stoop

FOR PROPERTY MANAGERS AND THEIR SOFTWARE

The Stoop partner feed

One JSON feed of your rentals, the same for every partner, whatever software you run. This page is everything your team needs to build it.

Version 1 · The latest copy is at stooprent.com/partners/feed

> **Before you send a feed**
>
> Stoop imports listings only from property managers with a signed listing partner agreement. With it comes your feed key. To get started, email **hello@stooprent.com**.

How property managers send Stoop their rentals. It's one JSON document that lists every unit you want on Stoop, sent again whenever anything changes, or at least every hour. Every partner uses this same format, whatever software they run. Partners whose software already sends MITS 4.1 or 5.0 can send that instead.

Format	JSON, UTF-8, `Content-Type: application/json`
Schema	**`partner-feed.schema.json`**, also served at `https://api.stooprent.com/v1/partner-feed/schema.json` (JSON Schema 2020-12)
Sample	**`night-1.json`**, a fictional feed that passes every check
Money	Whole cents: `165000` is $1,650.00
Dates	`YYYY-MM-DD`, in the property's local time. Timestamps are ISO 8601 with an offset.

Size	At most 25 MB, 5,000 properties and 5,000 units per property
Before you start	A signed listing partner agreement. Stoop imports nothing without one.

Sending it

Pick either way. Both accept the same document.

Stoop pulls it. You serve the feed at an `https://` address of your own. Stoop fetches it every hour with `Authorization: Bearer <token>`, using a token you and Stoop agree on during setup (Stoop keeps it in its secret store, never in its database). Answer `200` with the feed. Answering `304` to `If-None-Match` is optional.

You push it. `PUT https://api.stooprent.com/v1/partner-feed` with `Authorization: Bearer stoop_feed_...`, the feed key Stoop gives you. Push whenever something changes, and at least once an hour. Stoop answers `202` with a delivery ID and the same report the check gives, and imports the feed within a minute.

Check without sending. `POST https://api.stooprent.com/v1/partner-feed/check` with your feed key and a feed checks it and stores nothing. The report says which properties and units would be rejected and why, path by path:

```
{
  "format": "stoop.partner-feed",
  "version": 1,
  "properties": 2,
  "units": 5,
  "accepted": 4,
  "rejected": [
    { "type": "listing", "id": "U-205", "path": "properties.0.units.3.bathrooms",
"message": "Whole or half baths: 1, 1.5, 2" }
  ]
}
```

A feed that isn't JSON, isn't this format or version, or is larger than 25 MB is refused as a whole (`400`, or `413` for size). A wrong property or unit is rejected on its own, and the rest of the feed still imports.

Every feed is the whole list

Each feed lists **every unit you currently offer on Stoop**, not just what changed. That's how Stoop knows when a unit comes off the market:

You	Stoop
Leave a unit out	Ends the listing once two feeds in a row have left it out and a day has passed. One missed feed never ends anything.
Send `"status": "on_hold"` (a holding fee, an approved application)	Takes it out of renters' decks right away; it comes back when you send it as available again
Send `"status": "leased"`	Ends the listing right away. You can leave it out of later feeds.

A feed with fewer than half the units of the last good one is treated as broken: Stoop imports what's in it but ends nothing. Fix the feed, or tell Stoop if you really did take most of your units off.

IDs are yours and must stay the same from feed to feed. A unit ID is unique across the whole feed, and it names the unit, not the vacancy: when a unit comes back on the market months later, send the same ID and Stoop opens a new listing for it.

