RadioWeb

Administration Guide

Version 1.1.10, 2026-11-24

Contents

Overview 1
- The Chapters at a Glance 1
- Terms 1

Installation 2
- System Requirements 2
- Initial Setup 3

Signing In and Overview 6
- Signing In 6
- Navigation 6
- Overview (Dashboard) 6

Sign-in & Security 8
- Setting up a second factor 8
- Backup codes 12
- Signing in with a second factor 12
- Managing factors 14
- Lost device 14
- Resetting a user’s sign-in factors 14
- Emergency: no administrator can sign in 15
- Operating notes 15

Master Data 16
- Customers 16
- Referrers 17
- Users 18

Tokens 21
- Integration Tokens 21
- QR Code Import 22
- Releases 23

Admin 26
- Import Status 26
- Print Templates 26
- Language Management 40
- Audit Log 41
- Archives 42
- HL7 Communication 43
- DirectView 52
- Licence & Modules 57
- Settings 59

Customer Area 61
- Releases 61

Creating Access Codes . . . 61
Uploading Prior Examinations (Upload) . . . 63
Portal Import (DLX) . . . 64
ZIP Upload . . . 64
Tracking progress . . . 64
Setup . . . 65
Operation . . . 65
Integration . . . 66
Overview . . . 66
PACS: VEPRO . . . 67
RIS: Mesalvo . . . 69
GDT/HL7: gdt-hl7-bridge . . . 71
Portal-Launch . . . 97

Overview

This guide is intended for administrators of the Patient Portal and for customers (practice or clinic) who operate the portal for their referrers and patients.

The Chapters at a Glance

Installation

System requirements and the form for the initial setup.

Signing In and Overview

Signing in, structure of the navigation and the dashboard with the key figures.

Master Data

Customers, Referrers and Users with roles.

Tokens

Integration tokens for external systems, QR code import and release management.

Admin

Import Status, Print Templates, Language Management, Audit Log, Archives and Settings.

Customer Area

Features for practices and clinics.

Integration

Connecting PACS (VEPRO), RIS (Mesalvo) and practice management systems (gdt-hl7-bridge, Portal-Launch).

Terms

Term	Meaning
Customer	Radiology practice/clinic or site for which the portal is operated
Referrer	Referring physician or practice
Release / access code (token)	Time-limited access to examinations in the format `XXXX-XXXX-XXXX`, usually with a QR code
Integration token	Long-lived access key for an external system (PACS, practice management system)
DICOM value	Value in the DICOM image data that assigns customers (InstitutionName) and referrers (ReferringPhysicianName) to their studies

Installation

System Requirements

Here you will find the recommended system requirements for the patient portal, based on the number of concurrent users and the selected operation mode.

Hardware Scenarios

The hardware requirements depend primarily on the number of concurrent users.

Scenario	Concurrent Users	RAM	CPU
Small	5–10	8 GB	4 vCPU
Medium	10–20	16 GB	6–8 vCPU
Higher	20–50	32 GB	12–16 vCPU

Storage Requirements

Mode: Query/Retrieve (Recommended)

It is generally recommended that the patient portal operates in **Query/Retrieve (Q/R)** mode. In this mode, all data is requested directly from the PACS as needed.

- **Storage Requirement**: Serves only as a cache.
- **Recommendation**: approx. **1 TB** (500 GB System + 500 GB Cache).

Mode: DICOM Store (Data Storage in Portal)

If data is actively sent to the patient portal and stored there, the following estimates for storage requirements apply.

Retention Period 3 Months (~90 Days)

Size	Data Volume	Raw Data (90 Days)	Recommended SSD (approx.)
Small	5–10 GB/Day	450–900 GB (0.45–0.9 TB)	**2 TB**
Medium	20–50 GB/Day	1.8–4.5 TB	**6 TB**
Large	50–100 GB/Day	4.5–9 TB	**12 TB**

Retention Period 6 Months (~180 Days)

Size	Data Volume	Raw Data (180 Days)	Recommended SSD (approx.)
Small	5–10 GB/Day	900–1,800 GB (0.9–1.8 TB)	**4 TB**
Medium	20–50 GB/Day	3.6–9 TB	**12 TB**
Large	50–100 GB/Day	9–18 TB	**24 TB**

Initial Setup

Please fill out the following form for the initial setup of the patient portal.

1. Network Data

Network data for the patient portal in the local network:

Parameter	Value
IP Address	
Subnet Mask	
Gateway	
DNS Server	

2. DICOM Parameters / Operation Modes

DICOM Services (Store / ECHO)

Parameter	Default Value	Your Value
TCP Port for DICOM Services of the Portal	4104	

AET for Patient Portal

Parameter	Default Value	Your Value
AET Title	RADIOWEB	

Operation Modes

Please select the desired operation mode (a or b).

a) Query/Retrieve (Recommended)

The portal actively retrieves images from the PACS.

Parameter	Value

IP Address (Remote PACS)	
Port (Remote PACS)	
AET (Remote PACS)	

b) DICOM Store

The PACS actively sends images to the portal.

Parameter	Value
AET for DICOM Store (Remote PACS)	

3. Mail Settings

For sending emails (e.g., token distribution), the following SMTP data is required:

Parameter	Description	Your Value
`email.smtp.host`	SMTP Server Hostname (e.g. smtp.gmail.com)	
`email.smtp.port`	SMTP Server Port (e.g. 587)	
`email.smtp.username`	SMTP Username	
`email.smtp.password`	SMTP Password	
`email.from.address`	Sender Email Address	
`email.from.name`	Sender Name	

4. Token Security

Configuration of permissions for access token creation.

Permissions

Who can create?	Referrer	Patient	Study	None
PACS User (Customer)	☐	☐	☐	☐
Referrer	☐	☐	☐	☐
Patient	☐	☐	☐	☐
Study Access (Study)	☐	☐	☐	☐

Legend:

- **Referrer**: All patients and studies of a referrer
- **Patient**: All studies of a patient (without needing to create a new share for each study)
- **Study**: Share at study level

- **None**: Cannot create further shares

Configuration Options

For each token type (Patient, Referrer, Study), the following parameters can be set:

Patient

Setting	Editable (User decides)	Fixed Default Value (if not editable, please choose only ONE option)
Authentication	☐ Yes	☐ Date of Birth ☐ Password ☐ None
Standard Validity	☐ Yes	☐ Unlimited ☐ 7 Days ☐ 14 Days ☐ 30 Days ☐ 60 Days ☐ 90 Days ☐ 180 Days ☐ 365 Days

Referrer

Setting	Editable (User decides)	Fixed Default Value (if not editable, please choose only ONE option)
Authentication	☐ Yes	☐ Password ☐ None
Standard Validity	☐ Yes	☐ Unlimited ☐ 7 Days ☐ 14 Days ☐ 30 Days ☐ 60 Days ☐ 90 Days ☐ 180 Days ☐ 365 Days

Study

Setting	Editable (User decides)	Fixed Default Value (if not editable, please choose only ONE option)
Authentication	☐ Yes	☐ Date of Birth ☐ Password ☐ None
Standard Validity	☐ Yes	☐ Unlimited ☐ 7 Days ☐ 14 Days ☐ 30 Days ☐ 60 Days ☐ 90 Days ☐ 180 Days ☐ 365 Days

Example: A patient wants to share a study. They create a `Study` token. The system checks `security.group.patient.allowedtokens` (includes `study`). The authentication method is determined by `token.type.study.authentication` (e.g., `BIRTHDATE` requires the patient's date of birth for verification). The validity is pre-set by `token.type.study.validity_days`.

Signing In and Overview

Signing In

Administrators sign in in the **Login** area of the login page with **e-mail** and **password** (if the login page only shows the access code, via the link **Sign in with email and password** below the card) and confirm the sign-in with a second factor (passkey, authenticator app or backup code, see Sign-in & Security). At the first sign-in you set up this second factor. The first administrator account is created during installation (see the installation guide in the developer documentation); create further accounts under Users.

Navigation

The header groups the administration features into three menus:

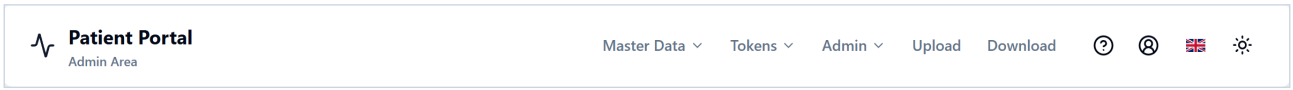


Figure 1. Header in the admin area

Menu	Entries
Master Data	Customers, Referrers, Users
Tokens	Integration Tokens, QR Code Import, Releases
Admin	Import Status, Print Templates, Language Management, Audit Log, Archives, Settings

In addition, **Download** gives you access to the desktop and integration apps, sample templates and the manuals.

On the right of the header there are four icons:

- **Help** (question mark): quick guide, FAQ, manuals and contact,
- **Account menu** (person icon): your e-mail address, Sign-in & Security and **Logout**,
- **Language** (flag) and **Light/Dark**.

Imprint and privacy are at the bottom of every page, the version number (with a link to the release notes) in the top right above the header.

Overview (Dashboard)

After signing in you see the **Overview** with the key figures.

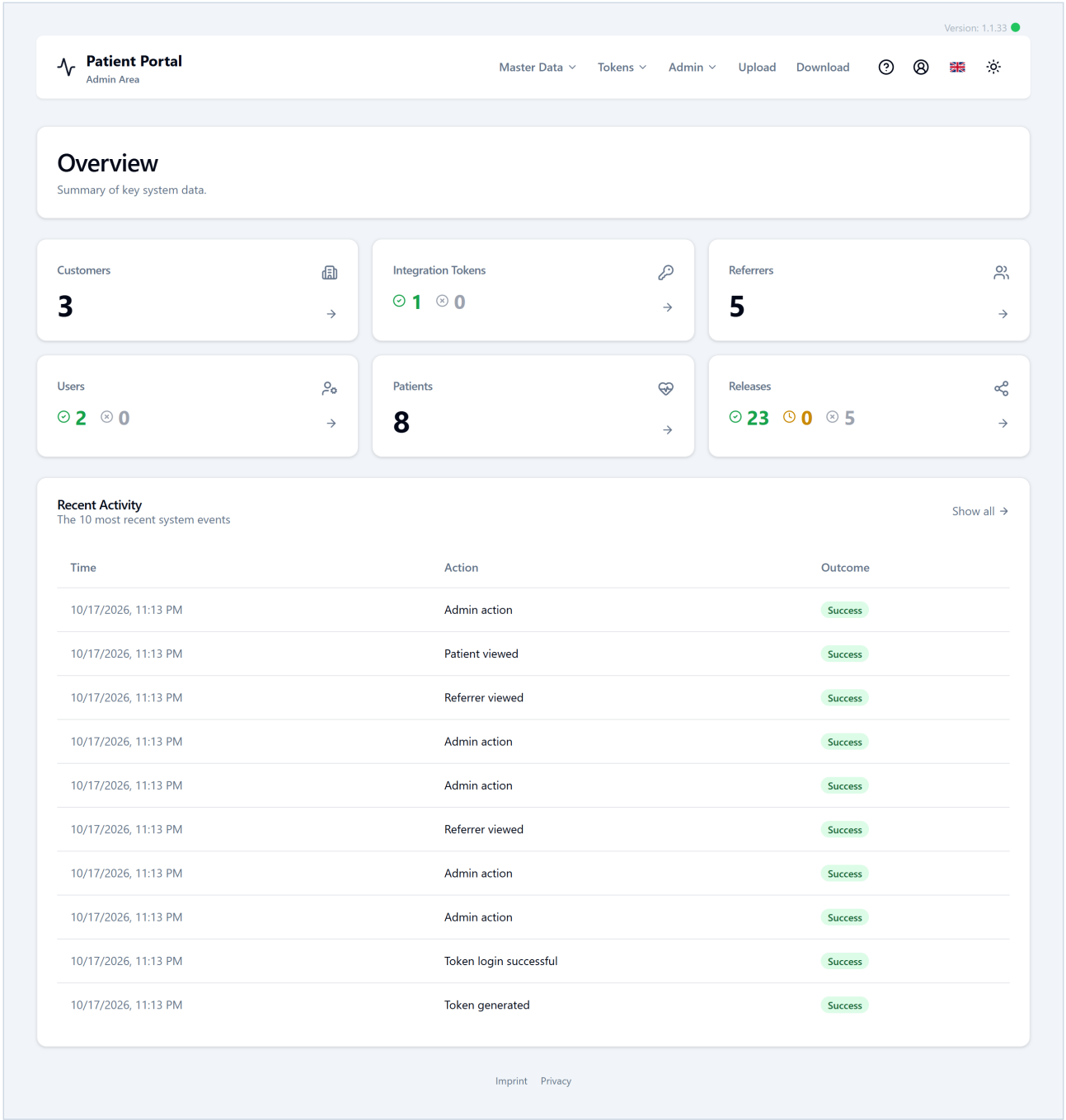


Figure 2. Overview after signing in

- The tiles show the number of **customers**, **referrers**, **patients**, **users**, **integration tokens** and **releases** – for tokens, users and releases split into active (green), expired (yellow) and revoked or inactive (grey). Clicking a tile opens the corresponding management view.
- **Recent Activity** lists the ten newest entries of the audit log; **Show all** opens the full log.

Sign-in & Security

All sign-ins with email address and password – administrators, customer logins of the practices and permanent logins of referrers and patients – require a second factor (NIS2, Art. 21(2)(j)). Sign-ins with an access code, a study share or an integration token are not user accounts and are not affected.

To protect health data, the password alone is not enough for signing in with **e-mail and password**: every account needs a **second factor**. Someone who knows your password still cannot sign in with it.

> Signing in with an **access code** (share code, token) is not affected and works as before.

You can choose between:

Passkey (recommended)

The sign-in is confirmed with your fingerprint, face recognition or your device PIN – no code to type and no additional app. Passkeys work with current smartphones, tablets and computers (Windows Hello, Apple Touch ID/Face ID, Android) and with security keys. You can add several passkeys, for example for your smartphone and your workplace PC.

Passkeys are considered particularly secure because they are protected against phishing: a passkey only works on the real address of the portal and therefore cannot be stolen through a fake website – unlike a password or a typed-in code. Your fingerprint and face are only used to unlock the passkey on your device; they never leave the device and are not sent to the portal.

Authenticator app

A free app on your smartphone generates a new 6-digit code every 30 seconds, which you enter when signing in. The portal recommends **Microsoft Authenticator** and **Google Authenticator**; any other app following the common standard (TOTP) works as well.

With the first factor you also receive ten **backup codes** for emergencies.

Setting up a second factor

When you first sign in with e-mail and password, the portal automatically opens the page **Sign-in & security** with the notice **Please set up a second sign-in factor**. It briefly explains why this is necessary and introduces the two options. As long as no second factor is set up, all other areas are locked. Later you can always reach the page via the account menu (person icon in the top right) under **Sign-in & security**.

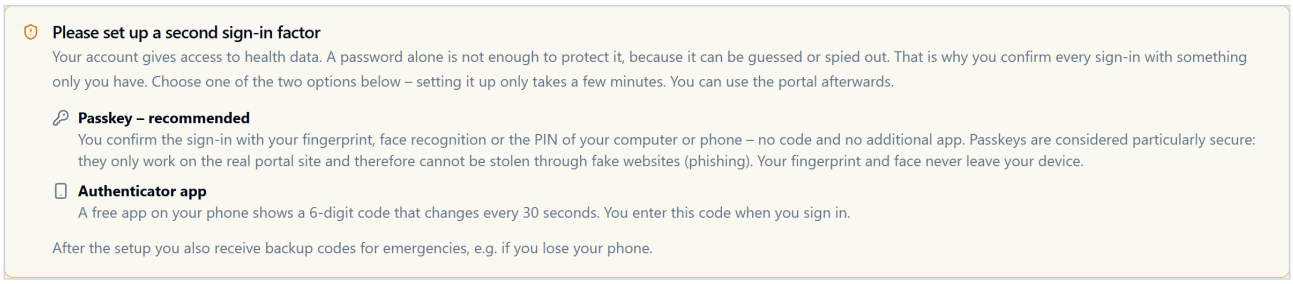


Figure 3. Notice at the first sign-in

Set up one of the two factors – or both, then you have a fallback.

Adding a passkey

1. Open **Sign-in & security** on the device you want to sign in with from now on (computer, smartphone or tablet).
2. Under **Passkeys**, click **Add passkey**.
3. Keep or change the suggested name (for example "office laptop") – it helps you tell the passkeys apart later. Enter your current password and click **Create passkey**.

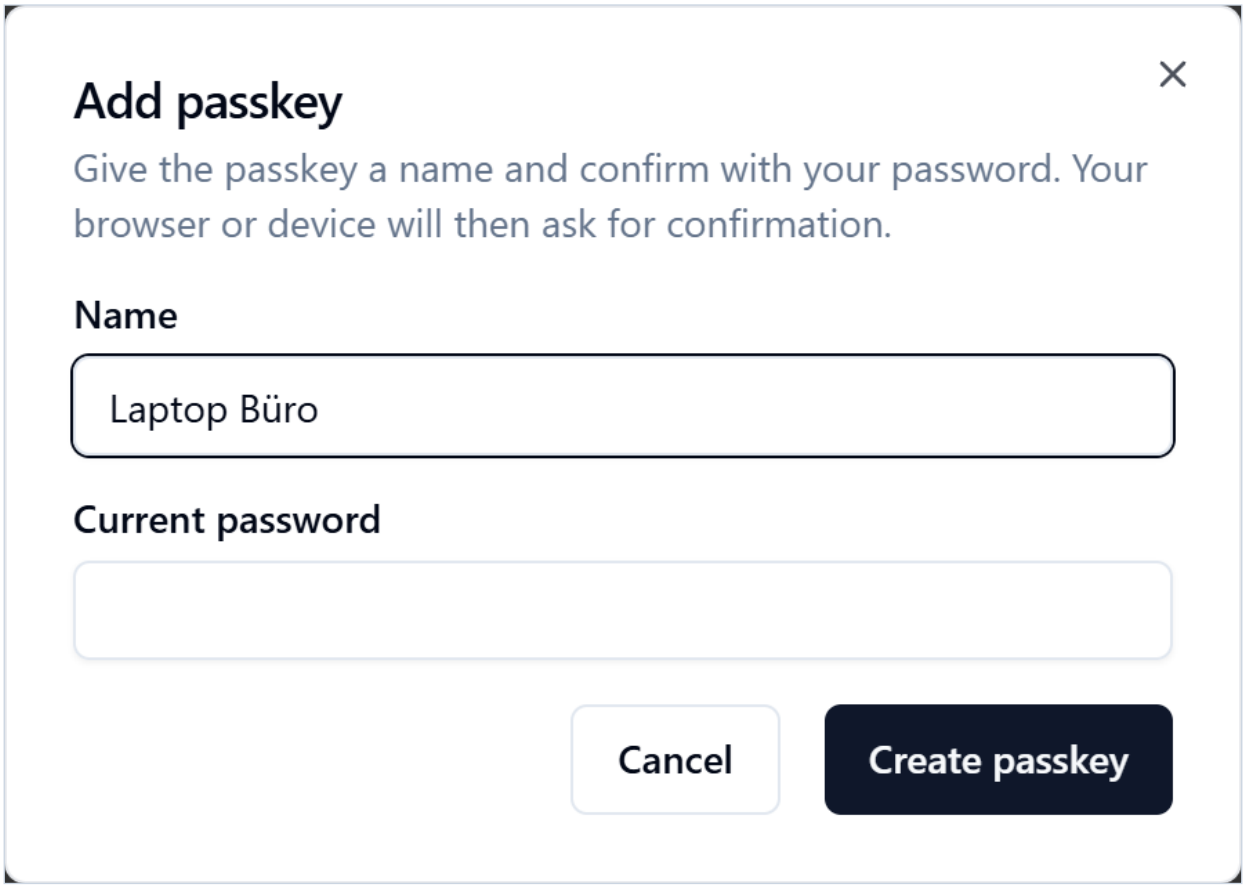


Figure 4. Adding a passkey

4. Your browser or operating system now opens its own window. Depending on the device:
   - **Windows:** confirm with Windows Hello (face, fingerprint or Windows PIN).
   - **Mac, iPhone, iPad:** confirm with Touch ID, Face ID or the device passcode. The passkey is stored in the iCloud Keychain.
   - **Android:** confirm with fingerprint, face or screen lock. The passkey is stored in Google Password Manager.
   - **Storing the passkey on your phone while working at the computer:** choose "Use another device" or "Phone or tablet" in the window and scan the QR code shown with your phone.
5. The new passkey appears in the list. The label **synced** means it is also available on your other devices via your Apple or Google account.

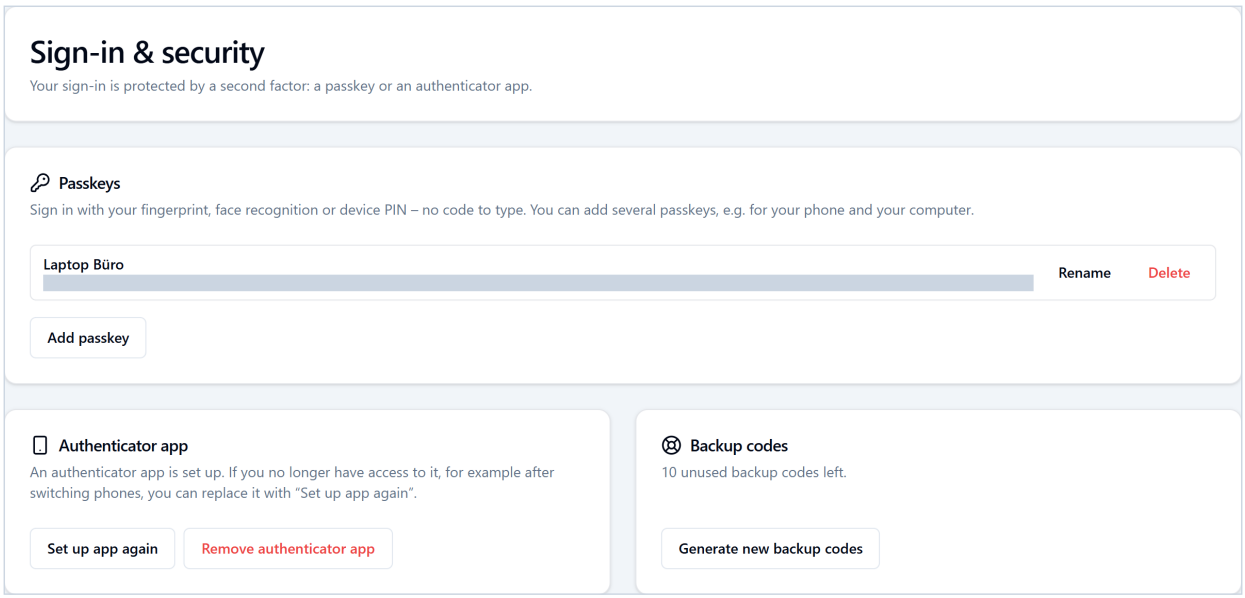


Figure 5. Passkeys, authenticator app and backup codes on the page Sign-in & security

Add a passkey for each device you regularly sign in with – or use a synced passkey. If the browser asks where to store it, your device's password manager is the easiest choice for most users.

Setting up an authenticator app

1. Under **Authenticator app**, click **Set up authenticator app** and enter your current password. The setup guides you through three steps.

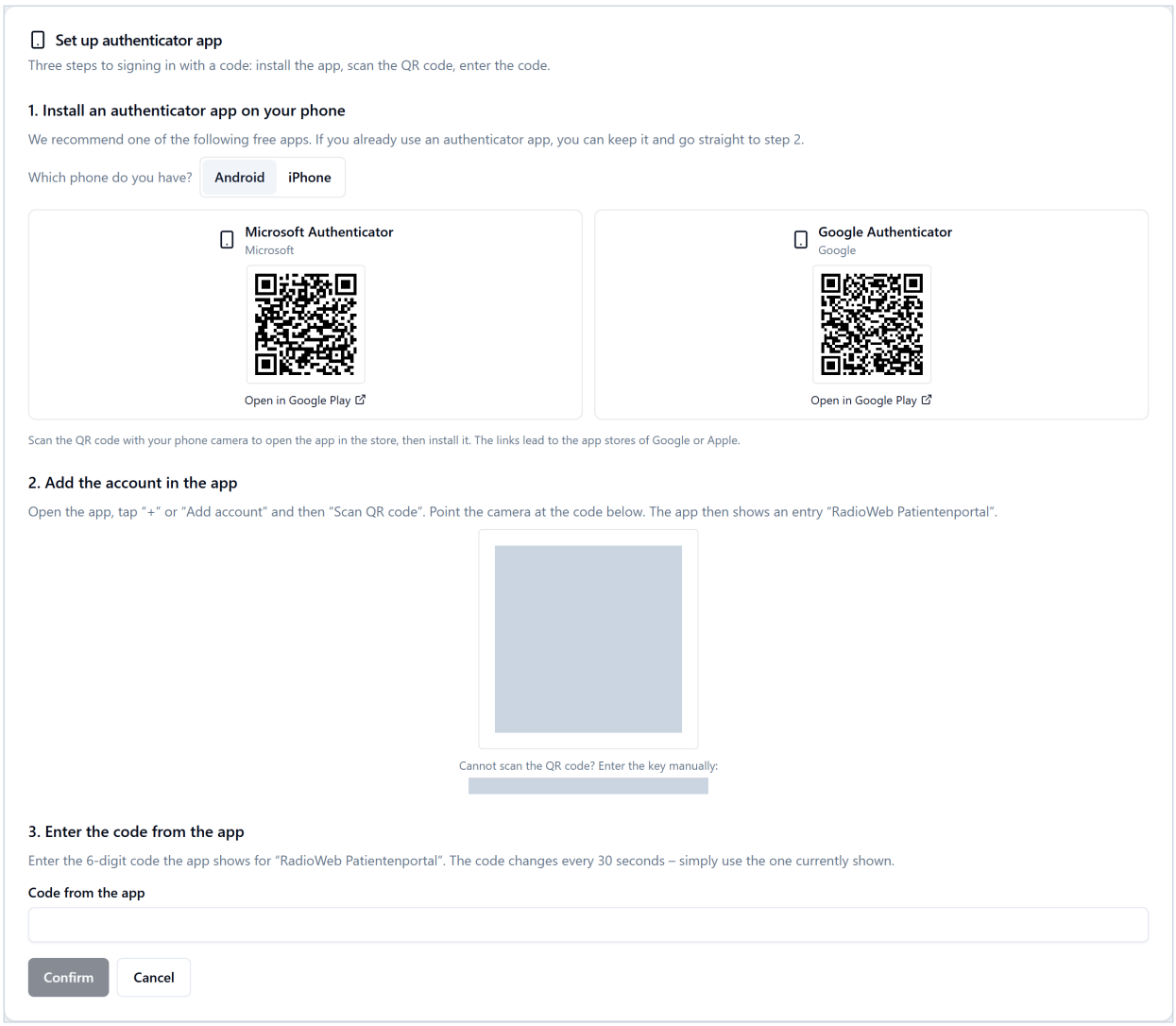


Figure 6. Setting up the authenticator app (QR code and key are hidden here)

2. **Install the app:** the portal recommends two free apps available for Android and iPhone: **Microsoft Authenticator** and **Google Authenticator**. Choose your phone (Android or iPhone) and scan the QR code of the app you want with your phone camera – it opens the app in Google Play or the App Store. If you open the portal on the phone itself, the links lead to the store directly. You can keep using an authenticator app you already have.
3. **Add the account:** in the app, tap "+" or "Add account" and scan the QR code in the portal. If the code cannot be scanned, enter the key shown below it manually in the app. On a phone, **Open directly in the authenticator app** opens the app without scanning. The app then shows the entry "RadioWeb Patientenportal".
4. **Enter the code:** enter the 6-digit code currently shown in the app and click **Confirm**. The code changes every 30 seconds – simply use the current one.



Choose an app with a backup function (with Microsoft and Google via your account) so the entries are kept when you get a new phone.

