Informationen zur DEMIS Docker Testumgebung

⚠ Die DEMIS Docker Testumgebung wurde abgekündigt. Bitte nutzen Sie die DEMIS Live-Testumgebung!

DEMIS wird als lauffähige Docker Testumgebung ausgeliefert. Diese ist vorkonfiguriert und wird mit passend konfiguriertem DEMIS-Adapter und DEMIS-Importer bereitgestellt.

Inhalt

System Requirements

Die notwendigen Mindestanforderungen an Ihr Testsystem finden Sie auf den den entsprechenden Support-Seiten des Herstellers (Windows oder Mac).

Vorbereitungen

Zur Ausführung der Docker Testumgebung benötigen Sie Docker. Sie können für Windows Docker Desktop nutzen (https://docs.docker.com/docker-for-windows/install/). Weitere Informationen entnehmen Sie bitte der Dokumentation unter https://docs.docker.com/get-docker/.
Sie müssen sich nicht mit einer Docker ID anmelden!

Die benötigte demis.yml Datei finden Sie hier:
https://github.com/gematik/DEMIS-Test_Environment

Den vorkonfigurierten DEMIS-Adapter und Importer finden Sie unter https://nexus.prod.ccs.gematik.solutions/service/rest/repository/browse/DEMIS/

Ausführung

Die Docker Compose Datei (demis.yml) lädt die Docker Container von Dockerhub herunter und setzt die einzelnen Services zu dem DEMIS System zusammen, sodass die lokale Ausführung des DEMIS System für Anwender einfach zu handhaben ist.

Auf der Kommandozeile in das Verzeichnis wechseln und folgendes Kommando fährt die Docker DEMIS Umgebung hoch:

```
docker-compose --file .\demis.yml up --detach
```

*) In V2 von Compose ist Compose direkt in docker integriert. Daher ist "docker compose" statt "docker-compose" zu verwenden.

Bei der ersten Ausführung werden die Images, die die einzelnen Komponenten darstellen, einmalig heruntergeladen und lokal bei Ihnen gespeichert.

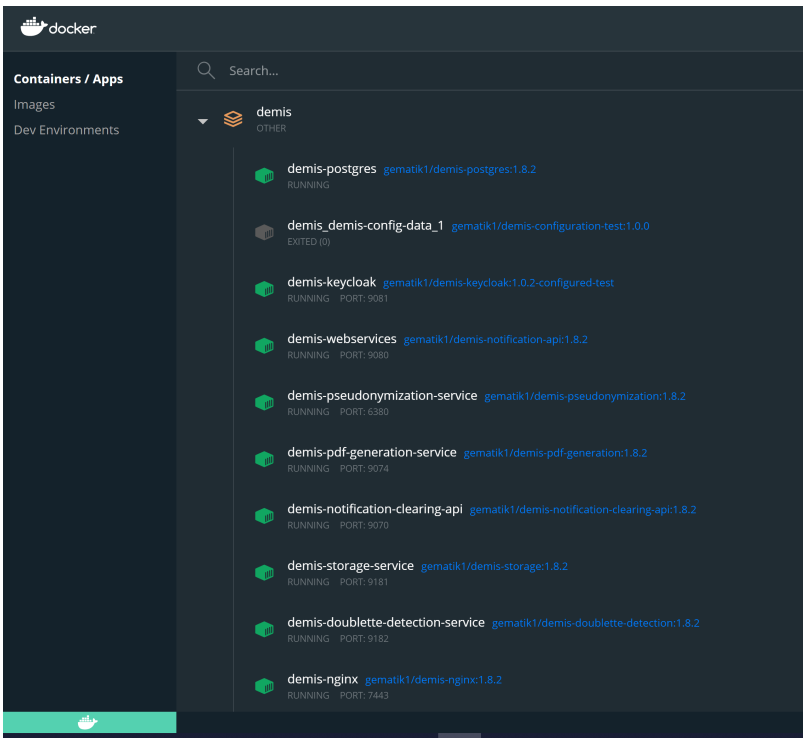
Die Docker Testumgebung ist vollständig hochgefahren, wenn Sie hinter jedem Service ein "done" sehen und die Kommandozeile wieder für eine Eingabe frei ist:

```
PS D:\>  docker-compose --file .\demis.yml up --detach
Docker Compose is now in the Docker CLI, try `docker compose up`

Creating network "default_default" with the default driver
Creating network "demis-network" with the default driver
Creating volume "default_demis-config-volume" with default driver
Creating demis-postgres                ... done
Creating demis-postgres-ui             ... done
Creating default_demis-config-data_1 ... done
Creating demis-keycloak                ... done
Creating demis-pdf-generation-service      ... done
Creating demis-storage-service             ... done
Creating demis-doublette-detection-service ... done
Creating demis-webservices                 ... done
Creating demis-pseudonymization-service    ... done
Creating demis-notification-clearing-api   ... done
Creating demis-nginx                       ... done
PS D:\>
```

