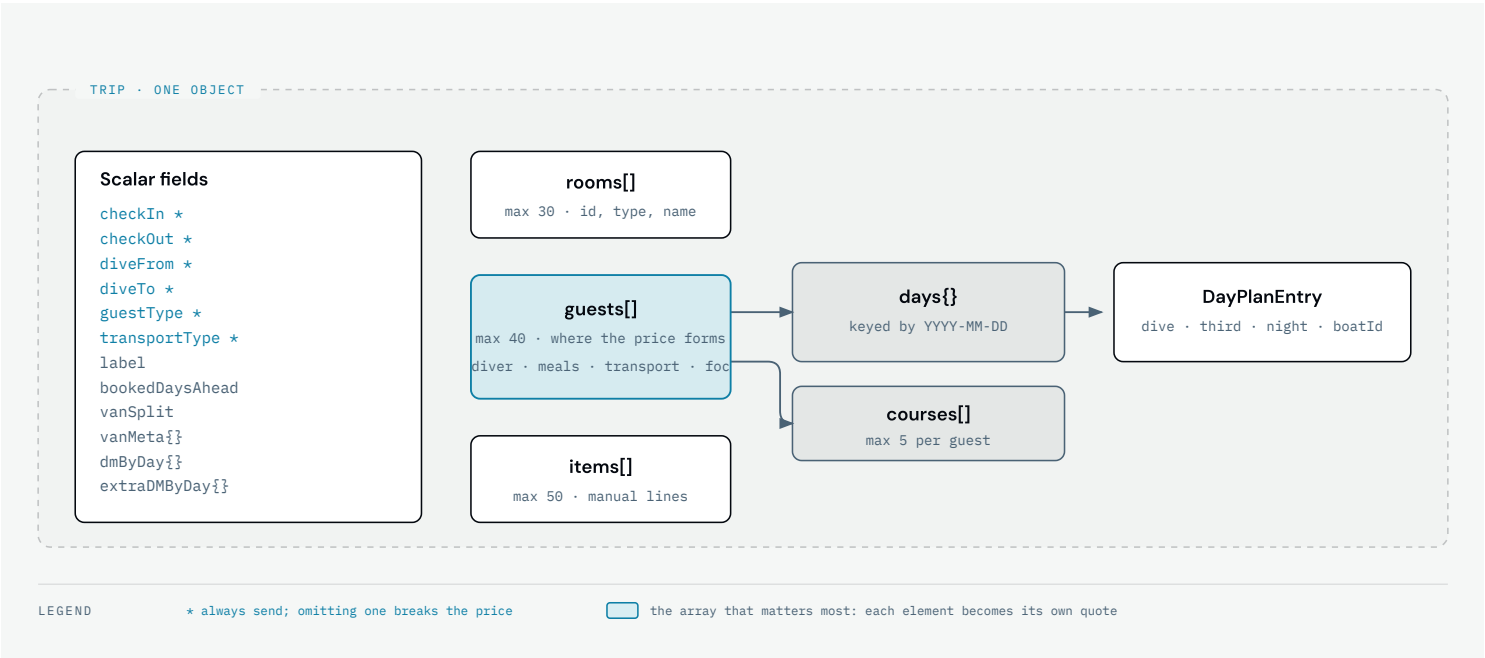
Same guest, same dive day of 2 Dec. The left one omits the dive window, the right one has it.

<pre>// A – thiếu diveFrom/diveTo {"checkIn": "2026-12-01", "checkOut": "2026-12-03", "guests": [{"diver": true, "days": {"2026-12-02": {"dive": true}}}]}</pre>	<pre>// B – có diveFrom/diveTo {"checkIn": "2026-12-01", "checkOut": "2026-12-03", "diveFrom": "2026-12-02", "diveTo": "2026-12-02", "guests": [{"diver": true, "days": {"2026-12-02": {"dive": true}}}]}</pre>
<pre>→ HTTP 200 → diveDates: ["2026-10-06", "2026-10-07", "2026-10-08"] → lines: room 11,000 → catRev.dive: 0 → warnings: []</pre>	<pre>→ HTTP 200 → diveDates: ["2026-12-02"] → lines: room 11,000 dive 10,000 → catRev.dive: 10,000 → warnings: ["no boat picked yet"]</pre>

Side A raises not one warning. This is the class of error that reaches a paying guest.