Backup codes

After the first factor, the portal shows ten backup codes once (format `XXXXX-XXXXX`). With a backup code you can sign in when you cannot use your passkey or app – for example after losing your smartphone.

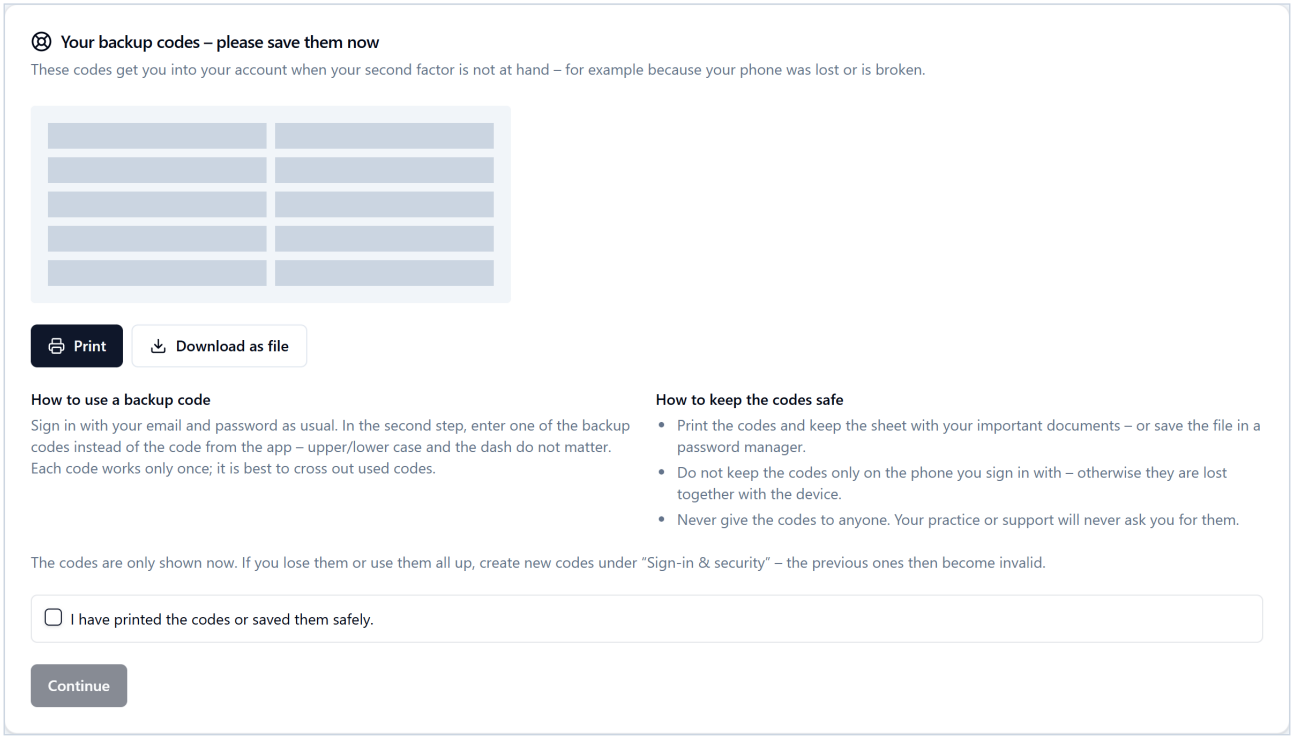


Figure 7. Backup codes after the setup (codes hidden here)

Using a backup code

Sign in with e-mail and password as usual and, in the second step, enter a backup code instead of the app code. Upper/lower case and the dash do not matter. Each code works exactly once – it is best to cross out used codes.

Keeping backup codes safe

- Click **Print** and keep the sheet with your important documents – or download the codes with **Download as file** and save the file in a password manager. The printout and the file also contain short instructions.
- Do not keep the codes only on the phone you sign in with – otherwise they are lost together with the device.
- Never give the codes to anyone. Your practice or support will never ask for your backup codes.

Only after you confirm that you have printed or safely saved the codes does **Continue** take you back to the overview. The page then shows how many unused codes you have left. **Generate new backup codes** gives you a new set; the previous codes become invalid.

Signing in with a second factor

1. Enter your e-mail address and password as usual and click **Sign in**.
2. In the step **Confirmation required**, click **Use passkey** and confirm the prompt of your device – or

enter the code from the authenticator app or a backup code and click **Confirm**.

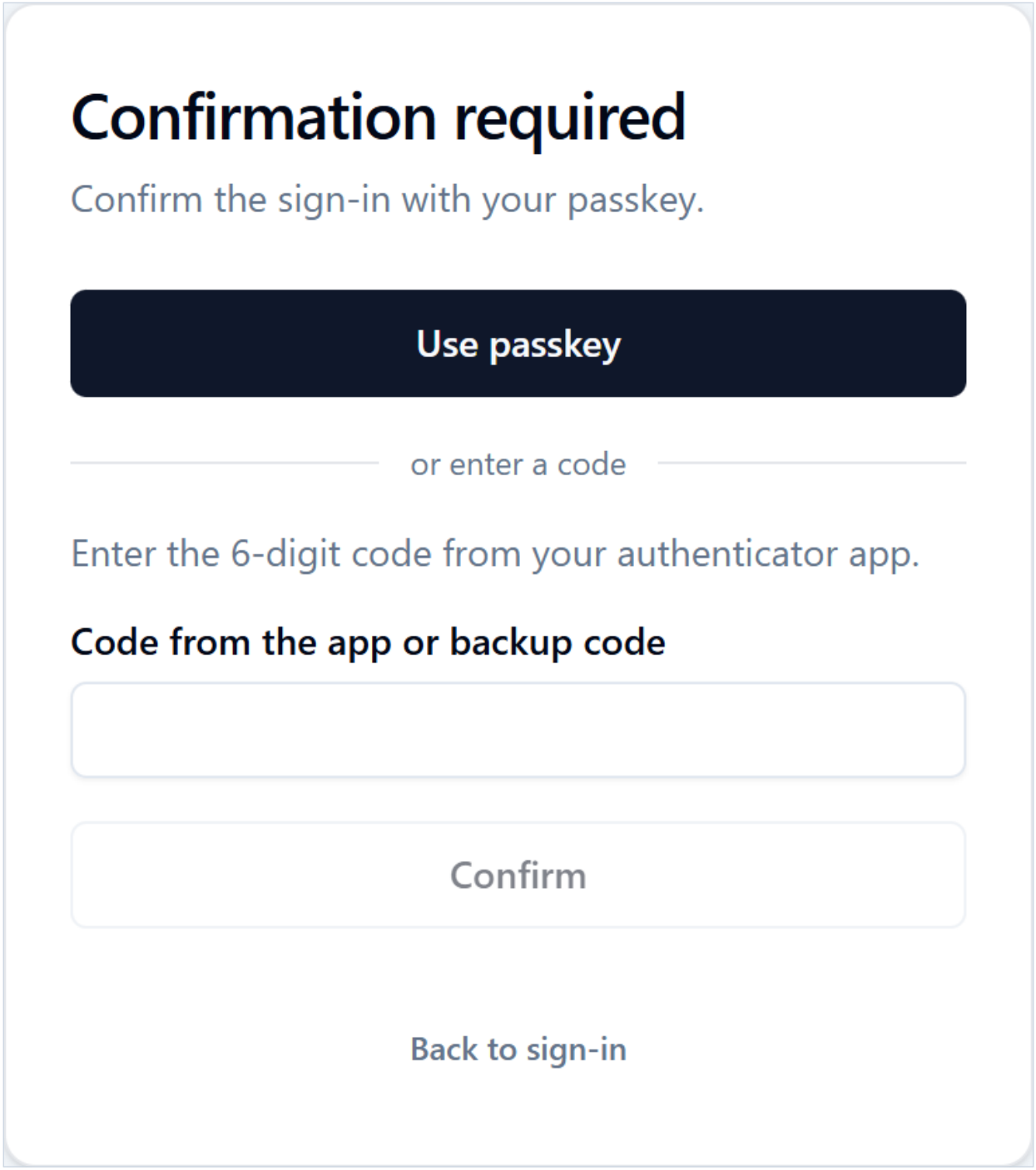


Figure 8. Second step of the sign-in

With a passkey you can also sign in without a password: click **Sign in with a passkey** on the login page or pick the passkey directly from the suggestion of the e-mail field. If the passkey is on your phone but you sign in at the computer, choose "Use another device" in the browser window and scan the QR code with your phone.

After five failed attempts (wrong password or wrong code), the account is locked temporarily: first for 15 minutes, longer after further failed attempts (at most 4 hours).

Managing factors

- **Rename** and **Delete** for individual passkeys and **Remove authenticator app** are on the page **Sign-in & security**.
- The last remaining factor cannot be removed – set up another one first. This way your account is never protected by the password alone.
- **Set up app again** replaces the authenticator app, for example after switching phones. The setup works as the first time; the previous app stays valid until you have confirmed the code of the new app. After that, only codes of the new app are accepted and your sign-ins on other devices are ended.
- Changes to the factors require your password. If your last confirmation was more than ten minutes ago, the portal also asks for a passkey or code (**Confirm your identity**).
- When a factor is removed, all other sessions of your account are ended.
- You receive an e-mail about every change – in the language of your last sign-in. If you did not make the change yourself, contact your practice or the administrator immediately.

Lost device

1. Sign in with e-mail, password and a **backup code**.
2. Open **Sign-in & security** and replace the lost factor:
    - **Authenticator app:** under **Authenticator app**, click **Set up app again** and set up the app on the new phone. Codes of the lost phone are no longer valid afterwards.
    - **Passkey:** first add a new passkey (or set up an authenticator app), then delete the passkey of the lost device.
3. Check how many backup codes you have left and generate new ones if needed.

If you have no backup code left, the portal administrator can reset your sign-in factors. At the next sign-in you then set up a new second factor.

If someone else changes your factors, you receive an e-mail – contact your practice or the administrator immediately.

Resetting a user’s sign-in factors

If a user lost their device and their backup codes, reset their sign-in factors under Users:

1. Verify the person’s identity through a second channel (for example by calling back the known phone number).
2. Click the shield icon **Reset sign-in factors** in the user’s row and confirm.

All passkeys, the authenticator app and the backup codes are removed, and running sessions end immediately. At the next sign-in the user must set up a new second factor. The column **Second factor** of the user list shows which factors a user has set up; accounts without a factor are marked **none**. The action is recorded in the audit log (**Sign-in factors reset**).

Emergency: no administrator can sign in

If all administrators have lost their second factor and their backup codes, the operator resets the factors of one administrator directly in the database. Replace `admin@example.com` with the administrator's email address:

```
-- docker compose exec postgres psql -U postgres -d patientenportal
SET search_path TO patientenportal;
BEGIN;
DELETE FROM user_passkey      WHERE user_id = (SELECT id FROM users WHERE email =
'admin@example.com');
DELETE FROM user_backup_codes WHERE user_id = (SELECT id FROM users WHERE email =
'admin@example.com');
UPDATE users SET two_factor_method = 'NONE', totp_secret = NULL,
two_factor_confirmed_at = NULL,
                token_version = token_version + 1
 WHERE email = 'admin@example.com';
COMMIT;
```

At the next sign-in the administrator sets up a new second factor.

Operating notes

- **Passkeys are bound to the domain** under which the portal is opened. It is set via `WEBAUTHN_RP_ID` (domain) and `WEBAUTHN_ORIGINS` (address including `https://`); without them, the first entry of `CORS_ALLOWED_ORIGINS` is used. If the domain changes later, the existing passkeys stop working – users then sign in with the authenticator app or a backup code and add new passkeys.
- The secrets of the authenticator apps are stored encrypted with `FIELD_ENCRYPTION_KEY` (the installation script generates the key automatically). Keep the key together with the backup – without it, the configured authenticator apps are unusable after a restore.
- Security notices by e-mail (factor added, removed, reset) are sent via the e-mail delivery configured in the settings.

Master Data

Customers

A **customer** is a radiology practice or clinic, or a site, for which the portal provides examinations. Customers group referrers, users, releases, integration tokens and their own print templates.

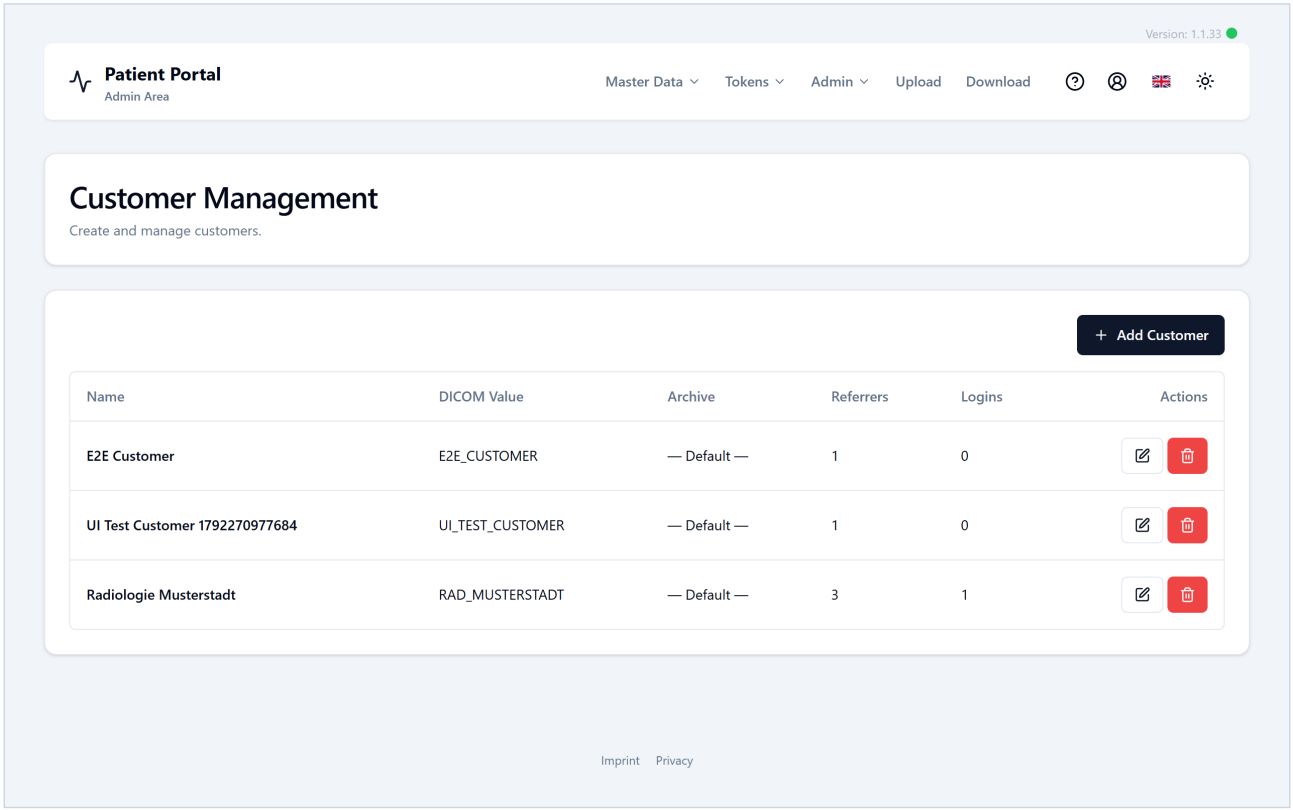


Figure 9. Customer management

For each customer the table shows name, DICOM value, assigned archive and the number of referrers and logins.

Creating and Editing Customers

1. Click **Add Customer** (or the pencil icon for an existing customer).
2. Fill in the fields:

Field	Meaning
Customer Name	Display name, e.g. "Radiology Sampletown".
DICOM Value	Value of the DICOM attribute **InstitutionName** (0008,0080) that marks this customer's studies in the PACS. Among other things it is used as a filter for PACS queries and must match the entry in the image data exactly.
Archive	PACS archive from which this customer's studies are retrieved. — *Default* — uses the default archive from archive management.

3. Save.



The licence defines how many customers can be created (one without a licence). Once the quota is reached, **Add Customer** is locked; existing customers stay usable.

Deleting Customers

Click the red trash icon and confirm the security prompt.



Before deleting, check whether referrers, users or integration tokens are still assigned to the customer.

Referrers

Referrers are the referring physicians and practices. They access the examinations of their patients through the referrer portal (see the Referrer Guide).



The referrer portal is a licensed module. While it is not active, the menu item **Master Data → Referrers** is hidden.

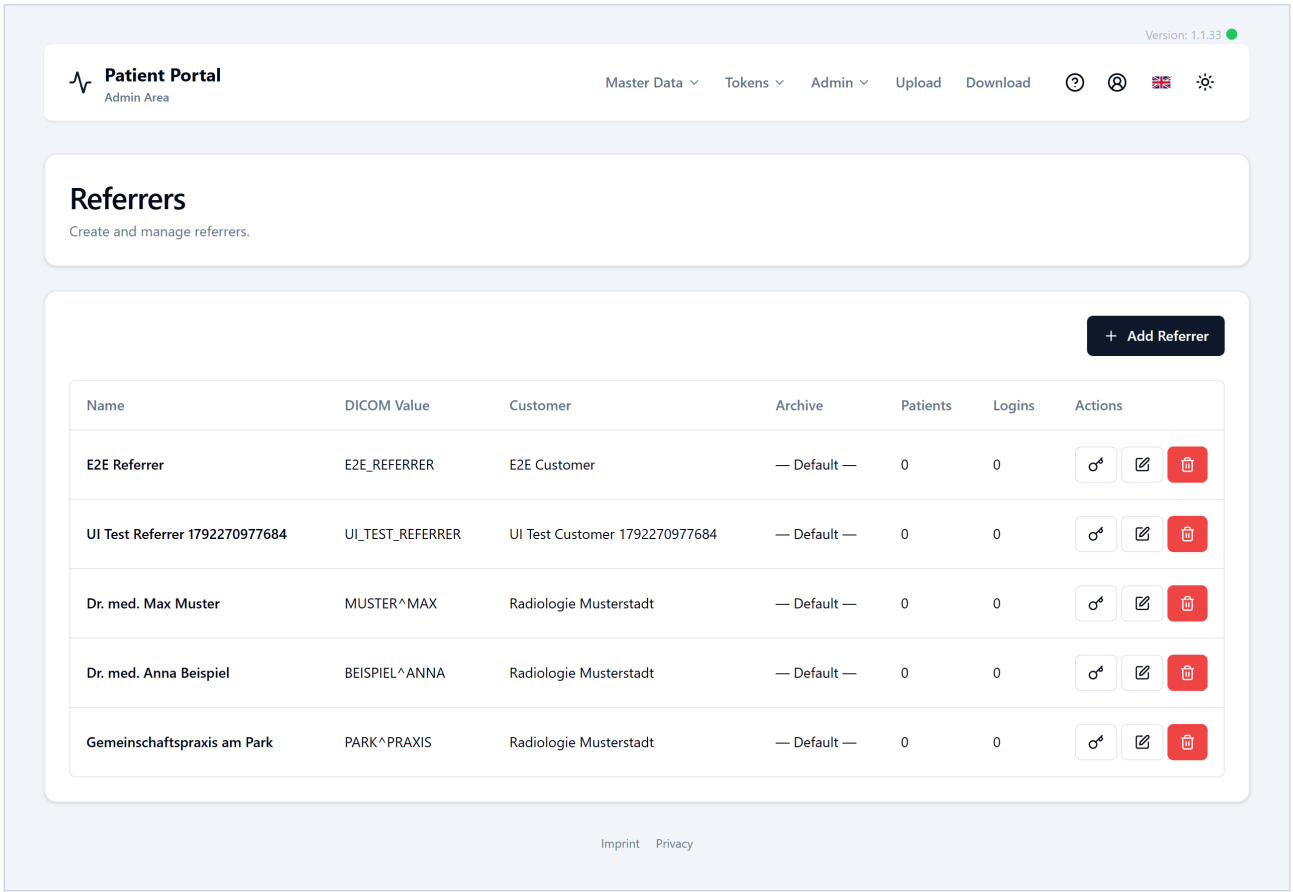


Figure 10. Referrer management

The table shows name, DICOM value, customer, archive and the number of patients and logins.

Creating and Editing Referrers

1. Click **Add Referrer** (or the pencil icon).
2. Fill in the fields:

Field	Meaning
Referrer Name	Display name, e.g. "Dr. Anna Example".
DICOM Value	Value of the DICOM attribute **ReferringPhysicianName** (0008,0090) as entered in the RIS/PACS for this referrer's examinations (e.g. `EXAMPLE^ANNA`). The referrer's PACS view shows **only** studies with exactly this value.
Customer	Practice/clinic the referrer belongs to.
Archive	PACS archive for this referrer's queries; — *Default* — uses the customer's archive or the default archive.

3. Save.



If a referrer does not find any patients in the PACS view, the DICOM value almost always does not match the entry in the image data. Check spelling, order and separators (^).

Creating Access for a Referrer

Use the key icon (**Generate Token**) to create an access code for the referrer:

1. Choose the **Validity (Days)** and the **Authentication Type** (*None* or *Password*, then enter the password).
2. Generate the token and copy the **Generated Token**.
3. Give the code to the referrer. With it, the referrer can sign in and then set up a permanent login with e-mail and password (if allowed in the settings).

Deleting Referrers

Click the trash icon and confirm the security prompt.

Users

Users lists all accounts that sign in with e-mail and password. Accesses via access codes (tokens) only appear here once a permanent login has been set up from them.

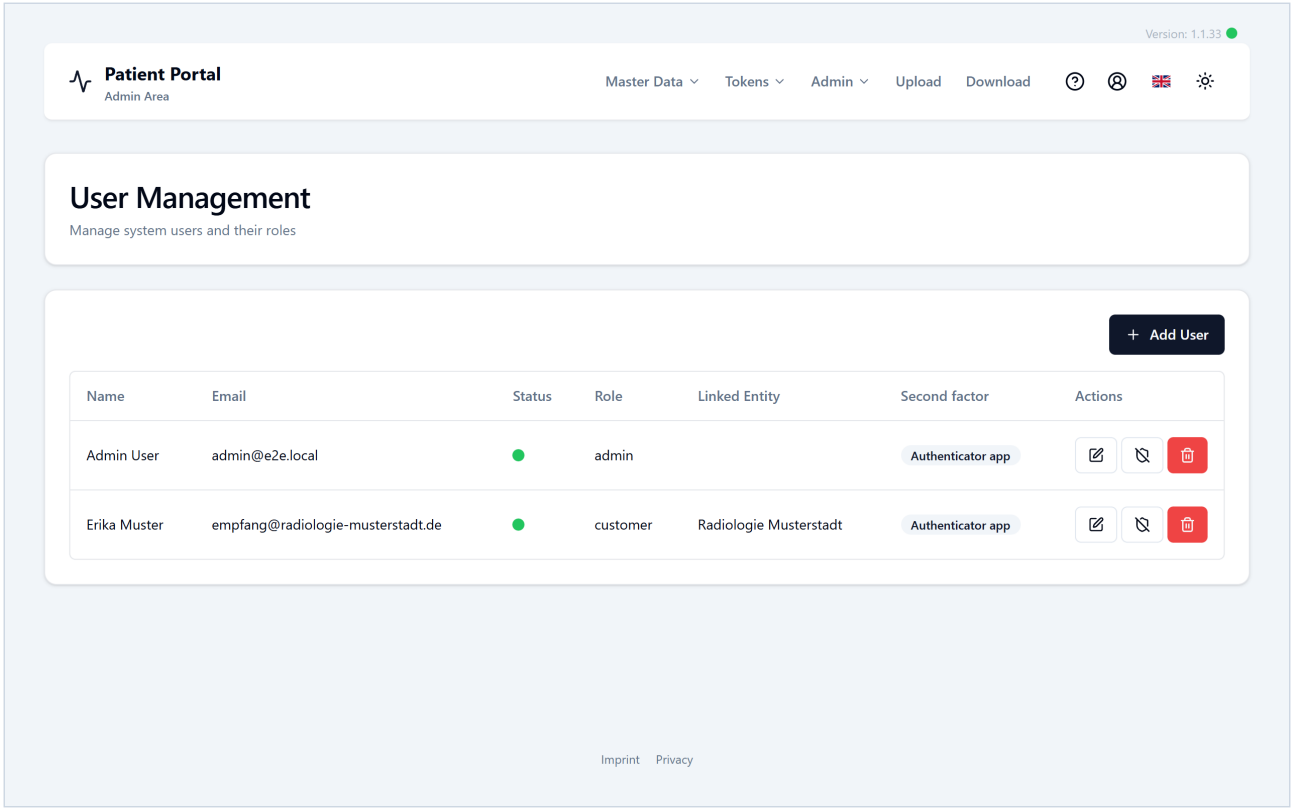


Figure 11. User management

The table shows name, e-mail, status (green = active), role and the linked entity (customer, referrer or patient).

Roles

Role	Access
`admin`	Full access to all administration features and settings.
`customer`	Practice/clinic: releases of the own customer, upload and downloads (see Customer Area). Must be linked to a customer.
`referrer`	Referrer: PACS view of the own patients, releases, upload, downloads. Must be linked to a referrer.
`patient`	Patient: view, download and share own examinations. Must be linked to a patient.

Creating and Editing Users

1. Click **Add User** (or the pencil icon).
2. Enter **First Name**, **Last Name**, **Email** and a **Password**. When editing, leave the password empty to keep the current one.
3. Choose the **Role**. No role is preselected; when editing, the field shows the stored role.
4. Depending on the role, choose the matching entry in **Link to Customer**, **Link to Referrer** or **Link to Patient**. These fields are search fields: type part of the name, for patients also the birth date or patient ID. For patients the result list shows birth date and patient ID to tell patients with the same

name apart. For customer, referrer and patient the link is required; without it the user cannot be saved.

5. Use **Active** to lock or unlock the account.
6. Save.

Every user sets up a second factor at the first sign-in (Sign-in & Security). The column **Second factor** shows the state; the shield icon resets a user's factors, for example after they lost their device (see Resetting sign-in factors). A new password set by the administrator ends all running sessions of the user.



The password rules (minimum length, upper/lower case, digits, special characters) are defined in the settings under **Security**.

Email Addresses and Letter Case

The letter case of the email address does not matter at sign-in: `Max.Muster@Practice.com` and `max.muster@practice.com` lead to the same account. The portal therefore always stores email addresses in lower case. An address that differs from an existing account only in letter case counts as already taken.

The update that introduced this behavior converted existing addresses to lower case once. Accounts whose addresses differ only in letter case, for example `Max@practice.com` and `max@practice.com`, are excluded. These accounts stay unchanged, and the account with capital letters can no longer sign in. The backend reports such accounts in the log on every start with `Email case collision between user ids: …`.

To resolve such a duplicate:

1. Find the accounts in the **Users** table by the IDs from the log or by the email address.
2. Decide which account stays in use.
3. Delete the account that is no longer needed, or give it a different email address.
4. Open the remaining account and save it. This converts its address to lower case.

Tokens

Integration Tokens

Integration tokens are long-lived access keys for external systems, e.g. a PACS workstation (VEPRO VIM), a practice management system via Portal-Launch or the gdt-hl7-bridge. They allow automatic sign-in without a password dialog and are always assigned to a customer.

