

Bruikbaarheidstest met \$everything-operator op FHIR-servers  
<https://medmij.atlassian.net/browse/AF-84>

Bruikbaarheidstest met \$everything-operator op FHIR-servers  
[https://dev.azure.com/Medmij/Anulap/\\_git/local-env-fhirservers](https://dev.azure.com/Medmij/Anulap/_git/local-env-fhirservers)

----


Koppelvlak-PGO  
[https://dev.azure.com/Medmij/Anulap/\\_git/local-env-pgokoppelvlak](https://dev.azure.com/Medmij/Anulap/_git/local-env-pgokoppelvlak)  
[https://dev.azure.com/Medmij/Anulap/\\_git/pgo-koppelvlak-bron](https://dev.azure.com/Medmij/Anulap/_git/pgo-koppelvlak-bron)  
[https://dev.azure.com/Medmij/Anulap/\\_git/pgo-koppelvlak-doel](https://dev.azure.com/Medmij/Anulap/_git/pgo-koppelvlak-doel)


 main 

 / Type to find a file or folder...


Files











 Set up build

 Clone



Contents History



Name ↑	Last change	Commits
 docs	Monday	<a href="#">e324c778</a> Added: export testdata, run_all and results Patrick de Kleijn
 scripts	54m ago	<a href="#">bcaf69cb</a> Added: tests for wildcards and type-filters Patrick de Kleijn
 servers	11h ago	<a href="#">9488540a</a> Updated: operator plugin configuration for Firely Server Patrick de Kleijn
 testdata	Yesterday	<a href="#">7c5088e0</a> Updated: defaults to R4 for improved import: Patrick de Kleijn
 testresults	11h ago	<a href="#">f36ca3b2</a> Updated: result-paths configuration for STU3 Patrick de Kleijn
 .gitignore	Monday	<a href="#">5a05267e</a> Added: compare tool, capabilities report, support STU3 and R4 Patrick de Kle...
 docker-compose.yaml	Yesterday	<a href="#">7c5088e0</a> Updated: defaults to R4 for improved import: Patrick de Kleijn
 docker-down.sh	Friday	<a href="#">6a8719f2</a> Added: debugging tools, Medplum, LinuxForHealth Patrick de Kleijn
 docker-up.sh	Friday	<a href="#">6a8719f2</a> Added: debugging tools, Medplum, LinuxForHealth Patrick de Kleijn
 README.md	54m ago	<a href="#">bcaf69cb</a> Added: tests for wildcards and type-filters Patrick de Kleijn

## FHIR servers for testing

This repository contains setup script for FHIR resource servers widely in use. It comes with a few test scripts to validate functionality and behaviour of server software.

More specifically we used this repo to see if operations like [\\$export](#) and [FHIR Documents](#) return similar results to advice MedMij participants in querying their resource server.

## Resource servers

Five server configurations are included in this repository. The servers are started in containers with Docker Compose and are accessible through the following URLs. For testing purposes, authentication and secure communication are taken out from configuration. Medplum however, requires authentication and LinuxForHealth requires HTTPS.

# FHIR servers for testing

---

This repository contains setup script for FHIR resource servers widely in use. It comes with a few test scripts to validate functionality and behaviour of server software.

More specifically we used this repo to see if operations like `$export` and `FHIR Documents` return similar results to advice MedMij participants in querying their resource server.

## Resource servers

---

Five server configurations are included in this repository. The servers are started in containers with Docker Compose and are accessible through the following URLs. For testing purposes, authentication and secure communication are taken out from configuration. Medplum however, requires authentication and LinuxForHealth requires HTTPS.

Name	FHIR capabilities, API endpoints	Web application, documentation
Net tools	net_tools	<a href="https://hub.docker.com/r/someguy123/net-tools">https://hub.docker.com/r/someguy123/net-tools</a>
Microsoft SQL	:1433	
PostgreSQL	:5432	
Redis	:6379	
HAPI	GET <a href="http://localhost:8100/fhir/metadata">http://localhost:8100/fhir/metadata</a>	<a href="http://localhost:8100/">http://localhost:8100/</a>
Medplum *	GET <a href="http://localhost:8200/fhir/R4/metadata">http://localhost:8200/fhir/R4/metadata</a>	<a href="https://github.com/medplum/medplum/tree/main/packages/app">https://github.com/medplum/medplum/tree/main/packages/app</a>
Firely Server	GET <a href="http://localhost:8300/metadata">http://localhost:8300/metadata</a>	<a href="http://localhost:8300/">http://localhost:8300/</a>
Microsoft FHIR	GET <a href="http://localhost:8400/metadata">http://localhost:8400/metadata</a>	