What a feed looks like


```
{
  "format": "stoop.partner-feed",
  "version": 1,
  "generatedAt": "2026-11-16T10:00:00-08:00",
  "properties": [
    {
      "id": "P-200",
      "name": "Birch Commons",
      "type": "apartment",
      "address": { "line1": "2100 Example Street", "city": "Bellingham", "state":
"WA", "postalCode": "98225" },
      "pets": { "policy": "cats_and_dogs", "maxPets": 2, "dogWeightLimitLbs": 40,
"depositCents": 30000 },
      "fees": [
        { "id": "wsg", "label": "Water, sewer and garbage", "category":
"utilities", "frequency": "monthly",
          "amountCents": 5000, "per": "occupant", "required": true, "covers":
["water", "sewer", "trash"] }
      ],
      "utilities": { "included": [], "heating": "electric", "waterHeating":
"electric", "cooking": "electric" },
      "units": [
        { "id": "U-201", "label": "201", "bedrooms": 1, "bathrooms": 1,
"squareFeet": 640,
          "rentCents": 145000, "depositCents": 145000, "availableOn": "2026-11-20",
          "photos": [{ "url": "https://photos.example.com/p200/u201/living.jpg",
"caption": "Living room" }] }
      ]
    }
  ]
}
```


Properties

Field		
`id`	required	Your ID for the building or house

Field		
`name`		"Birch Commons"
`type`	required	`apartment`, `townhome`, `house` or `room`. Duplexes, triplexes and condos are `apartment`.
`address`	required	`line1` (street, no unit), `city`, `state` (`WA`), `postalCode` (`98225`). Stoop places it on the map itself.
`description`		Plain text about the building
`amenities`		From the **amenity list**
`photos`		Of the building, best first
`pets`		The pet policy for every unit here, unless a unit has its own
`fees`	required	**Every** fee a renter here can be charged. `[]` says there are none.
`utilities`	required	What the rent includes and how the homes are heated
`units`	required	The units you offer here

Units

Field		
`id`	required	Your ID for the unit, unique across the feed
`label`	required	The unit number renters see, `"204"`; `""` for a whole house
`type`		Overrides the property's type
`floor`		`1` is the ground floor
`status`		`available` (the default), `on_hold` or `leased`

Field		
`bedrooms`	required	`0` for a studio
`bathrooms`	required	Whole or half: `1`, `1.5`
`squareFeet`		
`rentCents`	required	Monthly rent, before any fee
`depositCents`		Security deposit
`availableOn`	required	First move-in date; today or earlier means now
`minLeaseMonths`		
`pets`		Replaces the property's pet policy for this unit
`description`		Plain text about the unit. Used instead of the property's.
`amenities`		Added to the property's
`photos`		Of this unit, best first, shown before the property's
`fees`		Added to the property's. A fee with the same `id` replaces the property's.
`utilities`		Replaces the property's for this unit
`url`		The unit's page on your website, for your leasing team

Fees

Stoop shows renters one honest monthly price: rent, every required monthly fee and an estimate of utilities. So the feed has to list every fee, and say how each one is charged. Stoop works out the price for each renter's household.

Field		
`id`	required	Unique within the property's list, and within the unit's
`label`	required	What renters see: "Water, sewer and garbage"
`category`	required	`utilities`, `pet`, `parking`, `insurance`, `admin`, `amenity`, `technology` or `other`
`frequency`	required	`monthly` or `one_time`. Send other recurring fees as their monthly amount: $120 a year is `1000` a month.
`amountCents`	one of	A fixed amount
`percentOfRent`	one of	A share of the rent, up to two decimals: `2` is 2%
`per`		`unit` (the default), `occupant` or `pet`: what the amount is multiplied by
`required`	required	`false` only for fees a renter can choose, like a parking spot or storage
`onlyWith`		`pet`, `cat` or `dog`: charged only to households with those animals
`covers`		Utility fees only: which utilities it pays for (`water`, `sewer`, `trash`, ...)
`waivedWith`		`own_renters_insurance`: a renter can show their own policy instead

How Stoop uses them:

- **The monthly price** a household sees is rent, plus every `required` `monthly` fee that applies to it (per occupant times its size, per pet times its pets), plus the utilities estimate. A required fee that can be waived still counts, because the renter pays for coverage either way.
- **Optional fees** are listed beside the price, not in it.

- **One-time fees** are listed as move-in costs, with the deposits.
- **Assistance animals are not pets.** Stoop never applies pet fees, deposits or limits to them, and renters never list them as pets.