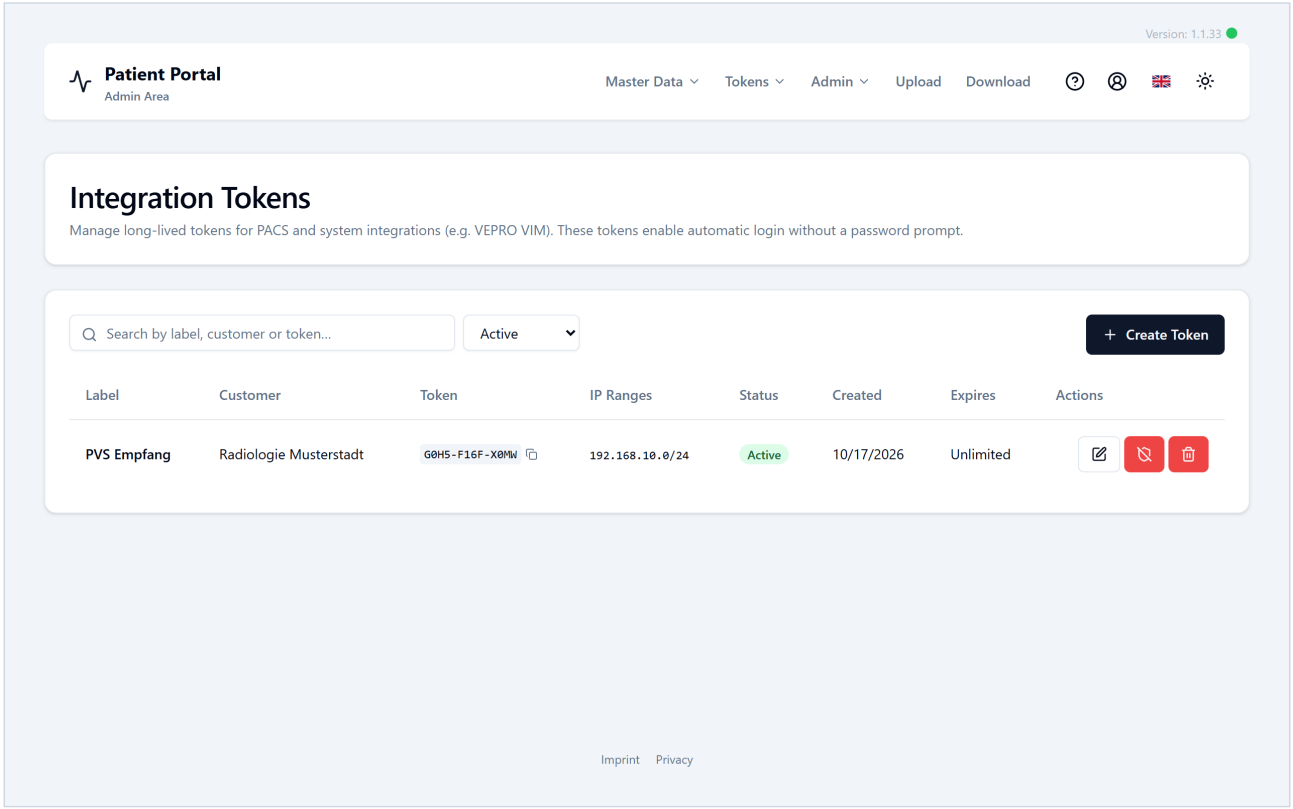


Figure 12. Integration tokens

Use the search field to filter by label, customer or token, and the selection to filter by status.

Creating a Token

1. Click **Create Token**.
2. Fill in the fields:

Field	Meaning
Label	What the token is used for, e.g. "VEPRO Workstation 1" or "PMS Reception".
Customer	The customer on whose behalf the external system works.
Allowed IP Ranges (CIDR)	Comma-separated IPv4 or IPv6 addresses or network ranges (CIDR) from which the token may be used, e.g. `192.168.2.0/24, 10.0.0.5, 2001:db8:10::/48`. Single addresses count as `/32` (IPv4) or `/128` (IPv6). The portal rejects invalid entries when saving.

Field	Meaning
Validity (Days)	Leave empty for unlimited validity.

3. Create the token. It is displayed – copy it immediately; it is shown in full only once. For VBS-based integrations an example configuration is shown as well.



Only leave the IP ranges empty in exceptional cases. A token without IP restriction can be used from anywhere; the portal warns you when creating it.

Editing, Revoking, Deleting a Token

- **Pencil icon** – change label, IP ranges or **expiry date**.
- **Revoke** (shield icon) – the token becomes invalid immediately but remains visible for traceability (status *Inactive*).
- **Delete** (trash icon) – removes the token permanently.

QR Code Import

QR Code Import takes over access codes that an external system (e.g. the PACS when printing reports) has already generated and printed. Patients can then sign in to the portal with these codes.

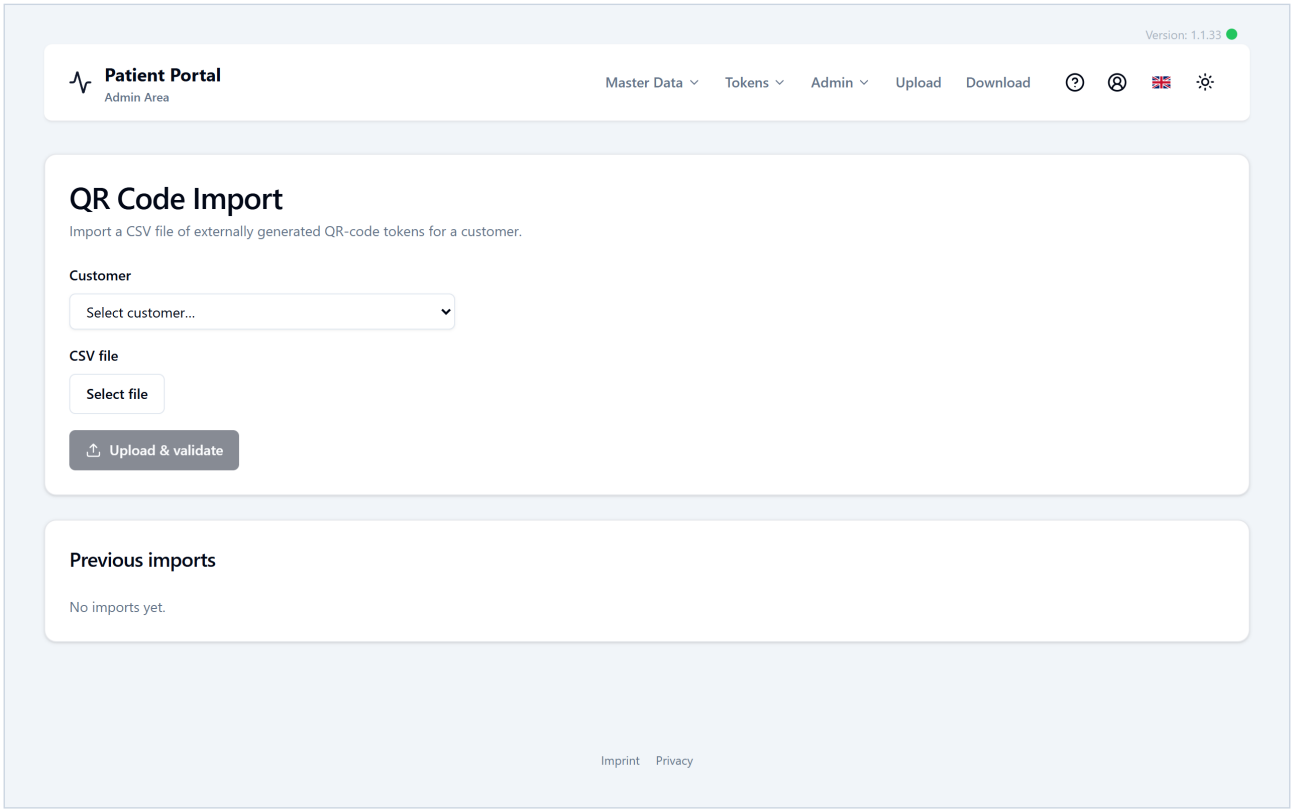


Figure 13. QR code import

CSV Format

The CSV file contains one code per line, separated by semicolons:

```
code;expirationdate;studyuid
ABCD-EFGH-JKLM;20270131;1.2.840.113619.2.55.3.1234.1
```

- `code` – the printed access code
- `expirationdate` – expiry date in the format `YYYYMMDD`
- `studyuid` – StudyInstanceUID of the shared examination

A header line is optional and detected automatically.

Running an Import

1. Choose the **Customer** and the **CSV file** and click **Upload & validate**.
2. The portal checks every line (format, duplicate codes in the file and in the database, existence of the study in the PACS). You see the progress under **Previous imports**.
3. Open the import via **View**: the details show valid and invalid lines with the reason.
4. Click **Run import** to take over the valid codes.

Running imports can be **cancelled** and **resumed** later; completed or failed imports can be **deleted**.

Releases

Release Management shows all patient and study releases (access codes) that have been issued – regardless of who created them (practice, referrer, patient).

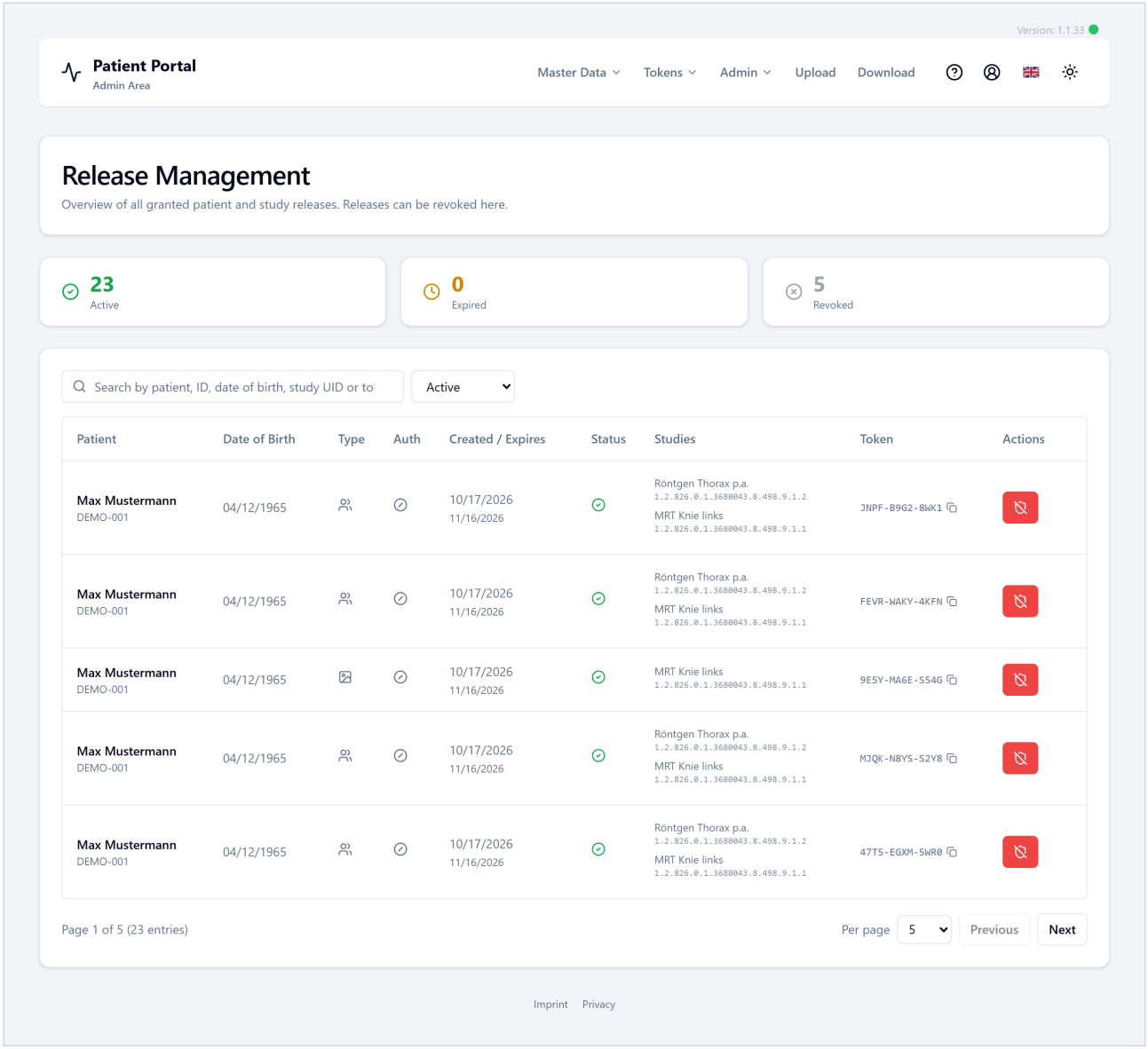


Figure 14. Release management

- The tiles at the top count **active**, **expired** and **revoked** releases.
- Use the search field to find releases by patient, patient ID, date of birth, study UID or token; the selection next to it filters by status.
- The columns show patient, date of birth, **type** (icon: patient or study release), **auth** (protection: none, date of birth, password), creation and expiry date, status, included examinations and the token (with copy icon).

Revoking a Release

Click the red icon in the **Actions** column and confirm. The access becomes invalid immediately.



This is the fastest way if a printout with an access code has been lost or handed to the wrong person.

Printing a Release Again

For active releases, the printer icon in the **Actions** column opens the print view with QR code and access link, using the same print template as on creation. The validity shown is the time remaining until the expiry date.



A password protecting the release is only stored encrypted and therefore does not appear on the reprint. Give it to the patient by other means.

Every reprint is logged in the audit log as `TOKEN_REPRINTED`.

Checking a Release in the Audit Log

The log icon in the **Actions** column opens the audit log filtered on this release as actor and without a date restriction, i.e. with all entries of this release. Logins with the access code and everything that follows (viewing, downloading) are logged with the release ID as actor ID. This shows, for example, whether studies have already been downloaded through exactly this release.

The icon is only shown to administrators, since only they may see the audit log.

Customers see the releases of their own customer under Releases in the customer area.

Admin

Import Status

Studies that reach the portal via upload (DLX import or ZIP upload) can be forwarded automatically to a **foreign studies archive** (DICOM C-STORE). **Import Status** shows these transfers.

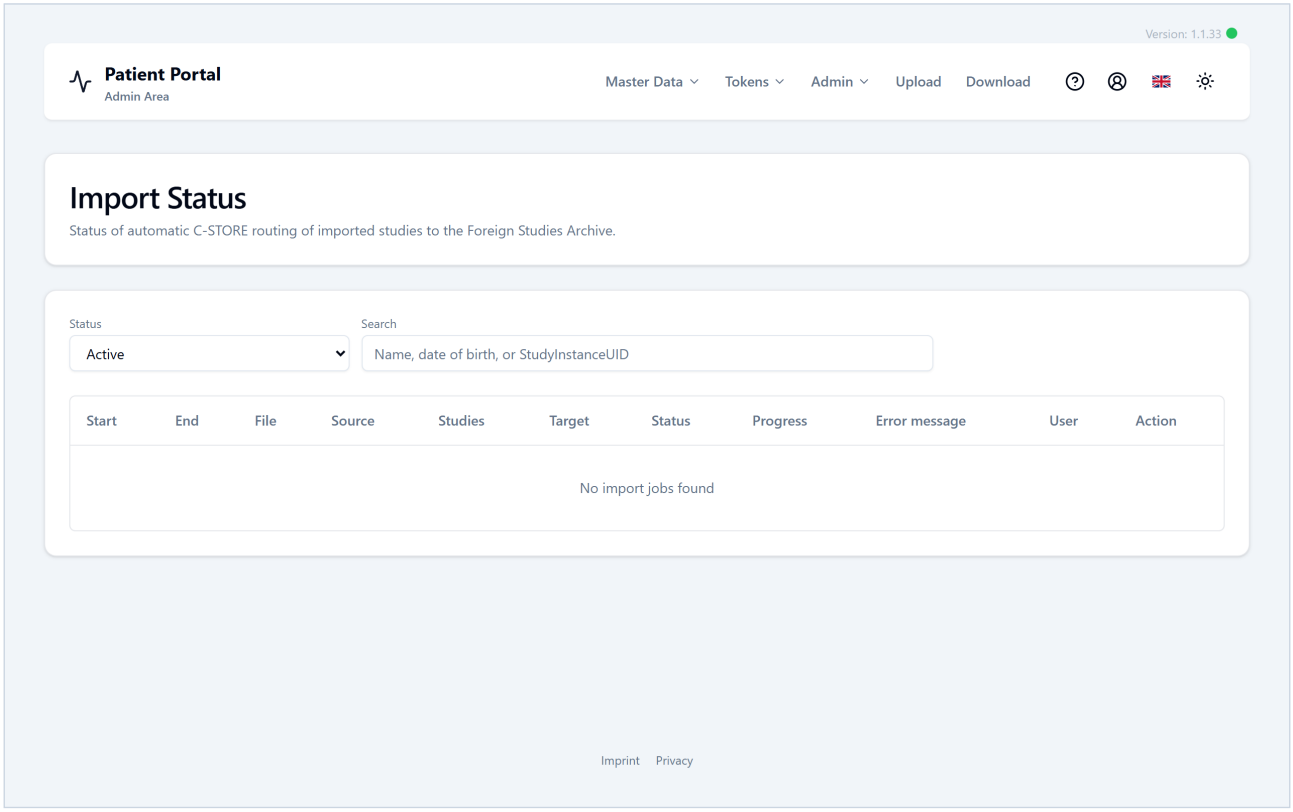


Figure 15. Import Status

- **Status** filter: active, completed or failed jobs.
- **Search** by name, date of birth or StudyInstanceUID.
- The table shows start and end, file, source (DLX, ZIP), number of studies, target, status, progress, error message and the user who triggered it. Failed transfers can be restarted via the **Action** column.

For which user groups DLX import and automatic forwarding are active is defined in the settings under **Import**.

Print Templates

This guide describes how to adapt print, PDF and e-mail templates for access links as well as the system e-mails (permanent login, security notices) to the corporate identity (CI) of a practice or customer. It is intended for administrators with basic HTML and CSS knowledge.

Overview

Whenever the portal issues an access link (printout with QR code, PDF via the share API, delivery by e-mail), an HTML template is rendered with the Handlebars template engine. Each template consists of two parts:

- **HTML template** – the layout with placeholders such as `{{patient.lastName}}` and texts such as `{{i18n "common.title"}}`.
- **Template translations (JSON, optional)** – only texts that exist exclusively in this template.

All default texts (title, subject, introduction, field labels …) live centrally in the `templates` translation namespace. They are translated into every installed language via the language packs and can be adjusted under **Admin → Language Management** with an override – without changing the template (see Translations).

In addition, there are **assets** (logo, stylesheet, images, fonts) that are uploaded centrally and can be used in all templates.

Templates are managed under **Admin → Print Templates** (`/admin/print-templates`).

Template Types

A template is assigned to one or more template types (template keys):

Context	Type	Usage
Patient	`patient_print`	Browser printout after generating a patient token (token generator).
	`patient_pdf`	PDF of the share API when called with `patid`. If missing, `patient_print` is used.
	`patient_email`	E-mail delivery of a patient token.
Study	`study_print`	Browser printout after generating a study token.
	`study_pdf`	PDF of the share API when called with `studyuid` or `accnbr`. If missing, `study_print` is used. Also used as the PDF attachment of e-mails.
	`study_email`	E-mail delivery of a study token.
Referrer	`referrer_print`	Printout of a referrer access (referrer management, token generator).
	`referrer_pdf`	Currently unused (reserved).
	`referrer_email`	E-mail delivery of a referrer token.
System e-mails	`account_activation`	Confirmation e-mail after creating a permanent login (patients and referrers), see System E-Mails.
	`account_welcome`	Welcome e-mail after the permanent login has been activated.

Context	Type	Usage
	`account_security`	Security e-mail to every e-mail login (administrators, practice, referrers, patients) when a sign-in factor changes.



Referrer types and patient/study types are mutually exclusive in the UI because they receive different data. A single template can, however, cover e.g. both `patient_print` and `study_print`. System e-mails receive their own data: a template covers exactly one system e-mail.

Resolution Order

For a template type, the first matching **active** template is used in this order:

1. **Customer template** – for the customer to which the token belongs.
2. **System template** – created under **Default Templates**, applies to all customers without their own template.
3. **Built-in template** – `backend/src/main/resources/templates/default-<type>.html`.

Assets are layered as well: system assets form the base, customer assets with the same key override them. A system logo stored under the key `logo` is therefore automatically replaced by the customer logo `logo`.



Per customer (or at system level) there must be **only one active template** per template type. Multiple active templates with the same type cause a server error during resolution. Deactivate old versions before activating a new one.

Step by Step: Creating a CI Template

1. **Upload assets.** Under **Print Templates**, select the customer (or **Default Templates** for all customers) and upload in the **Assets** section:
   - `logo` – practice logo (PNG, SVG, WebP or JPEG), with alt text.
   - `css` – optional stylesheet for print/PDF (see Colors and Stylesheet).
   - further images or fonts under freely chosen keys (e.g. `stamp`, `font_regular`).

   In the asset list, the eye icon (column **Preview**) opens images (PNG, JPEG, WebP, SVG) in a preview window. The pencil (column **Actions**) edits an existing asset: the alt text can be changed for every asset, and for CSS and SVG also the content in the code editor (SVG with live preview) – without uploading the file again.
2. **Create a new template. New Template** → enter a name, select template type(s) and use **Load default template** to start from the built-in template of the first selected type (see Editor).
3. **Adapt colors, font and layout** (see Implementing CI Elements).
4. **Check the preview.** The live preview on the right of the editor renders with sample data in every installed language (see Testing).

5. **Activate and save.** Several templates can also be switched together directly in the template list: select them via checkbox and choose **Activate … templates** or **Deactivate … templates**. Only one template per template type can be active – previously active templates of the same types are deactivated and named in the message; two selected templates of the same type cannot be activated together.
6. **Test all output channels for real** – printout, PDF and e-mail behave differently (see Differences per Output Channel).

Implementing CI Elements

Logo

```
{{#if assets.logo}}
<img class="logo" src="{{assets.logo.url}}" alt="{{assets.logo.alt}}" />
{{/if}}
```

Alternatively, the helper `{{assetUrl "logo"}}` returns only the URL. Always wrap assets in `{{#if ...}}` so the template also works without an uploaded asset.



Use SVG or PNG with sufficient resolution (at least 2× the display size) – otherwise the printout and PDF will be blurry.

Colors and Stylesheet

For print and PDF templates there are two options:

- **CSS directly in the template** in a `<style>` block – simplest if the template applies to a single customer only.
- **CSS as an asset** (key `css`, type `text/css`) – useful if one system template applies to all customers and each customer only uploads their own stylesheet.

The built-in templates already use the pattern "customer CSS, otherwise default CSS":

```
{{#if assets.css}}
<style>{{{assets.css.content}}}</style>
{{else}}
<style>
  /* default styles */
</style>
{{/if}}
```



CSS content must be included with **three** curly braces (`{{{assets.css.content}}}`). With two braces the content is HTML-escaped and the stylesheet has no effect.

Recommendation: define CI colors as CSS variables at the top of the stylesheet so they are maintained

in one place:

```
:root {
  --ci-primary: #004f9f;
  --ci-accent:  #e30613;
  --ci-text:    #1a1a1a;
}
h1         { color: var(--ci-primary); }
.header    { border-bottom: 3px solid var(--ci-accent); }
```

Fonts



Fonts must **not** be loaded from external servers (e.g. Google Fonts). The portal is subject to the GDPR; every external request transmits the patient's IP address to third parties. Upload fonts as a `font/woff2` asset or use a system font stack.

The URL of an uploaded font is only available in the HTML template, not inside a CSS asset (its content is not processed by Handlebars). The `@font-face` rule therefore belongs in the template's `<style>` block:

```
<style>
  {{#if assets.font_regular}}
  @font-face {
    font-family: "PraxisSans";
    src: url("{{assets.font_regular.url}}") format("woff2");
  }
  {{/if}}
  body { font-family: "PraxisSans", Arial, Helvetica, sans-serif; }
</style>
```

Many e-mail clients (Outlook, Gmail) ignore web fonts – always specify a fallback font.

Page Format for Print and PDF

```
@page { size: A4; margin: 2cm; }
@media screen { body { padding: 1.5rem; } }
```

The PDF is generated with Chromium in A4 format including background colors; `@page` rules are respected. The browser printout renders the template in an isolated `iframe` – the portal's styles do not affect the template.

Imprint and Contact Details

Instead of hard-coding addresses in the template, use the settings (see Settings in Templates). This keeps a system template valid for all customers:

```
<div class="footer">
  {{settings.legal.group.company.name}}
  {{#if settings.legal.group.company.street}}·
{{settings.legal.group.company.street}}{{/if}}
  {{#if settings.legal.group.company.city}}· {{settings.legal.group.company.zip}}
{{settings.legal.group.company.city}}{{/if}}
  {{#if settings.legal.group.contact.phone}}·
{{settings.legal.group.contact.phone}}{{/if}}
</div>
```

Translations

All visible texts are output via `{{i18n "…"}}` with their **full path** – never written directly into the HTML. The language is determined by the language selector when printing, by the selected language when sending e-mails and by the `lang` parameter of the share API for PDFs; any installed language is possible.

The `templates` Namespace

The default texts of all templates live in the `templates` translation namespace. Shared texts are under `common`; only subject, introduction and button text differ per audience:

Path	Text (en)
`common.title`	Your Access Link
`common.scan_qr`	Scan QR code or open link
`common.access_type_label`	Access Type
`common.access_type.PATIENT` / `.REFERRER` / `.STUDY`	Patient Access / Referrer Access / Study Access
`common.verification_label`	Access Protection
`common.verification_type.NONE` / `.BIRTHDATE` / `.PASSWORD`	No password / Date of birth / Password
`common.validity`, `common.unlimited`	Valid until, Unlimited
`common.patient`, `common.referrer`	Patient, Referrer
`common.included_studies`	Included Studies
`common.image_count` (plural)	1 image / 5 images
`common.series`	Series
`common.generated_on`	Generated on
`common.date_format`	`{{month}}/{{day}}/{{year}}` – format for `formatDate`
`patient.subject`, `referrer.subject`, `study.subject`	E-mail subject line

Path	Text (en)
`patient.intro`, `referrer.intro`, `study.intro`	E-mail introduction
`patient.open_link_button`, `referrer.open_link_button`, `study.open_link_button`	Label of the link button

Examples:

```
<h1>{{i18n "common.title"}}</h1>
<p>{{i18n "referrer.intro"}}</p>
{{i18n "common.verification_type" token.verificationType}}   {{!-- arguments are
joined with "." --}}
{{i18n "common.image_count" count=instanceCount}}            {{!-- plural form
matching the language --}}
{{i18n audience "intro"}}                                    {{!-- template for
several audiences --}}
```

`audience` contains the audience of the template type currently being rendered (`patient`, `referrer` or `study`). If a template covers e.g. `patient_email` and `study_email`, `{{i18n audience "subject"}}` returns the matching text for each. The subject of e-mails is always `<audience>.subject`.

Adjusting Texts

Do **not** adjust default texts in the template, but under **Admin → Language Management** as an override in the `templates` namespace (e.g. key `referrer.intro`). The override applies immediately to printout, PDF and e-mail of all templates and can be set per language.

Template Translations

For texts that only one template needs, write a key of your own into the HTML, e.g. `{{i18n "custom.footer"}}`. The editor automatically creates unknown keys in the template's translations JSON – empty and for **every installed language** (with `count=`, with all plural forms of the language). Entries that are still empty are shown as a warning in the editor and as `⚠custom.footer⚠` in the preview.

```
{
  "de": { "custom.footer": "Bitte bringen Sie diesen Ausdruck zum nächsten Termin
mit." },
  "en": { "custom.footer": "Please bring this printout to your next appointment." }
}
```

A template translation with a path from the namespace (e.g. `common.title`) overrides the default text for this template only.

Resolution Order

For each key, the first non-empty text from

1. the template translation of the language,
2. the override or language pack of the language (`templates` namespace),
3. the built-in default text of the language

is used, each along the language chain **selected language** → **fallback of the language pack** → `en` → `de`. If a key is missing everywhere, the key itself is output.



Translations must not contain HTML – the output is escaped differently per channel.