Bitte haben Sie etwas Geduld, bevor Sie Meldungen an das DEMIS System senden. Die Umgebung benötigt Zeit zum Hochfahren.
Falls Sie mit dem Adapter die Meldung "Status Error while sending to Notification API response Header HTTP 502 Bad Gateway" erhalten, ist die Umgebung noch nicht vollständig hochgefahren!

In Docker sieht die DEMIS Testumgebung so aus:



Zum Stoppen der Umgebung bitte folgenden Befehl ausführen:

```
docker-compose --file .\demis.yml down -v
```

*)

```
PS D:\> docker-compose --file .\demis.yml down -v
Stopping demis-nginx                        ... done
Stopping demis-pseudonymization-service     ... done
Stopping demis-notification-clearing-api    ... done
Stopping demis-webservices                  ... done
Stopping demis-storage-service              ... done
Stopping demis-pdf-generation-service       ... done
Stopping demis-doublette-detection-service  ... done
Stopping demis-postgres-ui                  ... done
Stopping demis-keycloak                     ... done
Stopping demis-postgres                     ... done
Removing demis-nginx                        ... done
Removing demis-pseudonymization-service     ... done
Removing demis-notification-clearing-api    ... done
Removing demis-webservices                  ... done
Removing demis-storage-service              ... done
Removing demis-pdf-generation-service       ... done
Removing demis-doublette-detection-service  ... done
Removing demis-postgres-ui                  ... done
Removing default_demis-config-data_1        ... done
Removing demis-keycloak                     ... done
Removing demis-postgres                     ... done
Removing network default_default
Removing network demis-network
Removing volume default_demis-config-volume
PS D:\>
```

Fachlicher Aufbau der Docker Testumgebung

Die Docker Testumgebung unterschiedet sich fachlich von der Produktivumgebung. Hierzu wurden statt 375 Gesundheitsämtern, insgesamt zehnTest-Gesundheitsämter angelegt, die jeweils alle gültigen Postleitzahlen unterstützen.

Test-Gesundheitsämter

Zur Erleichterung der Abbildung aller Gesundheitsämter und gültigen Postleitzahlen in Deutschland, stehen für die Testumgebung zehn Test-Gesundheitsämter zur Verfügung, die alle gültigen Postleitzahlen abdecken.

Test-Gesundheitsamt Name	Code	Postleitzahlenkreis
Test-Gesundheitsamt 01	1.test-oegd01	1xxxx 10115 - 19417
Test-Gesundheitsamt 02	1.test-oegd02	2xxxx 20095 - 29699
Test-Gesundheitsamt 03	1.test-oegd03	3xxxx 30159 - 39649 (außer Bielefeld)
Test-Gesundheitsamt 04	1.test-oegd04	4xxxx 40210 - 49849
Test-Gesundheitsamt 05	1.test-oegd05	5xxxx 50126 - 59969
Test-Gesundheitsamt 06	1.test-oegd06	6xxxx 60308 - 69518
Test-Gesundheitsamt 07	1.test-oegd07	7xxxx 70173 - 79879
Test-Gesundheitsamt 08	1.test-oegd08	8xxxx 80331 - 89619