Name	FHIR capabilities, API endpoints	Web application, documentation
IBM LinuxForHealth *	GET <a href="https://localhost:9443/fhir/metadata">https://localhost:9443/fhir/metadata</a>	<a href="https://localhost:9443/openapi/ui/">https://localhost:9443/openapi/ui/</a>

\* Note that we couldn't take out auth and https for Medplum and LinuxForHealth, but scripts are updated to work with tokens and local signed certificates. It's a bit of a grind in testing now. We'll get around that if more testing requires it.

## Start servers

Servers are configured in containers with [docker-compose.yaml](#). Use [docker-up.sh](#) and [docker-down.sh](#) scripts to start and stop the servers.

<input type="checkbox"/>			local-env-fhirservers	-	-	2.46%
<input type="checkbox"/>			cache_redis	redis:7	6379:6379 ↗	0.37%
<input type="checkbox"/>			fhir_firely	firely/server	8300:4080 ↗	0.01%
<input type="checkbox"/>			fhir_hapi	hapiproject/hapi:latest	8100:8080 ↗	0.27%
<input type="checkbox"/>			fhir_linuxforhealth	linuxforhealth/fhir-server	9443:9443 ↗	0.32%
<input type="checkbox"/>			fhir_medplum	medplum/medplum-server	8200:8200 ↗	0.01%
<input type="checkbox"/>			fhir_microsoft	healthcareapis/stu3-fhir-server	8400:8080 ↗	0.27%
<input type="checkbox"/>			net_tools	someguy123/net-tools		0%
<input type="checkbox"/>			sql_microsoft	mssql/server:2019-latest	1433:1433 ↗	1.21%
<input type="checkbox"/>			sql_postgres	postgres:16	5432:5432 ↗	0%

## Testdata folder with examples

---

Examples taken from Nictiz repositories with slight modifications like taking off the `<?xml />` headers to simplify import. For quick testing, the `StructureDefiniton`s are not imported, which should be in a future dataset to fully test MedMij compliancy. Even on permissive settings Firely Server applies validation well and stops importing when fields like `Observation.comments` are in the `*-examples.xml` files. Other servers like HAPI and Microsoft are more forgiving when validation is turned off (by default).

For this first round of tests, these examples will do. For other tests, we should review the dataset. Also, examples may benefit from an update to have resources link to a single Patient doing an `$export` operation, etc.

- Source: [Nictiz-STU3](#) and optionally R4 repositories.
- Import: [Test data](#) folder.

Patient resource	File name	Name
Patient/medmij-bgz-patient-01	medmij-bgz-patient-01.xml	Thomas van Beek
Patient/example	medmij-bgz-patient-02-example-id.xml	Ties van Oest
Patient/medmij-medication-patient-01	medmij-medication-patient-01.xml	Piet Jansen

