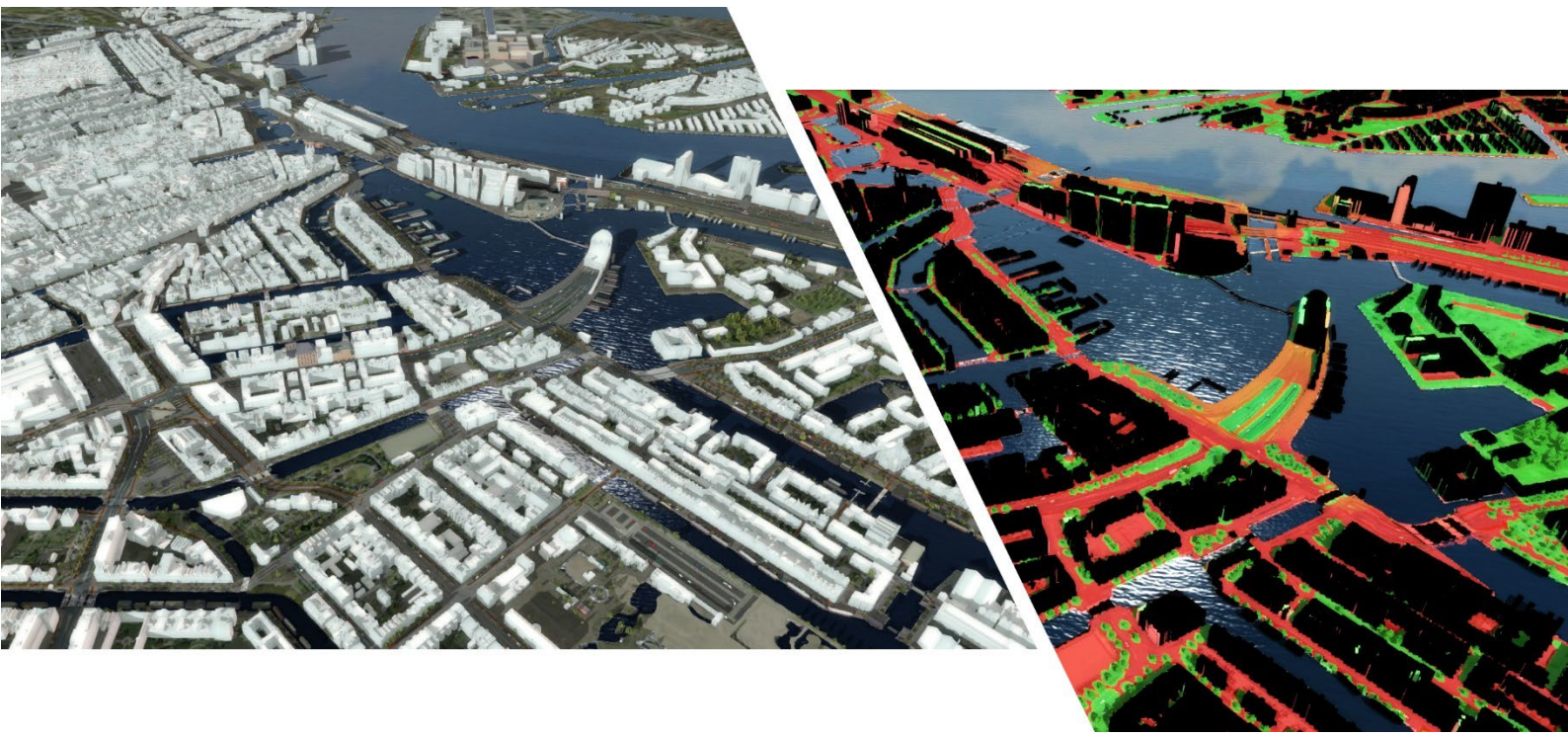
TYGRON PLATFORM

Accelerated Simulations for Engineers

Aan de slag met de API

CoP 2024, handout

Versie 4 november 2024



Inhoudsopgave

Webcalls naar de root API

Authenticatie

Authenticatie berust op http basic authentication, waarbij een “authorization” header aan requests wordt toegevoegd. De authenticatie die hierin wordt meegegeven is de gebruikersnaam en het wachtwoord, samengevoegd met een dubbele punt, base64 encoded. Bijvoorbeeld:

- Gebruikersnaam:
  - gebruiker@tygron.com
- Wachtwoord:
  - GeheimWachtwoord123
- Samengevoegd:
  - gebruiker@tygron.com:GeheimWachtwoord123
- Base64 encoded:
  - Z2VicnVpa2VyQHR5Z3Jvbi5jb206R2VoZWltV2FjaHR3b29yZDEyMw==
- Header:
  - Authorization: Basic Z2VicnVpa2VyQHR5Z3Jvbi5jb206R2VoZWltV2FjaHR3b29yZDEyMw==

Doel	Method	Url (en data)	Antwoord (en data)
Server		https://engine.tygron.com	
API	GET	/api/	HTML
Eigen gebruiker	GET	/api/myuser/	HTML
Eigen gebruiker	GET	/api/myuser/?f=JSON	JSON
Interacties omtrent account	GET	/api/event/user/	HTML
Interactie omtrent projecten	GET	/api/event/io/	HTML
Lijst met startbare projecten	POST	/api/event/io/get_domain_startable_projects/?f=JSON	JSON Data, waaronder fileName
Project sessie starten	POST	/api/event/io/start/?f=JSON [“EDITOR”, “demo_heat_stress”]	JSON Numeriek sessionId
Project sessie joinen	POST	/api/event/io/join/?f=JSON [sessionId, “EDITOR”]	JSON Data, waaronder apiToken

Toelichtingen:

In de lijst met startbare projecten vind je per project de “fileName”. Deze heb je nodig in de call voor het starten van Project Sessies.

Als voorbeeld voor het starten van een Project Sessie wordt gebruik gemaakt van de “demo_heat_stress”. Dit is de “fileName” van het respectievelijke Project.

Als voorbeeld voor het joinen van een Project Sessie wordt verwezen naar de “sessionId”. Deze moet vervangen worden met de verkregen informatie uit de sessie voor het starten van de Project Sessie.

Webcalls naar de session API

Authenticatie

Authenticatie berust op het meegeven van een api token als query parameter. Bijvoorbeeld:
https://engine.tygron.com/api/session/?token=1174ffe3jUF8Al3idygPxYjJrp0AvrAf
waarbij 1174ffe3jUF8Al3idygPxYjJrp0AvrAf de apiToken is.

Doel	Method	Url (en data)	Antwoord (en data)
Server		https://engine.tygron.com	
Session endpoint	GET	/api/session/?token=...	HTML
Lijst met alle datatypen	GET	/api/session/items/?token=...	HTML
Lijst met alle datatypen	GET	/api/session/items/?f=JSON&token=...	JSON
Lijst met alle Neighborhoods	GET	/api/session/items/neighborhoods/?f=JSON&token=...	JSON
Lijst met alle Overlays	GET	/api/session/items/overlays/?f=JSON&token=...	JSON
Lijst met interacties	GET	/api/session/event/?token=...	HTML
Lijst met Overlay interacties	GET	/api/session/event/editoroverlay?token=...	HTML
Info over interactie Overlay toevoegen	GET	/api/session/event/editoroverlay/add/?token=…	HTML
Overlay toevoegen	POST	/api/session/event/editoroverlay/add/?f=JSON&token=… [“TRAFFIC_NOISE”]	JSON overlayID
Testrun starten	POST	/api/session/event/editor/activate_testrun/?f=JSON&token=… [“true”, “true”]	No Content
Uitvoeren van een maatregel	POST	/api/session/event/participant/measure_plan_construction/?f=JSON&token=… [0, 2]	JSON logId
Afbeelding Overlay	GET	/web/overlay.png?id=57&token=…	PNG
Afbeelding Overlay (origineel)	GET	/web/overlay.png?id=57&maptype=CURRENT &token=…	PNG

Toelichting:

Bij het uitvoeren van een maatregel worden een ID van een Stakeholder meegegeven (de uitvoerende partij), en een ID van een Measure. Het verkrijgen van geldige Ids van deze datatypen kan op dezelfde manier als opvragen van data over Overlays.

Bij het downloaden van de afbeelding van de Overlay wordt de ID van de gewenste Overlay meegegeven als query parameter, niet als POST data payload.

