

# Fhir On Pi - 7 (Finale)



de linkerkant.

Na vijf afleveringen vol voorbereiding hebben we in de [vorige aflevering](#) eindelijk de HAPI Fhir-server gebouwd en geïnstalleerd.

Met de webbrowser heb je op het adres:

`http://<ip-adres van je Raspberry>:8080` de testclient van de HAPI Fhirserver gezien met alle “Fhir-resources” die je systeem kent aan

We zullen eerst de testclient eens bekijken (zie figuur, de kaders zijn er door mij ingetekend).

The screenshot shows the HAPI FHIR web interface. The top navigation bar includes 'Home', 'Server: Global Tester' (circled with a red 3), 'Source Code', and 'About This Server'. On the left, the 'Options' section (circled with a red 1) allows selecting 'Encoding' (XML or JSON) and 'Pretty' (On or Off). Below it is the 'Resources' list (circled with a red 2) showing counts for various FHIR resources like Observation, Patient, Encounter, etc. The main content area features the 'COMPANY NAME' logo and a table (circled with a red 4) displaying server information: 'UHN Test Server (R4 Resources)', 'HAPI FHIR Server - 5.3.0-SNAPSHOT/35f4d48eee/2020-12-03', and 'http://hapi.fhir.org/baseR4'. At the bottom, the 'Server Actions' section (circled with a red 5) includes buttons for 'Conformance' and 'History', along with input fields for 'Since' and 'Limit # (opt)'.

De kaders hebben (voor zover ik heb kunnen ontdekken) de volgende functie

Kader	Functie
1	Hiermee stel je in hoe je in deze client de antwoorden van de Fhir-server wil zien, je kunt kiezen tussen XML en JSON en je kunt ook nog aangeven of je wilt dat daarbij de leesbaarheid wordt vergroot. Ikzelf vind “JSON”-uitvoer die “pretty” is het makkelijkst te lezen.
2	Hier zie je de FHIR-“resources” in de server die je bevraagt. Je eigen FHIR-server is nog leeg, dus daar zie je (nog) geen aantallen, maar als je wisselt naar de global (test-) server (in kader 3) dan zie je enorme aantallen patienten, observations enz. (De schermafdruck hierboven is zo gemaakt).
3	Hier kies je tussen je eigen server (Local Tester) en de testserver op <a href="http://hapi.fhir.org">http://hapi.fhir.org</a> (Global Tester).

Kader	Functie
4	Hier zie je informatie over de Fhir-server waarmee je je testclient verbonden hebt.
5	In je testclient kun je hier het “conformance statement” opvragen van de fhir-server waarmee je bent verbonden. In dit statement vertelt de server (in FHIR-formaat) welke resources hij kent en hoe je die kunt benaderen. FHIR-servers kunnen (meestal) verschillende versies van FHIR aan, zoals DSTU2, DSTU3 en R4 maar ook binnen één versie zijn er vaak verschillen in de uitwisselingsstandaard. Met de “history”-knop vraag je op van wanneer tot wanneer wélke versie door de server wordt ondersteund.

## FHIR Resources

FHIR resources zijn de bouwstenen voor de gegevensuitwisseling met FHIR. FHIR resources zijn *grofkorrelig* (één resource omvat een grote hoeveelheid kenmerken). FHIR resources hebben ook een duidelijke relatie met de “werkelijke wereld”, het is in de meeste gevallen niet moeilijk om te bepalen welke resource(s) bij een bepaalde uitwisseling betrokken zijn. Maar makkelijker dan hier uitleg te krijgen is het om zelf een paar resources te bekijken.

