

Für Systembetreuer: Anleitung für die Konfiguration von DICOM4charly

ab Version 9.18.0

Stand 09.02.2023



charly
by solutio

Impressum

solutio GmbH & Co. KG
Zahnärztliche Software und Praxismanagement

Max-Eyth-Straße 42
71088 Holzgerlingen
Fon 07031 4618-700
Fax 07031 4618-99700

info@solutio.de
www.solutio.de

© solutio GmbH & Co. KG 2023. Das Dokument „Für Systembetreuer: Anleitung für die Konfiguration von DICOM4charly“ ist urheberrechtlich geschützt. Die Nutzungsrechte liegen bei der solutio GmbH & Co. KG, insbesondere das Vervielfältigen oder Verbreiten des Dokuments „Für Systembetreuer: Anleitung für die Konfiguration von DICOM4charly“ im Ganzen oder in Teilen ist – soweit nicht durch das Urheberrecht zwingend erlaubt – untersagt.

Dokumentversion: 20230209.062227-DICOM4charly-9.18.0

Support

Sie erreichen unseren Support jeweils Montag bis Freitag von 7:30 bis 18:00 Uhr.

Fon 07031 4618-800
Fax 07031 4618-99800

support@solutio.de

Inhalt

1	Über diese Anleitung	4
1.1	Workflow aus Anwendersicht	4
2	Voraussetzungen	5
2.1	Software-Voraussetzungen	5
2.2	Erforderliche Ports für DICOM4charly	5
2.3	Erforderliche charly-Rechte	6
3	Konfiguration	7
3.1	[Optional] Soldicom-Service für mehrere Mandanten konfigurieren	7
3.2	[Optional] Ports ändern	7
3.3	In charly: Lizenz freischalten	12
3.4	Im DICOM-Röntgensystem: Schnittstelle zu DICOM4charly konfigurieren	13
3.5	In DICOM4charly: DICOM-Röntgengerät und Service erstellen	13
3.6	In charly: Röntgengerät mit DICOM-Röntgengerät verknüpfen	16
4	Server	17
4.1	charly-Java-Server	17
4.2	Konfigurationsdatei "application.yml"	17
4.3	Service-Registry	18
4.4	Kommandozeilenwerkzeug für Serversteuerung	19
4.5	Soldicom-Service	22
4.5.1	Ports für die Schnittstelle	22
4.5.2	Parameter des Soldicom-Services	22
4.5.3	Mandantenfähigkeit	24
5	Beispiel-Konfigurationen für die Datei „application.yml“ des Soldicom-Services	26

1 Über diese Anleitung

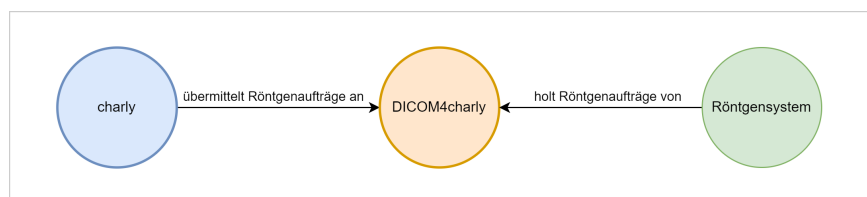
DICOM4charly ist die Schnittstelle zwischen charly und dem DICOM-Röntgengerät.

Hinweis: Diese Anleitung beschreibt die Konfiguration von DICOM4charly. Sie richtet sich an Systembetreuer mit grundlegenden Kenntnissen zu DICOM. Weitere Informationen zu DICOM4charly finden Sie in der integrierten Hilfe.

Hinweis: Alle notwendigen Informationen rund um den prinzipiellen Aufbau sowie die Verwaltung des Servers, haben wir in dem Kapitel [„Server“ auf Seite 17](#) für Sie zusammengefasst.

1.1 Workflow aus Anwendersicht

Das folgende Diagramm zeigt den grundsätzlichen Workflow mit DICOM4charly in stark vereinfachter Form aus Anwendersicht.



1. Ein Röntgenauftrag wird im charly-Röntgenkontrollbuch angelegt und an DICOM4charly übermittelt.
2. Ein Röntgensystem fragt DICOM4charly nach vorhandenen Röntgenaufträgen.
3. DICOM4charly übermittelt die zugehörigen Röntgenaufträge an das Röntgensystem.
4. Die Patienten- und Untersuchungsdaten stehen am Röntgengerät für die Durchführung der Röntgenuntersuchung bereit.
5. Nach Durchführung der Röntgenuntersuchung entfernt der Anwender den Röntgenauftrag aus DICOM4charly.

2 Voraussetzungen

Für die Verwendung von DICOM4charly müssen bestimmte Software- und Betriebsbedingungen erfüllt werden.

2.1 Software-Voraussetzungen

Für DICOM4charly benötigen Sie mindestens folgende Software-Versionen:

- charly ab Version 9.18.0
- PostgreSQL-Version ab 8.4

Tip: Die PostgreSQL-Version prüfen Sie, indem Sie in charly in der Menüleiste auf **Hilfe > SQL-Datenbank** klicken. Wenn die angegebene PostgreSQL-Version kleiner 8.4 ist, muss zunächst ein Datenbank-Upgrade durchgeführt werden. Wenden Sie sich dazu bitte an den Support der solutio GmbH & Co. KG.