Available Data

General

Placeholder	Content
`{{lang}}`	Current language (e.g. `de`, `en`, `ro`), e.g. for `<html lang="{{lang}}">`.
`{{audience}}`	Audience of the template type: `patient`, `referrer` or `study`.
`{{link}}`	Full access link.
`{{qrDataUrl}}`	QR code of the link as a `data:` URL, usable as `<img src="{{qrDataUrl}}">`.
`{{token.value}}`	Token value.
`{{token.type}}`	`PATIENT`, `REFERRER` or `STUDY`.
`{{token.verificationType}}`	`NONE`, `BIRTHDATE` or `PASSWORD`.
`{{token.validUntil}}`	Expiry date as `YYYY-MM-DD` (empty for unlimited validity) – output with `formatDate`.
`{{token.validityDays}}`	Validity in days (empty for unlimited validity).
`{{token.generatedAt}}`	Creation date as `YYYY-MM-DD` – output with `{{formatDate token.generatedAt}}`.
`{{patient.firstName}}`, `{{patient.lastName}}`, `{{patient.externalId}}`	Patient data (patient/study only).
`{{referrer.name}}`	Name of the referrer (referrer only).
`{{#each studies}}`	List of included studies (see below).
`{{assets.<key>.url}}`, `.alt`, `.content`	Assets; `content` only for CSS.
`{{settings.<path>}}`	Settings (see Settings in Templates).



Do not output `patient.birthDate` in templates. For `BIRTHDATE` tokens the date of birth

> is the access secret – a printout or e-mail containing the link **and** the date of birth defeats the access protection.

Studies

Within `{{#each studies}}` the following fields are available:

Field	Content
`description`	Study description (may be empty).
`studyDate`	Study date as `YYYY-MM-DD` – output with `{{formatDate studyDate}}`.
`studyTime`	Time of the study (`HH:MM:SS`).
`modalities`	Modalities, e.g. `CT` or `MR / CT`.
`modalitiesList`	Modalities as a list, e.g. `{{#each modalitiesList}}…{{/each}}`.
`accessionNumber`, `studyuid`	Identifiers.
`seriesCount`, `instanceCount`	Number of series or images.
`{{#each series}}`	Series with `seriesuid`, `modality`, `description`, `instanceCount`, `seriesTime` (can be switched off for the browser printout via `display.group.print.show_series`).

The number of printed studies can be limited with the setting `display.group.print.maxstudies`.

Differences per Output Channel

Templates are rendered in three places with different data. Robust templates therefore always check optional fields with `{{#if ...}}`.

Aspect	Print (browser)	PDF	E-mail
Renderer	Handlebars (JS) in the browser	Handlebars (Java), Chromium	Handlebars (Java)
Language	selectable	`lang` parameter (default `de`)	chosen when sending
Asset URLs	relative (`/api/…`)	embedded as `data:` URL	absolute (`frontend.url` + path)
`studies[].series`, `studyTime`, `modalitiesList`, `instanceCount`	yes	yes	yes
`patient`	patient/study	yes	patient/study
`referrer.name`	only from referrer management	no	yes

Settings in Templates

`settings` contains all **non-encrypted** system and customer settings with the prefixes `display.`, `frontend.`, `app.` and `legal.`. Customer settings override system settings. Dots in the key become nesting levels: `legal.group.company.name` becomes `{{settings.legal.group.company.name}}`.

Commonly used:

- `legal.group.company.name`, `.street`, `.zip`, `.city`, `.country`, `.representative`
- `legal.group.contact.phone`, `.email`, `.privacy_email`
- `legal.group.medical.responsible_name`, `.responsible_specialty`
- `legal.group.dataprivacy.dpo_name`, `.dpo_contact`
- `app.support_email`
- `frontend.url`

Helper Reference

Helper	Description
`{{i18n "path"}}`	Translation for the current language (see Translations). Several arguments are joined with `.`: `{{i18n "common.access_type" token.type}}`. With `count=` the plural form is selected and `{{count}}` is replaced.
`{{formatDate date}}`	Formats `YYYY-MM-DD` according to `common.date_format`. Other formats are output unchanged.
`{{assetUrl "key"}}`	URL of an asset (empty if not present).
`eq`, `ne`, `gt`, `lt`, `gte`, `lte`	Comparisons as a subexpression: `{{#if (eq token.type "PATIENT")}}…{{/if}}`.
`{{#and a b}}…{{/and}}`, `{{#or a b}}…{{/or}}`	Logical operators – use only as **block** helpers, not as a subexpression inside `#if`.
`#if`, `#unless`, `#each`, `#with`, `@index`, `@first`, `@last`	Standard Handlebars.



The browser printout uses Handlebars for JavaScript, PDF and e-mail use Handlebars for Java. Limit yourself to the helpers listed here and only compare values of the same type with `eq` (number with number, text with text).

System E-Mails

Besides access links, the portal sends three system e-mails about the account. Each exists only once – shared by patients and referrers. They are resolved (practice → system → built-in template) and designed exactly like the **`_email` templates and appear in the editor in the group *System e-mails**:

Type	Occasion and data
`account_activation`	A patient or referrer created a permanent login. Data: `link` (confirmation link), `expiresInHours` (validity, 24).
`account_welcome`	The e-mail address is confirmed, the login is active. Data: `link` (sign-in page).
`account_security`	A sign-in factor changed. Data: `event`, `detail` (e.g. the passkey name), `message` (ready-made sentence for the event), `occurredDate`, `occurredTime`, `link` (page **Sign-in & security**).

All system e-mails also receive `email`, `user.firstName`/`user.lastName`, `accountType` (`patient`, `referrer`, `customer` or `admin`), `lang`, `audience`, `assets` and `settings`. Use `accountType` to vary texts per recipient:

```
{{#if (eq accountType "referrer")}}<p>You find your referrals under …</p>{{/if}}
```

Which practice template applies?

- Referrers and practice logins: the practice of the account.
- Patients: the practice of the patient's most recently created access code.
- Administrators: always the system template.

Language

- Confirmation and welcome e-mail: the UI language when the login was registered.
- Security e-mail: the UI language of the last sign-in with e-mail and password.
- Any installed language is possible; if a text is missing, the fallback chain applies (see Translations).

Events of the security e-mail

The variable `event` has one of these values; its text is `account.security.events.<EVENT>` and can be overridden like any template text: `TOTP_ADDED` (authenticator app set up), `TOTP_REPLACED` (app set up again), `TOTP_REMOVED` (app removed), `PASSKEY_ADDED` and `PASSKEY_REMOVED` (with the passkey name as `{{name}}`), `BACKUP_CODES_REGENERATED`, `FACTORS_RESET` (an administrator reset all factors).

```
<p>{{i18n "account.security.events" event name=detail}}</p>
{{#if (eq event "FACTORS_RESET")}}<p>Please set up a new factor at your next sign-
in.</p>{{/if}}
```

In the editor preview, choose the event in the menu **Event**.

Texts (namespace `templates`)

`account.activation.`, `account.welcome.` and `account.security.*` – each with `subject` plus heading, text and button. The validity of the confirmation link is a plural form: `{{i18n "account.activation.validity" count=expiresInHours}}`.

E-Mail Template Specifics

E-mail clients (Outlook in particular) support only a subset of HTML and CSS. The built-in `*_email` templates therefore follow proven rules – keep them:

- Layout with `<table role="presentation">` instead of Flexbox/Grid, fixed width of 600 px.
- All styles **inline** in the `style` attribute; `<style>` blocks and CSS assets are removed by many clients.
- Buttons as `<a>` with background color and padding; additionally output the link as text below.
- No CSS variables, no web fonts without a fallback.
- Do **not** include the QR code (`qrDataUrl`) in e-mails – `data:` images are blocked by many clients.
- Logo URLs are made absolute automatically; the `frontend.url` setting must be set correctly for this, and the asset endpoint must be reachable from the internet.

If the setting `token.group.email.appendpdf` is enabled, a PDF (type `*_pdf` or `*_print`) is attached as well – the print/PDF rules apply to it.

Editor

The editor takes up almost the whole browser window: source and template translations (tabs) on the left, the live preview on the right. It only closes via **Cancel** or **Save** – clicking next to the window does not close it; with unsaved changes, **Cancel** (also `Esc`) asks for confirmation.

- `{{` opens the picker for **system variables** and **functions**. The description of the highlighted entry, including an example value from the preview data, is shown above the list. Inside `{{#each studies}}`/`{{#each series}}` the fields of the list item are offered, system variables there as `@root.⋯`.
- `{{i18n "` opens the picker for **translations**: template translations and default texts from the `templates` namespace, each with the text in the preview language.
- **Assets:** Uploaded assets appear in the `{{` picker as `assets.<key>.url`/`.alt` (images, with a thumbnail), `.url` (fonts) or `.content` (CSS). **Insert asset** (above the source) inserts the matching snippet at the cursor, e.g. `{{#if assets.logo}}<img src="{{assets.logo.url}}" alt="{{assets.logo.alt}}" />{{/if}}`, a `<style>` block for CSS or an `@font-face` rule for fonts.
- **Translations as a table:** The **Translations** tab shows the template translations as a table – one row per key, one column per installed language. Grey placeholders are the default texts (empty cell = default text); cells highlighted in yellow have neither a text nor a default. Keys can be added below the table (with suggestions from the namespace, to override a default text for this template only); rows the template does not use are marked. **JSON** still allows editing the raw format.
- **Editing:** Hovering over a placeholder shows its description or text plus actions: **Replace variable** (opens the picker for the selected variable), **Edit translation** (dialog with one field per installed language; the default text is shown as placeholder, empty = default text) and, for unknown variables, **Use as translation**. A **double-click** runs the first action.
- **New texts:** If you write e.g. `{{hinweis}}`, the editor marks the unknown variable. **Use as translation** (hover, double-click or the warning's quick fix) turns it into `{{i18n "hinweis"}}`, creates the key for all languages and opens the translation dialog.

- Unknown variables and syntax errors are marked in the source; empty translations in the translations tab (the number on the tab shows how many are missing). A syntax error prevents saving; with unknown variables the editor asks for confirmation.
- **Preview and source linked:** clicking an element in the preview selects its start tag in the source and scrolls to it (also from the **Translations** tab); on hover the element gets a dashed outline. Conversely, the cursor in the source highlights the corresponding element in the preview – for `{{#each}}` loops all repetitions. Text from `{{i18n …}}` or variables leads to the enclosing element.
- **Load default template** (above the source) loads the built-in default template of the first selected type; existing source is only replaced after confirmation.

Exporting and Importing Templates

Templates can be moved between installations (e.g. from a test to the production environment) – one at a time or several together in one file:

- **Export:** In the template list, the download icon of a row downloads the template as a JSON file – name, template types, HTML and template translations. For several templates, select them with the checkboxes (the checkbox in the table header selects all) and choose **Export … templates**; all selected templates end up in one file, shared assets only once. With **Include assets**, the assets the template uses (`assets.<key>…`, `assetUrl "<key>"`) are included as well; for customer templates the same layering as at render time applies (customer asset before system asset).
- **Import: Import templates** imports an export file into the currently selected level (customer or **Default Templates**). If the file contains several templates, the dialog lists them; all are preselected and can be deselected individually. Optionally the assets used by the selected templates are imported; existing assets with the same key stay unchanged unless **Overwrite existing assets with the same key** is selected.
- **Activation:** Without further options, the templates are created **inactive** – check them in the editor and then activate them (deactivating the previously active template of that type first). With **Activate after import** they are active right away; the previously active templates of the same types on that level are deactivated automatically (not deleted). This is not possible if several selected templates have the same type – the dialog points this out.

Export files of earlier versions (one template per file) can still be imported.

The import applies the same checks as creating a template in the editor (allowed template types, translation format, asset types PNG/JPEG/WebP/SVG/CSS/WOFF2, max. 2 MB per asset); an invalid file is rejected as a whole.

Sample templates

As a starting point for your own templates, there is a package of importable sample templates. It contains four designs: **Standard** (the built-in templates), **Modern**, **Professional** (letterhead) and **Clinical** (minimal, high contrast). Each design has four templates – patient/examination and referrer, each for print/PDF and for e-mail – with a placeholder logo, stylesheet and German/English texts. Patient and examination access share one template; it handles the differences (texts, series list) via the `audience` variable.

- **Where to get it:** for administrators on the portal's download page (card **Sample templates**, downloaded with SHA-256 checksum like the applications), as `sample-templates-<version>.zip` on `download.radioweb.de`, in the integration package of a release (folder `Sample-Templates/`) and in the repository under `integration/sample_templates/`
- **Import:** one file per design – `<design>/import/<design>.json` with all four templates and the assets – imported as described above, optionally activated right away with **Activate after import**
- **Logo:** if an asset `logo` exists on the level, all designs use that logo instead of the placeholder
- **License:** all templates and graphics are self-made and released under CC0 – free to use, including commercially

The package's `README.md` describes import and customisation in detail.

Testing

The live preview renders with sample data plus the assets and settings of the selected customer – in every installed language (**Language** selector) and, for templates covering several audiences, per audience. Texts are resolved exactly as when delivering. It is still no substitute for a real test of the output channels.

PDF preview: In the template list, the icon in the **Preview** column downloads a print/PDF template as a PDF with sample data – also a template that is still inactive. The PDF is generated in the backend like PDF delivery (page format, page breaks, embedded assets) and uses the assets and settings of the selected level and the current interface language. For templates covering patient and examination, the audience is chosen from the menu; e-mail templates have no PDF preview. Both previews show the customer's real practice data (settings) and assets for inactive templates too; the sample practice data only appears when no settings exist on that level.

Checklist before activating:

- ☐ Printout via the token generator checked in **all** languages in use.
- ☐ PDF generated via the share API (see developer documentation, if used) – logo visible, umlauts correct.
- ☐ Test e-mail sent to Outlook and a webmail client.
- ☐ Template checked without studies, with one and with many studies (page break).
- ☐ No external URLs (fonts, images, scripts) in the HTML.
- ☐ No hard-coded texts – everything via `i18n`.
- ☐ Only one active template per type and customer.

Limitations

- Assets: PNG, JPEG, WebP, SVG, CSS, WOFF2; maximum 2 MB per file.
- JavaScript in templates is not supported and must not be used.
- Templates and assets can only be maintained by users with administrator rights.

Language Management

The portal is available in German and English. **Language Management** lets you install additional languages and permanently adjust individual texts.

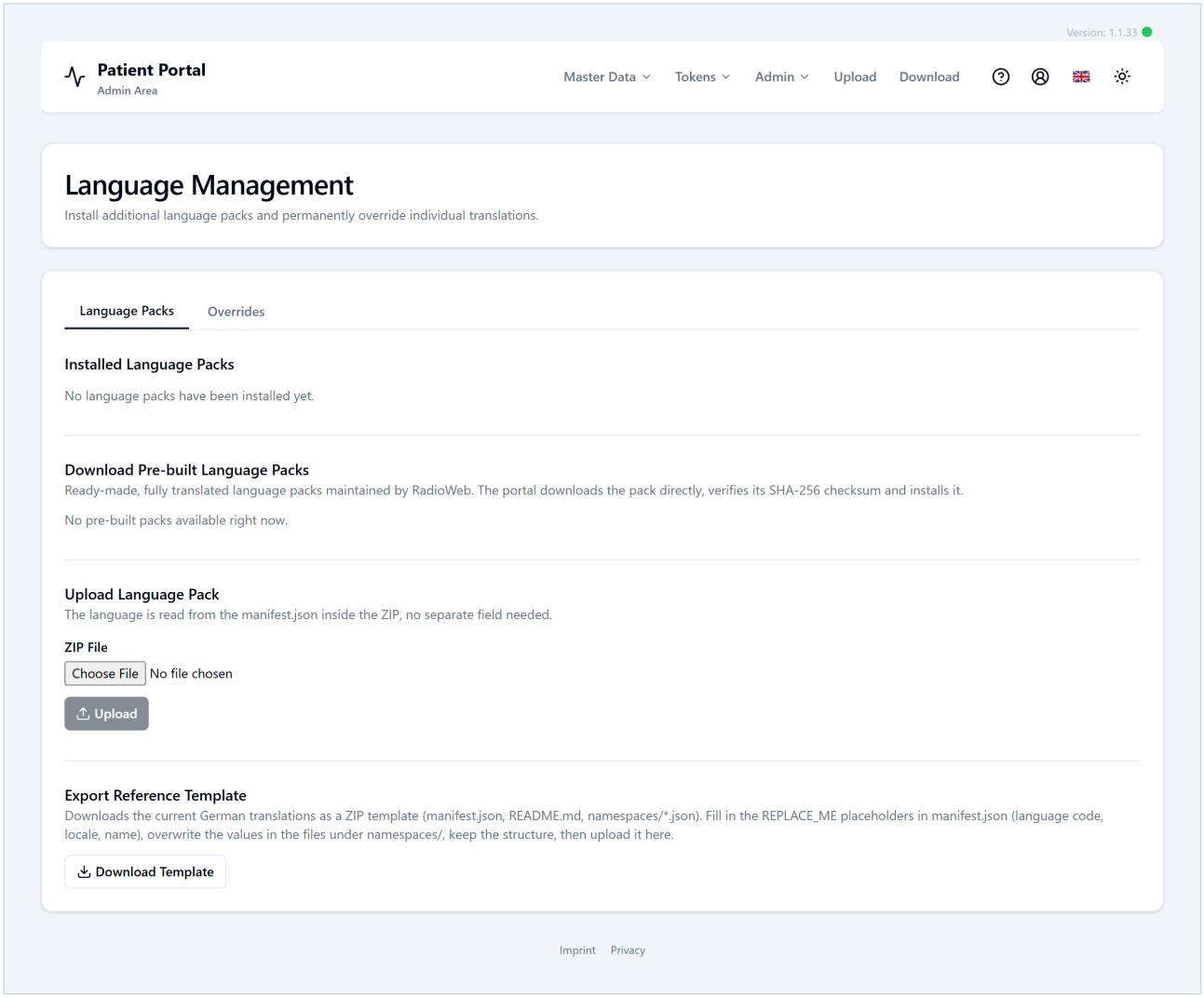


Figure 16. Language management, tab Language Packs

Language Packs

Installed language packs

List of installed languages with version and coverage; packs can be updated or removed.

Download pre-built language packs

Fully translated packs maintained by RadioWeb (e.g. Polish, Romanian, Russian, Turkish, Ukrainian). After confirmation the portal downloads the pack itself from `download.radioweb.de`, verifies the SHA-256 checksum and installs it. If the download server cannot be reached from the portal, you can download the pack in the browser and upload it afterwards.

Upload language pack

Upload a ZIP file with `manifest.json` and the translation files; the language is read from the manifest.

Export reference template

Downloads the current German texts as a ZIP template for your own translation. Fill in the `REPLACE_ME` placeholders in `manifest.json`, translate the texts under `namespaces/`, keep the structure and upload it afterwards.

Overrides

The **Overrides** tab lets you permanently override individual texts of a language – for example to adapt terms to the wording of your practice. Overrides are preserved when the portal is updated.



The texts of the print templates (namespace `templates`) can be adjusted here as well.

Audit Log

The **Audit Log** records all security-relevant accesses and actions in the portal: sign-ins, token checks, viewing patients and studies, downloads, releases and administrative actions. The entries are protected against subsequent modification (HMAC signature) and serve as evidence towards data protection authorities and patients.

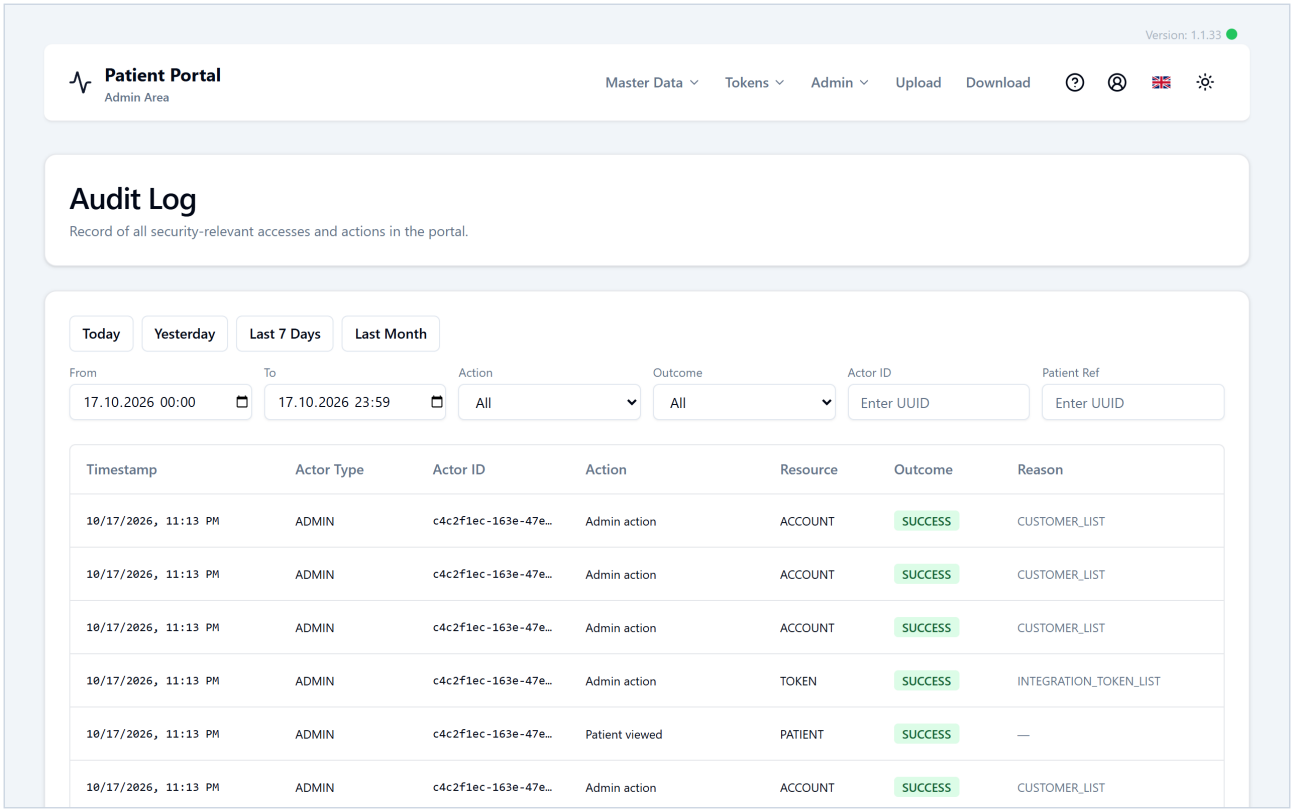


Figure 17. Audit log

Filtering

- **Shortcuts**: *Today*, *Yesterday*, *Last 7 Days*, *Last Month*, *All* (no date restriction)
- **From** / **To**: custom period
- **Action** (e.g. "Study viewed", "Token login successful") and **Outcome** (success/failure)

- **Actor ID**: all actions of a specific user or token
- **Patient reference**: all accesses to a specific patient – e.g. to answer a subject access request under Art. 15 GDPR

Columns

Column	Meaning
Timestamp	Date and time
Actor type	`ADMIN`, `CUSTOMER`, `REFERRER`, `PATIENT`, `STUDY` or `ANONYMOUS` (before sign-in)
Actor ID	Internal identifier of the user or token
Action	What happened
Resource	Affected object (e.g. `STUDY`, `TOKEN`, `ACCOUNT`)
Outcome	`SUCCESS` or failure
Reason	Additional information, e.g. `CUSTOMER_CREATE` or `PACS_QUERY_CACHED`

Archives

Archives lets you manage the PACS archives from which the portal retrieves examinations via DICOM query/retrieve. Customers and referrers without their own assignment automatically use the **Default** archive.

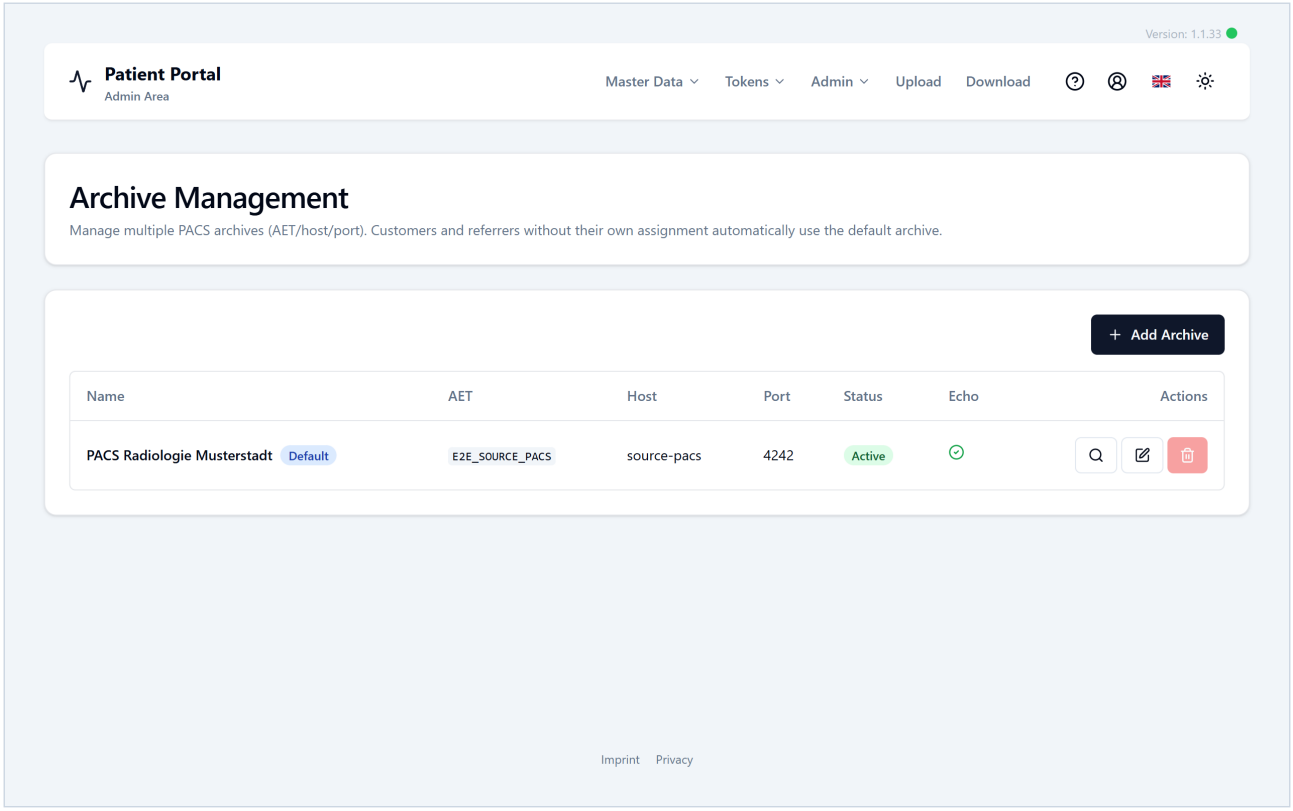


Figure 18. Archive management

The table shows name, AET, host, port, status and the result of the last connection test (**Echo**).

Creating and Editing an Archive

1. Click **Add Archive** (or the pencil icon).
2. Enter **Name**, **AET** (AE title of the PACS), **Host / IP** and **Port**, optionally a **Description**; use **Active** to enable or disable the archive.
3. Check reachability with **Test connection** (C-ECHO) and save.

> For the portal to be allowed to retrieve images, it must be registered in the PACS as a DICOM node – with the portal's AE title and port (see initial setup).

Query Test

The magnifier icon runs a test query against the archive: search by patient ID, name, study UID or accession number, optionally with a period. This lets you check whether a patient is found in the PACS and with which DICOM values (e.g. for assigning referrers).

> The default archive cannot be deleted.

PACS Query

Under `/admin/pacs` administrators additionally have a **PACS Query**: choice of archive, period, customer and referrer – with patient, study and series lists. This lets you check which data a specific customer or referrer gets to see.

DirectView

In the section **DirectView** of the archive dialog, you release an archive for embedding the image viewer into other applications and set the order, the foreign archive flag and the DICOM attribute of the case number. The column **DirectView** of the table shows the release. Details: DirectView.

HL7 Communication

The portal exchanges HL7 v2 messages with a RIS/HIS via MLLP – without the gdt-hl7-bridge. It sends the access card of an access code as **MDM^T02** (PDF as Base64 in OBX-5), cancels it with **MDM^T11** on revocation, accepts orders (**ORM^O01**) that create an access code right away, and takes over patient changes (**ADT^A08/A31**, **ADT^A40/A18/A34/A36**, **ADT^A47/A46**).