Het downloaden van de afbeelding van de Overlay geschied niet via /api/ maar via /web/. Dit heeft te maken met de mogelijkheid om deze informatie ook beschikbaar te stellen voor web viewers zonder volwaardige editing rechten die toegang tot de volledige API zou bieden.

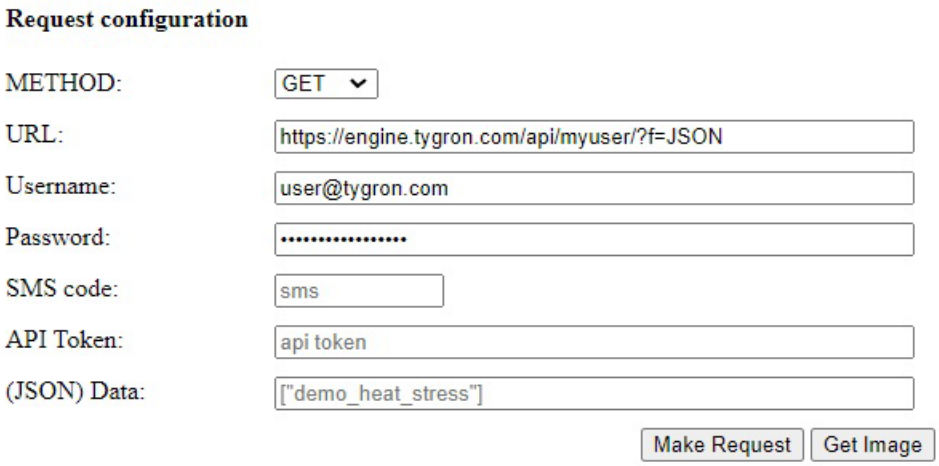
Voorbeelden in Tygron API test tool

Ten behoeve van de CoP stelt Tygron een API test tool beschikbaar. Deze draait volledig vanuit de browser en is beschikbaar op het volgende adres:

https://api.exp.tygron.com

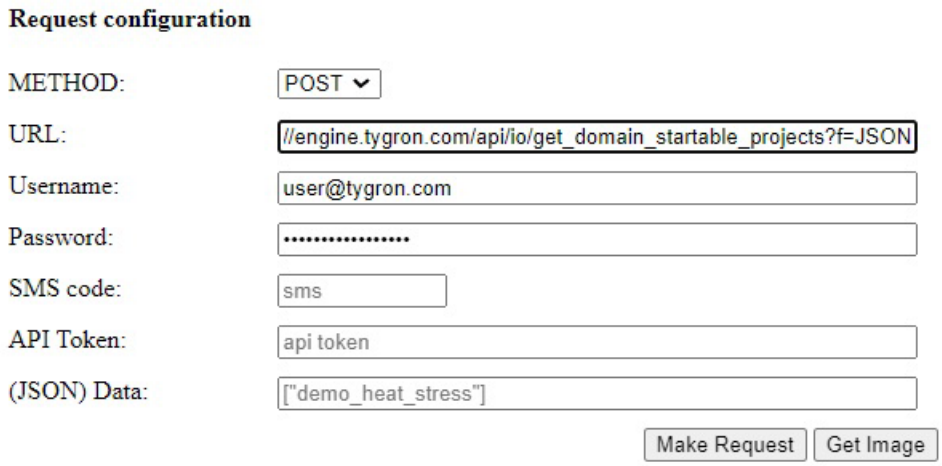
In deze tool kan ingevoerd worden naar welke URL een call gemaakt moet worden, met welke method, authenticatie, en data payload. Er wordt een voorbeeld weergegeven met de tekstuele inhoud van de request, en na het uitvoeren van de request wordt het resultaat weergegeven.

Root API (GET)


Request configuration
METHOD:
GET
URL:
https://engine.tygron.com/api/myuser/?f=JSON
Username:
user@tygron.com
Password:
SMS code:
sms
API Token:
api token
(JSON) Data:
["demo_heat_stress"]
Make Request
Get Image


Preview
GET /api/myuser/?f=JSON HTTP/1.1
Authorization: Basic dXNlckB0eWdyb24uY29tOlNlY3JldFBhc3N3b3JkMTIz
engine.tygron.com

Root API (POST) (zonder data)



Root API (POST) (met data)