2.2 Erforderliche Ports für DICOM4charly

Übersicht der Ports, die für die Kommunikation zwischen den Komponenten erforderlich sind:

Port	Beschreibung
10443	Standard für den SSL-Proxy. Wird vom charly-Updater in der Firewall geöffnet. Kann im charly-Updater umkonfiguriert werden. Führen Sie dazu den charly-Updater erneut aus.
10444 (TCP)	Standard für den DICOM4charly-Service für „MWL mit Auftragsdaten“. Über diesen Port kann ein Röntgensystem eine Anfrage an den Service stellen. Kann in der „application.yml“-Datei des Soldicom-Services geändert werden.

Port	Beschreibung
10445 (TCP)	Standard für den DICOM4charly-Service für „MWL für die Übertragung von Patientendaten“. Über diesen Port kann ein Röntgensystem eine Anfrage an den Service stellen. Kann in der „application.yml“-Datei des Soldicom-Services geändert werden.
14711 (HTTP)	Standard für den Messaging-Service. Wird vom charly-Updater in der Firewall geöffnet. Kann im charly-Updater umkonfiguriert werden. Führen Sie dazu den charly-Updater erneut aus.
14712 (STOMP)	Standard für den Messaging-Service. Kann in der „application.yml“-Datei des Messaging-Services geändert werden.

Für Änderungen an den Ports, siehe [„\[Optional\] Ports ändern“ auf Seite 7](#).

2.3 Erforderliche charly-Rechte

Für die Konfiguration von DICOM4charly benötigt Ihr charly-Benutzer mindestens folgende Zugriffsrechte in charly:

- Für den Zugriff auf das Lizenz-Fenster:
 - Einzelrecht Lizenz-Fenster: Lesen, Ändern, Neu
- Für den Zugriff auf den Karteireiter in den Stammdaten > Geräte > Geräte:
 - Einzelrecht Stammdaten: Lesen
 - Einzelrecht Stammdaten Geräte: Lesen, Ändern, Neu
- Für den Zugriff auf die Stammdaten in DICOM4charly:
 - Einzelrecht Patient: Lesen
 - Einzelrecht DICOM4charly STD Röntgengeräte: Lesen, Ändern, Neu, Löschen
 - Einzelrecht DICOM4charly STD Über diesen Dienst: Lesen

Die Zugriffsberechtigungen definieren Sie in charly in den Stammdaten > Praxis > Gruppen > Rechte.

3 Konfiguration

Um Röntgenaufträge an DICOM4charly übermitteln zu können, muss die Schnittstelle zunächst konfiguriert werden.

3.1 [Optional] Soldicom-Service für mehrere Mandanten konfigurieren

Für den Fall, dass in einer Praxis mehrere Mandanten mit DICOM4charly arbeiten möchten, deren Röntgengerät in seiner MWL-Anfrage **keine** Mandantenkennung liefert, muss die Konfiguration des Soldicom-Services angepasst werden.

Hinweis: Ausführliche Informationen zum Soldicom-Service sowie der Mandantenfähigkeit finden Sie hier: [„Soldicom-Service“ auf Seite 22](#)

3.2 [Optional] Ports ändern

Falls einer oder mehrere der [erforderlichen Ports für DICOM4charly](#) bereits belegt sind, konfigurieren Sie diese um:

- SSL-Proxy und Messaging-Service (HTTP) im charly-Updater
- Messaging-Service (STOMP) in der „application.yml“-Datei des Messaging-Services
- DICOM4charly-Services für „MWL mit Auftragsdaten“ und „MWL für die Übertragung von Patientendaten“ in der „application.yml“-Datei des Soldicom-Services

Hinweis: Kundenspezifische Konfigurationen müssen unter `ncjs` in dem Verzeichnis `conf2` abgelegt werden, da dieses Verzeichnis durch charly-Updates nicht überschrieben wird. Innerhalb des Verzeichnisses `conf2` muss für jeden Microservice, für den eine kundenspezifische Konfiguration hinterlegt wird, ein weiteres Verzeichnis mit dem Namen des Microservices angelegt werden. Dort wird die kundenspezifische „application.yml“-Datei

abgelegt.

Beispiel-Pfad einer kundenspezifischen „application.yml“-Datei mit Konfigurationen für den Soldicom-Service:

```
\Solutio\Server\ncjs\conf2\soldicom\application.yml
```

Ports für den SSL-Proxy und den Messaging-Service (HTTP) mit dem charly-Updater ändern

Hinweis: Sofern keine neuere charly-Version vorliegt, aktualisiert der charly-Updater ausschließlich den charly-Java-Server und nimmt dabei die Port-Änderungen vor. Falls eine neuere charly-Version verfügbar ist, aktualisiert der charly-Updater zusätzlich die charly-Version.

Voraussetzungen

- Der charly-Updater liegt auf dem charly-Server ([Download Windows](#) bzw. [Download macOS](#)).
- Der charly-Server verfügt über eine Internetverbindung.
- Das Ausführen des charly-Updaters ist vorbereitet:
 - Schalten Sie alle charly-Arbeitsplätze (Clients) aus.
 - Schließen Sie auf dem charly-Server alle Programme.
 - Führen Sie auf dem charly-Server eine Datensicherung durch.
 - Starten Sie den charly-Server neu.
 - Melden Sie sich mit Administratorrechten am charly-Server an.
 - Deaktivieren Sie auf dem charly-Server den Virenschanner für den Zeitraum des Updates.

