

SOAPify API v1 — Developer Documentation

Version: v1\ **Spec:** OpenAPI 2.0 (Swagger)\ **Base URL (dev):** `http://127.0.0.1:8000/\`
Consumes / Produces: `application/json` \ **Contact:** support@soapify.app

1) Overview

SOAPify provides endpoints to ingest clinical audio, transcribe it, extract structured **SOAP** drafts, manage dynamic checklists, finalize outputs, and share results with patients. The API also includes admin/ops tooling, analytics, hybrid search, and third-party integration helpers.

High-level capabilities

- **Auth:** Basic (optional) and JWT (access/refresh) flows.
 - **Encounters:** Create and read visit sessions.
 - **Uploads & Audio:** Chunked uploads or S3 presigned flows; commit and process.
 - **STT:** Batch transcribe by encounter or per-chunk.
 - **NLP:** Generate SOAP drafts; update sections and dynamic checklists.
 - **Outputs:** Finalize, list files, create patient links, and presign downloads.
 - **Checklist:** Catalog, templates, and per-encounter evaluations.
 - **Search:** Hybrid search + suggestions + analytics.
 - **Analytics:** System metrics, performance, and user activity.
 - **AdminPlus:** Health, logs, tasks, and exports.
-

2) Authentication & Authorization

SOAPify supports **HTTP Basic** (if enabled) and **JWT** via DRF SimpleJWT endpoints.

2.1 JWT token endpoints

- **POST** `/api/auth/token/` → obtain `{access, refresh}` \ **Body (JSON):** `{ "username": "<str>", "password": "<str>" }`
- **POST** `/api/auth/token/refresh/` → exchange `refresh` for new `access` \ **Body:** `{ "refresh": "<jwt>" }`
- **POST** `/api/auth/token/verify/` → verify a token \ **Body:** `{ "token": "<jwt>" }`

Use the access token in requests:

```
Authorization: Bearer <ACCESS_TOKEN>
```

2.2 Basic authentication (optional)

If Basic auth is enabled for your environment:

```
Authorization: Basic <base64(username:password)>
```

Note: Some operational endpoints under `/adminplus/` may require elevated privileges.

3) Common Conventions

- **Pagination:** List endpoints return `{count, next, previous, results}`.
- **Time:** Timestamps are ISO-8601.
- **Errors:** Standard HTTP codes; JSON bodies typically include a `detail` field.
- **Path Params:** Shown in `{curly}` braces.

4) Typical Workflows

4.1 Voice encounter → Finalized SOAP → Patient sharing

1. **Create encounter** `POST /api/encounters/create/`

2. **Upload audio** (choose one)

3. **Chunked:** `POST /api/uploads/session/create/` → `POST /api/uploads/chunk/`
(multipart) → `POST /api/uploads/commit/`

4. **S3 presigned:** `POST /api/uploads/s3/presign/` → client uploads to S3 → `POST /api/uploads/s3/confirm/`

5. **Dedicated audio presign:** `POST /api/audio/presigned-url/` → upload → `POST /api/audio/commit/`

6. **Transcribe**

7. All chunks by encounter: `POST /api/stt/encounter/{encounter_id}/process/`

8. Or single chunk: `POST /api/stt/transcribe/`

9. **Review transcript**

10. Full: `GET /api/stt/encounter/{encounter_id}/transcript/`

11. Per chunk: GET /api/stt/transcript/{audio_chunk_id}/

12. Manual edit: PUT /api/stt/transcript/{segment_id}/

13. **Generate SOAP draft**: POST /api/nlp/generate/{encounter_id}/

14. **Fetch & refine draft + checklist**

15. Draft: GET /api/nlp/drafts/{encounter_id}/

16. Update a section: PUT /api/nlp/drafts/{encounter_id}/update-section/

17. Checklist items: GET /api/nlp/drafts/{encounter_id}/checklist/

18. Mark item status: PUT /api/nlp/drafts/{encounter_id}/checklist/{item_id}/

19. **Finalize**

20. Trigger finalization: POST /api/outputs/finalize/

21. Get finalized SOAP: GET /api/outputs/finalized/{encounter_id}/

22. **Deliver to patient**

23. Create patient link: POST /api/outputs/link-patient/

24. Public access: GET /api/outputs/access/{link_id}/

25. List files for encounter: GET /api/outputs/files/{encounter_id}/

26. Presign download: POST /api/outputs/download/{file_id}/

4.2 Search & suggestions

- Search: GET /api/search/?
q=<query>&page=<n>&page_size=<m>&encounter_id=<id>&content_type=<t>&date_from=YYYY-MM-DD&date_to=YYYY-MM-DD
- Suggestions: GET /api/search/suggestions/?q=<prefix>&limit=<n>
- Search analytics: GET /api/search/analytics/?days=30