> HL7 is a licensed module. While it is not active, **Admin → HL7** is hidden, inbound messages are answered with `AR` and no messages are sent. Messages already created stay queued.

Everything is under **Admin → HL7** with the tabs **Partners**, **Messages**, **Inbound mapping**, **Outbound mapping** and **Events**. Processing always follows the same chain:



Shares are also created without an HL7 order (token generator in the browser, PDF endpoint, HL7 endpoint). They raise the same events; the event subscriptions decide who gets which message. Without a matching subscription the portal sends nothing.

Example: order from the RIS, access card back

1. Create a **partner**: host/port of the RIS, **application** `RIS` (MSH-3 of the orders, MSH-5 of the replies), **customer** of the practice. Under **Advanced** the mapping **ORM Standard** is preset for `ORM^O01`.
2. **Events** → subscription "Access card to RIS": event **Share created**, source **Order (ORM)**, recipient **Sender of the order**, mapping **MDM^T02 Standard**.
3. Second subscription "Cancel to RIS": event **Share revoked**, sources **Share revoked** and **Order cancelled/replaced**, recipient **Sender of the order**, mapping **MDM^T11 Standard**.
4. Switch on **Settings** → **HL7** → **Receiving** and open the port (see Setting up receiving).

An `ORC|NW` now creates an access code and sends the access card back to the RIS as MDM^T02; `ORC|CA` revokes it and sends MDM^T11. Both show up under **Messages**.

Partners

A partner is a communication partner (RIS/HIS, ePA connector, document archive). There can be any number of them.

General

Field	Meaning
Host / port	MLLP receiver (TCP). TLS is not supported yet.
Application / facility	Identification of the partner, entered only once: MSH-5/MSH-6 when **sending**, the expected MSH-3/MSH-4 when **receiving** (empty facility = any). Separate components with `^`.
Customer	Customer of the access codes created from orders of this partner.
Active	Inactive partners pause delivery; messages stay queued and are delivered once the partner is active again. Nothing is accepted from inactive partners.

The portal's own identification (MSH-3/MSH-4 of all messages and acknowledgements it sends) is set under **Settings** → **HL7** → **Sender** (`hl7.group.sender.application`, `hl7.group.sender.facility`).

Advanced – preset, rarely changed:

Field	Meaning
HL7 version	MSH-12 (default 2.5)
Character set	UTF-8 or ISO-8859-1; transmitted as such and announced in MSH-18. Characters ISO-8859-1 cannot represent are sent without accents.
Processing ID	MSH-11: P (production), T (training), D (debugging)

Field	Meaning
Timeouts	Connection setup (10,000 ms) and waiting for the acknowledgement (30,000 ms)
Receiving	Accept messages from this partner (on by default) and allowed source addresses (IP/CIDR)
Inbound mapping per message type	e.g. `ORM^O01` → **ORM Standard**. Without a mapping the type is rejected with `AR`.

The button in the **Connection** column tests the TCP connection (no message is sent). Partners with event subscriptions or messages cannot be deleted, only deactivated.

Messages

The tab lists all sent and received messages with partner, customer, patient, source, status and attempts (filter by status, partner, source, direction, type; search by control ID or patient ID). Actions:

- **Details**: message (the Base64 content is shortened), event, event subscription and mapping, acknowledgement, error, SHA-256 checksum, download the PDF
- **Retry now**: deliver a queued message right away (e.g. once the partner is reachable again)
- **Cancel**: stop delivering a queued message
- **Resend**: create a finished message again from the access code as a **new** message (new control ID, same document number, same mapping) – for documents only while the access code is valid

Queue and delivery

Every message is stored together with the operation that caused it and delivered afterwards. The acknowledgement (ACK, MSA-1) is evaluated:

- `AA`/`CA` → **Delivered**
- `AR`/`CR` → **Failed** (no retry)
- `AE`/`CE` → **Failed**; with the setting `hl7.group.retry.ack_ae` retried instead
- no or unreadable acknowledgement, timeout, connection error → **Retry scheduled**: after 30 s, 1 min, 2 min … up to `hl7.group.retry.max_backoff_minutes`; **Failed** after `hl7.group.retry.max_attempts` attempts.

Automatic retries send the same message with the same control ID (MSH-10). If a message cannot be created (e.g. a PDF error or a mapping that does not yield a valid message), it is listed as **Failed** with the reason; the access code or the PDF is created anyway.

All operations are recorded in the audit log (`HL7_MESSAGE_*`, `HL7_DOCUMENT_CANCEL_QUEUED`, `HL7_INBOUND_*`, `HL7_PARTNER_*`, `HL7_MAPPING_*`, `HL7_SUBSCRIPTION_*`).

Retention (data protection)

Messages contain patient data and the access card. The content of finished messages is deleted after

`hl7.group.retention.payload_days` (default 30); status and checksum are kept. After `hl7.group.retention.message_days` (default 365) the entry is deleted completely. Queued messages are never deleted. Deleting a patient also deletes their HL7 messages; a cancel caused by the deletion is kept until it is delivered.

Events

An event subscription defines which event sends which message to which recipient. An event can have several subscriptions (e.g. to the RIS **and** to an archive), and several sources can use the same mapping.

Field	Meaning
Event	**Share created** (`SHARE_CREATED`, message with the access card), **Share revoked** (`SHARE_REVOKED`, cancel), **Patient changed** (`PATIENT_UPDATED`) or **Patients merged** (`PATIENT_MERGED`)
Sources	What caused the event (see below); at least one
Customer	Only shares of this customer (for patient events: patients with an access code of this customer), empty = all
Recipient	**Fixed partner** or **Sender of the order** (the partner the ORM came from; only for shares created from orders). Patient events only go to fixed partners, never to the sender of the ADT message.
Outbound mapping	Builds the message: **Share created** → MDM^T02, **Share revoked** → MDM^T11, **Patient changed** → ADT^A08, **Patients merged** → ADT^A40 or ADT^A47.
Language	Language of the access card; empty = language of the request or of the order

Event	Source	When
Share created	Token generator	Access code created in the browser – sent in the background, without asking the user (e.g. for an ePA or document archive)
Share created	PDF endpoint	Request to `/api/reports/pdf`. Do **not** combine with the HL7 delivery of the gdt-hl7-bridge, otherwise the RIS receives every document twice.
Share created	HL7 endpoint	`/api/reports/hl7` (interface description in the developer guide under **Interfaces** → **HL7 endpoint**). Without a subscription the endpoint refuses and creates no code.
Share created	Order (ORM)	Access code from an ORM order
Share revoked	Share revoked	Revocation in the share overview (admin, customer, patient)
Share revoked	Order cancelled/replaced	ORM cancel (`CA`, `OC`, `DC`, `DR`) or a repeated order with `repeat` = `REPLACE`
Share revoked	Patient deleted	Patient deleted together with their shares

Event	Source	When
Patient changed	Patient data (ADT)	ADT^A08/A31 or an order (ORM) changed the name or birth date of a known patient
Patients merged	Patient data (ADT)	ADT^A40/A18/A34/A36 or ADT^A47/A46 changed the patient ID or merged two patients

Cancel on revocation (MDM^T11)

On revocation, every recipient of a **Share revoked** subscription receives a cancel for every document **it** received for this access code, with the same **document number** (TXA-12) and availability `CA` (TXA-19), without the PDF:

- once per document – even if it was sent several times; revoking again sends nothing;
- partners that never received a document of the code get no cancel;
- documents of the code that are still queued are no longer delivered (status **Cancelled**), independent of the subscriptions.

TXA-12 contains a unique document number (`RW⋯`, the control ID of the first delivery). The order number or patient ID is in the event field `document.orderNumber`, the accession in TXA-14.

Outbound mapping

An outbound mapping builds a message from the data of an event. Built in are **MDM^T02 Standard** (access card), **MDM^T11 Standard** (cancel) and **ADT^A08/A40/A47 Standard (out)** (forwarded patient change); they are maintained by the portal and read-only. To change one, **copy** it and choose the copy in the event subscription.

- Each row writes a value into an HL7 field: the target is a path (`PID-5-1`, `OBX(2)-5`, `PID-3[2]-1`), the source an **event field** (e.g. `patient.lastName`) or a **fixed value**. Several sources: the first one with a value counts.
- Values are HL7-escaped automatically (`|` → `\F\`, `^` → `\S\` …); components and repetitions follow from the path.
- **MSH** and **EVN** are filled by the portal: sender from the settings, receiver = application/facility of the partner, timestamp, message type, control ID, version, character set.
- A segment is only written if at least one value from the event landed in it – so the OBX segments of a cancel without a PDF are left out although they contain fixed values.
- On the left the editor lists all event fields with sample values, below it shows the rendered message with the structure check (HAPI).

Event fields: `customer.`, `patient.externalId/lastName/firstName/birthDate`, `token.`, `link`, `study.(first study, study.count)`, `order.` (order number, accession, Study UID), `visitNumber`, `document.id` (document number, TXA-12), `document.orderNumber`, `document.title`, `document.filename`, `document.pdfBase64`, `document.completionStatus`, `document.availability` (`AV`/`CA`), `event`, `source`, `lang`.

Event fields of the patient events: `patient.externalId/lastName/firstName/birthDate` (new state),

`prior.externalId` (prior ID, merges only; MRG-1), `action`, `event`, `source`. ADT^A08 always contains a PV1 segment (patient class `N`).



The standard mappings put name and birth date into the HL7-compliant fields PID-5/PID-7. The template of the gdt-hl7-bridge uses PID-6/PID-8 – a RIS configured for the bridge may need an adapted copy.

Inbound mapping

An inbound mapping turns a received message into a **processing object**. The partner chooses one per message type under **Advanced**. **ORM Standard** and one **ADT^... Standard** each for A08, A31, A40, A18, A34, A36, A47 and A46 are built in.

- For each target field the sources are checked in order; the first one with a value counts. A source is an HL7 path (`SEG-field-component-subcomponent`, e.g. `PID-3-1`; `OBR(2)-4` = second OBR; `PID-3[2]-1` = second repetition; `PID-3[PI]-1` = the repetition with identifier type `PI` in component 5, see Several IDs in PID-3 / MRG-1; Z segments like any other) or a fixed value.
- Transformations per source, in order: trim spaces, upper/lower case, **value table** (e.g. `NW=CREATE`, with a value for everything else), default value, date (HL7 → ISO), prepend/append text.
- Paste a sample message in the editor (e.g. from 7edit): it is shown as a tree (segment → field → component). Select a source and click a path in the tree to take it over. Below, the editor shows the processing object the mapping yields for the sample message.

Orders (ORM^O01)

Processing object and default assignment (**ORM Standard**):

Field	Default	Meaning
`action`	`ORC-1` → value table	`CREATE` (`NW`, `SN`), `UPDATE` (`XO`, `SC`, `XX`, `SS`), `CANCEL` (`CA`, `OC`, `DC`, `DR`), otherwise `IGNORE` (acknowledged, nothing happens)
`orderNumber`	`ORC-2-1`, `OBR-2-1`	Order number, unique per partner
`fillerOrderNumber`	`ORC-3-1`, `OBR-3-1`	
`patient.id`	`PID-3-1`	The patient is created or updated
`patient.lastName` / `patient.firstName`	`PID-5-1` / `PID-5-2`	
`patient.birthDate`	`PID-7-1` → date	Required if the code is protected by the birth date
`accession`	`OBR-18`, `ORC-3-1`, `OBR-3-1`	
`studyUid`	`ZDS-1-1`	
`visitNumber`	`PV1-19-1`	Visit number, goes into the reply as event field `visitNumber` (PV1-19)

Field	Default	Meaning
`scope`	fixed `PATIENT`	Requested share level `PATIENT`, `ACCESSION` or `STUDY`
`scopeFallback`	–	Fallback levels, comma separated (e.g. `ACCESSION,PATIENT`)
`maxScope`	–	Widest allowed level
`lang`	–	Language of the access card
`repeat`	– (= `KEEP`)	Repeated order, see below

Example level from a Z segment: `scope` ← `ZRW-1` with the value table `P=PATIENT`, `A=ACCESSION`, `S=STUDY`.

Processing:

- **CREATE:** access code right away, event **Share created** (source **Order**). If the order number is already known and its code is active, `repeat` decides (see below). Studies that do not exist yet are added at login with the code (query at the PACS by accession or Study UID) and as soon as the images arrive in the portal – only studies of the same patient.
- **Patient data:** name and birth date are taken over from PID. If they change a known patient, the local Orthanc copy follows as with ADT^A08 (failure → `AE`, nothing kept) and the event **Patient changed** is raised (never back to the sender).
- **UPDATE:** accession, Study UID and patient data are taken over, the code stays; an unknown order (or one without an active code) is handled like **CREATE**.
- **CANCEL:** all access codes of the order are revoked, event **Share revoked** (source **Order cancelled/replaced**).
- **Customer:** the customer of the partner (**General**); without a customer the order is rejected with `AR`.

Repeated order: an order is known if partner and order number (`orderNumber`, otherwise `fillerOrderNumber`) match. A new code is always created if the order was cancelled or its code revoked. If the order still has an active code, `repeat` decides what another **CREATE** does:

`repeat`	Behaviour
`KEEP` (default, also empty)	Update the order (accession, Study UID), code and level stay. Acknowledgement "Order already known, access code unchanged".
`REPLACE`	Revoke the old code (cancel to its recipients via the source **Order cancelled/replaced**), create a new code with a newly decided level and send it as **Share created**.
`NEW`	Create an additional code; the old one stays valid. A later cancel revokes all codes of the order.

`repeat` can be a fixed value of the mapping (then it applies to all orders of the partner) or come from the message, e.g. `ZRW-2` with the value table `N=NEW`, `R=REPLACE` and `KEEP` for everything else. An order number always belongs to one patient: if it comes with another patient ID, the message is rejected with `AR` ("Order number already used for another patient") and nothing is changed.

Share level – the rule itself is fixed, the mapping only provides the values:

1. **Requested level** `scope` (empty = `PATIENT`; any value other than the three levels → rejection).
2. **Maximum level:** the narrower of `maxScope` and the customer's `token.pdf.max_scope`. A wider requested level is reduced to it.
3. **Identifier present?** Study needs the Study UID, accession the accession number. If it is missing, the levels from `scopeFallback` are tried in order (each within the maximum). Without a fitting fallback level the order is rejected with `AR`.

The acknowledgement (MSA-3) and the history show how the level was reached, e.g. "Access code created (ACCESSION): level STUDY; Study Instance UID missing for level STUDY → fallback ACCESSION".

Patient data (ADT)

The RIS reports patient changes via ADT. The source PACS receives the same messages from the RIS and stays the source of truth; the portal follows: its patient record, the shares hanging on it and the local Orthanc copy of the studies. There is a built-in inbound mapping per type (**ADT^A08 Standard**, **ADT^A31 Standard** ...); at the partner under **Advanced** choose the types the RIS/HIS sends, otherwise the type is rejected with `AR`.

Message	`action`	Behaviour
ADT^A08	`UPDATE`	Take over name and birth date from PID. **Note:** the birth date is the second factor of logins with the birth date – after a correction the new date applies.
ADT^A47	`CHANGE_ID`	The patient `prior.id` (MRG-1) gets the ID `patient.id` (PID-3).
ADT^A40	`MERGE`	Like A47; if the new ID already exists in the portal, studies, shares, patient account, orders and HL7 history move to that patient and the prior record is deleted.
ADT^A31	`UPDATE`	Update person information (master patient index); like A08. Many HIS send A31 instead of A08.
ADT^A18	`MERGE`	Merge (old, up to v2.3); like A40.
ADT^A34	`MERGE`	Merge, patient ID only (old); like A40.
ADT^A36	`MERGE`	Merge, patient ID and case number (old); like A40, the case number (MRG-5) is ignored.
ADT^A46	`CHANGE_ID`	Change patient ID (old, predecessor of A47); like A47.

A18, A34, A36 and A46 are withdrawn since HL7 v2.3.1 or v2.4, but many systems still send them; the portal processes them like their successors.

Deliberately **not** processed (rejected with `AR`): A23/A29 (delete – shares are not revoked automatically), A41/A42/A44/A45/A49/A50 (case, account, visit – unknown to the portal), A24/A37 (link/unlink), A01/A04/A05 (admission/registration). A30/A39 (like merge) and A28 (like update) can be added if a specific RIS needs them.

- Only patients the portal already knows are changed; ADT never creates patients. Unknown

patients are acknowledged with `AA` ("unknown … nothing to update").

- **Local Orthanc copy:** the patient's studies (and studies still carrying the old ID) are rewritten (PatientID, name, birth date). Study, Series and SOP Instance UIDs stay the same, only the Orthanc-internal ID changes. If this fails, the portal changes nothing and answers `AE`; the partner may send the message again.
- **Referrers' PACS view:** after successful processing the portal also takes name, birth date and patient ID over into the PACS cache of all referrers (entries with the new or the old ID and the patient's known studies). Referrers see the change right away, a search for the new name finds the study, and retrieving all studies of the patient queries with the new ID – without a new PACS query. A failure here is only logged and does not change the acknowledgement; the same applies to patient changes by order (ORM). Changes made only in the PACS (without ADT) appear after at most 30 days (older studies) or 24 hours (last week), or immediately via **Refresh** in the PACS view.
- If **both** patients have a patient account (login) when merging, the message is rejected with `AR` ("Both patients have a portal account"); an admin has to resolve this.
- Repetitions are harmless: an already merged patient gives `AA` "already merged".
- **Several pairs in one message:** A40 (and the old merges) may repeat the PID/MRG group. The portal processes all pairs one after the other in one transaction: if one pair fails (conflict → `AR`, Orthanc error → `AE`), **nothing** is kept; the acknowledgement names the pair ("Pair 2: …"). On success the acknowledgement and the history name the count, e.g. "2 patient pairs processed: 1) Patients merged; 2) Patient id changed". The mapping is applied to each pair separately – `PID-3-1` and `MRG-1-1` always mean the pair being processed (do not use `PID(2)-⋯` in own mappings). The preview in the mapping editor shows the first pair.
- Afterwards the changes raise the events **Patient changed** or **Patients merged** (see Events) – never back to the partner the change came from.
- Audit log: `PATIENT_UPDATE` or `PATIENT_MERGE`.

Samples: `backend/samples/hl7/adt/`.

Several IDs in PID-3 / MRG-1

The standard mappings take the first repetition (`PID-3-1`, `MRG-1-1`). If a HIS sends several patient IDs (e.g. HIS and RIS ID: `PID|1||K-100^^^KIS^PI~R-200^^^RIS^MR|⋯`) and the portal or the PACS only knows one of them, copy the mapping (**Inbound mapping** → **Copy**) and choose the right ID in the copy:

- **By position:** `patient.id` ← `PID-3[2]-1`, `prior.id` ← `MRG-1[2]-1`. Only suitable if the HIS always sends the IDs in the same order.
- **By identifier type (recommended):** `patient.id` ← `PID-3[MR]-1`, `prior.id` ← `MRG-1[MR]-1`. The first repetition whose component 5 (identifier type code, e.g. `PI`, `MR`, `PT`) matches is chosen; case does not matter. If there is none, the value is empty.
- `PID-3-1` can stay as a second source in case a sender omits the identifier type.

Then choose the copy instead of the standard mapping at the partner under **Advanced** – for every type the HIS sends (ORM too). Selecting by identifier type is only available for reading (inbound); outbound mappings address repetitions by number only.

Setting up receiving

Partners send to the portal via MLLP (TCP port 2575 in the backend container).

1. **Open the port:** set `HL7_INBOUND_PORT` in the `.env`, e.g. `HL7_INBOUND_PORT=2575` (default: `127.0.0.1` only), restart the stack and open the firewall for the address of the RIS.
2. **Switch on receiving: Settings → HL7 → Receiving** `hl7.group.inbound.enabled`. Switched off, the portal closes every connection right away, without an acknowledgement.
3. **Set up the partner: application** (= MSH-3 of the partner), optionally **facility** (MSH-4); under **Advanced** receiving on, allowed source addresses and one inbound mapping per message type.

Behaviour:

- A message is assigned to the partner via MSH-3 (and MSH-4, if set); the source address must be within the allowed ranges. Unknown senders get `AR` and are only recorded in the audit log (`HL7_INBOUND_REJECTED`) and the backend log.
- Message types without an inbound mapping at the partner are rejected with `AR`.
- Every message of a known partner is listed in the history (direction **Received**) with the acknowledgement sent, status **Processed** (`AA`) or **Rejected** (`AR`/`AE`), with the customer of the access code created or of the partner.
- If the partner sends an already **processed** message with the same control ID (MSH-10) again, it gets the same acknowledgement; it is not processed a second time. A **rejected** message changed nothing and is processed again when it is resent (e.g. after the settings were corrected).
- Character set: according to MSH-18 (`UNICODE UTF-8`, `8859/1`), otherwise the partner's character set. The acknowledgement is sent in the same character set.



Enter the allowed source addresses. Without them the portal accepts messages with the matching MSH-3 from any address.

DirectView

DirectView embeds the portal's HTML5 image viewer into another application and shows archive data there directly. The **calling application** can be, for example, a HIS, a RIS, wound documentation software, practice software or an application from veterinary medicine. It embeds the portal address `/directview` in an iframe.

The portal retrieves the images via DICOM query/retrieve into its image storage (Orthanc) and shows them with the built-in image viewer. Patients and studies are **not** created in the portal database. The images stay in image storage only as long as they are needed.

DirectView is a licensed module (see Licence & Modules). The menu item **Admin → DirectView** appears as soon as the module is active.

How a call works

1. The calling application opens `/directview` with a one-time code (recommended) or a DirectView

API key and exactly one search parameter (e.g. the Study Instance UID).

2. The portal looks up the examination in the **primary archive**. It must belong to exactly one patient; otherwise no images are shown for safety reasons.
3. The requested examination is loaded series by series; the image viewer opens as soon as it is complete.
4. Then the **prior studies** selected by the rules are loaded, first from the primary archive, then from the further archives in their order.
5. The header reports further loaded studies; **show** reloads the image viewer with them. In the study list, the user can **load** studies no rule selected one by one.

The user only sees studies of the patient of the call.

Prerequisites

Switch the module on

Switch DirectView on under Licence & Modules.

Release archives

Under Archives, set per archive in the section **DirectView**:

- **Release for DirectView**: only released archives are queried.
- **Order**: archives are queried in ascending order. The first released archive that is not a foreign archive is the **primary archive**.
- **Foreign archive**: archive with its own patient IDs (e.g. external or older images). The patient is matched there by family name, first given name and birth date, and by sex if present. Spellings with umlauts, hyphens or different letter case are normalised. Studies from there are marked "Matched by name/birth date" in the view.

  The procedure is deliberately chosen for foreign image archives: their data comes from referring institutions with their own patient IDs and is usually deleted again after about 30 days (FIFO). For a regional hospital, two different patients with the same name and birth date within that period are so unlikely that the remaining risk is acceptable. Do not use the option for archives with a permanent, large stock.

- **DICOM attribute of the case number**: which attribute is queried for a call with a case number (default `AdmissionID`).

Create an API key

Under **Admin** → **DirectView** → **API keys**, create a key of its own for every calling application. A DirectView API key opens DirectView only, no other function of the portal (unlike integration tokens).

- The key is shown **only once** on creation; the portal does not store it in plain text. Enter it in the calling application right away.

- **Allowed IP ranges** are mandatory: the networks of the devices the application runs on and, for launch codes, of its server.
- **Calls per minute** limit the load on archive and image storage if a key is misused.
- **User name required** refuses calls without a user name of the calling application.
- **Revoke** also ends running sessions of the key immediately.

Allow embedding

The operator enters the address of the embedding application into the variable `DIRECTVIEW_FRAME_ANCESTORS` (see Operation); otherwise the browser blocks the embedding.

Call from the application

Recommended: launch code (server to server)

The application's server requests a one-time code with the API key and shows only the returned address in the iframe. The key never reaches the browser, and the user name counts as confirmed in the audit log.

```
POST https://portal.klinik.de/api/directview/launch
X-DirectView-Key: <API_KEY>
Content-Type: application/json

{"studyuid": "1.2.276.123.456.789", "user": "<USER>"}
```

Response: `{"url": "https://portal.klinik.de/directview?launch=<CODE>", "expiresAt": "…"}`. The code is valid for 60 seconds and exactly once.

Alternative: API key in the iframe address

Only if the application cannot make server calls:

```
https://portal.klinik.de/directview?apikey=<API_KEY>&studyuid=1.2.276.123.456.789&user=<USER>
```

The key is then visible in the HTML of the application and in the browser; the user name counts as **unconfirmed** in the audit log. The portal removes the key from the address right away and masks it in its logs.

Parameters

Parameter	Meaning
`launch`	One-time code from `/api/directview/launch`
`apikey`	DirectView API key (only the alternative without launch code)

Parameter	Meaning
`studyuid`	Study Instance UID of the requested study
`accnbr`	Accession number of the examination
`caseid`	Case number (DICOM attribute configurable per archive)
`patid`	Patient ID in the primary archive; the newest study is shown
`user`	optional: user name in the calling application, recorded in the audit log

Exactly one of the parameters `studyuid`, `accnbr`, `caseid` or `patid` is required. Wildcards (`*`, `?`) are not allowed.

A call is valid as long as set under **Settings** (default: 60 minutes), and only on the device that opened it (IP address and browser). The access token is kept in the page's memory only: after a reload or the expiry, the user opens the images again from the calling application.

Rules for prior studies

Under **Admin** → **DirectView** → **Rules** you define which studies are loaded in addition to the requested one.

- **Trigger**: the rule applies when the requested study matches, e.g. modality `CT` and study description `thorax`. Empty fields always match.
- **Selection**: which studies of the same patient are loaded: modalities (`SAME` = same as the requested study), study description, body part (any, same or matching an expression), maximum age in months, maximum number of studies and for which archives.
- Description and body part are checked as case-insensitive **regular expressions**, e.g. `thorax|lung`. The body part comes from the DICOM attribute `BodyPartExamined` of the series (e.g. `CHEST`, `KNEE`). RE2 syntax applies (linear time): backreferences such as `\1` and lookaround such as `(?!…)` are not allowed.

All active rules whose trigger matches are applied; the setting **Maximum prior studies loaded automatically** limits the total. If no rule matches, no prior studies are loaded automatically.

Test rules shows, for a Study Instance UID or patient ID, which studies in the primary archive would be selected. No images are retrieved.

Settings

Setting	Meaning
Validity of a call	minutes during which the user can view images after opening them from the application (default 60)
Retention in image storage	hours after the last use after which DirectView images are deleted (default 24)
Maximum prior studies loaded automatically	upper limit per call (default 20)

Setting	Meaning
Concurrent retrievals	concurrent C-MOVE retrievals of all calls together (default 2)

Calls, image storage and audit log

- **Calls** shows the latest 100 calls with time, type, application user and status.
- **Image storage** shows the number and size of the DirectView images; **Clean up now** deletes images no longer needed right away. Studies the portal itself uses (e.g. for a release) are kept.
- The audit log records every call (application user), every retrieved study (with the patient ID from the PACS) and rejected foreign archive matches.

Operation

`DIRECTVIEW_FRAME_ANCESTORS` in the `.env` holds the addresses of the applications allowed to embed DirectView, separated by spaces, `https` only, without a path:

```
DIRECTVIEW_FRAME_ANCESTORS=https://kis.klinik.de https://visite.klinik.de
```

The CheckMK check `env_plausibility` reports invalid entries. The embedding permission only applies to `/directview` and the image viewer.

Chrome and Edge additionally check access to the local network (**Local Network Access**): if the embedding application runs under a public address and the portal under an address in the hospital network, the browser blocks the embedding (`ERR_BLOCKED_BY_LOCAL_NETWORK_ACCESS_CHECKS`) or asks the user for permission. Run both in the same network, or the application has to embed the iframe with `allow="local-network-access"`.