Vorgehensweise

1. Starten Sie den charly-Updater.
2. Folgen Sie den Anweisungen des charly-Updater-Assistenten bis zu der Maske für die Konfiguration der Ports für den SSL-Proxy sowie für den Messaging-Service.
3. Ändern Sie den bzw. die Ports.
4. Folgen Sie weiter den Anweisungen des charly-Updater-Assistenten.

- > Der charly-Updater führt die Port-Änderungen im charly-Java-Server durch.

STOMP-Port im Messaging-Service ändern

Vorgehensweise

1. Öffnen Sie **auf dem Server** folgendes Verzeichnis:
`<charly-Installationspfad>\Solutio\Server\ncjs\`
2. Legen Sie in diesem Verzeichnis das Verzeichnis „conf2“ und darin das Verzeichnis „messaging“ an:
`<charly-Installationspfad>\Solutio\Server\ncjs\conf2\messaging\`
3. Legen Sie in dem Verzeichnis „messaging“ eine Textdatei an und benennen Sie diese mit „application.yml“.
4. Kopieren Sie die folgende Einstellung in die application.yml:
`de.solutio.ncjs.messaging.broker.stomp.port:14712`
5. Ändern Sie den Port.
6. Damit der Messaging-Service die fertig bearbeitete application.yml verwendet, muss der Messaging-Service gestoppt und anschließend wieder gestartet werden:

- **Windows**

- Öffnen Sie die Administrator-Kommandozeile und geben Sie folgende Befehle ein:

```
cd <charly-Installationspfad>\Solutio\Server\ncjs
```

- Stoppen Sie den Messaging-Service:

```
acd.bat stop messaging
```

- Starten Sie den Messaging-Service:

```
acd.bat start messaging
```

- **macOS**

- Öffnen Sie das Terminal und geben Sie folgende Befehle ein:

```
cd <charly-Installationspfad>/Solutio/Server/ncjs
```

- Stoppen Sie den Messaging-Service:

```
sudo ./acd.sh stop messaging
```

- Starten Sie den Messaging-Service:

```
sudo ./acd.sh start messaging
```

7. Prüfen Sie, ob der Messaging-Service gestartet ist:

1. Öffnen Sie ein Browserfenster.

2. Geben Sie folgende URL ein: <http://localhost:8087>

- > Das Eureka-Dashboard öffnet sich und zeigt alle registrierten und gestarteten Microservices des charly-Java-Servers. Nach ca. 1 – 2 Minuten sollte hier auch der Messaging-Service mit dem Status „UP“ gelistet sein.

Ports der DICOM4charly-Services im Soldicom-Service ändern

Hinweis: Die folgende Schrittanleitung beschreibt die Vorgehensweise in einem charly mit einem Mandanten. Sofern Sie in Ihrer Praxis einen charly mit mehreren Mandanten und einem nicht-mandantenfähigen Röntgensystem einsetzen, müssen Sie in Schritt 4 der Schrittanleitung die Standard-Konfiguration entsprechend um Services für den oder die Mandanten erweitern.

Für weitere Informationen siehe [„Mandantenfähigkeit“ auf Seite 24](#).

Vorgehensweise

1. Öffnen Sie **auf dem Server** folgendes Verzeichnis:

```
<charly-Installationspfad>\Solutio\Server\ncjs\
```

2. Legen Sie in diesem Verzeichnis das Verzeichnis „conf2“ und darin das Verzeichnis „soldicom“ an:

```
<charly-Installationspfad>\Solutio\Server\ncjs\
conf2\soldicom\
```

3. Legen Sie in dem Verzeichnis „soldicom“ eine Textdatei an und benennen Sie diese mit „application.yml“.
4. Kopieren Sie die Standard-Konfiguration in die application.yml. Siehe [„Standard-Konfiguration für einen Mandanten“ auf Seite 26.](#)
5. Bearbeiten Sie die application.yml: Ändern Sie in der kopierten Standard-Konfiguration den bzw. die Ports in freie Ports.
6. Damit der Soldicom-Service die fertig bearbeitete application.yml verwendet, muss der Soldicom-Service gestoppt und anschließend wieder gestartet werden:

- **Windows**

- Öffnen Sie die Administrator-Kommandozeile und geben Sie folgende Befehle ein:

```
cd <charly-Installationspfad>\Solutio\Server\ncjs
```

- Stoppen Sie den Soldicom-Service:

```
acd.bat stop soldicom
```

- Starten Sie den Soldicom-Service:

```
acd.bat start soldicom
```

- **macOS**

- Öffnen Sie das Terminal und geben Sie folgende Befehle ein:

```
cd <charly-Installationspfad>/Solutio/Server/ncjs
```

- Stoppen Sie den Soldicom-Service:

```
sudo ./acd.sh stop soldicom
```

- Starten Sie den Soldicom-Service:

```
sudo ./acd.sh start soldicom
```

7. Prüfen Sie, ob der Soldicom-Service gestartet ist:
 - a. Öffnen Sie ein Browserfenster.
 - b. Geben Sie folgende URL ein: `http://localhost:8087`
 - > Das Eureka-Dashboard öffnet sich und zeigt alle registrierten und gestarteten Microservices des charly-Java-Servers. Nach ca. 1 – 2 Minuten sollte hier auch der Soldicom-Service mit dem Status „UP“ gelistet sein.