Test-Gesundheitsamt 09	1.test-oegd09	9xxxx 90402 - 99998
Test-Gesundheitsamt 10	1.test-oegd10	0xxxx 01067 - 09669
Test-Gesundheitsamt 11 besitzt kein Zertifikat und produziert den Fehler: "*500: Recipient certificate could not be loaded. Target office (Gesundheitsamt) probably not registered yet*."		Bielefeld: 33602 33604 33605 33607 33609 33611 33613 33615 33617 33619 33647 33649 33659 33689 33699 33719 33729 33739

Endpunkte, Zertifikate, User und Passwort

Endpunkte

Service	Endpunkte	Client ID, Passwort
idp.lab.tokenendpoint für Labore	https://localhost:7443/auth/realms/LAB/protocol/openid-connect/token	demis-adapter, secret_client_secret
fhir.basepath für Labore	https://localhost:7443/notification-api/fhir/	demis-adapter, secret_client_secret
idp.tokenendpoint für Gesundheitsämter	https://localhost:7443/auth/realms/OEGD/protocol/openid-connect/token	demis-importer, secret_client_secret
fhir.basepath für Gesundheitsämter	https://localhost:7443/notification-clearing-api/fhir/	demis-importer, secret_client_secret
idp.tokenendpoint für Krankenhäuser	https://localhost:7443/auth/realms/HOSPITAL/protocol/openid-connect/token	demis-test, secret_client_secret
fhir.basepath für Krankenhäuser (Hospitalisierungsmeldung)	https://localhost:7443/hospitalization/fhir/	demis-test, secret_client_secret
fhir.basepath für Krankenhäuser (Bettenbelegungsmeldung)	https://localhost:7443/reports/fhir/$process-report	demis-test, secret_client_secret

Die Endpunktadressen ändern sich entsprechend der Server URL und Portangabe bei einer eigener Konfiguration, siehe "Konfiguration der Server URL".

Zertifikate, User und Passwörter

Die Docker Testumgebung ist mit festen Zertifikaten ausgestattet. Diese finden Sie in obigem Adapter- und Importer Download und direkt hier:

Für	Zertifikate	User, Passwort	Kommentar
Laborseite	Zertifikate-Labor	test-lab999, W7JDGJOVJ7	
Krankenhausseite	Zertifikate-Krankenhaus	test-hosp999, NZRE6NH65C	zugeordnete IK Nummer **987654321** Dieser IK ist der folgende Test-Standort zugewiesen: <IK>987654321</IK> <Bezeichnung>Testkrankenhaus - gematik GmbH</Bezeichnung> <PLZ>10117</PLZ> <Ort>Berlin</Ort> <Straße>Friedrichstraße</Straße> <Hausnummer>136</Hausnummer> <BSNR>987654321</BSNR> <StandortId>**987654**</StandortId>

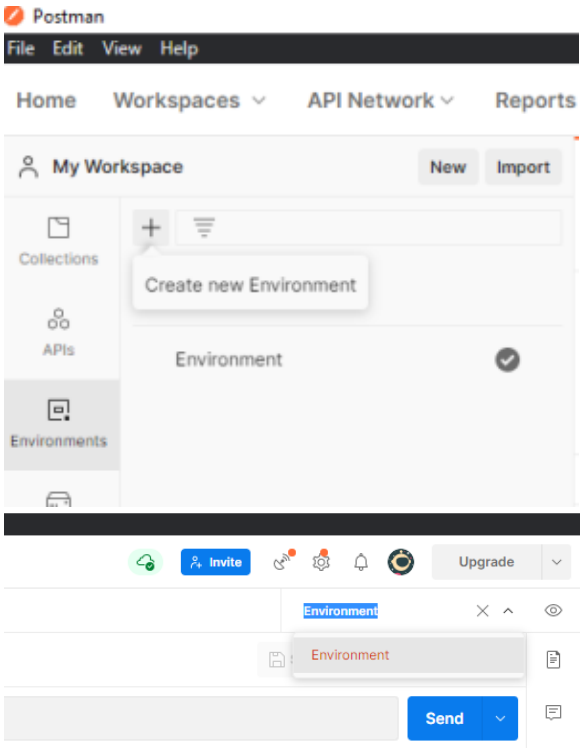
Gesundheitsamtseite	Zertifikate-Gesundheitsämter	1.test-oegd01, G9UVPJTXSV 1.test-oegd02, NCHR5LVL9N 1.test-oegd03, ER07OQEPH2 1.test-oegd04, N0HF3UJMPL 1.test-oegd05, ZTS6QXCVHI 1.test-oegd06, 98E3CNJ7GD 1.test-oegd07, LCHRB3RAN 1.test-oegd08, Y9C1INIKMV 1.test-oegd09, JJ76T3HXU1 1.test-oegd10, G8BSWZ3XLA 2.test-oegd01, 3XG8QMG2A3 2.test-oegd02, ID0FAC8QQJ 2.test-oegd03, K8B5D5UJ6K 2.test-oegd04, 1IXGPA0735 2.test-oegd05, 4151TE4GQO 2.test-oegd06, YR745PUF6G 2.test-oegd07, 4COKM5HOSZ 2.test-oegd08, 2ZRV5265CS 2.test-oegd09, 2XZJNZ78Y7 2.test-oegd10, PZUMJYLYPY	