## Import testdata

---

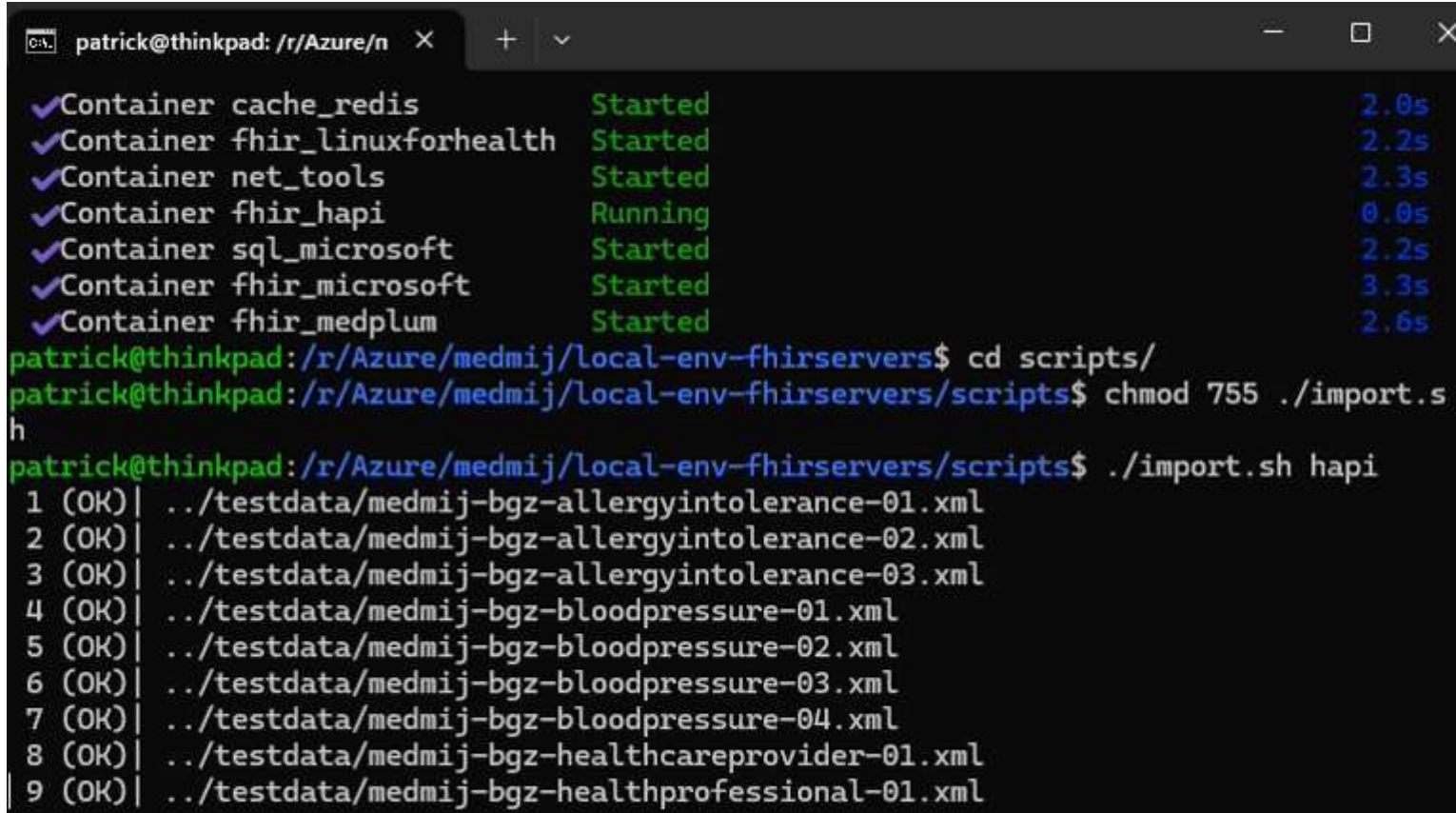
Medplum requires an authentication token to import data, some additional scripting would be useful to make this easy. Haven't looked into yet as we quickly want some first results. LinuxForHealth currently fails to connect, look into that later.

```
Usage: import.sh <server> [auth_token]
```

```
./scripts/import.sh hapi  
./scripts/import.sh medplum dXlBleGFtcGx1LmNvbTpzZNyZXQ=  
./scripts/import.sh firely
```

```
./scripts/import.sh microsoft
./scripts/import.sh linuxforhealth
```

Show video import datasets



```
patrick@thinkpad: /r/Azure/n x + v - □ X
✓Container cache_redis Started 2.0s
✓Container fhir_linuxforhealth Started 2.2s
✓Container net_tools Started 2.3s
✓Container fhir_hapi Running 0.0s
✓Container sql_microsoft Started 2.2s
✓Container fhir_microsoft Started 3.3s
✓Container fhir_medplum Started 2.6s
patrick@thinkpad:/r/Azure/medmij/local-env-fhirservers$ cd scripts/
patrick@thinkpad:/r/Azure/medmij/local-env-fhirservers/scripts$ chmod 755 ./import.s
h
patrick@thinkpad:/r/Azure/medmij/local-env-fhirservers/scripts$ ./import.sh hapi
1 (OK)| ../testdata/medmij-bgz-allergyintolerance-01.xml
2 (OK)| ../testdata/medmij-bgz-allergyintolerance-02.xml
3 (OK)| ../testdata/medmij-bgz-allergyintolerance-03.xml
4 (OK)| ../testdata/medmij-bgz-bloodpressure-01.xml
5 (OK)| ../testdata/medmij-bgz-bloodpressure-02.xml
6 (OK)| ../testdata/medmij-bgz-bloodpressure-03.xml
7 (OK)| ../testdata/medmij-bgz-bloodpressure-04.xml
8 (OK)| ../testdata/medmij-bgz-healthcareprovider-01.xml
9 (OK)| ../testdata/medmij-bgz-healthprofessional-01.xml
```

## Export testdata with script

Export runs an `$everything` and `$export` operation on the FHIR server. Tests with FHIR Documents too, and tests if pagination would work on these operations if requested.