3.3 In charly: Lizenz freischalten

Um DICOM4charly öffnen zu können, muss zunächst die Lizenz freigeschaltet werden.

DICOM4charly-Lizenz in charly freischalten

Voraussetzungen

- Sie haben die DICOM4charly-Lizenz der solutio GmbH & Co. KG erworben.

Vorgehensweise

1. Klicken Sie in charly in der Menüleiste auf Hilfe > Zusätzliche Lizenzen.
2. Aktivieren Sie in dem Fenster die Checkbox DICOM4charly und geben Sie in dem Feld daneben die Lizenz für die DICOM4charly-Schnittstelle ein.
3. Speichern Sie mit OK.
4. Starten Sie charly neu.
 - > Nach dem Neustart steht Ihnen in der Menüleiste unter Start die Option DICOM4charly zur Verfügung. Über diese Option rufen Sie DICOM4charly auf.

3.4 Im DICOM-Röntgensystem: Schnittstelle zu DICOM4charly konfigurieren

Hinweis: Für die Konfiguration der Schnittstelle zu DICOM4charly in Ihrem DICOM-Röntgensystem, lesen Sie bitte die Dokumentation des Röntgensystem-Herstellers.

Alle notwendigen Informationen für die Anbindung der DICOM4charly-Schnittstelle an das Röntgensystem, finden Sie in DICOM4charly > Stammdaten > Über diesen Dienst.

DICOM4charly bietet für Röntgensysteme folgende Services in der Rolle „Service Class Provider (SCP)“ an:

- MWL mit Auftragsdaten
- MWL für die Übertragung von Patientendaten

Der Hostname – die IP-Adresse des Servers in der Praxis – wird automatisch ermittelt und für beide Services gesetzt.

Die Kommunikation vom Röntgengerät zu DICOM4charly läuft standardmäßig über die Ports 10444 und 10445 (TCP). Sofern einer oder beide Ports in der Praxis bereits belegt sind, müssen die Ports in der „application.yml“-Datei des Soldicom-Services geändert werden. Siehe [„Ports der DICOM4charly-Services im Soldicom-Service ändern“ auf Seite 10](#).

3.5 In DICOM4charly: DICOM-Röntgengerät und Service erstellen

Die DICOM-Röntgengeräte und Services erstellen Sie in DICOM4charly > Stammdaten > Röntgengeräte. Dabei gelten folgende Regeln:

- Es muss mindestens ein DICOM-Röntgengerät angelegt werden. Darüber hinaus können beliebig viele weitere DICOM-Röntgengeräte angelegt werden.
- Pro DICOM-Röntgengerät muss mindestens ein Service angelegt werden. Darüber hinaus können pro DICOM-Röntgengerät beliebig viele weitere Services angelegt werden.

DICOM-Röntgengerät erstellen

Voraussetzungen

- Sie benötigen die Netzwerkadresse, unter der das DICOM-Röntgengerät im Netzwerk erreichbar ist.

Vorgehensweise

1. Klicken Sie in charly in der Menüleiste auf **Start > DICOM4charly**.
 - > DICOM4charly öffnet sich.
2. Klicken Sie in DICOM4charly auf **Stammdaten > Röntgengeräte**.
3. Um ein DICOM-Röntgengerät anzulegen, klicken Sie rechts neben dem Suchfeld auf den **Plus-Button +**.
 - > Die Ansicht zum Anlegen von DICOM-Röntgengeräten öffnet sich.
4. Geben Sie die Daten Ihres DICOM-Röntgengeräts ein:

Feld	Beschreibung
Hersteller	Name des Röntgengeräteherstellers
Gerätename	Produktname des Röntgengeräts
Zimmer	Standort des Röntgengeräts
Netzwerkadresse	Netzwerkadresse des Röntgengeräts

5. Klicken Sie auf **Speichern**.
 - > Das DICOM-Röntgengerät ist angelegt und wird in der Liste der Röntgengeräte angezeigt.

Nächste Schritte



- Um die Konfiguration abzuschließen, erstellen Sie für das DICOM-Röntgengerät mindestens einen Service.

Service für ein DICOM-Röntgengerät erstellen

Voraussetzungen

- Das DICOM-Röntgengerät, für das Sie einen Service erstellen möchten, ist erstellt. Für weitere Informationen siehe [DICOM-Röntgengerät erstellen](#).

Vorgehensweise

1. Klicken Sie in DICOM4charly auf Stammdaten > Röntgengeräte.
2. Klicken Sie in der Liste bei dem DICOM-Röntgengerät, für das Sie einen Service erstellen möchten, auf den Anzeigen-Button .
 - > Die Seite Röntgengerät anzeigen öffnet sich.
3. Um einen Service anzulegen, klicken Sie rechts neben dem Suchfeld auf den Plus-Button .
 - > Die Ansicht zum Anlegen von Services öffnet sich.
4. Geben Sie die Daten des Services ein:

Feld	Beschreibung
AE-Titel	(Application Entity Title) – Name des Services, über den der Service eindeutig identifizierbar ist
Röntgenverfahren	Röntgenverfahren (Modalität), das der Service bietet, z.B. „PX“ für „Panoramic X-Ray“
Port	Port, über den die DICOM-Kommunikation läuft
Rolle	Rolle des Services.

Hinweis: Für den Abruf der Röntgenaufträge hat der Service des Röntgengeräts immer die Rolle SCU. SCU (Service Class User) ist ein Gerät bzw. eine Applikation, die einen Dienst in Anspruch nimmt.