4.3 Checklist management

- Catalog: GET|POST /api/checklist/catalog/
- Template: GET|POST /api/checklist/templates/
- Link catalog items for template: GET /api/checklist/templates/{id}/catalog_items/
- Evaluate encounter: POST /api/checklist/evaluations/evaluate_encounter/
- Per-encounter evaluations: GET|POST /api/checklist/evaluations/
- Summary for encounter: GET /api/checklist/evaluations/summary/

4.4 Integrations & session windows

- Status across providers: GET /api/integrations/health/
 - OTP send/verify: POST /api/integrations/otp/send/, POST /api/integrations/otp/verify/
 - JWT window: GET /api/integrations/session/status/, POST /api/integrations/session/extend/
 - Helssa patients (read-only):\ GET /api/integrations/patients/search/, \ GET /api/integrations/patients/{patient_ref}/info/ (no PHI), \ POST /api/integrations/patients/{patient_ref}/access/
-

5) Endpoint Reference (Grouped)

5.1 Authentication

- POST /api/auth/login/ — Login; returns auth token (legacy).
- POST /api/auth/logout/ — Logout and delete auth token (legacy).
- POST /api/auth/token/ — Obtain JWT pair.
- POST /api/auth/token/refresh/ — Refresh access token.
- POST /api/auth/token/verify/ — Verify token validity.

5.2 Encounters

- GET /api/encounters/ — List user encounters.
- POST /api/encounters/create/ — Create an encounter.
- GET /api/encounters/{encounter_id}/ — Encounter details, including audio chunks.

5.3 Uploads & Audio

- POST /api/uploads/session/create/ — Create an upload session.
- POST /api/uploads/chunk/ — Upload a chunk (multipart/form-data).
- POST /api/uploads/commit/ — Commit uploaded chunks.
- GET /api/uploads/final/{session_id}/ — Final status for a session.
- POST /api/uploads/s3/presign/ — Presign S3 upload.
- POST /api/uploads/s3/confirm/ — Confirm S3 upload.
- POST /api/audio/presigned-url/ — Presign dedicated audio upload.
- POST /api/audio/commit/ — Commit audio file.

5.4 Speech-to-Text (STT)

- POST /api/stt/encounter/{encounter_id}/process/ — Transcribe all committed chunks.
- GET /api/stt/encounter/{encounter_id}/transcript/ — Full transcript (all chunks).
- POST /api/stt/transcribe/ — Transcribe a specific chunk.
- GET /api/stt/transcript/{audio_chunk_id}/ — Segments for a chunk.
- PUT /api/stt/transcript/{segment_id}/ — Update a segment text.

5.5 NLP — SOAP Drafts & Checklist

- **POST** `/api/nlp/generate/{encounter_id}/` — Start SOAP extraction.
- **GET** `/api/nlp/drafts/{encounter_id}/` — Get SOAP draft.
- **PUT** `/api/nlp/drafts/{encounter_id}/update-section/` — Update a section.
- **GET** `/api/nlp/drafts/{encounter_id}/checklist/` — Dynamic checklist for encounter.
- **PUT** `/api/nlp/drafts/{encounter_id}/checklist/{item_id}/` — Update item status.

5.6 Outputs — Files & Sharing

- **POST** `/api/outputs/finalize/` — Start finalization.
- **GET** `/api/outputs/finalized/{encounter_id}/` — Get finalized SOAP.
- **GET** `/api/outputs/files/{encounter_id}/` — List output files.
- **POST** `/api/outputs/download/{file_id}/` — Presign download URL.
- **POST** `/api/outputs/link-patient/` — Create patient link.
- **GET** `/api/outputs/access/{link_id}/` — Patient public access.

5.7 Checklist — Catalog, Templates, Evaluations

- **GET|POST** `/api/checklist/catalog/`
- **GET|PUT|PATCH|DELETE** `/api/checklist/catalog/{id}/`
- **GET|POST** `/api/checklist/templates/`
- **GET|PUT|PATCH|DELETE** `/api/checklist/templates/{id}/`
- **GET** `/api/checklist/templates/{id}/catalog_items/`
- **GET|POST** `/api/checklist/evaluations/`
- **GET** `/api/checklist/evaluations/summary/`
- **GET|PUT|PATCH|DELETE** `/api/checklist/evaluations/{id}/`
- **POST** `/api/checklist/evaluations/evaluate_encounter/`

5.8 Search

- **GET** `/api/search/` — Hybrid search (query, filters, pagination).
- **GET** `/api/search/reindex/` — (POST in spec) Reindex content for an encounter.
- **GET** `/api/search/suggestions/` — Prefix suggestions.
- **GET** `/api/search/analytics/` — Search analytics (by days).

