



Edu-V

Eenvoudige, veilige en betrouwbare
gegevensuitwisseling voor het onderwijs

Specificaties Koppelvlaktestfaciliteit

Edu-V

Afsprakenstelsel

Datum 22-07-2025

Inhoudsopgave

1	INLEIDING.....	3
2	KENMERKEN EN TRANSACTIEROLLEN.....	4
2.1	Kenmerken van de tool	4
2.2	Vier transactierollen.....	4
3	FUNCTIONALITEITEN.....	5
3.1	Beheermodule.....	5
3.2	Ingang voor leverancier (= toolgebruiker)	5
3.3	Rapportage module	6
3.4	Verzender test (Ontvanger wordt gesimuleerd).....	6
3.5	Ontvanger test (Verzender wordt gesimuleerd).....	7
3.6	Afnemer test (Bronstelsel wordt gesimuleerd)	8
3.7	Bronstelsel test (Afnemer wordt gesimuleerd)	8
3.8	Performance.....	8
3.9	OAuth/PKI test.....	8

1 Inleiding

Binnen het programma Edu-V maken scholen en leveranciers onderlinge afspraken die de eenvoudige, veilige en betrouwbare toegang tot en gebruik van digitale onderwijsmiddelen regelen voor het primair, voortgezet, speciaal en middelbaar beroepsonderwijs. Het [Edu-V Afsprakenstelsel](#) beschrijft de afspraken tussen publieke en private partijen over de uitwisseling van digitale gegevens tussen leer- en onderwijsmiddelen.

Leveranciers van systemen kunnen één of meerdere gegevensdiensten/API's realiseren die interoperabel zijn met systemen van andere leveranciers. Voor elke gegevensdienst kan een leverancier één of meerdere endpoints realiseren. Om als leverancier Deelnemer te kunnen worden van het Edu-V Afsprakenstelsel, moeten deze gegevensdiensten/API's voldoen aan de vastgestelde specificaties en eisen van het Edu-V Afsprakenstelsel.

Voor kwalificatie als Deelnemer moeten leveranciers een reeks testen succesvol doorlopen. Deze tests zijn:

- **Koppelvlaktest:** testen van de gegevensdienst(en)/API's **in isolatie**, met behulp van stubs en drivers.
- **Interoperabiliteitstest:** testen van de daadwerkelijke interactie met systemen van andere leveranciers.
- **Ketentest:** end-to-end test van (delen van) het gehele proces.
- **Productieacceptatietest:** laatste test vóór ingebruikname in de productieomgeving.

De **Koppelvlaktest** bestaat uit de twee onderdelen:

- Functioneel deel: de leverancier test zelfstandig de functionaliteit van elk endpoint van de gegevensdiensten/API's met behulp van stubs en drivers die de werking van het ontbrekende systeem van een andere leverancier simuleren.
- Niet functionele deel:
 - OAuth/PKI test: controle op correcte implementatie van authenticatie- en autorisatiemechanismen, inclusief het gebruik van PKI-certificaten en het correct afhandelen van tokenuitwisseling.
 - Performance test: beoordeling van de prestaties per endpoint, waarbij wordt getest op responsetijden.

Voor de Koppelvlaktest wil Edu-V een testvoorziening aanbieden: de koppelvlaktestfaciliteit.

Dit document bevat de specificaties voor die koppelvlaktestfaciliteit, hierna te noemen "tool", die functioneert als simulator. Leveranciers kunnen hun systeem hieraan koppelen om zowel het functionele als het niet-functionele deel van de Koppelvlaktest uit te voeren.

De specificaties van de Gegevensdiensten en API's zijn hier te vinden:

- [Edu-V gegevensdiensten - Afsprakenstelsel Edu-V - Confluence](#)
- [APIs - Afsprakenstelsel Edu-V - Confluence](#)