Rule for the BFF: always send a complete Trip, every field, empty ones included. Never send a partial patch to compute.

The shape of a Trip



The six starred fields are the tier-2 fields above. Not required by the spec, required in practice.

REFERENCE

Field by field

Trip

FIELD	TYPE	DEFAULT	NOTES
label	string	—	Group name, display only
guestType	retail agent instructor	retail	Sets the discount. Not the API caller's role
transportType	none roundtrip oneway	none	Defaults to no transport. Send it explicitly.
checkIn checkOut	string YYYY-MM-DD	demo trip	Nights = checkOut minus checkIn
diveFrom diveTo	string YYYY-MM-DD	Oct 2026	The gate: a guest dive day outside this window is not priced
bookedDaysAhead	number	0	Lead time in days, feeds the early-bird rule
rooms[]	Room[]	[]	Max 30
guests[]	Guest[]	[]	Max 40
items[]	CustomItem[]	[]	Max 50, free-form billed lines
vanSplit	equal vehicle	vehicle	Split the van bill per head or per vehicle
vanMeta{}	map	{}	Overrides time and price per van run, keyed vanA 2026-10-05 0
dmByDay{} extraDMByDay{}	map	{}	Dive masters and extra dive masters, per day

Guest

FIELD	TYPE	DEFAULT	NOTES
id	string	—	Must be unique in the trip; links to items.gids
name	string, ≤120	Guest	—
diver	boolean	true	Defaults to diving. A non-diver must be sent as false
meals	boolean	true	1,500 a day, never discounted
transport	boolean	true	Only has an effect when trip.transportType is not none
foc	boolean	false	Marks the free guest. Odoo counts the slots, a person picks who.
roomId	string	null	Must match a rooms[].id . No match, no room charge.
courses[]	dsd refresher ow aow rescue	[]	Max 5. All five price correctly, though rates lists only three.
days{}	map YYYY-MM-DD → DayPlanEntry	{}	Only dates inside diveFrom–diveTo are priced
arrive depart	string YYYY-MM-DD	""	Late arrival or early departure. Changes the per-night room split.
vanA vanD	string, ≤120	null	Trap: Sky's tool stores an integer. Sending a number is a 422.
comment	string, ≤500	""	—

Room · DayPlanEntry · CustomItem · VanMeta

OBJECT	FIELDS	NOTES
Room	id , type , name	type is standard, deluxe or suite. name binds a real room, e.g. "Deluxe B". A wrong name raises nothing, it just loses the binding.
DayPlanEntry	dive , third , night , boatId	All booleans except boatId. No boat still prices, it just adds a warning.
CustomItem	id , name , price , qty , mode , date , dateTo , gids[]	Free-form lines, for anything Odoo has no rate for. mode is a free string in the spec, but the tool only uses each and split .
VanMeta	date , time , price , foc	Overrides one specific van run. The map key is vanA date index .

Two different things called role

trip.guestType

- Lives in the payload, you send it
- Decides **how much**: agents get 30% off rooms
- retail, agent, instructor

Caller role (response.role)

- Odoo derives it from the access key; you cannot send it
- Decides **what you see**: cost, profit, the comparison model
- Anonymous calls today always return guest

CALLER ROLE	MODEL.KPIS.COST	MODEL.COSTS	RETAIL_MODEL	ASSUMPTIONS	SENDING ASSUMPTIONS
guest	null	null	null	null	Ignored, verified
agent	null	null	present	null	Ignored
staff	present	present	present	present	Accepted, deep-merged over the defaults

The three rows come from the spec. Only the guest row is verified, because keys for the other two roles do not exist yet. That is item one on the list for Phillip.