1. Kies (in kader 3) voor “Global Tester”, want we gaan patienten bekijken op de server van hapi.fhir.org.
2. Klik op het woord “Patient” onder “Resources” (kader 2). Het grote deel van het scherm rechtsonder verandert nu want we zoomen in op de Patient-resource.
3. Klik op het boxje onder “Search Parameters” en kies voor “name” als zoek-criterium. Voer dan “Bommel” in, in het vak er rechts naast (“parameter”).
4. Klik dan op “Search” en als het goed is wordt er één patient gevonden, met het ID 1722521. Scroll naar beneden en je ziet (in JSON-formaat) alle gegevens die over deze patient (Ollie B. Bommel) bekend zijn<sup>1</sup>.
5. Kijk bijvoorbeeld ook even naar de resource “AllergyIntolerance”, hier zie je -bijvoorbeeld- de overgevoeligheid die een patient heeft voor garnalen. (“Shrimps”, ik denk niet dat ze daar psychologen mee bedoelen 😊).

---

<sup>1</sup> Het kan zijn dat de global database ondertussen geleegd is, doe dan een “search” op bijvoorbeeld “johnson” of “jones”, je krijgt dan meer dan één patient te zien en kunt met een knop meer over die patient opvragen.

In aflevering ben ik kort ingegaan op “eenheid van taal” en “semantische interoperabiliteit”. FHIR biedt een (duidelijk) kader wáár (bij welke “resource”) bepaalde gegevens horen te worden overgedragen. Dat komt omdat FHIR met die resources als het ware een sterk vereenvoudigde (“helicopterview”) weergave biedt van alles wat met “zorg” te maken heeft. FHIR is wat dat betreft te vergelijken met de dozen en containers van een verhuisbedrijf: er zijn kisten voor schilderijen, kratten voor serviesgoed en dozen voor boeken.

Maar dat is niet genoeg om “semantisch interoperabel” te zijn en daarom gebruikt FHIR ook code-stelsels als SNOMED om aan te geven wát de inhoud betekent. Heel vaak zie je in de output van de FHIR-server dan, behalve de gegevens zelf, ook een (gestandaardiseerde) verwijzing naar het codestelsel dat bij die gegevens hoort.

Bijvoorbeeld voor de “huwelijkse staat” (“marital status”) wordt bij de patient hiernaast het codestelsel

```
{
  "text": "Social Security Number"
},
{
  "system": "http://hl7.org/fhir/sid/us-ssn",
  "value": "999-64-8855"
}
],
"name": [ {
  "use": "official",
  "family": "Aguilar229",
  "given": [ "Marco Antonio298" ]
} ],
"telecom": [ {
  "system": "phone",
  "value": "555-295-9732",
  "use": "home"
} ],
"gender": "male",
"birthDate": "2010-04-26",
"address": [ {
  "extension": [ {
    "url": "http://hl7.org/fhir/StructureDefinition/geolocation",
    "extension": [ {
      "url": "latitude",
      "valueDecimal": 41.94231266601543
    },
    {
      "url": "longitude",
      "valueDecimal": -87.69398435080656
    }
    ]
  } ],
  "line": [ "204 Will Center" ],
  "city": "Chicago",
  "state": "IL",
  "postalCode": "60607",
  "country": "US"
} ],
"maritalStatus": {
  "coding": [ {
    "system": "http://terminology.hl7.org/CodeSystem/v3-MaritalStatus",
    "code": "S",
    "display": "Never Married"
  } ],
  "text": "Never Married"
},
"multipleBirthBoolean": false,
"communication": [ {
  "language": [ {
    "coding": [ {
      "system": "urn:ietf:bcp:47",
      "code": "es",
      "display": "Spanish"
    } ],
    "text": "Spanish"
  } ]
} ]
} ]
}
```

<http://terminology.hl7.org/CodeSystem/v3-MaritalStatus> gebruikt en voor zijn spreektaal het stelsel “bcp:47” (zie: <https://tools.ietf.org/html/bcp47>) .

Met de structuur en de syntax van FHIR en met de verwijzingen naar gebruikte codestelsels en andere standaarden wordt een belangrijke stap naar semantische interoperabiliteit gezet. Met FHIR zeg je zo iets als: “ik communiceer in het Nederlands volgens het Groene Boekje, en voor de duiding baseer ik me op Winkler Prins Online augustus 2020”.