Usage: export.sh <testname> <server> [auth\_token]

```
./scripts/export.sh test1 hapi
./scripts/export.sh test1 medplum mytesttoken
./scripts/export.sh test1 firely
./scripts/export.sh test1 microsoft
./scripts/export.sh test1 linuxforhealth
```

Compare [test results](#) between FHIR servers.

Test	Description
test01	Export resources with <a href="#">Patient \$everything</a> operation.
test02	Is pagination supported on Patient <a href="#">\$everything</a> operation?
test03	Export resources with <a href="#">Patient/id/\$everything</a> operation.
test04	Is pagination supported with Patient/id/ <a href="#">\$everything</a> ?_count ?
test05	Export resources with <a href="#">\$export</a> operation.
test06	Is pagination supported with Patient/id/export?
test07	Export resources with <a href="#">FHIR Documents</a> .
test08	Is pagination supported with FHIR Documents?
test09	Can we modify <a href="#">\$everything</a> output with _include?
test10	Can we modify <a href="#">\$everything</a> output with _include fields?

Results:

- Firely: GET [http://localhost:8300/Patient/\\$everything](http://localhost:8300/Patient/$everything) 501 Not implemented

# Compare tool

---

Manual comparison of the output is a bit tough and the queries doesn't cover all options. For example, an export can use `_type` parameter to include resources. Authentication support, comparing number of resources, etc. We wrote a small tool in C# to help compare details for FHIR servers.

```
Linux:    compare
Windows:  compare.exe
```

```
Import:   compare.exe --import
          compare.exe --import firely
```

```
Compare:  compare.exe --compare
```

Source code is in the [tools/compare](#) folder. Compiled version is in the [tools/compare/bin](#) folder, but for this quick one-time exercise just running it from Visual Studio is fine.

Here's an example import of testdata on a single server. It takes about 4 minutes for the full set on all servers. Usually, we would import on all servers as a one-time job after `docker-up.sh` and keep using the databases of FHIR-servers until `docker-down.sh` and `clean.sh` removes the data folders (e.g. `servers/postgres/data`). You may choose to pause the FHIR servers as a group and hold on to the data for later use. That saves you time experimenting with the servers.

```
compare.exe --compare hapi
```

```
Importer Nictiz-testdata op HAPI:
```

```
Importer van 744 definities voor profielen, codelijsten enz.
```

```
- [ 1/744] Definitie: ext-AdditionalCategory.xml
- [ 2/744] Definitie: ext-Context-EpisodeOfCare.xml
- [ 3/744] Definitie: ext-CopyIndicator.xml
```



- [ 4/744] Definitie: ext-EdifactReferenceNumber.xml
- [ 5/744] Definitie: ext-Vaccination.PharmaceuticalProduct.xml
- [ 6/744] Definitie: nl-core-AbilityToDressOneself.xml
- [ 7/744] Definitie: nl-core-AbilityToDrink.DrinkingLimitations.xml
- [ 8/744] Definitie: nl-core-AbilityToDrink.xml
- [ 9/744] Definitie: nl-core-AbilityToEat.EatingLimitations.xml
- [10/744] Definitie: nl-core-AbilityToEat.xml
- [11/744] Definitie: nl-core-AbilityToGroom.xml
- [12/744] Definitie: nl-core-AbilityToPerformMouthcareActivities.Prosthesis.Product.xml
- [13/744] Definitie: nl-core-AbilityToPerformMouthcareActivities.Prosthesis.xml
- [14/744] Definitie: nl-core-AbilityToPerformMouthcareActivities.xml
- [15/744] Definitie: nl-core-AbilityToUseToilet.MenstrualCare.xml
- ...

Importeren van 252 voorbeelden.