A working example, copy and run

Two guests, two nights in a Standard room, one of them diving for one day, both on full board. This is the smallest request that still prices correctly.

```
curl -X POST https://casa-escondida-03-staging-web-37790335.dev.odoo.com/estimate-api/v1/estimate/compute \
-H "content-type: application/json" -d '{
  "trip": {
    "label": "Retail couple",
    "guestType": "retail",
    "transportType": "none",
    "checkIn": "2026-11-20", "checkOut": "2026-11-22",
    "diveFrom": "2026-11-21", "diveTo": "2026-11-21",
    "bookedDaysAhead": 30,
    "rooms": [ { "id": "r1", "type": "standard" } ],
    "guests": [
      { "id": "g1", "name": "Ana", "diver": true, "meals": true, "transport": false,
        "foc": false, "roomId": "r1", "courses": [],
        "days": { "2026-11-21": { "dive": true, "third": false, "night": false } } },
      { "id": "g2", "name": "Ben", "diver": false, "meals": true, "transport": false,
        "foc": false, "roomId": "r1", "courses": [], "days": {} }
    ],
    "items": []
  }
}'
```

The response, abridged:

```

{
  "ok": true,
  "role": "guest",
  "retail_model": null,          // chỉ staff và agent mới có
  "assumptions": null,         // chỉ staff mới có
  "model": {
    "N": 2,                     // số đêm
    "stayDates": ["2026-11-20", "2026-11-21"],
    "diveDates": ["2026-11-21"],
    "quotes": [
      { "g": { "id": "g1", "name": "Ana", ... }, // cả đối tượng khách, lặp lại
        "isFoc": false,
        "lines": [
          { "cat": "room", "label": "Standard A – 2 nights",
            "sub": "nightly rate ÷ that night's roommates, summed over your stay",
            "gross": 7600, "discs": [], "net": 7600 },
          { "cat": "meals", "label": "Full board – 2 days", "gross": 3000, ... },
          { "cat": "dive", "label": "Boat dives – Sat, Nov 21",
            "sub": "2-dive boat trip · 1 diver out", "gross": 10000, ... }
        ],
        "gross": 20600, "total": 20600, "discountTotal": 0 },
      { "g": { "id": "g2", "name": "Ben", ... }, "total": 10600, ... }
    ],
    "catRev": { "room": 15200, "meals": 6000, "dive": 10000, "course": 0,
                "transport": 0, "gear": 0, "extras": 0 },
    "kpis": { "revenue": 31200, "cost": null, "profit": null, "margin": null,
              "rpgn": 7800, "discounts": 0, "guests": 2, "nights": 2 },
    "foc": { "ok": true, "per": 5, "bracket": 0, "entitled": 0, "marked": 0,
             "stay": {...}, "dive": {...} },
    "warnings": [ { "level": "warn", "text": "Sat, Nov 21: no boat picked yet for Ana." } ],
    "dayPlans": [...], "vanRuns": [...], "presence": {...}, "covers": {...},
    "roomNames": {...}, "roomAvailability": {...}
  }
}

```

Reading the response

KEY	WHAT IT IS	WHAT THE UI DOES WITH IT
quotes[]	One block per guest, each with its money lines	This is what the guest sees. label and sub come pre-written in English, usable as is.
catRev	Totals by category: room, meals, dive, course, transport, gear, extras	The summary table, one row per category
kpis	Revenue, cost, profit, margin, revenue per guest-night	The last four only render for staff
foc	How many free slots are earned and how many are marked	When entitled exceeds marked , prompt staff to pick someone
warnings[]	Operational warnings, not errors	Show as reminders, never block the action
roomAvailability	Which rooms are free for this trip's dates	Feeds the room picker; saves a separate rooms call
dayPlans vanRuns presence	Per-day operational data	Heavy: each embeds the full guest object again. 27 of the 39 KB in the 7-guest example.

