

## Memo

### Aan

Bart Thonus, Dirk Schwanenberg, Flip Dirksen, Geri Wolters, Jackie Leng, Jan-Maarten Verbree, Jeroen Gerrits, Sander Loos.

**cc:** overige DD-Werkgroepleden

<b>Datum</b>	<b>Kenmerk</b>	<b>Aantal pagina's</b>
10 oktober 2019	11203680-DSC-22-v0.9	6
<b>Van</b>	<b>Doorkiesnummer</b>	<b>E-mail</b>
Stef Hummel	+31(0)6 1019 8112	Stef.Hummel@deltares.nl

### Onderwerp

Besprekingsverslag DD-API-werkgroep, sessie 12 september, web conference

### Aanwezig:

- zie 'Aan' en 'Van'

### Voorstelrondje

We verwelkomen twee nieuwe werkgroep-leden, Dirk en Jackie van Kisters. Iedereen stelt zich voor en schetst zijn achtergrond.

### ***DD-API als Pas-Toe-Of-Leg-Uit kandidaat (PTOLU)***

Eind november willen we de DD-API PTOLU-kandidaat aanbieden aan de CCvD-D. In feite is dit een proefkandidaatsstelling: het is voor het eerst dat de CCvD-D een API behandelt, dus men wil kijken wat daar bij komt kijken (b.v. meer dan bij een datamodel).

We moeten daartoe het volgende op orde hebben:

- De documentatie (specificaties, documentatie, en voorbeelden)  
Zie het volgende agendapunt.
- De organisatie (stuurgroep, werkgroep, architecture board)

M.b.t. de organisatie:

De **werkgroep** is goed bezet.

De **stuurgroep** bestaat momenteel uit de directeuren van de drie marktpartijen en de DD-programmadirecteur van RWS. Dit zou wat breder moeten worden. Voorkeur:

- Twee marktpartijen; dit kunnen de partijen van het eerste uur zijn (Nelen&Schuurman, HydroLogic), maar er het mag ook ingevuld worden door een nieuwe marktpartij.  
**Actie: Bart** vraagt na of HKV daar belangstelling voor zou hebben.
- Vertegenwoordiger vanuit RWS (**Actie: Stef** vraagt RWS wie dat volgend jaar is)
- Vertegenwoordiger vanuit een waterschap.  
Deze vertegenwoordiging ontbreekt momenteel.  
**Actie: Stef** vraagt bij Delfland en HHNK, en vraagt aan Koos Boersma en Michel Arendsen (zie 'architecture board' hieronder) of zij suggesties hebben.  
Idealiter is dit b.v. het hoofd van een inhoudelijke afdeling (waterbeheer).  
Deze persoon moet niet alleen de DD-API promoten, maar ook het mandaat hebben om financiering te regelen.

**Datum**  
10 oktober 2019

**Ons kenmerk**  
11203680-DSC-22-v0.9

**Pagina**  
2/6

Aan de invulling van de **architecture board** wordt momenteel gewerkt. Beschikbaar zijn Koos Boorsma, informatie-architect bij IHW, en Michel Arendsen, specialist monitoring bij Waterschap Rijn en IJssel.

Daarnaast zien we Geri graag in de Architecture Board. Hij is momenteel nog werkgroep-lid, maar dat is eigenlijk niet meer terecht, omdat hij niet meer bezig is met de DD-API (althans: niet inhoudelijk; wel ondersteunend bij het organiseren/genereren van de documentatie). Zijn brede technische kennis is een aanvulling binnen de board.

Daarnaast zou het goed zijn als er nog een tweede waterschapper in de board zou zitten. **Actie:** **Stef** vraagt Michel Arendsen of hij suggesties heeft.

## **Documentatie (mede n.a.v. het vorige punt)**

Tijdens de vorige bijeenkomst is besloten dat de RAML-specificaties van DD-API en DD-OPER uit elkaar moeten worden getrokken, om onduidelijkheden te voorkomen, met name in het wel of niet verplicht zijn van bepaalde attributen (sommige in DD-API wel en in DD-OPER niet, sommige juist andersom).