Request configuration

METHOD: POST

URL: https://engine.tygron.com/api/io/start?f=JSON

Username: user@tygron.com

Password: ••••••••••••••••

SMS code: sms

API Token: api token

(JSON) Data: ["EDITOR","demo_heat_stress"]

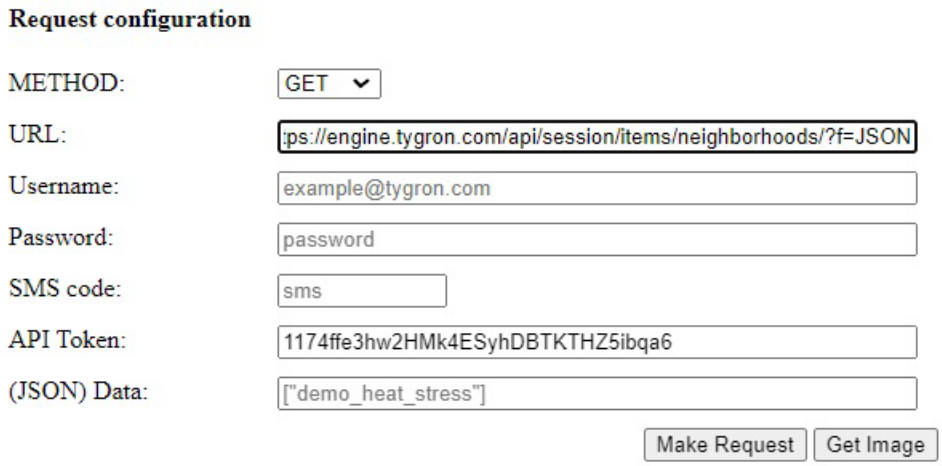
Make Request | Get Image

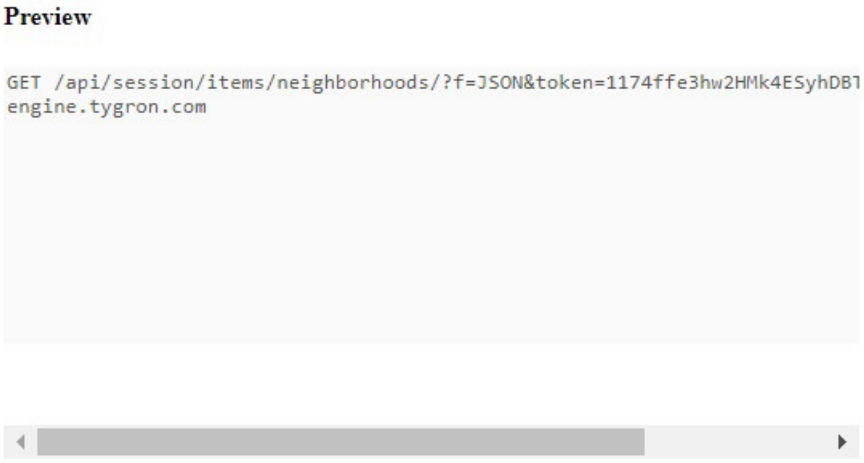
Preview

```
POST /api/io/start?f=JSON HTTP/1.1
Authorization: Basic dXNlckB0eWdyb24uY29tOlNlY3JldFBhc3N3b3JkMTIz
engine.tygron.com
Content-Type: application/json
Content-Length: 29

["EDITOR","demo_heat_stress"]
```

Session API (GET)





Session API (POST)

Request configuration

METHOD: POST

URL: https://engine.tygron.com/api/session/event/editoroverlay/add/?f=J

Username: example@tygron.com

Password: password

SMS code: sms

API Token: 1174ffe3hw2HMk4ESyhDBTKTHZ5ibqa6

(JSON) Data: ["TRAFFIC_NOISE"]

Make Request | Get Image

Preview

```
POST /api/session/event/editoroverlay/add/?f=JSON&token=1174ffe3hw2HMk4ES
HTTP/1.1
engine.tygron.com
Content-Type: application/json
Content-Length: 17

["TRAFFIC_NOISE"]
```

Voorbeelden in Python

Programmeer- en scripttalen kunnen gebruikt worden om http calls te maken naar web endpoints om interactie te automatiseren. Python is een taal die veel wordt gebruikt om dergelijke scripts op te stellen. Andere voorbeelden zijn R, Java, en PHP.