Note: In the spec, `reindex` is **POST** with body `{ encounter_id }`.

5.9 Analytics

- **POST** `/api/analytics/activity/` — Record user activity.
- **GET** `/api/analytics/alerts/` — Active alerts.
- **POST** `/api/analytics/alerts/check/` — Evaluate alert rules.
- **POST** `/api/analytics/alerts/{alert_id}/acknowledge/` — Ack alert.
- **POST** `/api/analytics/business-metrics/` — Metrics for period.
- **POST** `/api/analytics/metric/` — Record custom metric.
- **GET** `/api/analytics/overview/` — System overview.
- **GET** `/api/analytics/performance/` — API performance (days).

- **GET** `/api/analytics/users/` — User analytics (user_id?, days).

5.10 AdminPlus — Ops & Tasks

- **GET** `/adminplus/api/health/` — System health.
- **GET** `/adminplus/api/logs/` — Operation logs.
- **GET** `/adminplus/api/tasks/` — Task monitoring data.
- **GET** `/adminplus/api/tasks/stats/` — Task execution statistics.
- **POST** `/adminplus/api/tasks/cancel/` — Cancel running task.
- **POST** `/adminplus/api/tasks/retry/` — Retry failed task.
- **POST** `/adminplus/api/export/` — Export system data.

5.11 Users

- **GET|POST** `/api/users/` — List or create users (paginated).
- **GET|PUT|PATCH** `/api/users/{id}/` — Retrieve or update a user.

5.12 Integrations & Session

- **GET** `/api/integrations/health/` — All external integrations health.
- **POST** `/api/integrations/logout/` — Revoke JWT window.
- **POST** `/api/integrations/otp/send/` — Send OTP.
- **POST** `/api/integrations/otp/verify/` — Verify OTP (create JWT window).
- **GET** `/api/integrations/session/status/` — Current session & remaining time.
- **POST** `/api/integrations/session/extend/` — Extend current JWT window.
- **GET** `/api/integrations/patients/search/` — Search patients (Helssa, read-only).
- **GET** `/api/integrations/patients/{patient_ref}/info/` — Basic patient info (no PHI).
- **POST** `/api/integrations/patients/{patient_ref}/access/` — Request access to patient data.

6) Request & Response Examples

6.1 Obtain JWT tokens

```
curl -X POST \
  http://127.0.0.1:8000/api/auth/token/ \
  -H 'Content-Type: application/json' \
  -d '{"username":"doctor1","password":"....."}'
```

200 OK

```
{
  "access": "<jwt_access>",
```

```
"refresh": "<jwt_refresh>"
}
```

6.2 Create an encounter (minimal)

```
curl -X POST \
  http://127.0.0.1:8000/api/encounters/create/ \
  -H 'Authorization: Bearer <ACCESS>'
```

201 Created (example)

```
{ "encounter_id": "enc_12345" }
```

6.3 Transcribe all chunks for an encounter

```
curl -X POST \
  http://127.0.0.1:8000/api/stt/encounter/enc_12345/process/ \
  -H 'Authorization: Bearer <ACCESS>'
```

6.4 Generate SOAP draft

```
curl -X POST \
  http://127.0.0.1:8000/api/nlp/generate/enc_12345/ \
  -H 'Authorization: Bearer <ACCESS>'
```

6.5 Get finalized SOAP

```
curl -X GET \
  http://127.0.0.1:8000/api/outputs/finalized/enc_12345/ \
  -H 'Authorization: Bearer <ACCESS>'
```

Note: Bodies for some POST/PUT endpoints are implementation-specific and may be extended in server code. Where a schema is defined in the spec, it is documented below.