- [ 1/252] Voorbeeld: nl-core-AbilityToPerformMouthcareActivities.Prosthesis.Product-01.xml
- [ 2/252] Voorbeeld: nl-core-AbilityToDressOneself-01.xml
- [ 3/252] Voorbeeld: nl-core-AbilityToDrink-01.xml
- [ 4/252] Voorbeeld: nl-core-AbilityToDrink.DrinkingLimitations-01.xml
- [ 5/252] Voorbeeld: nl-core-AbilityToDrink.DrinkingLimitations-02.xml
- [ 6/252] Voorbeeld: nl-core-AbilityToEat-01.xml
- [ 7/252] Voorbeeld: nl-core-AbilityToEat.EatingLimitations-01.xml
- [ 8/252] Voorbeeld: nl-core-AbilityToEat.EatingLimitations-02.xml
- [ 9/252] Voorbeeld: nl-core-AbilityToGroom-01.xml
- [10/252] Voorbeeld: nl-core-AbilityToPerformMouthcareActivities-01.xml
- [11/252] Voorbeeld: nl-core-AbilityToPerformMouthcareActivities.Prosthesis-01.xml
- [12/252] Voorbeeld: nl-core-AbilityToUseToilet-01.xml
- [13/252] Voorbeeld: nl-core-AbilityToWashOneself-01.xml
- [14/252] Voorbeeld: nl-core-AdvanceDirective-01-Condition-01.xml
- [15/252] Voorbeeld: nl-core-AdvanceDirective-01-RelatedPerson-01.xml
- ...

After an import, the tool can compare results from the FHIR servers, show that in a markdown table, printed here:

```
compare.exe --compare
```

Vergelijken van testresultaten:

- Session met HAPI
- Session met Firely
- Session met Microsoft
- Session met Medplum
- Session met IBM

- Session met HAPI
- Session met Firely
- Session met Microsoft
- Session met Medplum
- Session met IBM

Zijn de FHIR-servers beschikbaar in lokaal netwerk?

Server	Beschikbaar
HAPI	online
Firely	online
Microsoft	online
Medplum	online
IBM	running, outdated

Welke export queries voor een externe client beschikbaar?

Capabilities HAPI	Beschikbaar
everything	beschikbaar
patient-everything	beschikbaar
export	
patient-export	beschikbaar
group-export	beschikbaar
anonymized-export	

Capabilities Microsoft	Beschikbaar
everything	
patient-everything	beschikbaar
export	beschikbaar
patient-export	beschikbaar
group-export	beschikbaar
anonymized-export	beschikbaar

Capabilities Firely	Beschikbaar
everything	beschikbaar
patient-everything	beschikbaar
export	beschikbaar
patient-export	beschikbaar

Capabilities Firely	Beschikbaar
group-export	beschikbaar
anonymized-export	beschikbaar

Wat geven de servers terug als we ze uitvragen?

Operations HAPI	Resources
1. \$everything	20
2. \$everything _count	1
3. \$everything id	20
4. \$everything id _count	1
5. \$export	-
6. \$export id	-
7. \$document	-
8. \$document id	-
9. \$everything id _include	20
10. \$everything id _includes	20
11. \$everything id _include=*	-
12. \$everything id _type	-

Operations Microsoft	Resources
1. \$everything	-

Operations Microsoft	Resources
2. \$everything _count	-
3. \$everything id	1
4. \$everything id _count	2
5. \$export	-
6. \$export id	-
7. \$document	-
8. \$document id	-
9. \$everything id _include	2
10. \$everything id _includes	2
11. \$everything id _include=*	-
12. \$everything id _type	-

Firely fully supports all FHIR operations defined by HL7 after a license key is obtained. Plugins and configuration of plugins decide what resources the server returns in operations like \$everything. Unfortunately, we most probably cannot override that configuration on a per-query basis with \_includes. (Same as for Microsoft, see table above. HAPI does allow adjusting the \$everything export with query \_includes.)

## Good support for \$everything operation, but differences in behaviour

---

Remarks 2024-02-18 on availability of operations in FHIR servers we're looking at here:

- **HAPI JPA** has **full support** for the \$everything operation. HAPI also has interesting alternative ways of passing multiple logical IDs out-of-spec, not relevant to our tests. By default bundles all resource types linked to Patient. I found HAPI to not support (the out-

of-specs) pagination on `$everything (_count)`, there is support for `$export` (which didn't work in this local install, but should be possible) and `$document` operations not available. There is mention of Bulk Data Export (= `$export`) in changelogs and code.