## Genoeg theorie

We gaan patienten toevoegen aan onze eigen FHIR-server. Daarvoor gaan we op de PC het programmaatje “Postman” installeren. Je vindt allerlei installatiepakketten op <https://www.postman.com/downloads/> . Je kunt ook een cloud-versie gebruiken, maar dat werkt waarschijnlijk niet omdat je

Raspberry niet vanaf het Internet bereikbaar is (en dat wil je ook zo houden waarschijnlijk).

Postman is -heel kort gezegd- een REST API client. REST staat voor “Representational state transfer” en we gebruiken REST eigenlijk over het hele internet. Als je een zoekmachine gebruikt en je kijkt naar de adresbalk, net nadat je op de “Enter” toets drukte, zie je iets als in onderstaande figuur.

<https://www.google.com/search?q=Texelse+Chocolaterie&oq=Texelse+Chocolaterie&aqs=chrome..69i57j0j15&sourceid=chrome&ie=UTF-8>

Het volledige adres dat je vanuit je webpagina naar de zoekmachine stuurde ziet er ongeveer als volgt uit:

<ip-adres van de zoekmachine>/search?q=<je zoekopdracht>&<extra info> .

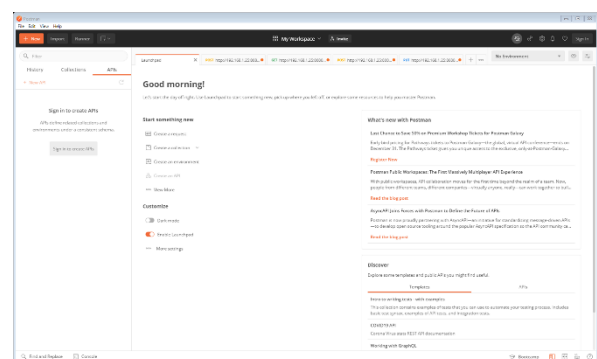
De opdracht die je het Internet opstuurde was een REST “GET” opdracht. Als je een webformulier invult, bijvoorbeeld bij een webwinkel, gebruik je een REST “POST” opdracht. Kortom: misschien heb je het nooit beseft, maar je gebruikt REST al een hele tijd! Als je meer wilt weten over REST kijk dan eens op [https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer) en [https://en.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol#Request\\_methods](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol#Request_methods).

Je begrijpt nu ook dat Postman niets met PostNL of Pieter Post (Postman Pete) te maken heeft, maar alles met de “POST” van REST. We gaan met Postman nu een nieuwe patient “posten” in onze eigen FHIR-server.



Start, als dat nog niet gebeurd is, Postman op je PC en kies voor “Create a request”.

In het scherm daarna (zie de afbeelding hiernaast) kun je kiezen voor één van de REST-opdrachten (bij mij begint Postman met een “GET”. Kies hier voor “POST” want we gaan een patient sturen en niet ophalen.