FHIR Schnittstelle - Postman - Collection

Zum direkten Testen der DEMIS Docker Testumgebung wurde eine Postman Collection bereitgestellt: DDTU.postman_collection.json

Vorbereitung/Einstellungen

Zur Nutzung der Speicherung der Tokens muss ein neues Environment angelegt werden (Name irrelevant) und auch in Postman ausgewählt werden:



Des Weiteren müssen die passenden Zertifikate in Postman importiert und ausgewählt sein.

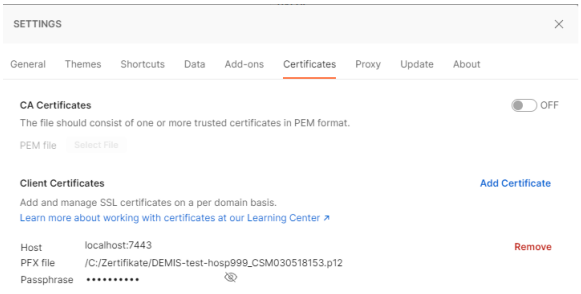
Für die Labormeldungen muss das Zertifikat test_lab999_*.p12 eingebunden werden.

Für die Hospitalisierungsmeldung und die Bettenbelegungsmeldung muss das Zertifikat test_hosp999_*.p12 eingebunden werden.

Postman lässt leider nur ein Zertifikat pro Host zu, deshalb muss das korrekte Zertifikat eingebunden werden.

Settings > Certificates > Client Certificates

Hinweis: Es darf kein " " Leerzeichen im Pfad zu dem Zertifikat sein! (Postman kann es sonst leider nicht korrekt verarbeiten.)



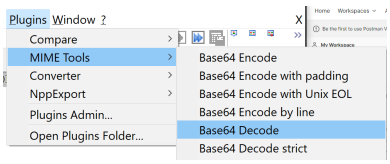
Um eine Meldung zu senden, muss zunächst der jeweilige "Token" Request ausgeführt werden. Dieser speichert den Token, der 10 Minuten gültig ist, in einer Environment Variable und dieser wird dann in den jeweiligen Requests genutzt.

PDF Response anschauen

Beim Postman kommt in der Respone die Meldungsquittung als Base64, z.B.:

Einfach hier den ganzen Inhalt von <data value="XXXX"></data> also die XXXX in notepad++ kopieren. In notepad++ alles markieren/auswählen und Base64 dekodieren.

Danach die Datei als *.pdf abspeichern.



Meldungen senden - Laborseite - DEMIS-Adapter

Zum Senden von Meldungen wird ein vorkonfigurierter DEMIS-Adapter mit dem Testlabor DEMIS mit der ID test-lab999 bereitgestellt.