Deze voorbeeldscripts zijn functioneel en illustratief voor het opzetten van een eigen script in Python of een andere taal, om daarmee calls te maken naar het Tygron Platform. De variabele informatie staat bovenaan aangegeven.

De voorbeelden kunnen subtiel verschillen voor het klaarzetten van data en het uitlezen van de resultaten.

Root API (GET)

```
import base64, json, urllib.request

username='username'
password= 'password' #open("password.txt", "r").read()

url='https://engine.tygron.com/api/myuser/'
method='GET'
queryparams = ['f=JSON']

#URL
url +='?'+'&'.join(queryparams)

#HEADERS
encoded =
base64.b64encode((username+':'+password).encode('ASCII')).decode('ASCII')
http_basic_authentication = 'Basic '+encoded
headers = {'Authorization' : http_basic_authentication}

#MAKE REQUEST
request = urllib.request.Request( url, headers=headers, method=method );

try:
        response = urllib.request.urlopen(request, timeout=60)
except Exception as err:
        response = err

status = response.status
reason = response.reason
content = response.read().decode('utf-8')

response.close()

#PRINT RESULTS
print(str(status) + ' : ' + reason)
print(json.loads(content));
```

Root API (POST) (zonder data)

```
import base64, json, urllib.request

username='username'
password= 'password' #open("password.txt", "r").read()

url='https://engine.tygron.com/api/io/get_domain_startable_projects'

method='POST'
queryparams = ['f=JSON']
data=[]

#URL
url +='?'+'&'.join(queryparams)

#HEADERS
encoded =
base64.b64encode((username+':'+password).encode('ASCII')).decode('ASCII')
http_basic_authentication = 'Basic '+encoded
headers['Authorization'] = http_basic_authentication

#MAKE REQUEST
request = urllib.request.Request(
                url, headers=headers, method=method,
                data=json.dumps(data).encode('utf-8')
        );
try:
        response = urllib.request.urlopen(request, timeout=60)
except Exception as err:
        response = err

status = response.status
reason = response.reason
content = response.read().decode('utf-8')

response.close()

#PRINT RESULTS
print(str(status) + ' : ' + reason)
print(content)
```

Root API (POST) (met data)

```
import base64, json, urllib.request

username='username'
password= 'password' #open("password.txt", "r").read()

url='https://engine.tygron.com/api/event/io/start/'

method='POST'
queryparams = ['f=JSON']
data=['EDITOR', 'demo_heat_stress']

#URL
url +='?'+'&'.join(queryparams)

#HEADERS
encoded =
base64.b64encode((username+':'+password).encode('ASCII')).decode('ASCII')
http_basic_authentication = 'Basic '+encoded
headers = {
        'Authorization' : http_basic_authentication,
        'Content-Type' : 'application/json; charset=UTF-8'
    }

#MAKE REQUEST
request = urllib.request.Request(
        url, headers=headers, method=method,
        data=json.dumps(data).encode('utf-8')
    );

try:
    response = urllib.request.urlopen(request, timeout=60)
except Exception as err:
    response = err

status = response.status
reason = response.reason
content = response.read().decode('utf-8')

response.close()

#PRINT RESULTS
print(str(status) + ' : ' + reason)
print( content )
```

Session API (GET)

```
import base64, json, urllib.request

token='apitoken'

url=' https://engine.tygron.com/api/session/items/neighborhoods'

method='GET'
queryparams = ['f=JSON', 'token='+token]

#URL
url +='?'+'&'.join(queryparams)

#HEADERS
headers = {
		'Content-Type' : 'application/json; charset=UTF-8'
	}

#MAKE REQUEST
request = urllib.request.Request(
		url, headers=headers, method=method
	);

try:
	response = urllib.request.urlopen(request, timeout=60)
except Exception as err:
	response = err

status = response.status
reason = response.reason
content = response.read().decode('utf-8')

response.close()

#PRINT RESULTS
print(str(status) + ' : ' + reason)
print( content )
```