For DirectView the image viewer runs under an address of its own (`/directview-viewer/`); the normal image viewer of the portal (`/viewer/`) stays protected against embedding.

The portal masks API keys, launch codes and access tokens in its own logs (nginx, Caddy). Upstream proxies of the hospital (e.g. Pangolin or a hospital reverse proxy) may still log the full address – so prefer the launch code or switch off logging of query parameters there.

Questions for the vendor of the calling application

For the connection, clarify with the vendor of the application (e.g. Telekom Healthcare for the HIS iMedOne / iMedOne Mobile):

1. Can the application make a server call (`POST` with its own header) before opening the iframe and show the returned address? Then use the launch code.
2. Which parameters are available for the call (Study Instance UID, accession number, case number, patient ID, user name)?
3. From which addresses do the server calls and the browsers of the devices it runs on come (for the allowed IP ranges)?

4. Under which address (origin) does the user interface that embeds the iframe run (for `DIRECTVIEW_FRAME_ANCESTORS`)?
5. Does the user interface of the application run in the same network as the portal? If it comes from the internet: can the iframe be embedded with `allow="local-network-access"`?
6. Are certificates and content security policies (CSP) of the embedded page supported, and is the iframe embedded with `allow="fullscreen; local-network-access"`?

Licence & Modules

Some functions of the portal are **licensed modules**:

Module	Function
DirectView	Embedding of the HTML5 image viewer into other applications for direct display of archive data, without storing in the portal (see DirectView)
HL7	HL7 communication with RIS and HIS: access codes from orders (ORM), patient data (ADT), sending and cancelling access cards (MDM) (see HL7 Communication)
Referrer portal	Sign-in and PACS view for referring physicians (see Referrers)
Customers (tenants)	Number of customers that can be created

A module is only **active** when it is **licensed** and **switched on**. Under **Admin** → **Licence & modules** you see the licence state and switch licensed modules on or off. Menu items and pages of a module that is not active (**Master Data** → **Referrers**, **Admin** → **HL7**, **Admin** → **DirectView**) are hidden.



From this version on, the referrer portal and HL7 require a licence as well. Install the licence file right after the update. Until then, referrer sign-ins are refused, HL7 messages are answered with `AR` and no HL7 messages are sent. Data and settings are kept.

Install the licence file

1. Open **Admin** → **Licence & modules**.
2. Click **Upload licence file** and choose the file you received from RadioWeb (`.json`).
3. The portal checks the signature and the portal address and then shows the licensee, the validity and the licensed modules.

A new licence file replaces the previous one. The upload is meant for installations without the RadioWeb Manager (offline).

Centrally managed licence (RadioWeb Manager)

For installations through the RadioWeb Manager, RadioWeb issues the licence in the manager. The updater installs it automatically and replaces it on renewal, without restarting the portal. The page then shows the source **Managed centrally** and a note that no upload is needed.

- A centrally managed licence takes precedence over an uploaded licence. While it is installed and valid, the portal rejects uploads.
- You still switch the modules on and off here.
- If the licence file is missing or invalid, a previously uploaded licence applies again.

Technically the file is `license.json` in the server's licence directory (`/etc/<product>/license`, `/run/radioweb/license` in the container). Without the manager, operators can also place the file there themselves.

Portal address

The licence is valid for specific portal addresses (host names). The setting **Frontend URL** (`frontend.url`) is decisive. If it does not match the licence, the licence is shown as **Invalid** and the modules are not active.

Status

Status	Meaning
Valid	Licence is fine
Expires soon	less than 30 days until the end
Expired – grace period	expired, the module keeps running for 14 days
Expired	Module is switched off
Invalid	Signature, key or portal address do not match
Not licensed	Module is not part of the licence

When a licence expires, only the modules are switched off. Configuration and data are kept and are available again as soon as a new licence is installed. HL7 messages already created stay queued and are sent as soon as the module is active again.

Switch modules on and off

In the table **Modules**, switch each licensed module on or off with **Switched on**. Modules that are not licensed cannot be switched on. DirectView is switched off after installation, the other modules are switched on.

Number of customers (tenants)

The module **Customers (tenants)** has no switch. The licence defines how many customers can be created:

- **Without a licence** for the module (or after the grace period) **one** customer is allowed.
- **With a licence** the licensed number (at least 1) applies or, without a number, an unlimited number.

The table shows how many customers exist and how many are allowed. Once the quota is reached,

Add customer in the customer management is locked. If there are more customers than the licence allows (e.g. after the licence expired), all existing customers stay fully usable; only new customers cannot be created.

Monitoring

The CheckMK check `license_status` warns 30 days before the licence or a switched-on module ends and for modules that are switched on but not licensed. From the end date it reports critical, also during the 14-day grace period in which the module still works. It provides the remaining days as metric `days_remaining`. All changes to the licence and the modules are recorded in the audit log.

Settings

Settings (menu **Admin**) configures the behaviour of the portal. The settings are organised in tabs; each setting shows its label, technical key (e.g. `app.support_email`), description and level (`SYSTEM`).

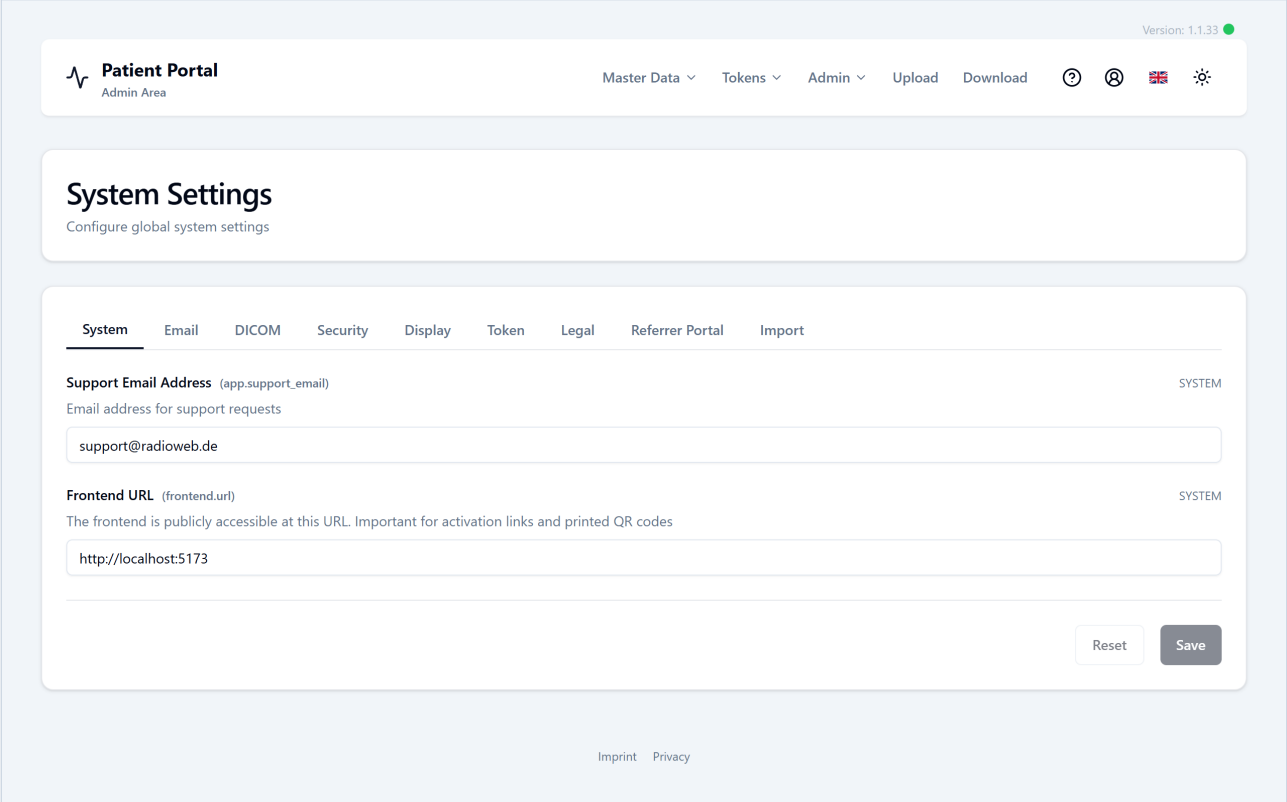


Figure 19. System settings

Tab	Typical settings
System	Support e-mail address
Email	SMTP server, credentials and sender for activation e-mails and for sending access links
DICOM	Operating mode of the archive: *Query/Retrieve* (retrieve images from the PACS on demand, recommended) or *Store* (data is sent to the portal and kept there)

Tab	Typical settings
Security	Password rules, session timeouts per role, which roles may create which access codes, create which accounts and download which data, permanent login for patients
Display	Default language, date format, colour scheme (light/dark/system), **Login page**: which sign-in forms (access code, e-mail/password) are shown and whether they are side by side or stacked
Token	Per token type (patient, study, referrer): active yes/no, protection (none, date of birth, password), default validity and selectable validity periods
Legal	Details for imprint and privacy policy: operator, contact, data protection officer, supervisory authority, medical chamber, register details (shown in the portal under **Imprint** / **Privacy**)
Referrer Portal	Settings of the PACS view for referrers
Import	DLX import and automatic forwarding per user group, target archive (AE title, host, port) for foreign studies – global or per user group; import list on the upload page: maximum number of entries per card (default 20) and time window in hours (default 24)

Change the desired values and click **Save**; **Reset** discards unsaved changes.



If the login page only shows the access-code sign-in (*Access code only*), administrators and practice logins switch to the e-mail/password sign-in with the link **Sign in with email and password** below the card; it can also be opened directly via `/login?form=password`. Conversely, `/login?form=token` shows the access-code sign-in; access-code links and QR codes work regardless of the setting.



A complete list of all settings with their default values is contained in the "System Settings" reference in the developer documentation.

Customer Area

Users with the role `customer` are staff of a practice or clinic (a customer). They sign in with e-mail and password; the account is created by an administrator under Users and linked to the customer.

The navigation contains:

Menu item	Function
Releases	All releases (access codes) of the own customer with status, search and the option to print releases again or revoke them.
Upload	Send prior examinations via DLX import or ZIP upload (if enabled for customers), see Uploading Prior Examinations.
Download	Desktop and integration apps (RadioWeb Bridge, gdt-hl7-bridge, Portal-Launch) and the administration guide as PDF.

Releases

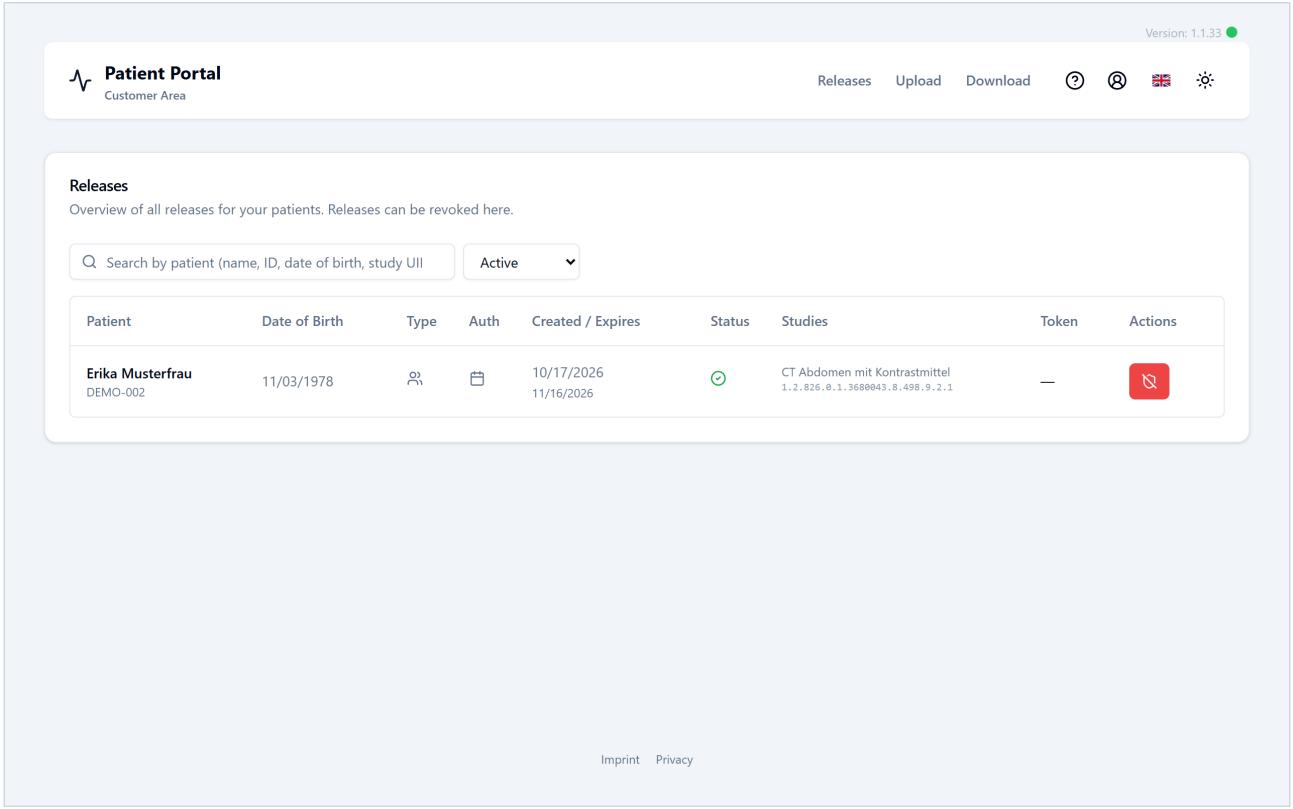


Figure 20. Releases in the customer area

The view corresponds to the administrators' release management but only shows releases of the own customer. Use the red icon to revoke a release immediately and the printer icon to print an active release again (see Printing a Release Again).

Creating Access Codes

Depending on the setting **Security**, customers may create access codes for referrers, patients and

studies. In practice this is usually done directly from the PACS or practice management system via the integrations (QR code when printing reports, Portal-Launch).

Uploading Prior Examinations (Upload)

Via **Upload** (`/import/upload`), users send images from other institutions to the portal: with **Portal Import** (DLX) directly from another portal, or as a **ZIP upload** from a CD or a download. The studies are stored in the portal, marked as foreign studies and – if configured – automatically forwarded to a Foreign Studies Archive (see Import Status).

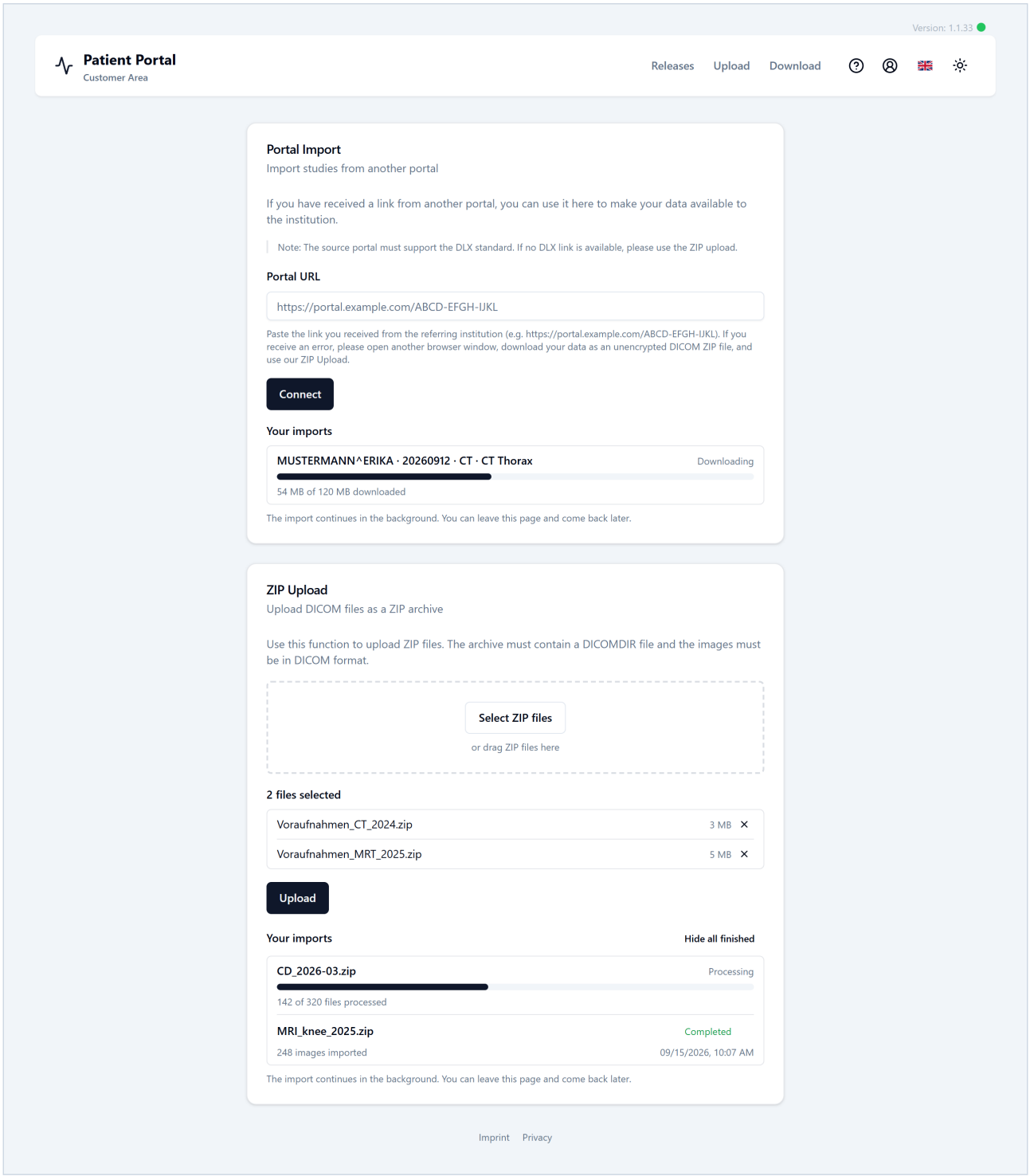


Figure 21. The upload page with running and completed imports

Portal Import (DLX)

If you have received a share link from another portal, the patient portal can fetch the studies directly from there. This requires the other portal to support the **DLX** (DICOM Link Exchange) standard.

1. Paste the link into the **Portal URL** field and click **Connect**.
2. If asked, answer the other portal's security questions (for example the date of birth) and click **Verify**.
3. Select the studies and click **Import studies**.

The portal then downloads the selected studies from the other portal itself; you can follow the progress under **Your imports** (see Tracking progress).

If no DLX portal is found at the address, a notice appears: open the link yourself, download the data there as an (unencrypted) DICOM ZIP file and use the ZIP upload.

ZIP Upload

1. Click **Select ZIP files** and choose one or more archives – or drag the ZIP files into the dashed area.
2. Check the list of selected files; click **×** to remove a file again.
3. Click **Upload**.

Several files are uploaded one after another; each file then appears as its own entry under **Your imports**. Files that are not ZIP files are ignored. If uploading a file fails, the remaining files are still uploaded; the failed file stays selected so you can try again.

The ZIP archive must contain images in DICOM format – as provided by most patient CDs and portals when downloading. The images may also be located in subfolders or in further ZIP files inside the archive. A single file may be at most 512 MB unless your practice has set a different limit. Split larger amounts of data into several ZIP files.

Tracking progress

While uploading, a bar shows how much has already been transferred (for several files: "file 2 of 5 …"). The portal then processes the data in the background. Under **Your imports** you can see the current state of each import:

- **Waiting** – the import will start shortly.
- **Downloading** – Portal Import only: the study is being fetched from the other portal.
- **Processing** – the images are being stored; the bar shows how many files have already been processed.
- **Completed** or **Failed** – with the number of imported images or the reason for the failure.

You do not have to wait until an import has finished: you can leave the page and come back later. Imports from the last 24 hours remain visible (the time window and number of entries are configurable).

You can hide finished entries yourself: hover over an entry and click the × (always visible on tablets and smartphones), or click **Hide all finished**. Imports that are still running always remain visible.

Setup

All settings are found under Settings in the **Import** category.

Setting	Default	Meaning
User groups with DLX import (`dlximport.enabled_groups`)	Admin, customer, referrer	For which roles the **Portal Import** card is shown.
User groups with ZIP import (`zipimport.enabled_groups`)	Admin, customer, referrer	For which roles the **ZIP Upload** card is shown. If neither option is enabled for a role, the **Upload** menu item is hidden.
Auto-routing (`dlximport.autorouting_groups`, `zipimport.autorouting_groups`)	–	Roles whose uploads are forwarded automatically to the Foreign Studies Archive (target: `import.foreign_studies.*`).
Upload page: maximum number of imports (`import.foreign_studies.upload_history.max_entries`)	20	Maximum number of imports the **Your imports** list shows per card (1–200).
Upload page: time window in hours (`import.foreign_studies.upload_history.window_hours`)	24	Imports from the last X hours are shown (1–720).

Operation

- Every import runs as a background job. Up to two imports are processed at the same time; further imports wait with the status **Waiting**.
- Finished imports are deleted at night after 7 days – or after the configured time window if it is longer. Users can hide them earlier themselves.
- If the server is restarted during an import, the portal resumes it after the start. If the uploaded file no longer exists (e.g. because the container was recreated), the import ends with a notice about the restart; the user then has to start it again.
- By default, a single upload file may be at most 512 MB. The operator of the installation sets the limit with `UPLOAD_MAX_SIZE_MB`. If a file exceeds it, the portal reports this when uploading.
- You can follow the forwarding to the Foreign Studies Archive under Import Status.

Integration

Overview

This document describes the integration of external systems with the Patient Portal.

The Patient Portal can be connected to the existing systems of a radiology practice. The integration scripts and applications are located in the `integration/` workspace of the monorepo, organized by system and vendor.

```
integration/
├──── apps/
│    └──── gdt-hl7-bridge/   # GDT and HL7 integration (Windows application)
├──── pacs/
│    └──── vepro/            # VEPRO PACS
└──── ris/
     └──── mesalvo/          # Mesalvo RIS
```

PACS

VEPRO

Buttons for sharing studies and creating QR codes directly from the VEPRO Archive Manager.

RIS

Mesalvo RadCentre

Link call from the RadCentre RIS to the Patient Portal, e.g. to create a QR code for the patient.

GDT / HL7

gdt-hl7-bridge

Windows application that monitors a GDT input directory, retrieves PDF reports from the portal, generates GDT response files and sends HL7 MDM^T02 messages via MLLP/TCP. Enables integration of the Patient Portal into any GDT 2.1-compatible practice management software.

portal-launch

Windows program that opens the Patient Portal directly from the PMS or RIS – without a separate login.

REST API

Share API (generate-share)

For custom integrations: REST endpoint for server-side generation of a patient or study share including a QR code, authenticated via an integration token. For cases not covered by VEPRO, `gdt-hl7-bridge` or `portal-launch`. The interface is described in the developer documentation (section "Interfaces").

PACS: VEPRO

Overview

The VEPRO integration allows users to share studies or patients directly from the VEPRO Archive Manager to the Patient Portal and generate a QR code for the patient.

Components

File	Description
`VIM/VIM.INI`	Configuration of the additional buttons in the VEPRO Archive Manager
`VIM/Scripts/VMedView.vbs`	VBScript executed when a button is clicked; opens the browser with the Patient Portal

Further Pages

- Installation – Setting up the buttons in VEPRO
- Create QR Code – Step-by-step guide for end users

Installation

Prerequisites

- VEPRO is installed and running
- The Patient Portal is reachable (URL known)
- An integration token has been created in the Patient Portal admin area under **Integration Tokens**

Step 1 – Deploy the Script

Copy the file `VMedView.vbs` from the workspace `integration/pacs/vepro/VIM/Scripts/` into the VEPRO VIM scripts directory, typically:

```
C:\VIM\Scripts\VMedView.vbs
```

Step 2 – Configure the Script

Open `VMedView.vbs` in a text editor and adjust the three constants at the top of the file:

```
Const PORTAL_BASE_URL   = "https://your-portal.example.com/generate-token"
Const BROWSER_PATH      = """C:\Program Files
(x86)\Google\Chrome\Application\chrome.exe"""
Const INTEGRATION_TOKEN = "XXXX-XXXX-XXXX"  '①
```

① Enter the integration token from the Patient Portal admin area.

Step 3 – Add Buttons to VIM.INI

Open the `VIM.INI` file (in the VEPRO configuration directory) and add the following lines in the `[VIMConfig]` section:

```
ArchiveManagerButtonCustom1 = Studie freigeben,
CALL_SCRIPT_METHOD(SendStudyToPatientPortal), Studie ins Patientenportal senden
ArchiveManagerButtonCustom2 = Patient freigeben,
CALL_SCRIPT_METHOD(SendPatientToPatientPortal), Patient ins Patientenportal senden
```

The ready-to-use `VIM.INI` from the workspace `integration/pacs/vepro/VIM/` can be used as a template.

Step 4 – Restart VEPRO

Restart VEPRO. The two new buttons **Studie freigeben** and **Patient freigeben** will then appear in the Archive Manager toolbar.

Result

After a successful installation, users can generate QR codes for patients or individual studies directly from VEPRO.

Next: Create QR Code

Create QR Code

This guide explains how a user creates and prints a QR code for a patient or an individual study directly from VEPRO.

Prerequisites

- VEPRO is open and the user is logged in
- The integration has been set up (Installation)

Workflow

Step 1 – Select a Record

In the **Archive Manager**, select the record you want to share:

- **Share Study** – click on a study in the list first
- **Share Patient** – selecting any study of the patient is sufficient

Step 2 – Click the Button

Click one of the two buttons in the toolbar:

Button	Effect
Studie freigeben (Share Study)	Shares exactly the selected study. The patient gains access to this one examination only.
Patient freigeben (Share Patient)	Shares all studies of the patient. The patient gains access to all their examinations in the portal.

A browser window opens automatically with the Patient Portal token generator.

Step 3 – Set Access Parameters (depending on configuration)

Depending on the portal settings, one or more of the following options may be presented:

Option	Meaning
Type	Type of sharing, e.g. study or patient
Authentication	Determines whether the patient must additionally enter their date of birth when opening the QR code
Validity Period	Time period for which the QR code is valid (e.g. 7 days, 30 days)



If the portal is configured to use default values, this step is skipped and the QR code is displayed immediately.

Step 4 – Display and Print the QR Code

After confirming the parameters, the QR code appears in the browser window.

1. Select the desired **print language** (German / English)
2. Click **Print**
3. The browser print dialog opens — select the desired printer and confirm

The printed slip contains the QR code along with a short explanatory text for the patient.

Result

The patient can scan the QR code with their smartphone and access their images in the Patient Portal directly — without a username or password.

RIS: Mesalvo

Overview

The Mesalvo integration allows users to add a link button directly in the RadCentre RIS to open the Patient Portal — for example, to generate a QR code for a patient.

Further Pages

- Create Link Button in RadCentre – Set up a button in MedDoku or the findings/examination module

Create Link Call

Introduction

This guide describes how to add a link button in RadCentre at various locations, for example to open the Patient Portal QR code generator directly from the RIS.

Add via MedDoku

1. Start RadCentre and log in with an admin user.
2. Select a test patient.
3. Double-click on an order — the selection window opens.
4. In the selection window press **SHIFT + F6** to open the cascade window. Then click the **New Entry** button.
5. A new entry appears (in the example row D, highlighted in blue). Enter the name of the new cascade entry there — pay attention to any existing numbering (e.g. A., B., C. → then D. "new name").
   Under **Action**, enter the command chain and confirm with **Apply**. Finally click **Save**.

   > Command chain for the Patient Portal QR code call:
   > `y_shellexec("open", "https://pp.kesrbtz.de/generate-token?patid="+alltrim(oGapPatient:Nr))`
   > For other command chains, support from Mesalvo may be required depending on what is to be called.

6. Quit RadCentre and restart it so the change takes effect.
7. To verify, repeat steps 2 and 3 — the new entry should now appear in the list.

