

Wilson Agent Bridge

Wilson decides who to write and drafts the note. Your agent sends it, and tells Wilson what came back.

Wilson recommends communications, called Today cards: who to write, why now, and a ready-to-send draft in the owner's voice. Wilson never sends them itself. The agent bridge lets **another agent** pull those recommendations, send them over any channel (email, WhatsApp, SMS, and so on), and report back what happened, including the person's reply. Wilson then holds the full back-and-forth on each person's timeline, and its next drafts are written with that context.

Base URL: the Wilson deployment, for example `https://app.colinrobertson.dev` .

Authentication

Every request carries a per-workspace key:

```
Authorization: Bearer wsk_...
```

Keys are created by the workspace owner in **Settings** › **CRM** › **Agent bridge**. The key is shown once at creation (only a SHA-256 hash is stored) and can be revoked there at any time. A key is scoped to exactly one workspace.

Rate limit: 240 requests per minute per key, answered with `429` and `retryable: true` . Any 5xx carrying `retryable: true` is safe to retry with backoff.



1 Pull recommendations

GET /api/agent/v1/recommendations
GET /api/agent/v1/recommendations?format=csv

PARAM	VALUES	DEFAULT
status	live, acted, dismissed, expired, all	live
kind	comma-separated action kinds	all
limit	1 to 200 (whole numbers)	60
format	json, csv	json

JSON response

```
{
  "workspace": { "slug": "acme", "company_name": "Acme Foundation" },
  "generated_at": "2026-09-05T16:00:00.000Z",
  "status_filter": "live",
  "count": 4,
  "amount_filtered": 0,
  "people_filtered": 1,
  "snoozed_filtered": 0,
  "unsurfaced_filtered": 0,
  "duplicates_collapsed": 0,
  "window_truncated": false,
  "recommendations": [
    {
      "id": "6b2e...-uuid",
      "kind": "reconnect",
      "status": "pending",
      "rank_score": 78,
      "effective_rank": 81.4,
      "why": "You have not spoken since the spring gala and she just published a new piece.",
      "effort_minutes": 5,
      "created_at": "2026-09-05T11:15:00.000Z",
      "expires_at": "2026-09-10T11:15:00.000Z",
      "recipient": {
        "contact_id": "9f1c...-uuid",
        "person_id": "77aa...-uuid",
        "name": "Jane Rivera",
        "email": "jane@example.org",
        "phone": "+1 555 010 1234",
        "preferred_channel": "whatsapp",
        "company": "Rivera Consulting",
        "title": "Principal",
        "role": "donor",
        "warmth_stage": "engaged"
      },
      "message": {
        "subject": "Your new piece",
        "body": "Hi Jane, ...\n\nColin",
        "suggested_channel": "whatsapp"
      },
      "open_in_wilson": "https://app.colinrobertson.dev/relationships?connector=9f1c..."
    }
  ]
}
```



How the live list is chosen

`live` is the same eligible pool Wilson's own Today page composes from: pending, unexpired, past any scheduled hold-back. It excludes the workspace's own team, skip-tier people, snoozed people (someone parked until a future date; booking-cycle cards are exempt, because a dated event is not cadence), internal chores, and any second card for a human who already has a higher-ranked one in the list. One human, one message. Cards whose text carries a currency amount are withheld, since some workspaces are under terms that forbid amounts leaving the app.

Every withheld row is counted in the response (`amount_filtered` , `people_filtered` , `snoozed_filtered` , `unsurfaced_filtered` , `duplicates_collapsed`), never dropped silently.

The page then seats a curated four to eight of that pool per day. The bridge hands you the whole ranked pool and lets your agent decide how many to act on. Live rows are sorted by `effective_rank` , the page's own ordering key (age decay, touch debt, booking-cycle rules), so sending from the top is the closest match to what a person would see.

`acted` , `dismissed` and `expired` are history: a record, newest first, one row per card, with `effective_rank: null` . A card past its expiry that Wilson's sweep has not yet re-labelled is reported as expired, never as pending. `status=all` returns the live rows first, then history. A pending card that is held back, expired, or for a snoozed person is never returned as actionable under any filter.

Notes for senders

- `message.body` is the **complete** message, greeting to sign-off, already in the owner's voice. Send it as-is or lightly adapted to the channel.
- `suggested_channel` is the person's stated preference when Wilson has one, else whatever their contact details support (email, then SMS). Email and phone may both be null. That is still a valid recommendation the owner might act on in person, so skip it or route it however your agent can.
- CSV is RFC 4180, UTF-8 with a byte-order mark so Excel reads diacritics correctly. Free-text columns carry a spreadsheet formula guard: a value starting with `=` , `+` , `-` or `@` is prefixed with an apostrophe so it cannot execute when opened. Identifiers, numbers, dates, URLs and `recipient_phone` are never altered. Use JSON when you need byte-exact text.
- Recommendations expire after about five days. Do not queue them for later: pull fresh before sending, and report outcomes promptly.