ERRORS

Four shapes, all of them 422

```
// thiếu trường bắt buộc
{"detail":[{"type":"missing","loc":["body","trip"],"msg":"Field required","input":{}}]}

// sai enum
{"detail":[{"type":"enum","loc":["body","trip","guestType"],
  "msg":"Input should be 'retail', 'agent' or 'instructor'",
  "input":"vip",
  "ctx":{"expected":"'retail', 'agent' or 'instructor'"}]}]}

// sai kiểu ← đây là bẫy vanA
{"detail":[{"type":"string_type","loc":["body","trip","guests",0,"vanA"],
  "msg":"Input should be a valid string","input":0}]}

// vượt giới hạn
{"detail":[{"type":"too_long","loc":["body","trip","guests"],
  "msg":"List should have at most 40 items after validation, not 80",
  "input":[ ...tồn bộ payload được trả lại... ]}]}
```

All four follow the Pydantic convention: detail is an array and each entry's loc points straight at the offending field. The BFF should map loc to a form path and highlight that input. Note the last shape echoes the whole payload back in input , so do not log the error verbatim.

Timeouts and retries

SITUATION	MEASURED	PROPOSAL
compute, back to back	0,36–0,40 s	—
compute, first call after idle	7,55 s	Time out at 15 seconds, not 8
compute, retry	—	Retry once on 502 or 503. Safe, because compute writes nothing.
submit, retry	—	Never retry automatically. With no Idempotency-Key, a retry creates a second folio.

POST /V1/BOOKING/SUBMIT

The one endpoint that writes

```
{
  "contact": { "name": "Ana Cruz", "email": "ana@example.com", "phone": "+63..." },
  "trip":      { ...đúng Trip đã gửi cho compute... }
}

→ { "success": true, "folio_id": 162, "order_ids": [418, 419] }
```

WHAT TO KNOW	DETAIL
Required	contact.name , contact.email , trip
Guest list	Phillip said on 14 Sep he would add it. Today's spec has none, so individual guest names do not reach Odoo.
De-duplication	Odoo folds every open request from the same customer into one record. One agency sending two different groups in a week gets them merged.
Double click	No Idempotency-Key yet. The BFF must lock it itself and not rely on Odoo.
Status afterwards	There is no endpoint to ask whether a folio has been paid. That is item five for Phillip.

Nobody has called this endpoint

The example above is built from the spec, not from a live call, because calling it creates a real folio in Phillip's staging. Before the first attempt: tell Phillip, and use an obviously fake email he can clean up.

rates and rooms

GET /v1/estimate/rates

```
{
  "roomRates": { "standard": { "1pax": 5500, "2pax": 7600 },
                 "deluxe":   { "1-2pax": 11200, "3-4pax": 16400.0100000000002 },
                 "suite":    { "1-2pax": 14200, "3-4pax": 18400 } },
  "diveTiers": { "1": 10000, "2": 5500, "3": 4500, "4": 3600 },
  "mealRate": 1500,
  "courseRates": { "dsd": 5500, "ow": 22000, "aow": 18000 }, // thiếu refresher, rescue
  "transport": { "roundtrip": 13000, "oneway": 6500 },
  "terms": { "depositPct": 50 },
  "roomNames": { "standard": [16 phòng], "deluxe": [4], "suite": [4] }
}
```

- `diveTiers` keys off the divers in the water that day, not the headcount.
- `courseRates` is **missing two kinds**: refresher and rescue. The engine still prices them correctly at 5,500 and 20,000. Do not build the course picker from this.
- Deluxe 3–4 pax returns 16400.0100000000002. Round it in the BFF before display.

GET /v1/estimate/rooms?check_in=2026-11-20&check_out=2026-11-22

```
{ "ok": true,
  "rooms": { "standard": ["Standard A"... "Standard P"], "deluxe": [...], "suite": [...] },
  "available": { "standard": [...], "deluxe": [...], "suite": [...] }, // null nếu không truyền ngày
  "check_in": "2026-11-20", "check_out": "2026-11-22" }
```

Casa has exactly 24 rooms: 16 standard, 4 deluxe, 4 suites. That matches Sky's tool exactly. The same data also rides inside the compute response as `roomNames` and `roomAvailability`, so the pricing flow needs no separate call.