7) Data Models (Schemas)

7.1 Authentication

- **TokenObtainPair**
- `username` (*string, required*)

- `password` (string, required)
- **TokenRefresh**
- `refresh` (string, required) → returns `access` (string)
- **TokenVerify**
- `token` (string, required)

7.2 ChecklistCatalog

Fields: `id` (ro), `title`, `description`, `category` [subjective|objective|assessment|plan|general], `priority` [low|medium|high|critical], `keywords` (object), `question_template`, `is_active`, `created_at` (ro), `updated_at` (ro)

7.3 ChecklistEval

Fields: `id` (ro), `encounter` (int), `catalog_item` (ChecklistCatalog), `catalog_item_id` (int), `status` [covered|missing|partial|unclear], `confidence_score` (0.0-1.0), `evidence_text` (str), `anchor_positions` (object), `generated_question` (str), `notes` (str), `created_at` (ro), `updated_at` (ro)

7.4 ChecklistTemplate

Fields: `id` (ro), `name`, `description`, `specialty`, `is_default`, `catalog_items_count` (ro), `created_at` (ro), `updated_at` (ro)

7.5 User & UserCreate

- **User:** `id` (ro), `username`, `email`, `first_name`, `last_name`, `role` [doctor|admin], `phone_number`, `updated_at` (ro)
- **UserCreate:** `username`, `password`, `email?`, `first_name?`, `last_name?`, `role?`, `phone_number?`

8) Query Parameters (Selected)

- **Pagination:** `page`, `page_size`
- **Search:** `q`, `encounter_id`, `content_type` (transcript|soap|checklist|notes), `date_from`, `date_to`
- **Analytics:** `days` (defaults vary by endpoint)

9) Operational & Admin Endpoints

- **Health:** `/adminplus/api/health/` (GET) — service readiness.
- **Logs:** `/adminplus/api/logs/` (GET) — operational logs.
- **Tasks:** list `/adminplus/api/tasks/`, stats `/adminplus/api/tasks/stats/`, cancel `/adminplus/api/tasks/cancel/` (POST), retry `/adminplus/api/tasks/retry/` (POST)

- **Export:** `/adminplus/api/export/` (POST) — export system data.

Security: These typically require administrative credentials and should not be exposed publicly.

10) Status Codes

- **200 OK** — Successful read/update.
 - **201 Created** — Resource created / job enqueued.
 - **204 No Content** — Deleted / empty success.
 - **400 Bad Request** — Invalid input.
 - **401 Unauthorized** — Missing/invalid credentials.
 - **403 Forbidden** — Insufficient permissions.
 - **404 Not Found** — Resource not found.
 - **409 Conflict** — State conflict.
 - **422 Unprocessable Entity** — Validation failed.
 - **429 Too Many Requests** — Throttled.
 - **5xx** — Server error.
-

11) Security & Compliance Notes

- **PHI Handling:** `/api/integrations/patients/{patient_ref}/info/` returns *no PHI*; access to detailed patient data requires explicit access requests.
 - **Links:** Patient links (`/api/outputs/link-patient/`) should be time-bound and scoped.
 - **Tokens:** Treat JWTs as secrets; renew using refresh tokens and session extension endpoints.
-

12) Changelog & Versioning

- **v1** (current): endpoints as documented in this guide. Future breaking changes will be versioned.
-

13) Appendix — Curl Snippets

13.1 Search with filters

```
curl -G http://127.0.0.1:8000/api/search/ \
-H 'Authorization: Bearer <ACCESS>' \
--data-urlencode 'q=chest pain' \
--data-urlencode 'content_type=transcript' \
--data-urlencode 'date_from=2025-01-01' \
```

```
--data-urlencode 'page=1' \  
--data-urlencode 'page_size=20'
```

13.2 Checklist evaluation for an encounter

```
curl -X POST \  
  http://127.0.0.1:8000/api/checklist/evaluations/evaluate_encounter/ \  
  -H 'Authorization: Bearer <ACCESS>' \  
  -H 'Content-Type: application/json' \  
  -d '{"encounter": 123, "catalog_item_id": 10, "status": "covered"}'
```

13.3 Presign + commit (S3 path)

```
curl -X POST http://127.0.0.1:8000/api/uploads/s3/presign/ \  
  -H 'Authorization: Bearer <ACCESS>' -H 'Content-Type: application/json' \  
  -d '{"filename": "rec.wav", "content_type": "audio/wav"}'  
# → upload to returned URL, then  
curl -X POST http://127.0.0.1:8000/api/uploads/s3/confirm/ \  
  -H 'Authorization: Bearer <ACCESS>' -H 'Content-Type: application/json' \  
  -d '{"key": "..."}'
```

14) Glossary

- **Encounter:** A visit/session identifier used to correlate audio, transcript, SOAP drafts, checklists, and outputs.
- **SOAP:** Subjective, Objective, Assessment, Plan clinical note structure.
- **STT:** Speech-to-Text.
- **NLP Draft:** Auto-generated SOAP content prior to clinician edits & finalization.

Next steps: Integrate gradually—start with auth, then create an encounter, upload & transcribe a short audio, generate a draft, and inspect the finalized outputs before enabling patient links in production.