5. Klicken Sie auf Speichern.
 - > Der Service ist für das DICOM-Röntgengerät angelegt und wird in der Liste der Services angezeigt.

3.6 In charly: Röntgengerät mit DICOM-Röntgengerät verknüpfen

Das DICOM-Röntgengerät, das Sie in DICOM4charly angelegt haben, verknüpfen Sie mit dem korrespondierenden Röntgengerät aus charly. Diese Verknüpfung ist notwendig, damit aus charly über die DICOM4charly-Schnittstelle ein Röntgenauftrag an ein DICOM-Röntgengerät übermittelt werden kann.

In charly ein Röntgengerät mit einem DICOM-Röntgengerät verknüpfen

Vorgehensweise

1. Gehen Sie in charly unter Stammdaten > Geräte > Geräte.
2. Wählen Sie aus der Geräteliste das Röntgengerät, das Sie mit dem DICOM-Röntgengerät verknüpfen möchten.
 - > Der Reiter DICOM wird eingeblendet.
3. Gehen Sie in den Reiter DICOM.
 - > In der Dropdownliste Röntgengerät sehen Sie alle DICOM-Röntgengeräte, die Sie in DICOM4charly angelegt haben und für die Sie jeweils mindestens einen Service erstellt haben.
4. Wählen Sie aus der Dropdownliste Röntgengerät, das DICOM-Röntgengerät, das Sie mit dem Röntgengerät aus charly verknüpfen möchten.
5. Um die Einstellungen zu speichern, klicken Sie auf OK.
 - > Wenn Sie im Röntgenbuch bei der Neuanlage eines Röntgenauftrags dieses Röntgengerät wählen, wird der Röntgenauftrag an DICOM4charly übermittelt.

4 Server

4.1 charly-Java-Server

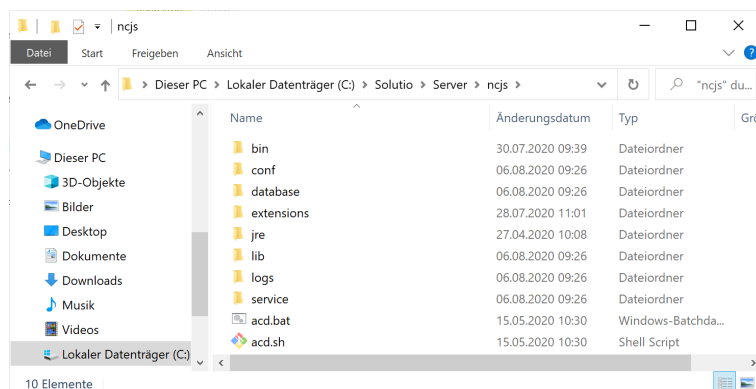
Der charly-Java-Server (ncjs) bezeichnet die Summe aller Microservices und die für deren Betrieb erforderliche Infrastruktur, welche auf dem Server der Zahnarztpraxis betrieben wird.

Hinweis: Der charly-Java-Server ist automatisch in charly enthalten und wird im Rahmen der charly-Updates laufend aktualisiert.

Installationspfad charly-Java-Server (ncjs)

Alle Komponenten des charly-Java-Servers befinden sich auf dem charly-Server unter folgendem Pfad:

<charly-Installationspfad>\Solutio\Server\ncjs



4.2 Konfigurationsdatei "application.yml"

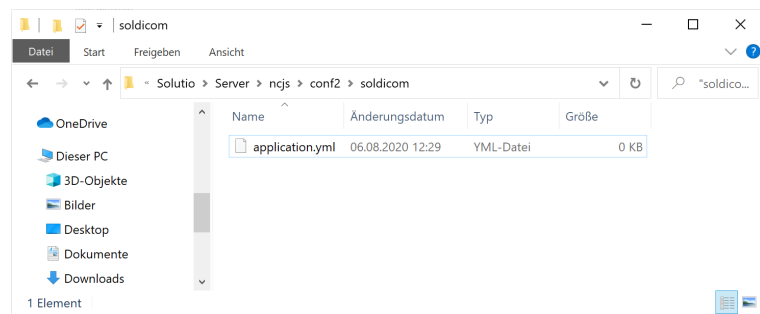
Microservices werden mit Hilfe der Datei `application.yml` konfiguriert. Für jeden Microservice gibt es eine eigene „application.yml“-Datei.

Kundenspezifische Konfigurationen müssen unter `ncjs` in dem Verzeichnis `conf2` abgelegt werden, da dieses Verzeichnis durch charly-Updates nicht überschrieben wird. Innerhalb des Verzeichnisses `conf2` muss für jeden Microservice, für den eine kundenspezifische Konfiguration hinterlegt wird, ein weiteres

Verzeichnis mit dem Namen des Microservices angelegt werden. Dort wird die kundenspezifische „application.yml“-Datei abgelegt.