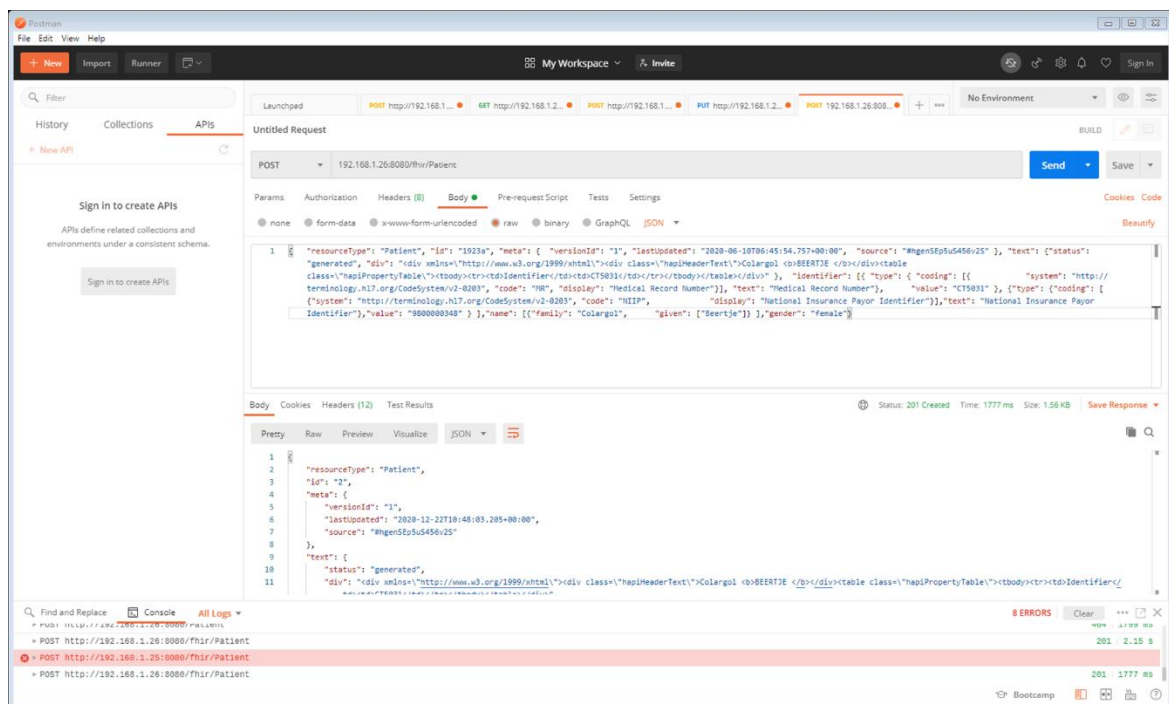
In de *adresbalk* van Postman zet je het volgende:

<ip-adres van je Pi>:8080/Patient

(vergeet vooral de :8080 niet, want dan krijg je een “connection refused” terug. Bij mijn Pi is dit dus: 192.168.1.25:8080/fhir/Patient

Voor de “body” van je bericht kies je voor “raw” en voor “JSON”(zie afbeelding hieronder) en dan klik je met je muis in het “body”-veld en zet daar het volgende neer:

```
{ "resourceType": "Patient", "id": "1923a", "meta": { "versionId": "1", "lastUpdated": "2020-06-10T06:45:54.757+00:00", "source": "#hgenSEp5uS456v2S" }, "text": { "status": "generated", "div": "<div xmlns='\"http://www.w3.org/1999/xhtml\"'><div class='\"hapiHeaderText\"'>Colargol <b>BEERTJE </b></div><table class='\"hapiPropertyTable\"'><tbody><tr><td>Identifier</td><td>CT5031</td></tr></tbody></table></div>" }, "identifier": [ { "type": { "coding": [ { "system": "http://terminology.hl7.org/CodeSystem/v2-0203", "code": "MR", "display": "Medical Record Number" } ], "text": "Medical Record Number", "value": "CT5031" }, { "type": { "coding": [ { "system": "http://terminology.hl7.org/CodeSystem/v2-0203", "code": "NIIP", "display": "National Insurance Payor Identifier" } ], "text": "National Insurance Payor Identifier", "value": "9800000348" } }, "name": [ { "family": "Colargol", "given": [ "Beertje" ] }, "gender": "female" }
```



Druk op de blauwe “Send” knop en na korte tijd krijg je onderin Postman het antwoord van je FHIR-server te zien; Beertje Colargol is ingevoerd in je Fhir-server.

Verlaat Postman, start je browser, ga naar <ip-adres van je Pi>:8080, kies voor “local tester”, klik op de Patient-resource, zoek op naam (of family-name, dat mag ook) en dan op “Colargol”.

