TOP-KT-012a - FHIR REST API Foutafhandeling

Versiegeschiedenis

Versie	Datum	Status	Wijzigingen
0.2.0	16 Jan 2023	concept	Herschreven op verzoek. • Individuele status codes verwijderd. • Meer referenties naar de FHIR API • Code ranges toegelicht.
0.1.0	19 Dec 2022	concept	Delete response code aangepast

Beschrijving

Bij de interactie met de FHIR REST API kunnen er op verschillende niveaus fouten ontstaan. Er kunnen zich problemen voordoen met de authenticatie, de autorisatie, met de input of met de status of beschikbaarheid van de resource. In al deze situaties moet de client van de FHIR REST API op de juiste manier geïnformeerd worden over de fout.

Overwegingen

Informatief vs. beveiliging

Over het algemeen geldt dat meer informatie in een foutmelding altijd beter is. In koppeltaal zien we dit niet anders. Echter, er is hier een belangrijke uitzondering op: en dat zijn de meldingen rondom beveiliging en/of het ontbreken van de juiste credentials. De (fout) meldingen van koppeltaal FHIR API zijn daarop in te delen in twee groepen; een groep van alles wat zich voordoet met een aanvraag (request) die niet geautoriseerd is en een groep van alle meldingen van een aanvraag (request) die wel geautoriseerd is. Om te voorkomen dat er informatie van de server en de gegeven wordt vrijgegeven aan onbevoegden geldt dat voor de eerste groep de meldingen minimaal en summier zijn. Voor de tweede groep, dus voor geautoriseerde aanvragen (requests), zijn de meldingen juist zo gedetailleerd en uitgebreid mogelijk.

Meerdere status codes zijn goed

De FHIR API definieert de HTTP status codes die voor de verschillende aanvragen door de API client verwacht kunnen worden. Het wordt van de FHIR resource service verwacht dat

deze van deze codes gebruik maakt. De FHIR resource service heeft uit de lijst van toegestane codes de keuze, het is de bedoeling de code die het meest van toepassing is te gebruiken. Dit heeft als gevolg dat het voorkomt dat, voor dezelfde fout bij verschillende FHIR resource services andere foutcodes worden gegeven. We leggend dit uit met een aantal voorbeelden.

Foute Accept Header: 400 of 415.

Een onbekende waarde in de accept header kan met twee verschillende http status codes worden aangegeven. Een 415, Media-type niet ondersteund wordt typisch gegeven als er het MIME-Type in de header wel herkend wordt, maar niet wordt ondersteund, bijvoorbeeld `application/pdf`. De 400, Onjuiste aanvraag wordt gegeven als de MIME-Type helemaal niet wordt herkend, dus bijvoorbeeld `application/vnd.lotus-wordpro`. Of een MIME-Type wordt herkend als zodanig hangt af van de implementatie van de MIME-Types in de FHIR resource service. Zowel een 400 als een 415 zijn in deze situatie correct.

If-Match header probleem: 409 of 412

Een If-Match header die niet overeenkomt met de verwachtte waarde duidt op een FHIR resource die in de tussentijd is aangepast. In deze situatie moet de client van de FHIR API op de hoogte gesteld worden van het feit dat de aanpassing die deze op de FHIR resource probeert uit te voeren op een verouderde versie van de resource gebeurt. Afhankelijk van die implementatie van de locking van de FHIR resource service kunnen er twee codes worden verzonden in dit geval. Maakt de FHIR resource service gebruik van pessimistisch locking; geeft deze een code 409 terug, maakt de FHIR resource service gebruik van optimistische locking, geeft deze een 412 terug. Zowel de status code 409 als de status code 412 zijn correct.

Als client van de FHIR API is het af te raden afhankelijk te zijn van specifieke error codes. Het is dan ook sterk af te raden code of tests te schrijven die van specifieke status codes

afhankelijk zijn, eenvoudigweg omdat deze per FHIR resource service implementatie of configuratie kunnen verschillen.

Toepassing en restricties

De FHIR API

De FHIR resource service implementeert de FHIR API en volgt zoveel mogelijk de HTTP status codes zoals in deze API worden vastgelegd. Primair verwijzen we naar de documentatie van FHIR API. Een overzicht van welke status codes kunnen voorkomen wordt in onderdeel 3.1.0.17 van de FHIR API documentatie uiteengezet.

Binnen koppeltaal maken we duidelijk onderscheid tussen de type fouten die kunnen voorkomen. In het geval van autorisatie fouten moet de FHIR resource service terughoudend zijn met het geven van informatie. In de meldingen van fouten die zich voordoen na de autorisatie kan de FHIR service uitgebreider zijn in de meldingen. Het is daarom van belang eerst te controleren op autorisatie/authenticatie en daarna de resource te controleren op fouten.