- **HAPI Plain** might be in use at PGO as a more flexible FHIR-interface built-in on other software (maybe for testing purposes). Good to know is that **HAPI Plain** won't support `$everything` operation and others out-of-the-box: requires a manual operation method, but fairly simple to implement.
- **Firely Server** has **full support** of `$everything` and other operations, and it requires a non-trial-licence and turning on a plugin at `appsettings.default.conf`. This took me by surprise and makes it difficult to test for now: we may request a temp licence with these plugins allowed for further testing. Didn't do that for now. Firely supports `$export` and `$document` operations too. Firely by default *filters out Practitioner* from `$everything` for privacy considerations, but there are configuration options for which resources to include with the Patient compartment. We can manipulate includes in the client query with `_include` and `_revinclude` too, but couldn't test if this overrides the server settings. Most likely takes a secure approach and prevents if not configured.
- **Microsoft FHIR Server** has **full support** for the `$everything` operation. Unfortunately, the server does not support `_include` on `$everything` operation. The newer server version is named 'Azure API for FHIR' and lets the system level Bulk **`$export operation`** export to Azure Blob Storage, which would make `$export` less suitable for a scenario of transferring data between PGOs.
- **Medplum FHIR supports** the `Patient/id/$everything` syntax. Supports `$export`, probably `$document` operations too. The **Composition-document bundle** is built-in the source code, so `$document` should work too. Haven't tested, but it means that Medplum at least supports returning of document bundles.
- **IBM LinuxForHealth supports** the `$everything` operation. While it mentions the Java API is not stable, this just works fine. The project however didn't merge updates for a couple of years, nor communicated on status. EOL.
- Adding `_include` and `_revinclude` for Patient **`$everything operation compartment`** would have been a nice way to make sure all resources are included in the export. Unfortunately, Microsoft's implementation doesn't support this and Firely's implementation most likely requires fixed configuration in `appsettings`.

## Summary:

Trying to find a way for servers to behave similar in exporting data, a solution seem to increasingly require more custom work that just calling an endpoint when you look at behaviour of FHIR-servers.

1. The `Patient/$everything` operation is NOT widely available.

2. The `Patient/id/$everything` operation IS widely available. Unfortunate drawback is that different servers behave differently when including linked resources. Firely supports configuration of what exactly serverwide, HAPI returns all within Patient compartment, etc. Not all servers support `_includes` and `_pagination` on this operation. That is very unfortunate, because this way it is hard to get similar behaviour in exports of a patient's data.
3. Other export operations may have a server specific implementation too: Bulk Export operations and Compartment Document bundles are well supported by most servers, but trying to call these with an external PGO client will not always lead to a returned bundle. Microsoft's implementation exports to Azure storage only.
4. Most FHIR-servers allow you to simply built an operation plugin or such. That's an inconvenience, we would be asking for a `$everything_medmij` operation. That comes down to writing a small API.

Failing a.t.m. to find a single FHIR operation (through query parameters and server configuration) that behaves similar over server brands. We may have to resort to a custom operation or API returning a search bundle or FHIR Documents bundle. Probably that would be an acceptable amount of work for a PGO to implement, but it would be nice if we could find a way to do this in a single query with the standard FHIR operations.

### Alternatives to Patient-export in full:

We're discussing this in ontwerpdocument and in team, but quick overview:

1. Receiving PGO runs a `GET Patient/id/$everything` with source PGO and accept that there are differences in completeness of levensloopdossier. Best effort.
2. Receiving PGO runs [multiple queries for gegevensdienst](#) or [all resources linked to Patient compartment](#) with source PGO.
3. Receiving PGO calls a single API-endpoint on source PGO that returns a bundle of all resources linked to Patient compartment. Similar as (2) having the receiving PGO run queries. Now the source PGO is responsible for creating a bundle.

There is an advantage of shifting the responsibility for bundling to source (2 → 3), so source has control over what data is sent over. There's more control than with direct FHIR queries from receiving PGO to. And quite useful: it wouldn't require source PGO to offer a full FHIR server.