2 Report the outcome of a send

POST `/api/agent/v1/recommendations/{id}/outcome`

```
{
  "outcome": "sent",
  "channel": "whatsapp",
  "subject": "Your new piece",
  "body": "Hi Jane, ... (what was ACTUALLY sent, if you edited the draft)",
  "sent_at": "2026-09-05T16:03:00Z",
  "external_message_id": "wamid.abc123"
}
```

or, when the recommendation was not acted on:

```
{ "outcome": "skipped", "skip_reason": "bad_timing" }
```

- `channel` is required for a send: one of `email`, `imessage`, `whatsapp`, `sms`, `slack`, `phone`, `in_person`, `linkedin`, `other`.
- Echo `body` when you can. It is logged on the person's timeline as the owner's outbound message, which is what lets Wilson's next draft build on what was actually said.
- `external_message_id` makes the write idempotent, so retries will not double-log. It is namespaced separately from reply ids, so reusing a provider's message id for both a send and its reply is fine.
- The response's `stored` is `full` (the echoed words were kept), `metadata_only` (only that it happened), or `none` (nothing written, as with a skip).
- `sent_at` and `received_at` must be ISO 8601 **with a timezone** and a real calendar date, for example `2026-09-05T14:00:00Z`. Anything else is a `400 invalid_timestamp`. Omit it to mean now. A send cannot predate the recommendation it answers.
- `skip_reason` is optional: `not_relevant`, `wrong_person`, `bad_timing`, `already_handled`.
- Reporting twice, or racing someone acting in the app, returns `already_finalized: true` with the status the card actually has. That is not an error.

What it changes in Wilson: the card is marked acted or dismissed, a touch lands on the person's timeline, their cultivation clock moves forward, and the outcome feeds the same analytics as an act inside the app.

3 Import a reply

POST /api/agent/v1/replies

```
{
  "recommendation_id": "6b2e...-uuid",
  "channel": "whatsapp",
  "body": "So glad you saw it! Would love to catch up, how is Thursday?",
  "from": "Jane Rivera",
  "received_at": "2026-09-06T09:14:00Z",
  "external_message_id": "wamid.def456"
}
```

Identify who replied with the **first** of: `recommendation_id` (the card the reply answers, best when you have it), `contact_id`, `prospect_id`, or `email`. Email matching is case-insensitive. Several records holding one address are treated as one person unless their names contradict each other: a shared first or last name, or a variant such as "Beth" and "Elizabeth Howard", is the same person, while two unrelated names on a shared inbox are not. When they genuinely differ you get `409 ambiguous_person` with candidates, so resend with `contact_id`. If nobody matches you get `404 person_not_found`: add the person in Wilson first.

What it changes in Wilson: a reply event with the text lands on the person's timeline, where Wilson's drafters read conversation history and the ask-readiness gate looks for an answer. If the person is also in the pipeline, their last-contacted clock advances, forward only, so importing an old conversation never rewinds it. Replies do **not** count as a touch by the owner, because a reply is them contacting you.

`external_message_id` makes the import idempotent, answering `deduped: true` on retries. Always send it when your channel provides one.

Do not double-feed email

If the workspace's Gmail is connected, Wilson already logs email with the people it knows. Import replies only for the channels Wilson cannot see: WhatsApp, SMS, LinkedIn, a separate sending address, and the like.

Message text policy

Whether Wilson **keeps the words** of imported replies and echoed sends is the workspace owner's decision, not the calling agent's: **Settings › CRM › Agent bridge › "Keep message text from the bridge"**. It is **on by default**, because the words come from your own agent about your own conversations, and keeping them is what lets the next draft build on what was actually said.

- **On** (the default): the text lands on the timeline where Wilson's drafters read it, and the response says `stored: "full"`. An agent can still tighten a single call with `metadata_only: true`.
- **Off**: Wilson logs that a reply came, on which channel, with its subject line, and nothing a drafter could quote. A body you send is discarded on the server and the response says `stored: "metadata_only"` with `stored_reason: "workspace_policy"`. Because there are then no words to tell one reply from the next, a metadata-only import requires `external_message_id` or `received_at`.

Workspaces whose agreement keeps message content out of Wilson are set to off, and any owner can switch it off at any time. Either way, this never changes how Wilson treats a mailbox it syncs itself: that stays metadata only.

The loop, end to end

```
# 1. pull today's recommendations
curl -s -H "Authorization: Bearer $WSK" \
  "$BASE/api/agent/v1/recommendations" | jq '.recommendations[0]'

# 2. send on your channel of choice, then tell Wilson
curl -s -X POST -H "Authorization: Bearer $WSK" -H 'Content-Type: application/json' \
  -d '{"outcome":"sent","channel":"whatsapp","external_message_id":"msg-1"}' \
  "$BASE/api/agent/v1/recommendations/$CARD_ID/outcome"

# 3. when they answer
curl -s -X POST -H "Authorization: Bearer $WSK" -H 'Content-Type: application/json' \
  -d '{"recommendation_id":"'$CARD_ID'", "channel":"whatsapp", "body":"Thanks, Thursday works!"}' \
  "$BASE/api/agent/v1/replies"
```

A signed-in person can also download the same list as a spreadsheet, without a key, from **Settings › CRM › Agent bridge**.