Add via Examination or Findings Module

1. Start RadCentre and log in with an admin user.
2. Select the **Findings** or **Examination** module.
3. In the lower right area click the **three dots** — a window opens.
4. Click the **New** button.
5. Enter the desired link name under **Label**. Select **External** or **URL** as the type, then enter the desired link. If required, insert a placeholder using the **Placeholder** button. Finally click **Save**.
6. Restart RadCentre, log in again and open the examination module again.
7. Click the **+/-** button, select **Add** and then select the newly created link from the list.

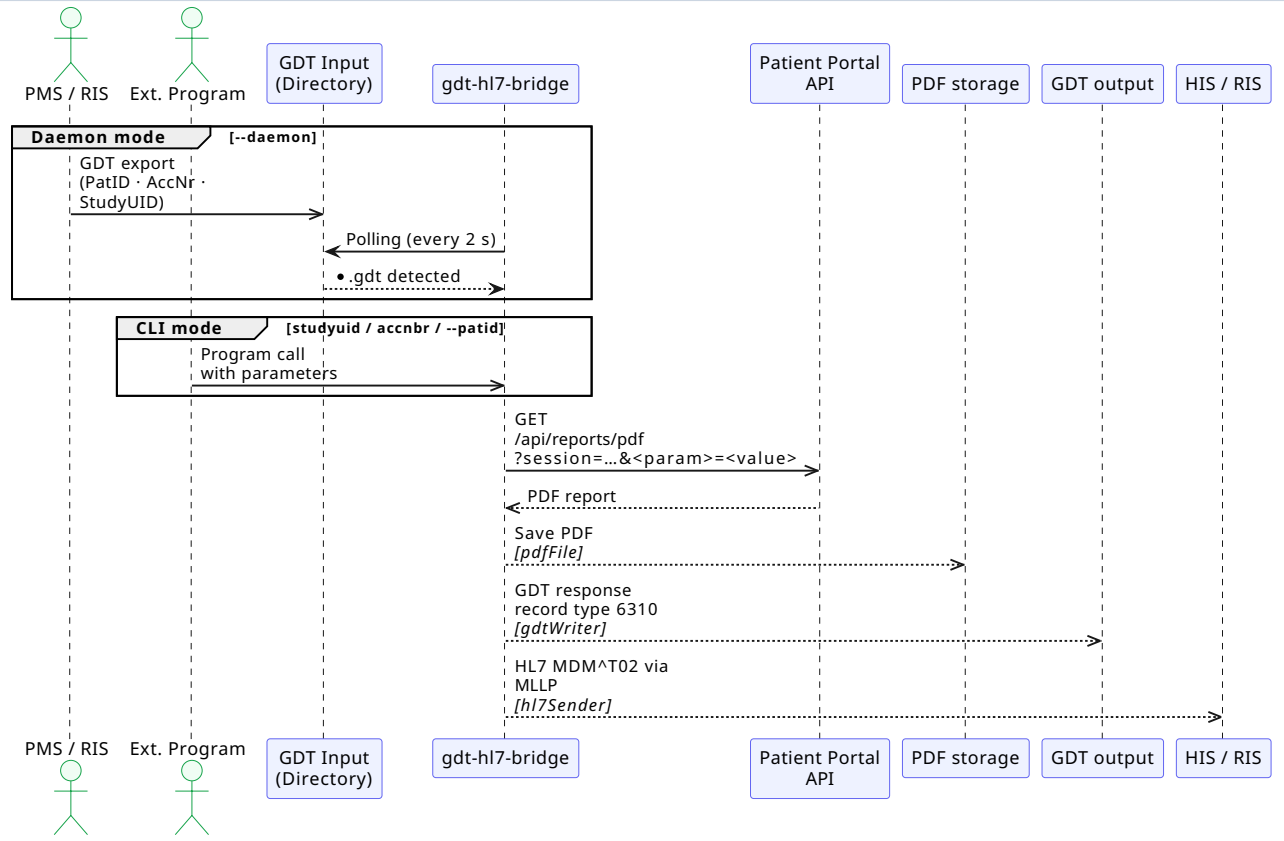
GDT/HL7: gdt-hl7-bridge

Overview

`gdt-hl7-bridge` is a Windows application that retrieves PDF reports from the Patient Portal and optionally generates a GDT response file and/or an HL7 MDM^T02 message. The application supports two invocation modes:

- **Daemon mode** (`--daemon`): Runs in the background and monitors a GDT directory. Each incoming GDT file from the practice software automatically triggers the retrieval.
- **CLI mode** (without `--daemon`): Single invocation with direct parameters (`--studyuid`, `--accnbr`, `--patid`). Enables integration into any external program without a GDT interface.

How It Works



Invocation Modes

Daemon Mode

```
gdt-hl7-bridge.exe --daemon
```

Starts the GDT polling service. Runs continuously in the background and automatically processes all incoming `*.gdt` files in `gdtImport.gdtInputDir`. Suitable for operation as a Windows service or autostart task.

CLI Mode

```
gdt-hl7-bridge.exe --studyuid 2.25.186760260011452398476108131496539831798
gdt-hl7-bridge.exe --accnbr 186B7928
gdt-hl7-bridge.exe --patid 3209648408 --nachname Mueller --vorname Max
```

Executes the pipeline once and exits with code `0` (success) or `1` (error). Suitable for direct integration from RIS, PACS, or worklist systems without the calling software needing to write GDT files.

Help

```
gdt-hl7-bridge.exe
gdt-hl7-bridge.exe --help
```

Displays help text. Also shown when called without any parameters.

Directory Structure

```
gdt-hl7-bridge.exe
config/
  config.json          ①
  config.json.example  ②
templates/
  gdt-output.template  ③
  mdm-output.template  ④
nssm.exe               ⑤
install.bat
uninstall.bat
```

① Active configuration file (not checked in, contains API token)

② Template with all available fields

③ Handlebars template for the GDT response file

④ Handlebars template for the HL7 MDM^T02 message

⑤ Windows service wrapper (Non-Sucking Service Manager)

Modules

Output modules can be enabled independently in `config.json`. Both invocation modes use the same enabled modules.

Module	Function
`pdfFile`	Download the PDF report from the portal and save it locally
`gdtImport`	Monitor the GDT input directory (daemon mode only)

Module	Function
`gdtWriter`	Generate a GDT response file (record type 6310)
`hl7Sender`	Send an HL7 MDM^T02 message via MLLP/TCP and/or save as file

Further Pages

Installation

Package contents, prerequisites, firewall requirements, Windows service setup, first start, and testing.

Configuration

Complete reference for `config.json` — all modules, field mapping, priority logic, filename templates, and GDT response template.

Operation

CLI reference, processing workflow, error handling, HL7 message queue, and logging.

HL7 MDM^T02

Protocol overview, MLLP transport, and template reference for the MDM^T02 message.

Installation

Package Contents

`gdt-hl7-bridge` is delivered as a ZIP package and contains the following files:

```
gdt-hl7-bridge.exe        ← Executable (standalone, no .NET / Java required)
config\
  config.json.example     ← Configuration template
templates\
  gdt-output.template     ← GDT response template (GDT 2.1, record type 6310)
  mdm-output.template     ← HL7 MDM^T02 template (only needed when hl7Sender is
enabled)
nssm.exe                  ← Non-Sucking Service Manager (Windows service wrapper)
install.bat               ← Install as service (run as Administrator)
uninstall.bat             ← Uninstall service (run as Administrator)
samples\
  input-patid.gdt         ← Test file: lookup by patient number
  input-accnbr.gdt        ← Test file: lookup by accession number
  input-studyuid.gdt      ← Test file: lookup by DICOM Study UID
```

The package is available for download under *Patient Portal → Download → GDT / HL7 Interface*.

Prerequisites

- Windows 10 or later (64-bit)

- Network access to the Patient Portal (HTTPS, standard port 443)
- If `hl7Sender.enabled: true`: Network access to the HL7 receiver (TCP, configured port)
- Customer API token — generated in the portal admin under *Settings* → *Integration*
- Practice management software must have write access to `gdtImport.gdtInputDir` and read access to `gdtWriter.gdtOutputDir`



If a firewall is in place between the installation machine and the Patient Portal or HL7 receiver, the following outbound connections must be allowed:

Destination	Port / Protocol	Purpose
Patient Portal host	443 / HTTPS (or customer-specific port)	PDF retrieval via the portal API
HL7 receiver (HIS / RIS)	Configured TCP port (e.g. 2333)	MLLP message delivery (only if `hl7Sender.enabled: true`)

Setting Up the Directory Structure

Extract the ZIP package into a directory of your choice, e.g. `C:\RadioWeb\`. The resulting structure:

```
C:\RadioWeb\
├─── gdt-hl7-bridge.exe
├─── nssm.exe
├─── install.bat
├─── uninstall.bat
├─── config\
│    ├─── config.json.example
│    └─── config.json          ①
├─── templates\
│    ├─── gdt-output.template
│    └─── mdm-output.template  ②
└─── samples\
     ├─── input-patid.gdt
     ├─── input-accnbr.gdt
     └─── input-studyuid.gdt
```

① Must be created manually from `config.json.example` (contains the API token, not included in the package)

② Only required if `hl7Sender.enabled: true` or `hl7Writer.enabled: true`



The installation path is freely configurable. All directories are specified as absolute paths in `config.json`.

Creating the Configuration

1. Copy `config\config.json.example` to `config\config.json`.
2. Open `config\config.json` in a text editor.
3. Fill in at least the following required fields:

```
{
  "portalUrl": "https://portal.your-practice.com",  ①
  "apiToken":  "XXXX-XXXX-XXXX",                    ②

  "gdtImport": {
    "enabled":      true,
    "gdtInputDir":  "C:\\RadioWeb\\GDT\\Input"       ③
  },

  "gdtWriter": {
    "enabled":      true,
    "gdtOutputDir": "C:\\RadioWeb\\GDT\\Output",     ④
    "senderId":     "RadioWeb"
  },

  "pdfFile": {
    "enabled": true,
    "pdfDir":  "C:\\RadioWeb\\PDFs"                  ⑤
  }
}
```

① URL of the Patient Portal (no trailing slash)

② Integration token — generated by the portal administrator under *Settings → Integration*

③ GDT input directory: the practice software writes its GDT export files here

④ GDT output directory: the practice software reads the GDT response files from here

⑤ Directory for locally stored PDF reports

All further options (HL7, logging, field mapping) are described in Configuration. Unused modules can remain disabled with `"enabled": false`.

> Directories are created automatically when first needed — they do not need to be created in advance.

Setting Up as a Windows Service (Recommended)

The preferred method is to install `gdt-hl7-bridge` as a Windows service using the included **NSSM** wrapper. The service starts automatically at system boot and is monitored by the Windows Service Manager.

1. Open a **Command Prompt as Administrator** (`cmd.exe → right-click → Run as Administrator`).

2. Change to the installation directory:

```
cd C:\RadioWeb
```

3. Run the installation script:

```
install.bat
```

The script installs the service `GDT-HL7-Bridge` and starts it immediately.

Action	Command (as Administrator)
Start service	`net start GDT-HL7-Bridge`
Stop service	`net stop GDT-HL7-Bridge`
Check service status	`sc query GDT-HL7-Bridge`
Uninstall service	`uninstall.bat`

Alternative: Task Scheduler

If Administrator access for service installation is not available, the Windows Task Scheduler can be used as an alternative.

1. Open the **Task Scheduler** (`taskschd.msc`).
2. Create a new **Task** (not a basic task):
   - **General**: Enable *Run whether user is logged on or not*
   - **Trigger**: At system startup
   - **Action**: Start a program
     - Program/script: `C:\RadioWeb\gdt-hl7-bridge.exe`
     - Arguments: `--daemon` ← **Required — without this argument no daemon starts**
     - Start in: `C:\RadioWeb\`
   - **Settings**: Enable *Run task as soon as possible after a scheduled start is missed*

First Start / Verifying the Configuration

Start the application manually from the command prompt to verify the configuration:

```
C:\RadioWeb\gdt-hl7-bridge.exe --daemon
```

On successful configuration, the following output appears (depending on enabled modules):

```
gdt-hl7-bridge gestartet [--daemon]
```

```
Input:     C:\RadioWeb\GDT\Input
GDT Out:   C:\RadioWeb\GDT\Output
PDFs:      C:\RadioWeb\PDFs
HL7 TCP:   ris-server:2333
HL7 Queue: queue/hl7 (retry 60000ms)
Interval:  2000 ms
```

Press `Ctrl+C` to stop the application. Then start the service or scheduled task.

If the startup fails with an error message, check the following:

Error Message	Cause / Solution
`config/config.json not found`	Configuration file missing or not in the `config\` subdirectory next to the `.exe`
`config.json missing "portalUrl"`	Required field `portalUrl` not set in the configuration
`config.json "gdtImport.gdtInputDir" missing`	`gdtImport` is enabled but `gdtInputDir` is not set
`--daemon requires gdtImport.enabled: true`	`gdtImport.enabled` is set to `false`; in daemon mode `gdtImport` must be active
`Template not found: templates/…`	Template file missing or `templateFile` path in the configuration points to a non-existent file
`Failed to compile template`	Syntax error in the Handlebars template

Testing the Installation

After a successful start, the installation can be verified using one of the included test files.

Prerequisites

The test files contain sample values (patient IDs, accession numbers). Make sure a matching patient or study exists in the portal for the values used, or adjust the test file accordingly.

Procedure

1. Open File Explorer and navigate to `C:\RadioWeb\samples\`.
2. Copy `input-patid.gdt` to the configured `gdtImport.gdtInputDir` (e.g. `C:\RadioWeb\GDT\Input\`).
3. Wait up to one polling interval (default: 2 seconds).

Expected result:

- The file disappears from `GDT\Input\` and appears in `GDT\Input\done\`
- A new PDF file is present in the configured `pdfDir`
- A new `.gdt` response file is present in the configured `gdtOutputDir`

In case of error:

- The file is in `GDT\Input\error\` with a timestamp suffix (e.g. `input-patid_20260701-143052.gdt`)
- A `.err` file with the error message is located next to it

Common errors on first test:

Error message in .err file	Cause
`HTTP 401`	API token invalid or not active
`HTTP 400 [TOKEN_NOT_FOUND]`	API token not registered in the portal
`HTTP 400 [TOKEN_EXPIRED]`	The API token's validity has expired — extend it under *Settings → Integration* or create a new token
`HTTP 400 [TOKEN_IP_RESTRICTED]`	The API token is restricted to IP ranges, and the public IP of the installation computer is not among them
`HTTP 400 [PATIENT_ID_MISMATCH]`	The study found belongs to a different patient than the one in field 3000 — the practice software's patient number and the PatientID in the PACS do not match
`HTTP 400 [STUDY_NOT_FOUND]`	No study exists in the portal for the patient ID in the test file
`HTTP 503 [ORTHANC_CONNECTION_ERROR]`	PACS is not reachable (only for PACS-dependent queries)
`ECONNREFUSED` / `ETIMEDOUT`	Portal URL not reachable — check network or firewall settings

Configuration

All configuration for `gdt-hl7-bridge` is done via the file `config\config.json`. A template with all available fields is provided as `config\config.json.example`.

The configuration is modular: each functional block is controlled by its own object with an `enabled` flag. Unused modules can be disabled without removing their configuration.

Full Example

```
{
  "portalUrl": "https://portal.your-practice.com",
  "apiToken":  "XXXX-XXXX-XXXX",

  "pdfFile": {
    "enabled":     true,
    "pdfDir":      "C:\\RadioWeb\\PDFs",
    "pdfFileName":
"{{patid}}_{{nachname}}_{{vorname}}_{{accnbr}}_{{timestamp}}.pdf"
  },

  "gdtImport": {
    "enabled":       true,
    "gdtInputDir":   "C:\\RadioWeb\\GDT\\Input",
    "pollIntervalMs": 2000,
```

```
    "mapping": {
      "patid":        "3000",
      "nachname":     "3101",
      "vorname":      "3102",
      "geburtsdatum": "3103",
      "accnbr":       "8402",
      "fallnummer":   "8402",
      "studyuid":     "8410"
    },
    "required": ["patid"],
    "priority":  ["studyuid", "accnbr", "patid"]
  },

  "gdtWriter": {
    "enabled":      true,
    "templateFile": "templates/gdt-output.template",
    "gdtOutputDir": "C:\\RadioWeb\\GDT\\Output",
    "gdtFileName":  "{{patid}}_{{accnbr}}_{{timestamp}}.gdt",
    "senderId":     "RadioWeb"
  },

  "hl7Sender": {
    "enabled":      true,
    "templateFile": "templates/mdm-output.template",
    "host":         "ris-server",
    "port":         2333,
    "queue": {
      "enabled":         true,
      "dir":             "queue/hl7",
      "retryIntervalMs": 60000
    }
  },

  "hl7Writer": {
    "enabled":      false,
    "templateFile": "templates/mdm-output.template",
    "hl7OutputDir": "C:\\RadioWeb\\HL7\\Output",
    "hl7FileName":  "{{patid}}_{{accnbr}}_{{timestamp}}.hl7"
  },

  "log": {
    "enabled":  true,
    "level":    "info",
    "file":     "logs/gdt-hl7-bridge-%DATE%.log",
    "maxSize":  "10m",
    "maxFiles": "30d"
  },

  "vars": {
```

```
    "institute": "Dres. Mueller Schmidt",

    "doc6302": "",
    "doc6304": "Radiological Report",

    "SF2":  "Dres. Mueller Schmidt",
    "SF3":  "1 Example Street",
    "SF6":  "00000 Anytown",
    "SF7":  "Germany",
    "SF16": "+49 0815 4711"
  }
}
```

Connection

Field	Required	Description
`portalUrl`	yes	Base URL of the Patient Portal, e.g. `https://portal.your-practice.com`. No trailing slash.
`apiToken`	yes	Customer integration token. Generated in the portal admin under *Settings* → *Integration* and validated server-side.

Module: `pdfFile`

Controls the download and local storage of the PDF report.

Field	Required	Default	Description
`enabled`	no	`false`	Enable the module. If `false`, no PDF is downloaded.
`pdfDir`	yes (if enabled)	–	Directory where downloaded PDF reports are stored. Created automatically when needed.
`pdfFileName`	yes (if enabled)	–	Filename for the saved PDF. Handlebars template, see Template Variables.

Module: `gdtImport`

Controls monitoring of the GDT input directory and reading of GDT fields.

Field	Required	Default	Description
`enabled`	no	`false`	Enable the module. Without this module, the application does not monitor any directory.
`gdtInputDir`	yes (if enabled)	–	Directory monitored for incoming `*.gdt` files. Created automatically on startup.
`pollIntervalMs`	no	`2000`	Polling interval in milliseconds.

Field	Required	Default	Description
`mapping`	yes (if enabled)	–	Object: mapping key → GDT field ID. See Field Mapping (`gdtImport.mapping`).
`required`	yes (if enabled)	–	Array of mapping keys that must be present in the input file.
`priority`	yes (if enabled)	–	Array of mapping keys in descending priority for the portal API call.
`scope`	no	portal setting	Share level of the created access code: `patient`, `accession`, `study` or `auto`. See Share Level (`gdtImport.scope`).
`scopeValues`	no	–	Translates values of the GDT field `mapping.scope` into share levels, e.g. `{"P": "patient", "A": "accession"}`.

Module: `gdtWriter`

Generates a GDT response file (record type 6310) with the local PDF path.

Field	Required	Default	Description
`enabled`	no	`false`	Enable the module.
`templateFile`	no	`templates/gdt-output.template`	Path to the Handlebars template (relative to the application directory or absolute).
`gdtOutputDir`	yes (if enabled)	–	Directory for GDT response files. Created automatically when needed.
`gdtFileName`	yes (if enabled)	–	Filename for the GDT response file. Handlebars template, see Template Variables.
`senderId`	no	`RadioWeb`	Identifier of the sending application in the GDT response file (field 8316).

Module: `hl7Sender`

Sends an HL7 MDM^T02 message via MLLP/TCP to an HL7 receiver.

For a complete description of the HL7 message format, see HL7 MDM^T02.

Field	Required	Default	Description
`enabled`	no	`false`	Enable the module.
`templateFile`	yes (if enabled)	–	Path to the Handlebars template for the HL7 message.
`host`	yes (if enabled)	–	Hostname or IP address of the MLLP listener.
`port`	yes (if enabled)	–	TCP port of the MLLP listener.

Field	Required	Default	Description
queue – Message queue on send failure			
`queue.enabled`	no	`false`	Enable the queue. On connection failure, the message is stored in `queue.dir` instead of being lost. Full description: HL7 Message Queue.
`queue.dir`	yes (if queue.enabled)	–	Directory for queued HL7 messages. Created automatically.
`queue.retryIntervalMs`	no	`60000`	Interval in milliseconds at which the daemon drains the queue. In CLI mode, the queue is drained immediately on each invocation.

Module: `hl7Writer`

Writes the HL7 MDM^T02 message as a text file to a local directory. Useful for testing or downstream file-based processing. Can be enabled independently of `hl7Sender`.

Field	Required	Default	Description
`enabled`	no	`false`	Enable the module.
`templateFile`	yes (if enabled)	–	Path to the Handlebars template for the HL7 message.
`hl7OutputDir`	yes (if enabled)	–	Output directory for HL7 files. Created automatically when needed.
`hl7FileName`	yes (if enabled)	–	Filename for the HL7 output file. Handlebars template, see Template Variables.

Module: `log`

Writes all runtime output to a daily rotating log file. Console output remains active — this module supplements it with a persistent log file.

Field	Required	Default	Description
`enabled`	no	`false`	Enable file logging.
`level`	no	`info`	Minimum log level: `error`, `warn`, `info`, `verbose`, `debug`.
`file`	no	`logs/gdt-hl7-bridge-%DATE%.log`	Path to the log file (relative to the application directory or absolute). `%DATE%` is replaced with the current date (`YYYY-MM-DD`). The directory is created automatically.
`maxSize`	no	`10m`	Maximum file size per log file. Units: `k` (KB), `m` (MB), `g` (GB).
`maxFiles`	no	`30d`	Retention period. Suffix `d` = days (e.g. `30d`), without suffix = maximum number of files (e.g. `14`).

`vars` – Global Template Variables

The `vars` object defines freely configurable variables available in *all* templates (GDT and HL7). Typical use cases are practice or institution data that rarely changes.

```
"vars": {
  "institute": "Dres. Mueller Schmidt",
  "doc6302": "",
  "doc6304": "Radiological Report",
  "SF2":  "Dres. Mueller Schmidt",
  "SF3":  "1 Example Street",
  "SF6":  "00000 Anytown",
  "SF7":  "Germany",
  "SF16": "+49 0815 4711"
}
```

Keys can be freely chosen and are accessible in templates directly as `{{institute}}`, `{{SF2}}`, etc.

Two keys have a special role in the default GDT response template:

Key	Meaning
`doc6302`	GDT field 6302 (report request marker). If empty or not set, the field is omitted.
`doc6304`	GDT field 6304 (document title). If empty or not set, the field is omitted.

Field Mapping (`gdtImport.mapping`)

The `mapping` object maps logical identifiers (mapping keys) to the corresponding GDT field IDs. All mapped fields are read from the input file and are available as template variables.

Key	GDT Field ID	Meaning
`patid`	`3000`	Patient number
`nachname`	`3101`	Patient last name
`vorname`	`3102`	Patient first name
`geburtsdatum`	`3103`	Date of birth (GDT format DDMMYYYY)
`accnbr`	`8402`	Order number / accession number
`fallnummer`	`8402`	Case number (alias for `accnbr`, same field ID)
`studyuid`	`8410`	DICOM Study UID (field assignment is vendor-dependent)
`scope`	any (optional)	Share level per request, see Share Level (`gdtImport.scope`). Only set it if the practice software can fill a free field.



GDT field IDs can be adjusted if the sending practice software uses different field IDs.

Required Fields (`gdtImport.required`)

The `required` array contains the mapping keys that must be present in the input file. If a required field is missing, the file is moved to `error/` with an error message.

```
"required": ["patid"]
```

Priority (`gdtImport.priority`)

The `priority` array determines which value is used for the API call when multiple mapped fields are present. Exactly the first key for which a value is found in the input file is used.

```
"priority": ["studyuid", "accnbr", "patid"]
```

Table 1. Examples

Fields present in GDT	Parameter used	Query
`studyuid` + `patid`	`studyuid`	`?studyuid=2.25.186…`
`accnbr` + `patid`	`accnbr`	`?accnbr=186B7928`
`patid` only	`patid`	`?patid=3209648408`

Share Level (`gdtImport.scope`)

When a PDF is requested, the portal creates an access code. The share level determines which studies this access code shares.

Value	Shared studies
`patient`	All studies of the patient (patient share).
`accession`	All studies of the accession number. When requested by `studyuid`, these are all studies with the same accession number as that study.
`study`	Only the requested studies (study share): exactly that study when requested by `studyuid`, all studies of the accession number when requested by `accnbr`.
`auto`	`accession` if the GDT file contains an accession number or Study UID, otherwise `patient`.

The level is determined in this order:

1. GDT field `mapping.scope`, if it is filled in the input file. Values are translated via `scopeValues`, otherwise used as they are.
2. `gdtImport.scope` (in the profile or the template, see Profiles (`profiles`)).
3. Without either, the portal setting **PDF Access Code Share Level** (`token.pdf.max_scope`, default `PATIENT`) applies.

The portal setting is also the widest level allowed: if it is set to `ACCESSION`, for example, the portal rejects requests with `patient`.

In the following cases, the input file is moved to `error/` together with an `.err` file, and no access code is created:

- invalid value in the GDT field,
- `accession` or `study` without accession number and Study UID in the file (`SHARE_SCOPE_REQUIRES_STUDY`),
- level wider than allowed by the portal (`SHARE_SCOPE_NOT_ALLOWED`).

In CLI mode, the level is passed with `--scope patient|accession|study|auto`.

Share level via a GDT field (e.g. 6200)

```
"mapping": {
  "patid":  "3000",
  "accnbr": "8402",
  "scope":  "6200"
},
"scopeValues": { "P": "patient", "A": "accession", "S": "study" },
"scope": "auto"
```

Profiles (`profiles`)

With profiles, one bridge installation runs several independent flows, e.g. one per button in the practice software. Each profile has its own input directory and can configure all modules differently, e.g. the share level, a different HL7 port or an additional `hl7Writer`.

- **Without `profiles`**, the whole configuration is the only profile (`default`). Existing configurations keep working unchanged.
- **With `profiles`**, everything outside `profiles` is only a template and does not run itself. Only the profiles run.
- A profile contains only its differences from the template:
  - Objects are merged, i.e. `"hl7Sender": { "port": 2334 }` only changes the port.
  - Arrays and single values replace the template value (e.g. `priority`).
  - `null` removes a template value.
- `"enabled": false` directly in the profile disables the whole profile.
- `log` is global and must not appear in profiles. Log lines contain the profile name, e.g. `Processing [accession]: …`.
- `portalUrl` and `apiToken` can differ per profile, e.g. for a second practice with its own integration token.
- Keys starting with `_` are comments and are ignored.

```
{
  "portalUrl": "https://portal.your-practice.com",
  "apiToken":  "XXXX-XXXX-XXXX",
  "log":       { "enabled": true, "level": "info", "file": "logs/gdt-hl7-bridge-
%DATE%.log" },

  "pdfFile":   { "enabled": true, "pdfDir": "C:\\RadioWeb\\PDFs", "pdfFileName":
"{{patid}}_{{accnbr}}_{{timestamp}}.pdf" },
  "gdtImport": {
    "enabled":  true,
    "mapping":  { "patid": "3000", "accnbr": "8402", "studyuid": "8410" },
    "required": ["patid"],
    "priority": ["studyuid", "accnbr", "patid"]
  },
  "gdtWriter": { "enabled": true, "gdtOutputDir": "C:\\RadioWeb\\GDT\\Output",
"gdtFileName": "{{patid}}_{{timestamp}}.gdt" },
  "hl7Sender": {
    "enabled": false, "templateFile": "templates/mdm-output.template", "host":
"ris-server", "port": 2333,
    "queue": { "enabled": true, "dir": "queue/hl7" }
  },
  "hl7Writer": { "enabled": false, "templateFile": "templates/mdm-output.template",
"hl7OutputDir": "C:\\RadioWeb\\HL7\\Output", "hl7FileName":
"{{patid}}_{{timestamp}}.hl7" },

  "profiles": {
    "patient": {
      "gdtImport": { "gdtInputDir": "C:\\RadioWeb\\GDT\\Patient", "scope":
"patient" }
    },
    "accession": {
      "gdtImport": { "gdtInputDir": "C:\\RadioWeb\\GDT\\Accession", "scope":
"accession" },
      "hl7Sender": { "enabled": true, "port": 2334, "queue": { "dir": "queue/hl7-
accession" } },
      "hl7Writer": { "enabled": true }
    }
  }
}
```

In the example, `patient` creates a patient share and a GDT response. `accession` only shares the accession number, additionally sends HL7 to port 2334 and also stores the HL7 message as a file.