## Net tools

---

Some notes helpful in testing connections with containers in the local Docker subnet.

```
su -  
apt update  
apt install -y postgresql-client-common postgresql-client redis-tools  
  
nc -zv sql_postgres 5432  
telnet sql_postgres 5432  
pg_isready -h sql_postgres -p 5432  
psql -h sql_postgres -p 5432 -U medplum -d medplum
```



# PGO-Koppelvlak

Bruikbaarheidstest \$everything-operator ([AF-84](#))

<https://medmij.atlassian.net/wiki/spaces/AS/pages/168034305/PGO-koppelvlak+Q1-2025>

februari 2025

# Waar het team aan werkt

Tijdens voorverkenning kwam naar voren dat er verschillende opties zijn in **uitwisselmethode** en **uitwisselgegevens**. In **alpha-fase** bekijken we de **haalbaarheid van implementeren opties** uitwisselen tussen Dienstverleners Persoon.

- ✔ **Deelnemers bevragen** in expertsessie over ons eerste voorstel uit voorverkenning. PDF/A.
- ✔ **Beschikbaarheid componenten** bekijken (authenticatie en resourceservers). Vragenlijst MedCom.
- ✔ **Testen op technische beperkingen** in FHIR-resource servers van deelnemers. \$everything, \$export
- ✔ **Gebruikers bevragen** in betafase: gebruikersinteractie beperkt en sterk afhankelijk van deelnemers.

<https://medmij.atlassian.net/wiki/spaces/AS/pages/168034305/PGO-koppelvlak+Q1-2025>

# Opties in uitwisselmethode

Drie opties voor uitwisselen. We kozen in technisch ontwerp voor het overbrengen van een FHIR-bundel met gegevens 'endpoint-to-endpoint' met voorkeur boven het downloaden-uploaden van een bestand.

1. **Overbrengen portabiliteitsrapport** is de huidige situatie. Een bevragende PGO krijgt de adresgegevens van een andere PGO om dezelfde verzoeken aan DVA's uitvoeren.

**Voordeel:** helpt bij lokalisatie. **Nadeel:** verlies van levensloop na veranderingen bij leverende deelnemers, sommige gegevens niet meer beschikbaar na twee jaar.

2. **Downloaden bestand** (zip, tar) bij PGO en uploaden bij een ander PGO.

**Voordeel:** patiënt beschikt over archiefbestand. **Nadeel:** handmatige gebruikersoverdracht, gegevens raken mogelijk onbedoeld buiten stelsel, buiten scope uitwisseling Afsprakenstelsel.

3. **Overbrengen FHIR-bundel** of resource in 'endpoint-to-endpoint'.

✓ **Voordeel:** gebruiker hoeft zelfs niets technisch te doen, past binnen uitwisseling Afsprakenstelsel.  
**Nadeel:** beschikbaarstellende PGO's moeten (net als DVA) een autorisatie- en resourceserver plaatsen.

*getest*

# Opties in uitwisselmethode

In het overbrengen van een FHIR-bundel met gegevens 'endpoint-to-endpoint' twee varianten denkbaar:

**3a. Verzamelinstructie bij bevragende PGO leggen** past goed bij hoe PGO's nu DVA's bevragen.

**Bijvoorbeeld:** Geef mij alle gegevens gelinkt aan patiënt. **Of:** geef mij alle informatie gegevensdienst.

GET https://bron-pgo/fhir/Patient/123/\$everything

GET https://bron-pgo/fhir/Observation?subject=Patient/123

*getest*

**3b. Verzamelinstructie bij leverende PGO leggen** past goed bij meer controle voor leverende PGO, situaties waar FHIR niet goed past of een bundel (met document) gegenereerd moet. **Bijvoorbeeld:** Geef mij een PDF waarin alle patiëntgegevens staan. **Of:** Maak archiefbestand op maat voor patiënt.

GET https://bron-pgo/api/maak\_pdf\_voor\_patient?subject=123

GET https://bron-pgo/api/maak\_archiefbestand?taal=nl&user=123

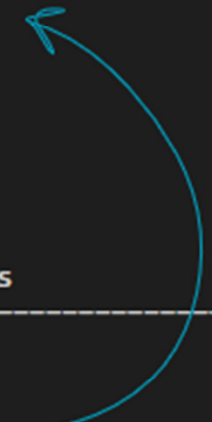
# Testscenario: \$everything, \$export, \$document

In Afsprakenstelsel vragen PGO's bij DVA's met FHIR-queries informatie op over een patient. Dit volgt een serie afgesproken verzoeken volgens gegevensdienst waarvoor is gekwalificeerd.