Beispiel: Pfad einer kundenspezifischen „application.yml“-Datei mit Konfigurationen für den Soldicom-Service:

`\Solutio\Server\ncjs\conf2\soldicom\application.yml`



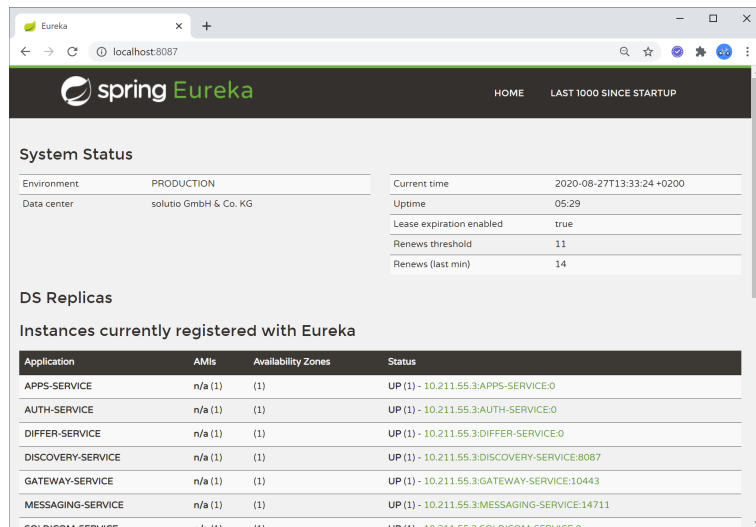
4.3 Service-Registry

Die Service-Registry fasst die Microservices, ihre Instanzen und ihre Lokationen in einer Datenstruktur zusammen. Die Microservices werden unter einem logischen Namen registriert und lassen sich anschließend über diesen Namen ansprechen.

Eine Liste aller registrierter und gestarteter Microservices ist auf der Weboberfläche einsehbar. Diese können Sie wie folgt aufrufen:

1. Öffnen Sie **auf dem Server** ein Browserfenster.
2. Geben Sie folgende URL ein: `http://localhost:8087`

Hinweis: Falls Sie den Port für den Discovery-Service über den charly-Updater geändert haben, geben Sie stattdessen diesen geänderten Port in der URL an.



The screenshot shows the Spring Eureka web interface. The top navigation bar includes the 'spring Eureka' logo and links for 'HOME' and 'LAST 1000 SINCE STARTUP'. The main content is divided into two sections: 'System Status' and 'DS Replicas'.

System Status

Environment	PRODUCTION	Current time	2020-08-27T13:33:24+0200
Data center	solutio GmbH & Co. KG	Uptime	05:29
		Lease expiration enabled	true
		Renews threshold	11
		Renews (last min)	14

DS Replicas

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
APPS-SERVICE	n/a (1)	(1)	UP (1) - 10.211.55.3:APPS-SERVICE-0
AUTH-SERVICE	n/a (1)	(1)	UP (1) - 10.211.55.3:AUTH-SERVICE-0
DIFFER-SERVICE	n/a (1)	(1)	UP (1) - 10.211.55.3:DIFFER-SERVICE-0
DISCOVERY-SERVICE	n/a (1)	(1)	UP (1) - 10.211.55.3:DISCOVERY-SERVICE-8087
GATEWAY-SERVICE	n/a (1)	(1)	UP (1) - 10.211.55.3:GATEWAY-SERVICE:10443
MESSAGING-SERVICE	n/a (1)	(1)	UP (1) - 10.211.55.3:MESSAGING-SERVICE:14711
SOLIDCOM-SERVICE	n/a (1)	(1)	UP (1) - 10.211.55.3:SOLIDCOM-SERVICE-0

4.4 Kommandozeilenwerkzeug für Serversteuerung

Alle Microservices werden im Rahmen eines charly-Updates durch den charly-Updater registriert und gestartet. Für den Fall, dass ein Microservice z.B. für eine Konfigurationsänderung manuell gestoppt und wieder gestartet werden muss, gibt es das Tool ACD.

Das ACD-Tool besteht aus jeweils einer Skriptdatei für Windows (`acd.bat`) und macOS (`acd.sh`). Die Skriptdateien sind grundsätzlich in Funktion und Oberfläche identisch. Das Skript muss über die Kommandozeile **im Administratormodus** ausgeführt werden. Dazu muss in der Kommandozeile der Pfad zu dem Verzeichnis geöffnet sein, in dem die Skriptdateien (`acd.bat` bzw. `acd.sh`) liegen.

Die Skriptdateien befinden sich unter: `\Solutio\Server\ncjs` .

Aufruf

Der Aufruf beginnt immer mit der Angabe des Skripts gefolgt von dem eigentlichen Befehl. Die Syntax lautet wie folgt:

- **Windows**

```
acd.bat <Befehl> <ggf. Short Name Microservice>
```

- **macOS**

```
sudo ./acd.sh <Befehl> <ggf. Short Name Microservice>
```

Es gibt Befehle, die für alle bekannten Microservices gleichzeitig durchgeführt werden und Befehle, mit denen Sie nur einen bestimmten Microservice ansprechen. Für diese „Einzelbefehle“ müssen Sie den Short Name des Microservices angeben. Die Short Names der Microservices finden Sie heraus, indem Sie in der Kommandozeile folgenden Befehl eingeben:

- **Windows**

```
acd.bat list
```

- **macOS**

```
sudo ./acd.sh list
```

Als Ergebnis erhalten Sie eine Liste aller bekannten Microservices.

Beispiel: Im Folgenden ein Beispiel für den Auth-Microservice. Der Short Name ist die Angabe hinter **Name**. In diesem Fall **auth**:

```
Name: auth
Memory: 32m/256m
Filename: auth-service-app-1.3.0-SNAPSHOT.jar
Full name: NCJS Auth
State: RUNNING
```

Mit dem Short Name können Sie nun einen „Einzelbefehl“ für den Auth-Microservice absetzen.