2 Kenmerken en transactierollen

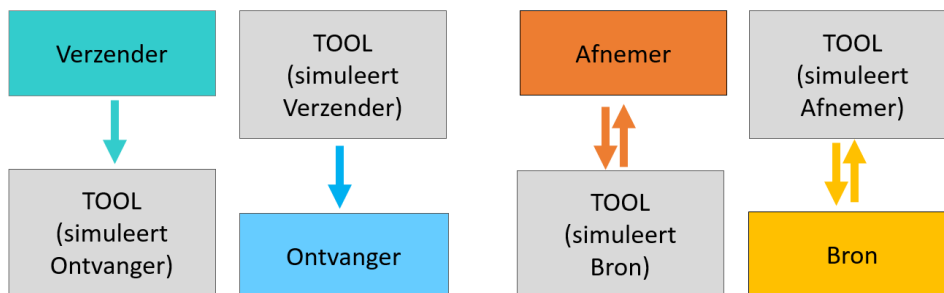
2.1 Kenmerken van de tool

De tool heeft de volgende kenmerken op hoofdlijnen:

- De tool is webbased
- De systemen van de leveranciers zijn eenvoudig te koppelen aan de tool.
- De tool fungeert als “stub” (nep-server) en “driver” (nep-client)
- De tool kan na afloop van de test testrapporten genereren
- De tool is gebruikersvriendelijk, heeft een goede performance, en nog nader te bepalen mate van andere kwaliteiten (zoals onderhoudbaarheid, schaalbaarheid, beveiliging, enz)
- De tool bevat een Beheermodule

2.2 Vier transactierollen

Per endpoint zijn er 4 transactierollen mogelijk. Dit zijn: Verzender, Ontvanger, Afnemer en Bron.



Verzender = systeem dat data pusht naar het ontvangende systeem

Ontvanger = systeem dat data ontvangt van het verzendende systeem

Afnemer = systeem dat data ophaalt bij een bronsysteem (de afnemer stuurt een request naar het Bronsysteem en krijgt een response terug).

Bronsysteem = systeem dat data terugstuurt op basis van de ontvangen request van afnemer.

De tool dient functionaliteiten te hebben op 4 niveaus:

- [Verzender test](#): de tool simuleert het ontvangende systeem.
- [Ontvanger test](#): de tool simuleert het verzendende systeem.
- [Afnemer test](#): de tool simuleert een gegevensaanbieder (een bronsysteem).
- [Bronsysteem test](#): de tool simuleert een gegevensafnemer (Afnemer).

3 Functionaliteiten

Dit hoofdstuk beschrijft in hoofdlijnen de functionaliteiten die de tool minimaal moet ondersteunen.

3.1 Beheermodule

De beheermodule moet minimaal de volgende functionaliteiten ondersteunen:

Gebruikersbeheer

- Mogelijkheid voor Admin rol om gebruikers aan te maken, te beheren, en specifieke rechten toe te kennen op basis van rollen (zoals Admin, Beheerder, Leverancier, enz.)

Specificatiebeheer van referentiecomponenten, gegevensdiensten en endpoints

- Beheren van de lijst met referentiecomponenten: Toevoegen, wijzigen, verwijderen van componentnamen.
- Beheren van de lijst met gegevensdiensten/API's en bijbehorende Endpoints: Toevoegen, wijzigen, verwijderen van namen gegevensdiensten/API's en endpoints.
- Beheren van de samenhang tussen referentiecomponenten, gegevensdiensten/API's, Endpoints en transactierollen.
- Voor elk endpoint moet de Admin de specificaties kunnen vastleggen:
 - Relevante velden / parameters (veldnamen)
 - Specificaties per veld (zoals verplicht/optioneel, datatype, toegestane waarden, enum waarden, validatieregels, etc)
 - Eventuele specifieke businesslogica tussen velden
- Ook deze specificaties moeten opgeslagen en beheerd kunnen worden in een database.
- Belangrijk: specificaties mogen niet hardcoded worden in de programmatuur, zodat de code niet onderhouden hoeft te worden bij wijzigingen aan de specificaties (van o.a. endpoints).
 - Dit stelt de tool in staat om de juiste verificaties uit te voeren.
 - Dit stelt de tool in staat om bij sommige transactierol testen standaard testcases automatisch te genereren op basis van deze kenmerken en regels (nice to have)
 - Op basis van de informatie over de endpoints moet de tool automatisch een scherm kunnen genereren waarmee de toolgebruiker de waarde van velden (van een door het eigen systeem te ontvangen JSON bestand) kan bewerken.