**Patient Resource** Server: Local Tester Source Code About

**Bundle**

- CapabilityStatement
- CarePlan
- CareTeam
- CatalogEntry
- ChargeItem
- ChargeItemDefinition
- Claim
- ClaimResponse
- ClinicalImpression
- CodeSystem
- Communication
- CommunicationRequest
- CompartmentDefinition
- Composition
- ConceptMap
- Condition
- Consent
- Contract
- Coverage
- CoverageEligibilityRequest
- CoverageEligibilityResponse
- DetectedIssue
- Device
- DeviceDefinition

**< Response** **✓ HTTP 200**

**Response Headers**

date: Tue, 22 Dec 2020 18:52:14 GMT  
 x-request-id: N4yDm2VVR13gm  
 last-modified: Tue, 22 Dec 2020 18:52:13 GMT  
 keep-alive: timeout=20  
 transfer-encoding: chunked  
 x-powered-by: HAPI FHIR 5.2.0 REST Server (FHIR Server; FHIR 4.0.1/NA)  
 connection: keep-alive  
 content-type: application/fhir+json;charset=utf-8

**Result Body** JSON bundle (1684 bytes)

Bundle contains 1 / 1 entries

ID	Updated
Patient/2_history/1	10:48:03

[Read](#) [Update](#)

**Raw Message**

```
{
  "resourceType": "Bundle",
  "id": "OpDm2c-8062-464c-8387-30c0faf31785",
  "meta": {
    "lastUpdated": "2020-12-22T18:52:13.589+00:00"
  },
  "type": "searchset",
  "total": 1,
  "link": [
    {
      "relation": "self",
      "url": "http://localhost:8080/fhir/Patient?pretty=true&name=Colargol"
    }
  ],
  "entry": [
    {
      "fullUrl": "http://localhost:8080/fhir/Patient/2",
      "resource": {
        "resourceType": "Patient",
        "id": "2",
        "meta": {
          "versionId": "1",
          "lastUpdated": "2020-12-22T18:48:03.205+00:00",
          "source": "HAPIFHIRvgt5X9RE5P"
        },
        "text": {
          "status": "generated",
          "div": "div>welcome=http://www.u3.org/1999/xhtml</div><div class='hapiHeaderText'>Colargol</div><div><table class='hapiPropertyTable'><tbody><tr><td>Identifier</td><td>CT5831</td></tr></tbody></table></div>"
        },
        "identifier": [
          {
            "type": "coding",
            "system": "http://terminology.hl7.org/CodeSystem/v2-0203",
            "code": "MR",
            "display": "Medical Record Number"
          },
          {
            "text": "Medical Record Number"
          },
          {
            "value": "CT5831"
          }
        ],
        "type": {
          "coding": [
            {
              "system": "http://terminology.hl7.org/CodeSystem/v2-0203",
              "code": "NIIP",
              "display": "National Insurance Payor Identifier"
            }
          ]
        }
      }
    }
  ]
}
```

Beertje Colargol is inderdaad toegevoegd aan je Fhir-database. Op dezelfde manier zou je haar ook in de database van <http://hapi.fhir.org> kunnen toevoegen!



Nou, en daarmee zijn we aan het eind gekomen van deze reeks over het installeren van een Fhir-server op de Raspberry Pi. Er is op Internet een ongelooflijke hoeveelheid materiaal te vinden over de Raspberry, over FHIR, over de HAPI Fhirserver, over REST en ga zo maar door, dus als je in deze richting verder wilt dan is er meer dan voldoende te vinden.

Ik zal van deze reeks van zeven afleveringen nog een (ingekorte) Engelstalige variant maken, die vooral gericht is op de installatie van de FHIR-server en ik ben van plan om ook nog te proberen om een andere FHIR-server (die van InterSystems) op de Pi te krijgen. Maar dat (en nog veel meer plannen) komt dan na de Kerst.

Fijne Kerstdagen, een gezond en gelukkig 2021 en veel succes en plezier!

Bob