Code ranges

De FHIR api geeft in het response een status code mee. Deze status codes staan niet vast, de FHIR resource service mag tot bepaalde hoogte zelf bepalen welke code in welke situatie van toepassing is. Het is dan ook te adviseren de status code van het antwoord van de FHIR resource service op range niveau te checken, de uitzondering hierop zijn de 401 en 403 status codes. De volgende ranges zijn van toepassing:

- 2xx: De 200 range kan worden beschouwd als een succesvolle status code. De FHIR resource service kan in bepaalde situaties een 201 terugsturen als een nieuwe entiteit is aangemaakt.
- 3xx: De 300 range zal niet direct van de FHIR resource service verwacht worden. In een infrastructuur kan het voorkomen dat een 301 of 302 gebruikt wordt na verhuizingen of om verkeer naar een https endpoint te upgraden.
- 4xx: deze range komen fouten voor waarvan de oorzaak bij de aanvrager ligt. De 401 en 403 codes worden gebruikt voor de autorisatie- en authenticatiefouten en liggen vast. Hoe en wanneer de andere fouten kunnen voorkomen hangt sterk af van de actie die wordt ondernomen.
- 5xx: deze range zou bij normaal gebruik en een stabiele FHIR resource service niet mogen voorkomen. Deze fouten hebben betrekking op de onverwachte fouten die in de FHIR resource service of de omliggende infrastructuur voorkomen.

Autorisatie fouten.

Bij autorisatie fouten moet de dienstverlener (server) *zo min mogelijk informatie weggeven.* Er valt onderscheid te maken tussen twee aspecten die invloed hebben op het verlenen van toegang:

1. Authenticatie: het vaststellen van de identiteit van de aanvrager. Deze doet zich voor als de identiteit door middel van de client_assertion niet kan worden vastgesteld. Als dit zich voordoet dient de service een `401 Unauthorized` terug te sturen.
2. Autorisatie: het matchen van de rechten van de aanvrager met de actie die deze uitvoert. In dit geval is de identiteit van de aanvrager vastgesteld, maar heeft de aanvrager geen recht om de gevraagde actie uit te voeren. In dit geval dient de service een `403 Forbidden` terug te sturen.

De tabel hieronder geeft een overzicht van de situaties die zich kunnen voordoen.

Autorisatie	Authenticatie	HTTP Statuscode
x	x	401 Unauthorized
x	√	403 Forbidden

Overige fouten en status codes

De tabel hieronder geeft een niet uitputtend overzicht van de status codes (groen) en foutcodes (rood). Sommige rijen hebben meerdere status codes, omdat in dat geval meerdere status codes goed zijn. Deze tabel geeft een informatief overzicht en is niet normatief.

Actie	Beschrijving	Resource state	HTTP Statuscode
GET /Resource	Read all van type <Resource>	Nul of meer resources bestaan	200 OK
GET /Resource/<id>	Read van één resource	Resource bestaat	200 OK
GET /Resource/<id>	Read van één resource	Resource bestaat NIET	404 Not Found
GET /Resource/<id>	Read van één resource	Resource is soft-deleted	410 Gone
POST /Resource	Aanmaken van een resource	Resource is valide	200 OK 201 Created
POST /Resource	Aanmaken van een resource	Resource is NIET valide	422 Unprocessable Entity
PUT /Resource/<id>	Updaten van een resource	Resource is valide, bestaat en `If-Match` header bevat de laatste versie	200 OK
PUT /Resource/<id>	Updaten van een resource	Resource is NIET valide	422 Unprocessable Entity

PUT /Resource/<id>	Updaten van een resource	Resource bestaat NIET	404 Not Found
PUT /Resource/<id>	Updaten van een resource	Resource is valide, bestaat en `If-Match` header bevat NIET de laatste versie	409 Conflict 412 Precondition Failed
DELETE /Resource	Deleten van een resource	Resource bestaat	200 (met OperationOutcome) 204 (zonder OperationOutcome)
DELETE /Resource	Deleten van een resource	Resource bestaat NIET	404 Not Found

De OperationOutcome resource

FHIR definieert een **OperationOutcome** resource die gebruikt kan worden om specifieke gedetailleerde verwerkbare (fout) informatie over te brengen. De **OperationOutcome** kan worden geretourneerd met elke HTTP 4xx- of 5xx-reactie, maar dit is niet vereist - veel van deze fouten kunnen worden gegenereerd door een generieke server dat ten grondslag ligt aan een FHIR resource service.

Het is de bedoeling om, met uitzondering van de 401 en 403 status codes, de operation outcome zo informatief mogelijk te maken. Zo is het behulpzaam om bij validatiefouten zoveel mogelijk detail van het validatieprobleem mee te geven.

Eisen

ERR - Eisen (en aanbevelingen) voor foutafhandeling

Voorbeelden

De **OperationOutcome** resource bevat minimaal de volgende velden, beginnend met minimaal 1 issue en een severity die het type response aangeeft: fatal (fataal), error (fout) , warning (waarschuwing) of information (informatie)

OperationOutcome


```
{
   "resourceType": "OperationOutcome",
   "issue": [
       {
           "severity": "error",
           "code": "processing",
           "diagnostics": "Invalid request: The FHIR endpoint on this server does not know how to handle GET operation[Patient] with parameters [[wrong_parameter]]",
           "expression": ["Patient.identifier[2].value"]
       }
   ]
}
```


Links naar gerelateerde onderwerpen

Links over status codes

Aanvullende informatie over het gebruik van HTTP Statuscodes in REST APIs is te vinden op:

- HL7 status codes
- HTTP Status Codes (restfulapi.net)
- HTTP Status Codes (restapitutorial.com)
- https://nl.wikipedia.org/wiki/Lijst_van_HTTP-statuscodes