Testcase beheer

Beheren van testcases: toevoegen, wijzigen, verwijderen van testcases, voor de:

- Functionele test per endpoint per transactierol
- OAuth / PKI test voor de Client
- OAuth / PKI test voor de Server
- Performance test

Testdata beheer

- Mogelijkheid om testdata te beheren en eerdere gemaakte testcases (waaronder JSON bestanden) te hergebruiken bij het testen

3.2 Ingang voor leverancier (= toolgebruiker)

- Mogelijkheid voor toolgebruiker om in te loggen
- Mogelijkheid dat leverancier (toolgebruiker) contact- en productinformatie opgeeft.

- Dashboard informatie voor leverancier, waarin leverancier kan zien welke testen (bijvoorbeeld klaargezet/toegewezen door de Beheerder) gedaan moeten worden, wat de voortgang is, e.a.

3.3 Rapportage module

- Testrapporten kunnen automatisch worden gegenereerd. De toolgebruiker kan aangeven voor welke geteste endpoints een gedetailleerd rapport gewenst is. Het testrapport wordt als één PDF-bestand gegenereerd en bevat de testresultaten van de geselecteerde endpoints.
- Het rapport bevat de volgende onderdelen:
 - Samenvatting: het rapport opent met een overzichtelijke samenvatting van de belangrijkste bevindingen.
 - Exit-criteria: het rapport toont een "GO" indien er voldaan is aan de GO/NO GO criteria (nog vast te stellen). Eventueel kan de status ook visueel worden weergegeven met kleuren (groen = GO, oranje = aandachtspunten, rood = NO GO).
 - Details: onder de samenvatting volgt een gedetailleerd overzicht van de uitgevoerde testcases, inclusief:
 - De onderdelen waarop is getest;
 - De resultaten (Passed/Failed);
 - Eventuele bevindingen met foutmeldingen en waarschuwingen, inclusief ernstcategorieën.
- Testrapporten kunnen één van meerdere statuses hebben, zoals: Nieuw, Ingediend, Afgewezen, Goedgekeurd, etc.
- De toolgebruiker kan de status van een testrapport aanpassen. Een wijziging in status kan automatisch een actie of melding triggeren voor een andere toolgebruiker. Voorbeeld: wanneer een leverancier de status wijzigt naar Ingediend, wordt de Beheerder hiervan automatisch op de hoogte gesteld.

3.4 Verzender test (Ontvanger wordt gesimuleerd)

- De Verzender kan gekoppeld worden aan de tool om data (JSON bestanden) te verzenden.

1. Invoer:

- De tool kan de door de Verzender gestuurde data ontvangen.

2. Verificatie JSON:

- De tool vergelijkt de ontvangen data (JSON-bestanden) met de specificaties uit het afsprakenstelsel. Dit is onder andere:
 - Schema-validatie: Het controleren of het JSON-bestand voldoet aan het juiste format en structuur (JSON Schema Definition).
 - Business logica check, controleren of het JSON-bestand voldoet aan de functionele eisen en regels: of de juiste velden aanwezig zijn, de aanwezigheid van verplichte velden, of verplichte velden daadwerkelijk een waarde bevatten, of de waarden correcte / valide zijn, of het de juiste datatype is, veldlengte (?), e.d., afhankelijkheden tussen verschillende gegevens (semantisch, e.a.), e.a.

3. Simuleren:

- De tool geeft de juiste HTTP-statuscode en API-responsen (succesvolle terugmeldingen en foutmeldingen) terug
- Protocoltest: controleert of de gegevens veilig worden gestuurd (bijvoorbeeld via een versleutelde verbinding (zoals HTTPS)).

4. Output:

- De tool geeft een overzicht (samenvatting) van de uitgevoerde verificaties, met per JSON-bestand de aantallen checks, aantallen gevonden fouten / inconsistenties / afwijkingen. Dit kan worden opgedeeld in ernstcategorieën:
 - Blokkerende fouten. Dit zijn fouten die de testvoortgang blokkeren (bijv JSON voldoet niet aan de JSON Schema definition)
 - Kritieke fouten die moeten worden opgelost voordat de JSON verder kan worden gebruikt.
 - Waarschuwingen voor potentiële problemen die niet kritisch zijn, maar wel aandacht vereisen.
- De toolgebruiker kan aangeven om een gedetailleerd rapport te laten genereren (in PDF) van de uitgevoerde testcases. Zie 3.3.