MAPPING

Sky's form to Odoo's Trip

IN SKY'S TOOL	IN ODOO'S TRIP	WHAT THE BFF MUST DO
trip.label , checkIn , checkOut , diveFrom , diveTo , guestType , transportType , bookedDaysAhead , vanSplit	Identical names	copy across
rooms[].id , .type	Same, plus Odoo's name	Copy across; leave name empty until the UI picks real rooms
guests[].vanA , .vanD int	vanA , vanD string	Must convert; a number is a 422
items[].mode = each split	free string	Keep to those two values; do not invent new ones
trip.agencyName	absent	Keep it in the draft store. Odoo drops it in silence, verified.
trip.opsNotes (desk, dive, kitchen)	absent	Keep in the draft store until Phillip opens a place for it
trip.manual (ghi đè giá từng khách)	absent	Turn into an items line, or drop the feature in v1
trip.groupId	absent	Internal key, stays in the draft store
assumptions (bảng giá sửa tay)	Present, staff only	Never send it from a guest-facing surface. It is ignored, verified.

FOR PHILLIP

Ten items, the first three are blockers

#	REQUEST	WHY	LEVEL
1	Declare <code>securitySchemes</code> and issue keys for the three roles	Without it the staff role cannot be tested and the BFF cannot know which header to send	blocker
2	<code>Idempotency-Key</code> on submit	Two clicks means two folios. It also replaces today's de-dupe, which wrongly merges two groups from one agency.	blocker
3	Guest list in submit, with <code>external_ref</code>	Phillip promised this on 14 Sep. Without it guest names never reach Odoo and nothing reconciles.	blocker
4	<code>rates_version</code> in the compute response	So a snapshot can prove which rate card it used	needed
5	<code>GET /v1/booking/{folio_id}</code>	To learn whether a folio was sent or paid, and when the draft may be deleted	needed
6	Response examples in the spec	With no examples the mock returns nonsense strings. Three examples would do.	needed
7	<code>vanA</code> and <code>vanD</code> : number or string	The tool stores numbers, the spec demands strings. One side has to change.	needed
8	<code>courseRates</code> missing refresher and rescue	The engine prices them but the rate card hides them. The UI cannot build a course list.	needed
9	Defaults inherited from the demo trip	Omit <code>diveFrom</code> and you inherit October 2026 rather than nothing. This should be an error, or at least a warning.	dangerous
10	Slim the response	<code>vanRuns</code> , <code>presence</code> and <code>dayPlans</code> re-embed whole guests: 27 of 39 KB.	can wait

The `tn-casa-quotation-estimator` repo already carries three tests asserting that items 1, 2 and 4 are still missing. They turn red the day Phillip ships one, and that is how we find out.

UNBLOCKED

A mock that returns real numbers

The spec has no examples, so Prism invents nonsense strings. The fix does not wait for Phillip: save the real responses above as example files.

```
contracts/odoo/
estimate-api.v1.json      # spec đã ghim, md5 14c078e9
examples/
  compute.retail-couple.json # 2 khách, 2 đêm, 1 ngày lặn → 4 KB
  compute.agent-group.json  # 7 khách, 4 đêm, FOC, khóa học → 39 KB
  compute.course.json       # refresher + open water
  rates.json                # bảng giá thật
  rooms.json                # 24 phòng thật
  errors.422-*.json         # bốn dạng lỗi

npm run mock -w contracts    # Prism, cổng 4010, trả ví dụ thật
npm run spec:check -w contracts # so spec staging với bản đã ghim, lệch là thoát 1
```

Contract pod's job

- Commit the six example files above
- Point Prism at the examples instead of letting it invent
- Add a test: a trip without `diveFrom` must yield zero dive revenue, so we learn the day Phillip fixes it

BFF pod's job

- One builder that always emits a complete Trip, never omitting the six starred fields
- Convert `vanA` and `vanD` from number to string
- Map the `loc` in a 422 onto a form field path
- Round money before display

Source: estimate-api v1 on staging, md5 14c078e9, 14 live calls on 17 Sep 2026 from Vietnam. The submit endpoint has never been called. Money in Philippine pesos.