- ✓ **Voorstel:** Doe verzoeken op dezelfde wijze voor PGO naar PGO. **Nadeel:** Dubbele informatie ontstaat bij ophalen langs meerdere gegevensdiensten. Ontdubbelen? We willen een patient de volledige levensloop in dossier laten meenemen. **Vraag:** Is gebruik van FHIR Patient \$everything operator een optie?
- ✓ **Vragen:** Gebruiken PGO-deelnemers resourceservers die \$everything en \$export ondersteunen of gebruiken veel PGO's slechts een FHIR-client? **Antwoord:** ...
- ✓ **Testen:** Geven FHIR-servers als Firely Server, HAPI/Smile en Microsoft Azure FHIR Server de resultaten op dezelfde wijze terug? **Antwoord:** Nee, alle resourceservers ondersteunen FHIR Patient \$everything, maar welke resources precies terugkomen is wat bewerkelijk; vereist soms applicatiebrede configuratie.

Welke export queries voor een externe client beschikbaar? Wat geven de servers terug als we ze uitvragen?

Capabilities HAPI	Beschikbaar	Operations HAPI	Resources
everything	beschikbaar	1. <code>\$everything</code>	20
patient-everything	beschikbaar	2. <code>\$everything _count</code>	1
export		3. <code>\$everything id</code>	20
patient-export	beschikbaar	4. <code>\$everything id _count</code>	1
group-export	beschikbaar	5. <code>\$export</code>	-
anonymized-export		6. <code>\$export id</code>	-
		7. <code>\$document</code>	-
Capabilities Microsoft	Beschikbaar	8. <code>\$document id</code>	-
everything		9. <code>\$everything id _include</code>	20
patient-everything	beschikbaar	10. <code>\$everything id _includes</code>	20
export	beschikbaar		
patient-export	beschikbaar	Operations Microsoft	Resources
group-export	beschikbaar	1. <code>\$everything</code>	-
anonymized-export	beschikbaar	2. <code>\$everything _count</code>	-
		3. <code>\$everything id</code>	1
Capabilities Firely	Beschikbaar	4. <code>\$everything id _count</code>	2
everything	beschikbaar	5. <code>\$export</code>	-
patient-everything	beschikbaar	6. <code>\$export id</code>	-
export	beschikbaar	7. <code>\$document</code>	-
patient-export	beschikbaar	8. <code>\$document id</code>	-
group-export	beschikbaar	9. <code>\$everything id _include</code>	2
anonymized-export	beschikbaar	10. <code>\$everything id _includes</code>	2



configureerbaar met `?_type`

Microsoft FHIR te configureren met `_type`. Firely met `appsettings.default.json`. Anderen met `_include`, soms leidt `_include` tot foutmelding.

# Opties in uitwisselmethode

In het overbrengen van een FHIR-bundel met gegevens 'endpoint-to-endpoint' twee varianten denkbaar:

**3a. Verzamelinstructie bij bevragende PGO leggen** past goed bij hoe PGO's nu DVA's bevragen.  
Bijvoorbeeld: Geef mij alle gegevens gelinkt aan patiënt. Of: geef mij alle informatie gegevensdienst.

GET [https://bron-pgo/fhir/Patient/123/\\$everything](https://bron-pgo/fhir/Patient/123/$everything)  
GET <https://bron-pgo/fhir/Observation?subject=Patient/123>

getest

**3b. Verzamelinstructie bij leverende PGO leggen** past goed bij meer controle voor leverende PGO, situaties waar FHIR niet goed past of een bundel (met document) gegenereerd moet. **Bijvoorbeeld:** Geef mij een PDF waarin alle patiëntgegevens staan. **Of:** Maak archiefbestand op maat voor patiënt.

GET [https://bron-pgo/api/maak\\_pdf\\_voor\\_patient?subject=123](https://bron-pgo/api/maak_pdf_voor_patient?subject=123)  
GET [https://bron-pgo/api/maak\\_archiefbestand?taal=nl&user=123](https://bron-pgo/api/maak_archiefbestand?taal=nl&user=123)

getest

# Opties in uitwisselgegevens

- ✓ **Gegenereerde PDF/a** via gegevensdienst 51: **Verzamelen documenten 3.0**
- ✓ **Nieuwe gegevensdienst** toevoegen voor een **FHIR-bundle-format**.
- ✓ **Bestaande gegevensdiensten** **herbruik en kwalificatie**.

<https://medmij.atlassian.net/wiki/spaces/AS/pages/149782532/Technisch+Ontwerp+Q125-PGOK#Uitwisselgegevens>