Deze actie is ondertussen uitgevoerd. Het blijkt programmeurs van DD-API 2.0 inderdaad te hebben geholpen.

Toch moet er nog het nodige aan de documentatie gebeuren. Het gaat dan met name om het:

- toevoegen van meer voorbeelden
- verduidelijken van de betekenis van query-parameters (in de url) en attributen (in de response)
- het verbeteren en uitbreiden van de beschrijvende teksten op [www.digitaledelta.org](http://www.digitaledelta.org)

Stef gaat dit half oktober doen, en zal het resultaat ter review aanbieden aan de werkgroep.

*Opmerkingen ten tijde van het maken van het verslag:*

- *Ik heb ondertussen overleg gehad met de leden van de Architecture Board. Zij zullen de documentatie in detail reviewen.*
- *Jurgen was geen voorstander van het scheiden van de documentatie van DD-API en DD-OPER, omdat het risico bestaat dat de response uit elkaar gelopen.*

*In een overleg hebben Jurgen en ik geconstateerd dat scheiden desalniettemin de beste oplossing is. Consistentie kan worden geborgd door een script te maken dat checkt de delen van de response die gelijk moeten blijven na wijzingen nog steeds gelijk zijn.*

## **DD-OPER vs DD-API bij RWS**

Nu de DD bij RWS gebruikt wordt is ook bij anderen dan de RWS/MKK-DL-ontwikkelaars duidelijk geworden dat er verschillen zijn in de implementatie van de API's.

Het probleem is dat er in de [response van Matroos](#) per event een "aspects"-niveau en een "points" niveau aanwezig is. Dit wijkt af van alle andere implementaties, inclusief die van RWS's [Aqualarm](#). Reden voor deze tussenniveaus is dat er dan aan de ontvangstkant geen onderscheid hoeft te worden gemaakt tussen echte "aspectgroep"- en "multipoint"-data en de reguliere één quantity op één punt series; laatstgenoemde worden dan gezien als een aspectgroep met één aspect, een een multipoint-locatie met maar één punt.

Stef gaat hier 's middags overleg over voeren met RWS, en er op aandringen dat deze onnodige tussenniveaus' verdwijnen.

*Opmerkingen ten tijde van het maken van het verslag:*

*Ik heb hier ondertussen met RWS drie besprekingen gehad. De eerste was ik meteen 's middags na ons overleg. Op dat moment was de insteek van RWS:*

- *Intern, en in de puur operationele systemen, een respons met de genoemde tussenlagen.*
- *Extern een response zonder die tussen lagen, conform de bestaande DD-nodes.*

*Tijdens vervolgbesprekingen werd geconstateerd dat het toch wel vervelend is als een toekomstig consumerend systeem twee verschillende soorten response krijgt. Daarom is besloten om te kijken*

*wat er moet gebeuren om het bij nader inzien bij RWS toch te gelijk te maken aan de manier waarop de andere systemen het doen: geen aspecten-zet en geen multipoint-respons als het gaat om een gewone scalaire grootheid op een gewone locatie.  
Jurgen en ik zijn begonnen uit het uitwerken van de consequents voor deze wijziging binnen de MKK-DL; een analyse die eind oktober moet zijn afgerond.*

## **Locations filteren op grootheid**

VORtech en Deltares zijn viewers aan het bouwen, en onafhankelijk van elkaar leidde dat tot de vraag: is het niet mogelijk om een lijst te krijgen van de locaties waar series van een bepaalde grootheid aanwezig zijn.

Momenteel moet dat door in het /timeseries end point te filteren, en uit die response (lijst met de metadata van timeseries) de locaties te destilleren. Het zou handiger zijn de filtering aan het /locations end point en (met name) het /locations/geojson end point te kunnen meegeven, zodat men een geojson feature collection terugkrijgt die middels één statement in de viewer te visualiseren is.