Beispiel: Der Microservice mit dem Short Name **auth** soll über ACD gestoppt werden.

- **Windows**

```
acd.bat stop auth
```

- **macOS**

```
sudo ./acd.sh stop auth
```

Befehle

Befehl	Beschreibung
<code>list</code>	Listet alle bekannten Microservices mit folgenden Informationen: Name (= Short Name), Speicher, Dateiname, voller Name und Status.
<code>register</code>	Registriert einen Microservice als Systemdienst. Der Name (= Short Name) des Microservices muss angegeben werden.
<code>registerall</code>	Funktioniert wie der Befehl <code>register</code> , wird jedoch für alle bekannten Microservices ausgeführt.
<code>start</code>	Startet einen Microservice. Der Name (= Short Name) des Microservices muss angegeben werden. Um den Start-Befehl erfolgreich auszuführen, muss der Microservice bereits registriert sein.
<code>startall</code>	Funktioniert wie der Befehl <code>start</code> , wird jedoch für alle bekannten Microservices ausgeführt. Um den Start-Befehl erfolgreich auszuführen, müssen die Microservices bereits registriert sein.
<code>stop</code>	Stoppt einen Microservice als Systemdienst. Der Name (= Short Name) des Microservices muss angegeben werden.
<code>stopall</code>	Funktioniert wie der Befehl <code>stop</code> , wird jedoch für alle bekannten Microservices ausgeführt.

Befehl	Beschreibung
unregister	Meldet einen Microservice als Systemdienst ab. Der Name (= Short Name) des Microservices muss angegeben werden. Falls der Microservice noch läuft, führt diesen Befehl vor dem <code>unregister</code> ein <code>stop</code> durch.
unregisterall	Funktioniert wie der Befehl <code>unregister</code> , wird jedoch für alle bekannten Microservices ausgeführt. Für Microservices, die noch laufen, führt diesen Befehl vor dem <code>unregister</code> ein <code>stop</code> durch.

4.5 Soldicom-Service

Der Soldicom-Service bietet eine DICOM-Schnittstelle, welche die Übertragung von Patienten- und Auftragsdaten zwischen charly und DICOM-Röntgensystemen ermöglicht.

4.5.1 Ports für die Schnittstelle

Die Kommunikation vom Röntgengerät zu DICOM4charly läuft standardmäßig über die Ports 10444 und 10445 (TCP). Sofern einer oder beide Ports in der Praxis bereits belegt sind, müssen sie in der „application.yml“-Datei des Soldicom-Services geändert werden. Siehe [„Ports der DICOM4charly-Services im Soldicom-Service ändern“ auf Seite 10](#).

4.5.2 Parameter des Soldicom-Services

Für die Konfiguration des Soldicom-Services müssen in der der „application.yml“-Datei vorgegebene Parameter angegeben sein.

Parameter	Beschreibung
u-id	ist die u-id (Unique ID) der Schnittstelle – muss aufsteigend angegeben werden.

Parameter	Beschreibung
type	gibt den Schnittstellen-Type an. DICOM4charly bietet für beide Schnittstellen den Type „MWL_SCP“.
name	ist der Name der Schnittstelle – kann frei gewählt werden und dient dem Verständnis
ae-title	ist der AE-Title (Application Entity Title) der Schnittstelle – die maximale Länge von 16 Zeichen darf nicht überschritten werden. Hinweis: Nach einer Änderung des AE-Title im Soldicom-Service, muss diese Änderung auch in der Konfiguration des Röntgengerätes vorgenommen werden.
port	TCP-Port der Schnittstelle.
option-system-start	gibt an, ob die Option „System Start“ von der Schnittstelle unterstützt wird (1) oder nicht (0).
option-post-processing-pass-through	gibt an, ob die Option „Post Processing Pass-Through“ von der Schnittstelle unterstützt wird (1) oder nicht (0).
option-multi-mandant	gibt an, ob die Option „Multi-Mandant“ von der Schnittstelle unterstützt wird (1) oder nicht (0).
option-storage-commitment	gibt an, ob die Option „Storage Commitments“ von der Schnittstelle unterstützt wird (1) oder nicht (0).
only-patient-data	gibt an, ob ein MWL-SCP Patientendaten oder Auftragsdaten bereitstellt; 1 für reine Patientendaten oder 0 für vollständige Auftragsdaten.
active	gibt an, ob die Schnittstelle aktiv ist (true) oder nicht (false).
defaultTenant	gibt die Nummer des Mandanten an, von welchem die Schnittstelle die Patienten- und Behandlerdaten holt.

Ein Beispiel für die Standard-Konfiguration finden Sie [hier](#).

4.5.3 Mandantenfähigkeit

DICOM4charly ist mandantenfähig und standardmäßig für einen Mandanten (Mandant 1) vorkonfiguriert.

Hinweis: Das BDW-Profil sieht vor, dass ein Röntgengerät in seiner MWL-Anfrage angibt, für welchen Mandanten die Anfrage gilt. Nicht alle Röntgengerätehersteller haben die Mandantenfähigkeit umgesetzt.

