

74. Qlip-RFC: foutcodes

16okt23: Van: Floris Ruiterkamp <Ruiterkamp@qlip.nl>

Aan: 'Conny Graumans' <c.graumans@chello.nl>; 'Beuker, E. (Esther)'

<Esther.Beuker@frieslandcampina.com>

CC: Renze van de Kuilen <renze@join-data.nl>; 'bob.deruiter@frieslandcampina.com'

<bob.deruiter@frieslandcampina.com>; Wesley van de Kamp <kamp@qlip.nl>; 'Ralf Alderlieste' <ralf@join-data.nl>

Onderwerp: Voorstel uniforme foutcodes RFC en Qlip reactie Bob en Florisv2.xlsx

Hallo Conny en Esther.

Het uniformeringsvraagstuk is in 2 delen verdeeld.

- 1 stuk Rfc en Qlip
- 1 stuk door JD

Stukje context: error bericht opbouw JD

Het antwoord van JD richting de data opvrager is als volgt opgebouwd

Antwoord JD

Data afkomstig van bron 1, of error afkomstig van bron 1

Data afkomstig van bron n, of error afkomstig van bron N

/Antwoord JD + http CODE JD (samengesteld over alle bronnen heen).

De *http CODE JD (samengesteld over alle bronnen heen)*, wordt als volgt bepaald.

- Als alle bronnen antwoord geven en er zijn geen gevulde "error" elementen à http 200
- Als 1 van de bronnen een gevulde error element heeft à http 500, waarbij de dataopvrager dan dus **in de error** elementen moet kijken wat er aan de hand is.

In de bijlage is een voorstel gedaan hoe RFC en Qlip de fouten uniform gaan terug geven. JD moet nog een voorstel in leveren hoe de http errors van QLIP en RFC gemapt gaan worden naar de "JD http-code"

Voorbeeld:

Vraag 1

- data bron 1 geeft antwoord zonder fouten (http_code_bron_1=200)
- ➔ *http CODE JD = 200*

Vraag 2

- data bron 2 geeft timeout (http_code_bron_2=504)
- ➔ *http ERRORCODE JD = 500*

Vraag 3

- data bron 1 geeft antwoord zonder fouten (http_code_bron_1=200)
- data bron 2 geeft unautorised (http_code_bron_2=401)
- ➔ *http ERRORCODE JD = 500*

Van JD kregen we onderstaande tabel: Als er een meta object is hebben we dus een attribuut message en http-code. Zie verder de toelichting in de onderstaande tabel. Niet alle gevallen hebben op dit moment een meta object binnen de error message, misschien moeten we daar nog eens verder naar kijken als JoinData zijnde.

Status code bron / situatie	Error ID JoinData	Error meta http_code	Error meta message	Toelichting
Timeout	source_data_timeout	504	timeout reached	Gateway timeout
0 (Service unavailable)	source_data_unreachable	503	couldn't connect to server	Service unavailable
Timeout authentication server	source_auth_unreachable	N/A	N/A	Service unavailable, geen meta object
401 (Unauthorized)	source_auth_invalid_credentials	N/A	N/A	Zou betekenen dat JoinData credentials niet goed heeft staan. Eigenlijk zouden wij hier ook een 500 van moeten maken omdat een eindgebruiker er niets aan kan doen. Geen meta object
406 (Not Acceptable)	source_data_not_acceptable	500	Response body van bron	Wordt geprobeerd naar JSON te parsen
>= 400	source_data_request_failed	500	Response body van bron	Wordt geprobeerd naar JSON te parsen
Overig 200	internal_error Bron geeft een geaccepteerd antwoord. En er is geen error element ingevuld in de repons(es)	500	internal server error	

17okt23, uit de OC:

- Indien geen data beschikbaar is dan verdient het toch de voorkeur om een 200 code te rapporteren in plaats van een 204 c.q. 500 code. Nu wordt hiertoe nog een 500 code uitgeleverd, dat moet zo snel mogelijk worden afgebouwd. Bij een 500 code wordt de import namelijk direct onderbroken. RFC doet het al op de correcte wijze; voor Qlip betekent dit een change.
- **Floris** bereidt een email voor om deze change onder de aandacht te brengen van de data-afnemers (beschrijven van de change, aangeven per wanneer deze ingaat). **Renze** voorziet Floris van een lijst van data-opvragers voor wie deze change impact heeft.
- Renze stelt voor om in iedere JoinData response de identifier (het kvk) van de 'providing company' mee te geven zo kan de ontvangende partij zien of de dat van RFC of van één van de Qlip zuivelaars afkomstig is. Dit geldt voor alle (succesvolle en niet succesvolle) responses; dus niet uitsluitend voor error-meldingen.
Besluit: de vergadering is hiermee akkoord.
Renze levert Conny een voorbeeld van hoe de 'providing company id' meegegeven moet worden in de response.