Rules checked at startup

- All required fields are checked per profile (after merging). Error messages name the profile, e.g. `"profiles.accession.hl7Sender.host" missing`.
- Each profile needs its own `gdtImport.gdtInputDir`.

- Profiles whose HL7 messages go to different receivers (`host:port`) each need their own `hl7Sender.queue.dir`. The queue of a directory is always delivered to exactly one receiver.

In CLI mode, select the profile with `--profile <name>`. If there is only one profile, it is not required.



After every change, run `gdt-hl7-bridge --print-profiles` to check what each profile actually does. The output shows the merged values and marks values coming from the profile with *.

Template Variables

The following variables are available in filename templates (`pdfFileName`, `gdtFileName`, `hl7FileName`). All keys defined in `mapping` are also available, provided the corresponding field is present in the input file.

Variable	Content
`{{patid}}`	Patient number (GDT field 3000)
`{{nachname}}`	Last name (GDT field 3101)
`{{vorname}}`	First name (GDT field 3102)
`{{geburtsdatum}}`	Date of birth in format DDMMYYYY (GDT field 3103)
`{{accnbr}}`	Order number / accession number (GDT field 8402)
`{{fallnummer}}`	Case number – identical to `accnbr` (GDT field 8402)
`{{studyuid}}`	DICOM Study UID (GDT field 8410, vendor-dependent)
`{{scope}}`	Determined share level (`patient`, `accession`, `study`; empty if the portal setting applies)
`{{timestamp}}`	Unix timestamp in milliseconds
`{{date}}`	Current date in format `YYYY-MM-DD`
`{{time}}`	Current time in format `HH-MM-SS`
`{{inputFile}}`	Input filename without extension
`{{<key>}}`	Any variable from `vars` is available directly under its key name

GDT Response Template

The template `templates\gdt-output.template` controls the structure of the generated GDT response file in GDT 2.1 format. It uses Handlebars syntax.

The helper `{{gdtLine "FFFF" "content"}}` generates a correctly formatted GDT line including length prefix and CRLF. The length is calculated automatically: 3 (length field) + 4 (field ID) + content length + 2 (CRLF).

Default Template

```
{{~gdtLine "8000" "6310"~}}
```

```
{{~gdtLine "9206" "2.10"~}}
{{~gdtLine "8316" senderId~}}
{{~#if (lookup fields "8316")~}}{{gdtLine "8315" (lookup fields "8316")}}{{~/if~}}
{{~#if (lookup fields "8402")~}}{{gdtLine "8310" (lookup fields "8402")}}{{~/if~}}
{{~#if (lookup fields "3000")~}}{{gdtLine "3000" (lookup fields "3000")}}{{~/if~}}
{{~#if (lookup fields "3101")~}}{{gdtLine "3101" (lookup fields "3101")}}{{~/if~}}
{{~#if (lookup fields "3102")~}}{{gdtLine "3102" (lookup fields "3102")}}{{~/if~}}
{{~#if (lookup fields "3103")~}}{{gdtLine "3103" (lookup fields "3103")}}{{~/if~}}
{{~#if (lookup fields "3104")~}}{{gdtLine "3104" (lookup fields "3104")}}{{~/if~}}
{{~#if pdfPath~}}{{~#if doc6302~}}{{gdtLine "6302" doc6302}}{{~/if~}}{{gdtLine
"6303" "PDF"}}{{~#if doc6304~}}{{gdtLine "6304" doc6304}}{{~/if~}}{{gdtLine "6305"
pdfPath}}{{~/if~}}
```

Template Context

Variable	Content
`fields`	All GDT fields from the input file, accessible by field ID: `(lookup fields "3000")`
`pdfPath`	Absolute local path to the downloaded PDF file (empty if `pdfFile.enabled: false`)
`senderId`	Value from `gdtWriter.senderId`
`doc6302`	Value from `vars.doc6302` – optional report request marker (field 6302)
`doc6304`	Value from `vars.doc6304` – optional document title (field 6304)
`{{<key>}}`	All further variables from `vars`

Generated GDT Fields

Field ID	Meaning
`8000`	Record type: `6310` (examination data)
`9206`	GDT version: `2.10`
`8316`	Sender ID (`senderId`)
`8315`	Receiver ID – taken from input field 8316 (practice software identifier), only if present
`8310`	Order number of the sender – taken from input field 8402 (accession number), only if present
`3000` / `3101` / `3102` / `3103` / `3104`	Patient data from the input file (only if present)
`6302`	Report request marker (from `vars.doc6302`, optional)
`6303`	File format: `PDF` (always set when `pdfPath` is present)
`6304`	Document title (from `vars.doc6304`, optional)
`6305`	File reference – local path to the PDF file

HL7 MDM^T02

After each processed GDT request, the `hl7Sender` module sends an HL7 message of type **MDM^T02** (*Medical Document Management – Original Document Notification*). The message contains the PDF report as a Base64-encoded attachment as well as DICOM metadata from the incoming GDT request.

This allows downstream systems (HIS, RIS) to receive the report automatically without relying on the GDT response protocol.



The portal can also send MDM^T02 messages itself (partners and event subscriptions, persistent queue, history with resend), see HL7 Communication. Do not enable both paths for the same request (`hl7Sender` of the bridge and an event subscription with source **PDF endpoint** in the portal), otherwise the RIS receives every document twice.

Transmission Modes

Both output channels can be enabled independently and can be active at the same time.

Mode	Activation	Description
MLLP/TCP	`hl7Sender.enabled: true`	The message is sent to an HL7 listener via the MLLP protocol. Waits for an ACK (MSA segment), timeout 10 seconds.
File output	`hl7Writer.enabled: true`	The message is stored as a UTF-8 text file in `hl7Writer.hl7OutputDir`. Useful for testing or downstream file-based processing.

Message Format

The message conforms to HL7 v2.3.1, message type MDM^T02:

Segment	Content
`MSH`	Message header – sender `RADIOWEB|PORTAL`, timestamp, message ID
`EVN`	Event timestamp
`PID`	Patient data (ID, name, date of birth)
`PV1`	Visit information (case number)
`TXA`	Document announcement (type `RAD`, status `AU|AV`)
`OBX[1]`	PDF report as Base64 attachment (`ED` type, MIME `AP/PDF`)
`OBX[2]`	Document description (modality, study description from `vars`)

MLLP Protocol

MLLP (*Minimum Lower Layer Protocol*) is the standard transport protocol for HL7 messages over TCP. `gdt-hl7-bridge` frames every message as follows:

```
0x0B  ← Start Block (Vertical Tab)
<HL7 message as UTF-8 text>
0x1C  ← End Data (File Separator)
0x0D  ← Carriage Return
```

The receiver's MLLP listener must understand this framing (standard in all common HL7 engines).

Configuration

Example

```
"hl7Sender": {
  "enabled":      true,
  "templateFile": "templates/mdm-output.template",
  "host":         "ris-server",
  "port":         2333,
  "queue": {
    "enabled":         true,
    "dir":             "queue/hl7",
    "retryIntervalMs": 60000
  }
},

"hl7Writer": {
  "enabled":      false,
  "templateFile": "templates/mdm-output.template",
  "hl7OutputDir": "C:\\RadioWeb\\HL7\\Output",
  "hl7FileName":  "{{patid}}_{{accnbr}}_{{timestamp}}.hl7"
}
```

Fields

`hl7Sender` – **MLLP/TCP transmission**

Field	Required	Default	Description
`enabled`	no	`false`	Enable MLLP/TCP transmission.
`templateFile`	yes (if enabled)	–	Path to the Handlebars template for the MDM^T02 message (relative to the application directory or absolute).
`host`	yes (if enabled)	–	Host name or IP address of the HL7 receiver (MLLP listener).
`port`	yes (if enabled)	–	TCP port of the MLLP listener.
queue – message queue on transmission errors			

Field	Required	Default	Description
`queue.enabled`	no	`false`	Enable the queue. If transmission fails (connection refused, timeout), the message is stored in `queue.dir` instead of being lost.
`queue.dir`	yes (if queue.enabled)	–	Directory for buffered HL7 messages. Created automatically. In daemon mode the queue is flushed periodically as soon as the receiver is reachable again.
`queue.retryIntervalMs`	no	`60000`	Interval in milliseconds at which the daemon tries to flush the queue. Only active in daemon mode.

`hl7Writer` – file output

Field	Required	Default	Description
`enabled`	no	`false`	Enable file output. Independent of `hl7Sender`.
`templateFile`	yes (if enabled)	–	Path to the Handlebars template (can be identical to `hl7Sender.templateFile`).
`hl7OutputDir`	yes (if enabled)	–	Output directory for HL7 files. Created automatically if needed.
`hl7FileName`	yes (if enabled)	–	File name of the HL7 output file. Handlebars template, see Template Variables.

Template (`mdm-output.template`)

The template controls the exact structure of the MDM^T02 message and uses Handlebars syntax.

Default Template

```
MSH|^~\&|RADIOWEB|PORTAL|SCN|RIS|{{now}}||MDM^T02|{{msgId}}|P|2.3.1|
EVN||{{now}}
PID|||{{patientId}}|||{{patientName}}||{{birthDate}}
PV1||O|||||||||||||||||{{fallnr}}
TXA|1|RAD||{{now}}|||||||||{{docId}}||||{{filename}}|AU||AV
OBX|1|ED|18748-4^Diagnostic Imaging
Report^LN||qr2mdm^AP^PDF^Base64^{{pdfBase64}}||||||F|||||||{{studyUid}}
OBX|2|ST|^Document Description||VEPRO QR Code {{modality}} {{studyDescription}}
```

Template Variables

Variable	Source	Content
`{{now}}`	automatic	HL7 timestamp in the format `YYYYMMDDHHmmss` (e.g. `20260628143022`)
`{{msgId}}`	automatic	Unique message ID (Base-36-encoded Unix timestamp)

Variable	Source	Content
{{patientId}}	GDT field 3000	Patient number
{{patientName}}	GDT fields 3101, 3102	Name in HL7 format LastName^FirstName
{{birthDate}}	GDT field 3103	Date of birth in HL7 format YYYYMMDD (converted automatically from GDT format DDMMYYYY)
{{fallnr}}	GDT field 8402	Case number
{{studyUid}}	GDT field 8410	DICOM Study UID
{{docId}}	GDT field 8402 or 3000	Document ID (order number, fallback: patient number)
{{filename}}	automatic	File name of the locally stored PDF (without path)
{{pdfBase64}}	automatic	PDF file, Base64-encoded. The PDF is always downloaded when hl7Sender or hl7Writer is active – regardless of pdfFile.enabled.
{{<key>}}	vars	All variables from config.json → vars, e.g. {{modality}}, {{studyDescription}}

{{modality}} and {{studyDescription}} cannot be derived from the GDT request and must be set via vars in the configuration if needed.

GDT encodes dates of birth in the format DDMMYYYY; HL7 expects YYYYMMDD. The conversion for {{birthDate}} happens automatically.

Testing the Connection

A simple TCP listener can be used to test the MLLP receiver:

```
# Linux/macOS
nc -l 2333

# Windows (PowerShell, requires ncat or similar)
ncat -l 2333
```

After processing a GDT test file, the MLLP-framed HL7 message appears in the terminal. The listener does not need to send an ACK – gdt-hl7-bridge closes the connection after receiving FS CR or when the connection is closed.

Operation

CLI Reference

Usage

```
gdt-hl7-bridge.exe [--daemon] [--studyuid <uid>] [--accnbr <nr>] [--patid <id>]
                   [--nachname <name>] [--vorname <name>]
                   [--geburtsdatum <DDMMYYYY>] [--fallnummer <nr>]
                   [--help]
```

Parameters

Parameter	Mode	Description
`--daemon`	Daemon	Starts the GDT polling service. Runs continuously in the background.
`--studyuid <uid>`	CLI	DICOM Study UID as the search parameter for PDF retrieval.
`--accnbr <nr>`	CLI	Order number / accession number as the search parameter.
`--patid <id>`	CLI	Patient number as the search parameter.
`--scope <level>`	CLI	Share level of the access code: `patient`, `accession`, `study` or `auto`. Defaults to `gdtImport.scope`, otherwise the portal setting applies. See Share Level.
`--profile <name>`	CLI	Profile whose settings are used. Required if several profiles are configured. See Profiles.
`--print-profiles`	Check	Prints all profiles as the bridge uses them (template and profile merged): input directory, GDT fields, search order, share level and all outputs. Values set in the profile are marked with *, the integration token is masked. With `--profile <name>`, only that profile. Does not call the portal and does not create directories.
`--nachname <name>`	CLI	Last name – optional, for filename templates and HL7 context.
`--vorname <name>`	CLI	First name – optional, for filename templates and HL7 context.
`--geburtsdatum <DDMMYYYY>`	CLI	Date of birth in GDT format – optional, automatically converted to YYYYMMDD for HL7.
`--fallnummer <nr>`	CLI	Case number – optional, for filename templates and HL7 (PV1 segment).
`--help` / `-h`	both	Show help and exit. Also shown when called without parameters.

When multiple search parameters are provided, `gdtImport.priority` (default: `studyuid → accnbr → patid`) determines which one is used for the API call.

Exit Codes (CLI Mode)

Code	Meaning
0	Pipeline completed successfully
1	Error (no search parameter, HTTP error, MLLP timeout, etc.)

Examples

```
# Retrieve PDF by Study UID, send HL7 MDM
gdt-hl7-bridge.exe --studyuid 2.25.186760260011452398476108131496539831798

# Retrieve PDF by accession number, with patient data for HL7 context
gdt-hl7-bridge.exe --accnbr 186B7928 --patid 3209648408 --nachname Mueller
--vorname Max

# Call from a batch script with error check
gdt-hl7-bridge.exe --studyuid %STUDYUID%
if errorlevel 1 echo ERROR retrieving PDF
```

Directory Structure During Operation

`gdt-hl7-bridge` manages three subdirectories below `gdtImport.gdtInputDir`:

```
gdtInputDir/
 ├──── *.gdt                          ← New input files (placed here by the
practice software)
 ├──── done/
 │     └──── *.gdt                     ← Successfully processed files
 └──── error/
     ├──── *_YYYYMMDD-HHmmss.gdt      ← Failed files (timestamp appended to
filename)
     └──── *_YYYYMMDD-HHmmss.gdt.err  ← Error message as a text file (same base
name)
```

The `done/` and `error/` subdirectories are created automatically when first needed.

Processing Workflow

For each `*.gdt` file found, the following steps are executed:

1. **Stability check** – Files modified within the last 500 ms are skipped (protection against ongoing writes by the practice software).
2. **GDT parsing** – The input file is read in GDT 2.1 format (ISO-8859-1 encoding). Both compact format (without spaces) and standard format are supported.
3. **Field resolution** – All fields defined in `gdtImport.mapping` are extracted from the file.

4. **Required field check** – Missing required fields from `gdtImport.required` result in an error.
5. **Priority selection** – The first key from `gdtImport.priority` for which a value is found determines the API parameter.
6. **Share level** – The share level is determined from the GDT field `mapping.scope` or the profile's `scope` (see Share Level).
7. **PDF retrieval** – The PDF is downloaded from the portal at `/api/reports/pdf?session=…&<param>=<value>&scope=<level>`. HTTP redirects (301, 302, 307) are followed automatically. *(if `pdfFile.enabled`)*
8. **PDF saved** – The PDF is stored at `pdfFile.pdfDir/<pdfFileName>`. *(if `pdfFile.enabled`)*
9. **GDT response written** – The response file is created at `gdtWriter.gdtOutputDir/<gdtFileName>`. *(if `gdtWriter.enabled`)*
10. **HL7 MDM^T02 sent/saved** – The HL7 message is sent via MLLP/TCP *(if `hl7Sender.enabled`)* and/or saved as a file *(if `hl7Writer.enabled`)*. If sending fails and `hl7Sender.queue.enabled` is set, the message is written to the queue instead of aborting the processing.
11. **Cleanup** – The input file is moved to `done/`.

Concurrent Processing

Multiple GDT files in the input directory are processed concurrently (in parallel). Each file is tracked internally as a separate job; a file already being processed is skipped in the next poll cycle.

Error Handling

If an error occurs during processing (e.g. HTTP error when retrieving the PDF, missing required field, MLLP timeout, invalid GDT file), the following happens:

1. The error message is printed to the console.
2. A text file `<filename>.gdt.err` containing the error message is created in `error/`.
3. The input file is moved to `error/<filename>.gdt`.

Example after an HTTP 400 error

```
error/
 ├──── input-accnbr_20260701-141203.gdt
 └──── input-accnbr_20260701-141203.gdt.err   ← Content: "HTTP 400 von
https://portal.../api/reports/pdf?session=*** [STUDY_NOT_FOUND]: The requested
study was not found."
```

Example after an MLLP timeout

```
error/
 ├──── input_20260701-141203.gdt
 └──── input_20260701-141203.gdt.err   ← Content: "MLLP timeout after 10000ms
connecting to ris-server:2333"
```



The session token in the portal URL is automatically masked as * in all log output and error files.

HL7 Message Queue

When `hl7Sender.queue.enabled: true` is configured, HL7 messages are not lost on connection failure.

Situation	Behavior
Receiver unreachable (ECONNREFUSED, timeout)	Message is saved as a `.hl7` file in `queue.dir`. Processing is considered successful — the GDT file is moved to `done/`.
Daemon mode	Every `queue.retryIntervalMs` milliseconds (default: 60 s), the daemon attempts to drain the queue. On the first failure, it stops and retries at the next interval.
CLI mode	After processing, the queue is drained once immediately. The next invocation repeats this.
Receiver becomes reachable again	Queued messages are resent in chronological order and removed from the queue after a successful ACK.

```
queue/hl7/
 ├──── 1751111295123-a4f2b.hl7   ← pending message
 └──── 1751111301456-c9d1e.hl7   ← pending message
```

Logging

The application writes all messages to standard output (`stdout`). Additionally, the `log` module in `config.json` can enable a daily rotating log file — see Module: log.

The log file rotates daily and contains timestamps:

```
2026-06-28 13:48:15 [INFO] gdt-hl7-bridge started [--daemon]
2026-06-28 13:48:15 [INFO]   Input:    C:\RadioWeb\GDT\Input
2026-06-28 14:12:03 [INFO] Processing: input.gdt
2026-06-28 14:12:03 [INFO]   HL7:    send failed (connect ECONNREFUSED ...) —
queued as 1751111295123-a4f2b.hl7
2026-06-28 14:13:03 [INFO] Queue: 1 message(s) pending — retrying ris-server:2333
2026-06-28 14:13:03 [INFO] Queue: sent and removed 1751111295123-a4f2b.hl7
```



The `log` module handles rotation and retention automatically. Manual redirection of console output is no longer necessary.

Typical Console Output

Daemon mode – normal operation

```
gdt-hl7-bridge started [--daemon]
  Input:     C:\RadioWeb\GDT\Input
  GDT Out:   C:\RadioWeb\GDT\Output
  PDFs:      C:\RadioWeb\PDFs
  HL7 TCP:   ris-server:2333
  HL7 Queue: queue/hl7 (retry 60000ms)
  HL7 Out:   C:\RadioWeb\HL7\Output
  Interval:  2000 ms
Processing: input.gdt
  Lookup: patid=3209648408
  PDF:    C:\RadioWeb\PDFs\3209648408__1719475200000.pdf
  GDT:    C:\RadioWeb\GDT\Output\3209648408__1719475200000.gdt
  HL7:    sending to ris-server:2333
  HL7:    sent
  HL7:    C:\RadioWeb\HL7\Output\3209648408__1719475200000.hl7
  Done
```

Daemon mode – receiver unreachable, queue active

```
Processing: input.gdt
  Lookup: patid=3209648408
  HL7:    sending to ris-server:2333
  HL7:    send failed (connect ECONNREFUSED 10.0.0.1:2333) — queued as
1751111295123-a4f2b.hl7
  Done
Queue: 1 message(s) pending — retrying ris-server:2333
Queue: sent and removed 1751111295123-a4f2b.hl7
```

CLI mode

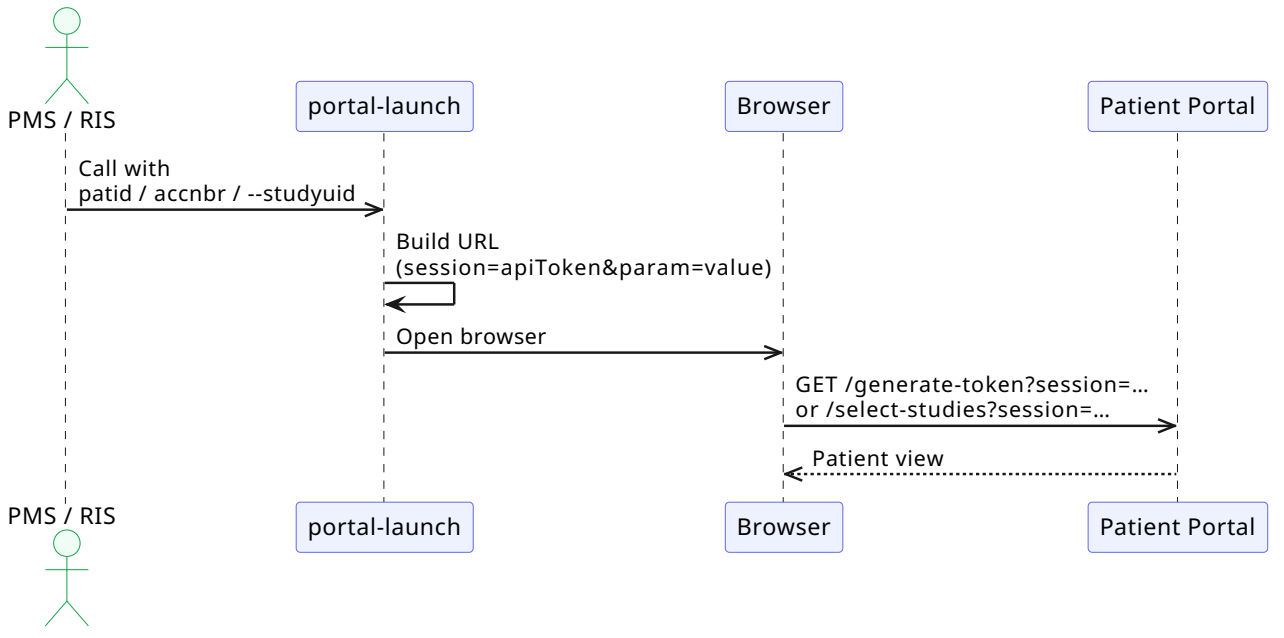
```
gdt-hl7-bridge [CLI]
  Lookup: studyuid=2.25.186760260011452398476108131496539831798
  PDF:    C:\RadioWeb\PDFs\3209648408__1719475200000.pdf
  HL7:    sending to ris-server:2333
  HL7:    sent
Done.
```

Portal-Launch

`portal-launch` is a Windows command-line application that opens the patient portal directly from practice management software (PMS), a RIS, or any other work application in the system's default browser (or a configured browser).

The application builds the correct portal URL including a session token and passes it to the browser — the user lands on the right page without a separate login step.

How It Works



Invocation

```
portal-launch.exe --patid 3209648408
portal-launch.exe --accnbr 186B7928
portal-launch.exe --studyuid 2.25.186760260011452398476108131496539831798
```

Depending on the parameter provided, the portal opens:

Parameter	Portal route	Behaviour
`--patid`	`/select-studies`	Shows all studies for the patient to choose from
`--accnbr`	`/generate-token`	Opens the study for the given accession number directly
`--studyuid`	`/generate-token`	Opens the study for the given Study UID directly

All Parameters

Parameter	Alias	Description
`--patid <id>`	`-p`	Patient ID (external ID)
`--accnbr <nbr>`	`-a`	Accession number (DICOM AccessionNumber)
`--studyuid <uid>`	`-s`	DICOM Study UID
`--config <path>`	`-c`	Path to `config.json` (default: `config\config.json` next to the `.exe`)

Parameter	Alias	Description
`--help`	`-h`	Display help text and exit

At least one of `--patid`, `--accnbr`, or `--studyuid` must be provided. The application exits with code `0` on success and `1` on error.

Prerequisites

- Windows 10 or later
- Network access to the patient portal (HTTPS, standard port 443)
- Customer API token — generated in the portal admin under *Settings → Integration*
- A browser installed (the system default browser is used unless `browser.executable` is configured)

Installation

Extract the ZIP package into a directory of your choice, e.g. `C:\RadioWeb-Launch\`. The resulting directory structure:

```
C:\RadioWeb-Launch\
 ├──── portal-launch.exe
 └──── config\
     └──── config.json.example
```

Creating the Configuration

1. Copy `config\config.json.example` to `config\config.json`.
2. Open `config\config.json` in a text editor.
3. Fill in at least the required fields:

```
{
  "portalUrl": "https://portal.your-practice.com",  ①
  "apiToken":  "XXXX-XXXX-XXXX"                     ②
}
```

① URL of the patient portal (no trailing slash)

② Integration token — generated by the portal administrator under *Settings → Integration*

Configuring the Browser (optional)

Without a `browser` configuration, the operating system's default browser is used. To open a specific browser at a defined window size:

```
{
```

```
  "portalUrl": "https://portal.your-practice.com",
  "apiToken":  "XXXX-XXXX-XXXX",

  "browser": {
    "executable": "C:\\Program Files\\Google\\Chrome\\Application\\chrome.exe",  ①
    "width":       1280,
②
    "height":      900
②
  }
}
```

① Absolute path to the browser executable

② Window size on open (only applied when `executable` is set)

Integration with PMS / RIS

`portal-launch.exe` can be called directly as an external program or macro from practice management software. Pass the patient ID or accession number as a parameter.

Example call from a RIS macro (Windows):

```
"C:\RadioWeb-Launch\portal-launch.exe" --accnbr %ACCESSION_NUMBER%
```

The exact syntax for calling external programs depends on your PMS/RIS software. Consult the documentation of your practice management or RIS application.

Configuration Reference

Field	Required	Default	Description
`portalUrl`	yes	–	URL of the patient portal. No trailing slash.
`apiToken`	no	–	Integration token. Passed as the `session` parameter in the URL. Without a token the user must log in manually.
`browser.executable`	no	–	Absolute path to a browser executable (e.g. Chrome, Edge). Without this field, `start` (system default browser) is used.
`browser.width`	no	–	Window width in pixels. Only applied when `browser.executable` is set.
`browser.height`	no	–	Window height in pixels. Only applied when `browser.executable` is set.

Error Handling

The application outputs error messages to stderr and exits with code 1.

Error message	Cause / Solution
`ERROR: At least one of --studyuid, --patid, or --accnbr is required.`	No lookup parameter provided. Pass at least one of the three parameters.
`ERROR: Config file not found: …`	`config\config.json` is missing. Create it from `config.json.example` and fill in the required fields.
`ERROR: config/config.json must contain "portalUrl".`	Required field `portalUrl` is missing from the configuration.
`ERROR: Failed to parse …`	`config.json` contains invalid JSON. Fix the syntax error.
`ERROR: Failed to open browser: …`	The browser could not be started. Check the `browser.executable` path or remove the field to use the system default browser.