Damit DICOM4charly die MWL-Anfrage eines Röntgengeräts in jedem Fall bearbeiten kann, unterscheidet DICOM4charly zwischen den folgenden Fällen:

- Wenn ein Röntgengerät eine MWL-Anfrage sendet, die eine Mandantenkennung enthält, werden die Daten zu dieser Mandantenkennung geliefert.
- Wenn ein Röntgengerät eine MWL-Anfrage sendet, die **keine** Mandantenkennung enthält, werden die Daten des Mandanten aus der Konfiguration im Soldicom-Service geliefert.

Falls die Praxis über mehrere Mandanten verfügt, muss die Konfiguration des Soldicom-Services angepasst werden.

Hinweis: Kundenspezifische Konfigurationen müssen unter `ncjs` in dem Verzeichnis `conf2` abgelegt werden, da dieses Verzeichnis durch charly-Updates nicht überschrieben wird. Innerhalb des Verzeichnisses `conf2` muss für jeden Microservice, für den eine kundenspezifische Konfiguration hinterlegt wird, ein weiteres Verzeichnis mit dem Namen des Microservices angelegt werden. Dort wird die kundenspezifische „application.yml“-Datei abgelegt.

Beispiel-Pfad einer kundenspezifischen „application.yml“-Datei mit Konfigurationen für den Soldicom-Service:

```
\Solutio\Server\ncjs\conf2\soldicom\application.yml
```

Konfiguration in einer Praxis mit mehreren Mandanten und einem nicht-mandantenfähigen Röntgengerät

Für den Fall, dass in einer Praxis mehrere Mandanten mit DICOM4charly arbeiten möchten, deren Röntgengerät in seiner MWL-Anfrage **keine** Mandantenkennung liefert, muss die Konfiguration des Soldicom-Services angepasst werden.

Hinweis: Durch die Konfiguration für jeden Mandanten entstehen zwei eigene Schnittstellen mit individuellen Ports, die dann im Röntgengerät für diesen Mandanten aufgerufen werden können.

Dabei gelten folgende **Regeln:**

- Pro Mandant zwei Blöcke:
 - Ein Block für die Schnittstelle, welche die MWL mit Auftragsdaten liefert
 - Ein Block für die Schnittstelle, welche die MWL für die Übertragung von Patientendaten liefert
- Pro Block muss die `u-id` hochgezählt werden
- Jeder Block benötigt einen unique `port`
- Jeder Block benötigt einen unique `ae-title` – die maximale Länge von 16 Zeichen darf nicht überschritten werden.

Hinweis: Nach einer Änderung des AE-Title im Soldicom-Service, muss diese Änderung auch in der Konfiguration des Röntgengerätes vorgenommen werden.

- Der `defaultTenant` entscheidet, von welchem Mandanten die Daten geholt werden

Eine Beispiel-Konfiguration für zwei Mandanten finden Sie [hier](#).

5 Beispiel-Konfigurationen für die Datei „application.yml“ des Soldicom-Services

Standard-Konfiguration für einen Mandanten

```
de:
  solutio:
    ncjs:
      soldicom:
        dicom-api:
          configuration:
            services:
              -
                u-id: 1
                type: MWL_SCP
                name: 'charly DICOM-Schnittstelle für MWL'
                ae-title: DICOM4CHARLY_MWL
                port: 10444
                option-system-start: 0
                option-post-processing-pass-through: 0
                option-multi-mandant: 1
                option-storage-commitment: 0
                only-patient-data: 0
                active: true
                defaultTenant: 1
              -
                u-id: 2
                type: MWL_SCP
                name: 'charly DICOM-Schnittstelle für die Übertragung von Patientendaten'
                ae-title: DICOM4CHARLY_PDE
                port: 10445
                option-system-start: 0
                option-post-processing-pass-through: 0
                option-multi-mandant: 1
                option-storage-commitment: 0
                only-patient-data: 1
                active: true
                defaultTenant: 1
```

Beispiel-Konfiguration für zwei Mandanten

```
de:
  solutio:
    ncjs:
      soldicom:
        dicom-api:
          configuration:
            services:
              -
                u-id: 1
                type: MWL_SCP
                name: 'DICOM-Röntgenaufträge Praxis Dr. Molar'
                ae-title: DICOM4CHARLY_MWL
                port: 10444
                option-system-start: 0
                option-post-processing-pass-through: 0
                option-multi-mandant: 1
                option-storage-commitment: 0
                only-patient-data: 0
                active: true
                defaultTenant: 1
              -
                u-id: 2
                type: MWL_SCP
                name: 'DICOM-Patientendaten Praxis Dr. Molar'
                ae-title: DICOM4CHARLY_PDE
                port: 10445
                option-system-start: 0
                option-post-processing-pass-through: 0
                option-multi-mandant: 1
                option-storage-commitment: 0
                only-patient-data: 1
                active: true
                defaultTenant: 1
              -
                u-id: 3
                type: MWL_SCP
                name: 'DICOM-Röntgenaufträge Praxis Dr. Zahn'
                ae-title: DICOM_TWO_MWL
                port: 10446
                option-system-start: 0
```

```
option-post-processing-pass-through: 0
option-multi-mandant: 1
option-storage-commitment: 0
only-patient-data: 0
active: true
defaultTenant: 2
-
u-id: 4
type: MWL_SCP
name: 'DICOM-Patientendaten Praxis Dr. Zahn'
ae-title: DICOM_TWO_PDE
port: 10447
option-system-start: 0
option-post-processing-pass-through: 0
option-multi-mandant: 1
option-storage-commitment: 0
only-patient-data: 1
active: true
defaultTenant: 2
```