Im Unterordner "data" befinden sich 10 Test-Meldungen, die jeweils einem passenden Test-Gesundheitsamt zugestellt werden. Zusätzlich befindet sich hier die Meldung "test-oegd11_INVALID.LDT", die dem Test-Gesundheitsamt 11 (Postleitzahlen von Bielefeld) zugestellt werden soll. Dies stellt den Negativ-Test dar und wird mit dem oben genannten Fehler abgelehnt.

Weitere Informationen zum DEMIS-Adapter finden Sie hier: DEMIS-Adapter

Die Meldungen in der lokal laufenden Docker Umgebung werden nach 30 Tagen gelöscht (vorausgesetz die Umgebung läuft auch Nachts).

DEMIS-Adapter 2.0.x

Die DEMIS Docker Testumgebung 1.1.0 unterstützt die Verarbeitung der neuen Profile (https://simplifier.net/demis). Zur Nutzung des neuen Profils für SARS-CoV-2 ist der DEMIS Adapter 2.0.0 (oder höher) mit entsprechenden Test-LDT und JSON Dateien hier zu finden (vorkonfigurierter DEMIS-Adapter und DEMIS-Importer).

Eine Rückwärtskompatibilität ist ebenfalls gegeben, sodass die Beispieldateien im vorherigen Profilformat im DEMIS-Adapter 1.7.1 auch weiterhin genutzt werden können

---

Meldungen abrufen - Gesundheitsamtsseite - DEMIS-Importer

Zum Abrufen der Meldungen wird ein vorkonfigurierter DEMIS-Importer bereitgestellt, der alle zehn Test-Gesundheitsämter, sowie die zehn Zweitzertifikate enthält.

Im Ordner "windows" bzw. "linux" finden Sie insgesamt 20 start.cmd Dateien, die jeweils das vorkonfigurierte Test-Gesundheitsamt darstellen und für dieses die Meldungen abruft.

Weitere Informationen zum DEMIS-Importer finden Sie hier: DEMIS-Importer

---

Konfiguration der Server URL

Bei Nutzung der DEMIS Docker Testumgebung außerhalb von "localhost" besteht ab dem Release 1.1.0 die Möglichkeit, die Server URL und den Port über eine Environment Variable in der demis.yml zu setzen. Bitte ersetzen Sie in allen Parametern "localhost:7443" mit Ihrer Server URL und Port. Fehler in der Konfiguration führen dazu, dass einzelne Komponenten nicht erreichbar sind!

Image: gematik1/demis-keycloak

- KC_HOSTNAME_ADMIN_URL: https://localhost:7443/auth/
- KC_HOSTNAME_URL: https://localhost:7443/auth/

Image: gematik1/demis-notification-clearing-api

- zusätzlicher environment Parameter: HAPI_OIDC_ISSUER=https://localhost:7443/auth/realms/OEGD
- zusätzlicher environment Parameter: HAPI_SERVER_ADDRESS=https://localhost:7443/notification-clearing-api/fhir/

Image: gematik1/demis-notification-entry-service

- zusätzlicher environment Parameter: HAPI_OIDC_ISSUER=https://localhost:7443/auth/realms/LAB,https://localhost:7443/auth/realms/HOSPITAL
- zusätzlicher environment Parameter: HAPI_SERVER_ISSUER=https://localhost:7443/notification-entry-service/fhir/

Befüllte Postgres Datenbank mit Beispielmeldungen

Die DEMIS Docker Testumgebung Release 1.1.0 bietet die Möglichkeit, eine mit Beispielmeldungen befüllte Datenbank zu nutzen. Hierzu nutzen Sie die demis.yml Datei des Releases 1.1.0. Die eingespielten Beispielmeldungen decken SARS-CoV-2, Influenza und Rotavirus ab und sind hier zu finden: Beispiele für Erregernachweismeldungen

Für jedes der zehn Test-Gesundheitsämter sind Beispielmeldungen enthalten, die die oben genannten unterstützten Profile darstellen. Sie können diese mit dem DEMIS-Importer abholen.

Image: gematik1/demis-postgres:1.8.2-configured-test

- zusätzlicher environment Parameter: PGDATA=demis_database

---

*) In V2 von Compose ist Compose direkt in docker integriert. Daher ist "docker compose" statt "docker-compose" zu verwenden.