27nov23, Renze aan Conny:

Ik zou je (issue 68) [red.: issue 68 heeft een vervolg gekregen in issue 74] nog een voorbeeld aanleveren over hoe de 'providing company id' meegegeven zou worden in onze response. Dit gaat er als volgt uit zien:

[

```
{
  "id": "cc734450-c654-40cd-827c-a2dbb911aca1",
  "providingCompany": {
    "scheme": "nl.kvk",
    "id": "12345678"
  },
  "meta": {
    "responseTime": 2.28
  },
  "errors": [],
  "data": [
    {
      "identifier": {
        "id": "NL 1111111111",
        "scheme": "nl-v1"
      }
    },
    {
      "identifier": {
        "id": "NL 2222222222",
        "scheme": "nl-v1"
      }
    }
  ]
}
```

27nov23: Van: Floris Ruiterkamp <Ruiterkamp@qlip.nl>

Aan: Graumans Conny Agro-connect <c.graumans@chello.nl>; Renze van de Kuilen JoinData <renze@join-data.nl>; Ralf Alderlieste <ralf@join-data.nl>

CC: Wesley van de Kamp <kamp@qlip.nl>; Beuker, E. (Esther) <esther.beuker@frieslandcampina.com>;

Ruiter, J.B. de (Bob) <bob.deruiter@frieslandcampina.com>

Onderwerp: Beschrijving uniforme foutcodes RFC en Qlip_def.xlsx

Hallo Conny, Hierbij de beschrijving van de uniforme fout-codes [red.:

20231127.Beschrijving_uniforme_foutcodes_RFC_en_Qlip_def.xlsx]. Ik heb er een uitleg aan toegevoegd.

@ Renze en Ralf nog 2 vragen: ik heb niet helder of de sheet nu het beeld geeft over hoe het nu werkt of hoe het moet gaan werken en op 1 regel in het gele blok staan 2 foutcodes: welke is de juiste?

Tot zover en groeten Floris

Omschrijving	HTTP-Code RFC en Qlip	HTTP-Code JD
Wanneer de zuivelfabrieken gewoon antwoord kunnen geven: zowel als er data is , maar ook als er geen data is.	200	200
wanneer meer verzoeken binnen komen dan " drempel waarde "	429	200
Als je geen token hebt om op te vragen	401	200
Als de parameters in de vraag niet begrepen worden (zoals fout in datum / onbekend locatie schema/ requestor niet gevuld, UBN niet gevuld/ negatief locatie nummer / datumleeg, ongeldige datum)		
(2023-10-11:BDR) FrieslandCampina controleert nu alleen de waardes achter het ?. Controle op de andere componenten in de API zouden dan nog gebouwd moeten worden, incl impact analyse op performance: Alternatief : RFC returns een 200 met een error message	400	200
Technisch probleem (payload te groot,storing esb, storing 3e partij, timeout, ongeldige transactiecode)	500	200
Als Wim een bericht type uit-zet (is echt een nood maatregel bij bijvoorbeeld performance issues van prod)	503	200
JoinData HTTP algemene foutcodes		
Ongeldig of verlopen token		403
		401
Niet meegeven van authentication header of niet meegeven van bearer token		403
Algemene fout		500
Company identificatie claim(s) niet gevonden in het token		400
Resource niet gevonden, onjuiste url e.d.		404

27nov23: Van: **Ralf Alderlieste** <ralf@join-data.nl>

Aan allen.

400/422

Wat betreft de 400/422 status codes die worden nu binnen het JoinData platform beide gebruikt, daarom heb ik ze beide vermeld.

Wellicht moeten we daar intern nog eens kritisch naar kijken of we dat nu zelf ook overal consistent toepassen.

Onderstaande geeft denk ik een goede uitleg over het verschil tussen beide. 422 zijn voornamelijk validatie fouten.



Yash Bajaj

Senior Software Engineer at Microsoft (company) (2023–present) · Author has **548** answers and **8.4M** answer views · 8y



Use 400 if the query is **syntactically** incorrect.

Use 422 if the query is **semantically** incorrect.

I'll show you a clean example :

Consider a GET REST call where you provide user_id as a parameter you hit the getDetails URL.

http://localhost:8080/getDetails?user_id=12345

- In this case if the details for the user_id 12345 exist then 200 OK status can be sent .
- If the details for user_id 12345 are not present .Return a 404 NOT FOUND status
- If you do not provide the user_id. That's syntactically wrong. You must return a 400 BAD REQUEST status
- If you provide the user_id but it's not valid i.e a -43.67 . The query is syntactically correct but semantically incorrect. Throw a 422 Unprocessable Entity status. This must be used for validation purposes.

401/403

Bij een ongeldig of verlopen token geven we nu een 403 terug en dat zou eigenlijk een 401 moeten zijn. Zal dit even met [@Renze van de Kuilen](#) kortsluiten/bekijken.

Hoop dat het zo duidelijk is, als er nog vragen zijn dan stel ze gerust.

Status 28nov23: In behandeling.