Deze filtering op twee manieren toegevoegd worden:

- Door gebruikmaking van het generieke filtering concept, dat reeds onderdeel van de specificaties is:  
/locations?quantityFilter=quantityId:eq:waterlevel
- Door de specs (versie 2.1) uit te breiden met een filter-parameter:  
/locations?quantityId=waterlevel

Gezien de hiervoor nu reeds aanwezige vraag heeft de tweede optie de voorkeur van de werkgroep. De gewenste functionaliteit wordt daarmee heel expliciet gemaakt.

**Actie:** **Stef** stelt hiervoor op github een change request op.

## **Gebruik van CORS-headers**

Bij bovengenoemde viewers (maar ook vorig jaar al toen we bezig waren met de authenticatie-pilot en Lizard moest worden bevraagd vanuit een demo-viewer) liep men er tegenaan de viewers niet rechtstreeks de API mogen benaderen.

Dit kan opgelost worden d.m.v. CORS-headers. Zie bijlage A voor achtergrond-informatie over CORS-headers.

Vraag aan de werkgroep is of we het gebruik van CORS-headers kunnen formaliseren.

Na enig overleg is de conclusie dat de huidige deelnemers daar te weinig kennis voor hebben.

We besluiten om de groep weer bij elkaar te roepen die de authenticatie/autorisatie heeft uitgewerkt.

**Actie:** **Stef** organiseert meeting van de autorisatie-groep.

Deelnemers: Casper (Nelen & Schuurmans), Bob of Bram (Hydrologic), Geri (EcoSys), Erik (Deltares), Werner (VORtech), Jeroen Grave (RWS).

Bart (HKV) en Dirk/Jackie (Kisters) reviewen het resultaat.

## **Afwijkingen DD-API van landelijke API-strategie**

Opvolging van het actiepunt van de vorige vergadering.

Uit de door Geri gemaakte analyse blijkt dat de API-richtlijnen waaraan de DD-API niet (volledig) voldoet:

- óf uitlegbaar zijn
- óf over authenticatie gaan (met name het gebruik van API-keys)

Besloten wordt om het omgaan met de authenticatie-issues uit te laten werken door de authenticatiegroep (zie vorig punt).

## **Status implementaties DD-API 2.0**

Overzicht van het beschikbaar komen/zijn van versie 2.0:

- *Nelen en Schuurmans (Lizard)*  
Eind oktober. Open data van een aantal instanties.
- *HydroLogic (HydroNet)*  
Ontwikkeling afgerond; voor de volledigheid nog verifiëren tegen verbeterde versie van de documentatie.  
Nog geen keuze over waar in te zetten; hangt af van opdrachtgevers.
- *HKV*  
Implementatie in Continu Inzicht. Data beschikbaar voor betrokken partijen; niet openbaar.
- *Kisters (WISKI)*  
Implementatie afgerond, draait op demo-databron. Binnenkort feedback van de waterschappen.  
Daarna implementatie bij de waterschappen. Waarschijnlijk open data van één van die waterschappen.
- *RWS*  
Implementatie op een aantal databronnen:  
LMW en ODS, beide DD-OPER, momenteel alleen intern.  
Matroos: DD-OPER (intern) en DD-API (extern). Deel van de data uit Matroos zal openbaar beschikbaar komen.
- *Deltares (FEWS)*  
Draait op Glossis, momenteel intern richting (openbare) viewers; op termijn openbaar.

## **Demo DD-ECO-query-builder**

Wordt vanwege tijdgebrek verplaatst naar de volgende bijeenkomst (en komt dan niet als laatste item op de agenda, maar als een van de eerste).

## ***Bijlage A Achtergrond-info bij gebruik van CORS-headers***

### ***Van Geri:***

CORS staat voor Cross-Origin Resource Sharing. Hieronder een korte beschrijving, zonder (te ver) in de techniek te duiken.

CORS is een mechanisme ingebouwd in moderne browsers om er voor te zorgen dat het een stuk moeilijker is om bepaalde aanvallen (Cross Site Forgery Requests) uit te voeren op de browser .