5. Gebruiksvriendelijkheid:

- Het systeem van de leverancier is eenvoudig te koppelen aan de tool.
- De User Interface is gebruikersvriendelijk (intuïtief).
- De toolgebruiker moet in staat zijn om de verificaties meerdere keren te kunnen laten uitvoeren .

3.5 Ontvanger test (Verzender wordt gesimuleerd)

- De Ontvanger kan gekoppeld worden aan de tool om data (JSON bestanden) te ontvangen en hier op te reageren met de juiste terugmelding.

1. Invoer:

- De tool kan JSON-bestanden genereren alsof ze van de Verzender afkomstig zijn. Dit kan handmatig (op het scherm velden kiezen, waarden opgeven en/of selecteren of juist leeg laten) of geautomatiseerd gebeuren op basis van keuzes (zie Beheermodule 3.1) van bijvoorbeeld eerdere testen.
- Bij het handmatige kunnen toolgebruikers testcases instellen waarmee de tool verschillende data/JSON-bestanden genereert, bijvoorbeeld een standaard gegevensstroom, foutmeldingen, of specifieke situaties, die later getest kunnen worden. Deze mogelijkheden moeten in te stellen zijn in de Beheermodule.
- Ook moet de toolgebruiker erbij kunnen zeggen wat het verwachte resultaat zou moeten zijn (bijvoorbeeld goed, of foutmelding als de tool (in de transactierol van “Verzender”) foutieve data verzendt om de robuustheid van de Ontvanger te waarborgen.
- In de tool moet het mogelijk zijn om bestaande JSON-bestanden (die bijvoorbeeld de leverancier van de Verzender heeft laten genereren en correct waren) kunnen (selecteren en) versturen zodat de Ontvanger hiermee kan testen.

2. Verificatie JSON:

- De tool vergelijkt de JSON-bestanden met de specificaties uit het afsprakenstelsel. Dit is onder andere schema-validatie en business logica check (zie punt 2 bij Verzender test) zodat de toolgebruiker kan verifiëren of het Ontvangende systeem op de juiste wijze omgaat met de ontvangen data.
- Daar waar de tool zelf JSON-bestanden genereert, moet de toolgebruiker in staat zijn om JSON-bestanden aan te kunnen passen op bijvoorbeeld fouten. Dit zou ook kunnen middels

een scherm met velden die vanzelf gegenereerd wordt op basis van de informatie per endpoint in de beheermodule.

3. Simuleren:

- De toolgebruiker moet de JSON-bestanden kunnen verzenden naar het Ontvangende systeem om te controleren of het systeem de data van de tool correct verwerkt en de juiste acties uitvoert op basis van de ontvangen data, bijv correcte verwerking, terugmelding, foutmelding.

4. Output:

Zie punt 4 bij Verzender test

5. Gebruikersvriendelijkheid

Zie punt 5 bij Verzender test

3.6 Afnemer test (Bronstelsel wordt gesimuleerd)

- De Afnemer kan gekoppeld worden aan de tool om JSON bestanden te verzenden en te ontvangen.

De specificaties voor de Afnemer test bestaan uit het versturen van een request en het correct ontvangen van een response. Voor de request zijn de specificaties van Verzender test van toepassing en voor de response de specificaties van de Ontvanger test.

3.7 Bronstelsel test (Afnemer wordt gesimuleerd)

- Het bronsysteem kan gekoppeld worden aan de tool om JSON bestanden te ontvangen en daarop te reageren met de juiste actie.

De specificaties voor de Bron test bestaan uit het verzenden van een request naar de Bron vanuit de tool (zie specificaties Ontvanger test) en het controleren van de response van de Bron (zie specificaties Verzender test).

3.8 Performance

- Met de tool kan er getest worden op performance (o.a. responsetijden).

De performance eisen staan hier: <https://edu-v.atlassian.net/wiki/spaces/AFSPRAKENS/pages/53805057/Dienstenniveaus#4.-----Performance>

3.9 OAuth/PKI test

- De tool kan fungeren als Client en als Server voor het testen van de OAuth/PKI.

De specificaties zijn hier te vinden:

[M2M identificatie, authenticatie en autorisatie - Afsprakenstelsel Edu-V - Confluence](#)