Examples:

```
[
  { "id": "wsg", "label": "Water, sewer and garbage", "category": "utilities",
"frequency": "monthly",
    "amountCents": 5000, "per": "occupant", "required": true, "covers": ["water",
"sewer", "trash"] },
  { "id": "pet-rent", "label": "Pet rent", "category": "pet", "frequency":
"monthly",
    "percentOfRent": 2, "per": "pet", "required": true, "onlyWith": "pet" },
  { "id": "insurance", "label": "Renters liability insurance", "category":
"insurance", "frequency": "monthly",
    "amountCents": 1250, "required": true, "waivedWith": "own_renters_insurance" },
  { "id": "parking", "label": "Parking permit", "category": "parking", "frequency":
"monthly",
    "amountCents": 2500, "required": false },
  { "id": "admin", "label": "Lease administration fee", "category": "admin",
"frequency": "one_time",
    "amountCents": 20000, "required": true }
]
```

For a $1,450 one-bedroom with those fees, a renter living alone sees $1,512.50 plus utilities. Two people with a dog see $1,591.50 plus utilities (two $50 utility fees, $29 pet rent, $12.50 insurance), and a $200 move-in fee.

Fees that depend on the city, like a fee one city doesn't allow, are simply left off that property's list.

Utilities

Each utility is one of three things:

1. **In the rent**: list it in `utilities.included`.
2. **Billed by you**: a fee with `category: "utilities"` and `covers`.
3. **Paid by the renter to the utility company**: everything else. Stoop estimates these.

Field		
`included`	required	From `electricity`, `gas`, `propane`, `oil`, `water`, `sewer`, `trash`, `internet`
`heating`		`electric` (baseboard, wall or furnace), `heat_pump`, `gas`, `propane` or `oil`
`waterHeating`		`electric`, `gas`, `propane` or `oil`
`cooking`		`electric`, `gas` or `propane`
`estimateCents`		Your own estimate of what renters pay utility companies each month. Stoop uses yours instead of its own.

Stoop's estimate comes from the local housing authority's published utility allowance schedule (form HUD-52667), by unit size, building type and fuel. Electric baseboard heat costs more there than a heat pump, so send `heating`. Internet isn't estimated. A unit Stoop can't estimate, and whose utilities you didn't estimate either, waits out of renters' decks until one of you can.

Photos

`https://` links to JPEG, PNG or WebP images, at most 15 MB each and 60 per unit, best first. Stoop copies them when it imports the feed, under your agreement, and serves its own copies. Set `"representative": true` on a photo of a similar unit, and Stoop labels it "Similar unit". A unit needs at least one photo (its own or the property's) to go live, and should have three.

Vocabularies

List	Values
Property types	`apartment`, `townhome`, `house`, `room`
Pet policies	`none`, `cats`, `dogs`, `cats_and_dogs`, `case_by_case`
Amenities	`in_unit_laundry`, `shared_laundry`, `dishwasher`, `air_conditioning`, `parking`, `ev_charging`, `elevator`, `step_free_access`, `gym`, `bike_storage`, `storage`, `balcony`, `yard`, `roof_deck`, `pool`, `concierge`, `furnished`
Fee categories	`utilities`, `pet`, `parking`, `insurance`, `admin`, `amenity`, `technology`, `other`
Utilities	`electricity`, `gas`, `propane`, `oil`, `water`, `sewer`, `trash`, `internet`

Objects take only the fields listed here. An unknown field rejects its unit or property, so a misspelled price can't pass as a missing one.

When a unit goes live

A unit that passes the schema still waits, out of renters' decks, until:

- its whole monthly price is known: rent, fees (once you've confirmed in writing that the feed lists every fee) and utilities
- Stoop has placed the address on the map
- at least one photo has been copied
- it has a move-in date, a pet policy, bedrooms and bathrooms

`GET https://api.stooprent.com/v1/partner-feed/deliveries/{id}`, with your feed key, says how a delivery's import went: how many units are live, and what each waiting unit still needs.

Versions

This is version 1. Stoop may add optional fields to it and accepts them before asking anyone to send them. Anything that would change the meaning of a field you already send is version 2, and Stoop accepts both versions for at least six months.