De browser stuurt een CORS request uit naar de server en verwacht een bepaald antwoord terug. Is het antwoord van de server onjuist, dan zal de browser weigeren het inhoudelijke verzoek naar de server uit te sturen.

Same-origin requests (waarbij de server en de browser in hetzelfde domein staan) hebben hier geen last van.

### **CORS heeft de volgende impact op de DD-API:**

De definitie van de huidige versie van de DD-API specificeert alleen de GET-method. Dat zou geen CORS-check moeten activeren. Zonder CORS aan de server-kant zou dit dus geen probleem moeten veroorzaken.

URLs zijn echter beperkt in lengte, afhankelijk van zowel browser als server, waardoor het geen optie is om grote hoeveelheden data als parameters mee te geven. Soms is dat echter wel nodig. Een voorbeeld is de DD-ECO-API waar soms geografische filters van meer dan 50MB gebruikt worden. Dat kan niet via een GET, maar moet via een POST.

In dat geval geldt dat er wél een CORS-check wordt gestuurd, omdat de data in JSON formaat wordt verzonden. POSTs die data via plain/text of via form-data sturen, zijn uitgesloten van een CORS-check (al zijn daar weer uitzonderingen op),

### **Dus moeten we CORS ondersteunen?**

Doet je implementatie geen POST, dan is het antwoord 'nee'. Doe je POST, en laat je browsers buiten je organisatie de API benaderen, dan is het antwoord 'ja'.

Doe je authenticatie, dan is het antwoord 'ja'.

### **Gevolgen van het ondersteunen van CORS**

Op zichzelf is het niet zo'n probleem om CORS te ondersteunen. De meest webserver-frameworks hebben standaard implementatie hiervoor.

Er zijn echter wel restricties: je moet namelijk aan de server-kant aangeven welke domeinen of IP adressen toegang hebben tot de service. Dit zal in de meeste gevallen in configuratie aan de server kant geregeld worden.

Echter, de configuratie kan behoorlijk groeien naarmate er domeinen moeten worden toegevoegd. Je zou geneigd zijn om maar alles open te zetten (allowedHosts op '\*'). Dat kan (al wordt het afgeraden), maar niet in combinatie met allowCredentials, m.a.w. dus niet als je authenticatie gebruikt.

***Van Werner Kramer (VORtech-collega van Jeroen): Z.O.Z.***

**Van Werner:**

Alle moderne browsers houden zich aan de same-origin-policy:

[https://developer.mozilla.org/en-US/docs/Web/Security/Same-origin\\_policy](https://developer.mozilla.org/en-US/docs/Web/Security/Same-origin_policy)

Om data van een andere origin dan de eigen open te stellen is CORS de standaard methode. Dit is van belang als vanaf een andere webpagina (bijv. viewer-a.com) een GET request wordt gedaan op de DD-API. In de request header wordt door de browser het volgende gezet:

Origin: viewer-a.com

Hierbij informatie over de CORS http headers.

<https://fetch.spec.whatwg.org/#http-cors-protocol>

<https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>

Zoals ik het interpreteer zijn alle DD-API requests "simple"

[https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS#Simple\\_requests](https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS#Simple_requests)

Hierdoor is het niet nodig om preflight CORS checks te ondersteunen. Dit zou in de toekomst kunnen wijzigingen als er bijv. ook POST methods in de DD-API komen waarbij er json data wordt meegestuurd. Mogelijk verandert dit ook bij het gebruik van autorisatie.

Het enige wat voor een "simple" CORS request nodig is om aan de http response header het volgende toe te voegen:

Access-Control-Allow-Origin: \*

De browser die het request heeft gedaan vergelijkt deze waarde met de eerder verzonden waarde Origin: viewer-a.com en vertrouwt de data alleen als deze overeenkomen of in dit geval als er een wild card wordt terug gegeven.

NB. Het is een security risk om --- in plaats van een wildcard --- de waarde uit Origin: viewer-a.com uit de request header als Access-Control-Allow-Origin: viewer-a.com in de response header terug te geven zonder deze te checken tegen een whitelist.