Session API (POST)

```
import base64, json, urllib.request

token='apitoken'

url=' https://engine.tygron.com/api/session/event/editoroverlay/add/ '

method='GET'
queryparams = ['f=JSON', 'token='+token]
data=['TRAFFIC_NOISE']

#URL
url +='?'+'&'.join(queryparams)

#HEADERS
headers = {
        'Content-Type' : 'application/json; charset=UTF-8'
    }

#MAKE REQUEST
request = urllib.request.Request(
        url, headers=headers, method=method,
        data=json.dumps(data).encode('utf-8')
    );

try:
    response = urllib.request.urlopen(request, timeout=60)
except Exception as err:
    response = err

status = response.status
reason = response.reason
content = response.read().decode('utf-8')

response.close()

#PRINT RESULTS
print(str(status) + ' : ' + reason)
print( content )
```

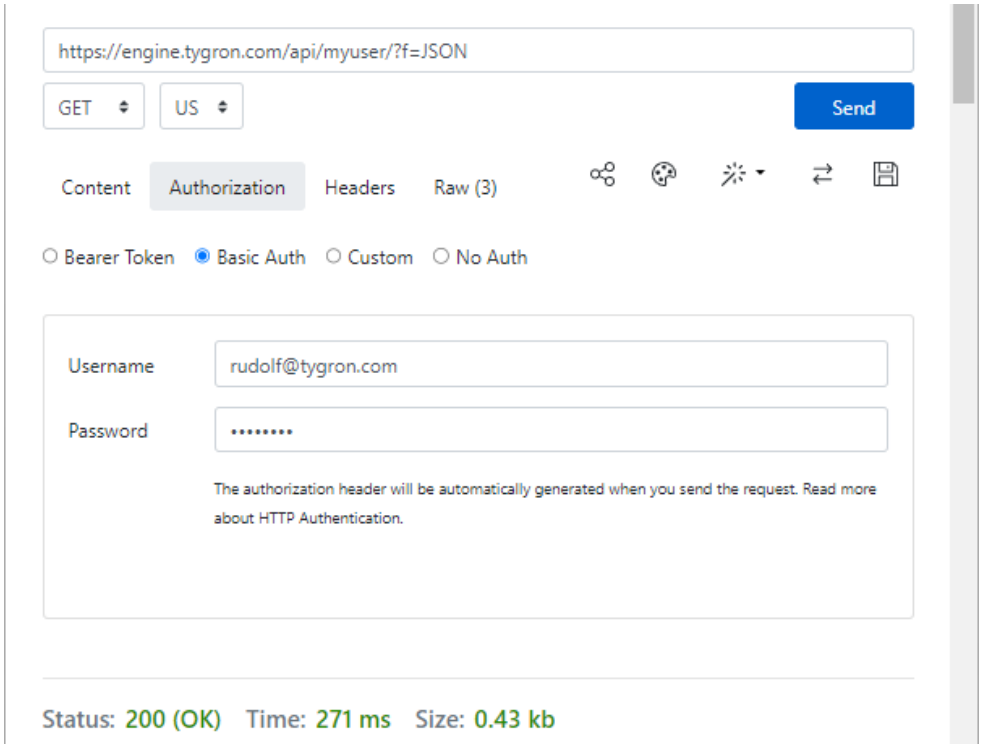
Voorbeelden in Reqbin

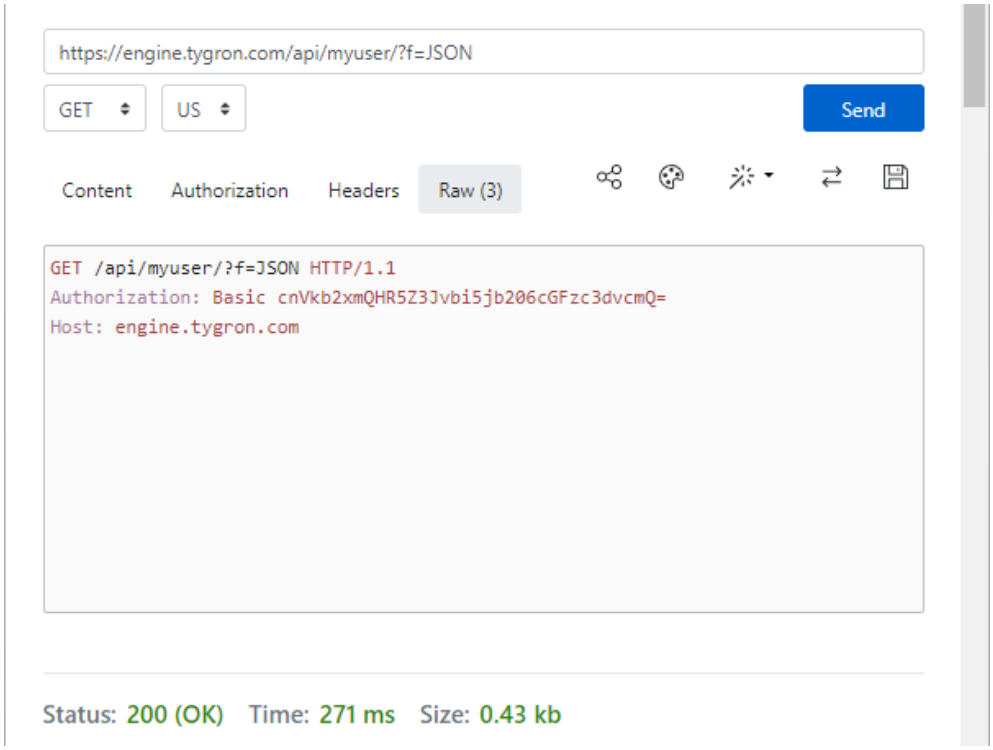
Er bestaan tools om http calls te maken naar web endpoints teneinde te testen hoe zo'n call opgebouwd moet worden en wat er van terug komt. Reqbin is zo'n tool. Een ander voorbeeld zijn bijvoorbeeld Postman.

Deze voorbeelden zijn screenshots, illustratief voor instellingen die in Reqbin opgezet kunnen worden om interacties met het Tygron Platform mogelijk te maken. De tool zelf behelst de logica, en de variabele informatie zijn de instellingen.

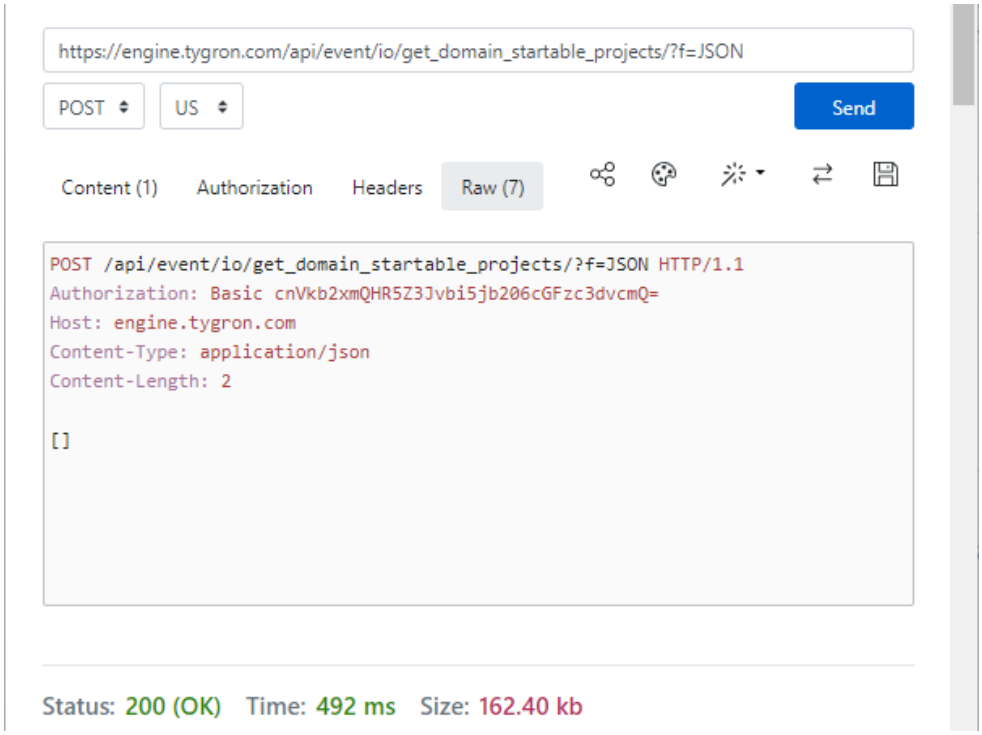
Reqbin is een gratis online dienst, en heeft daarom ook IP-based rate limiting. Indien toegang bestaat tot andere vergelijkbare tools zonder rate-limiting dan hebben die de voorkeur.

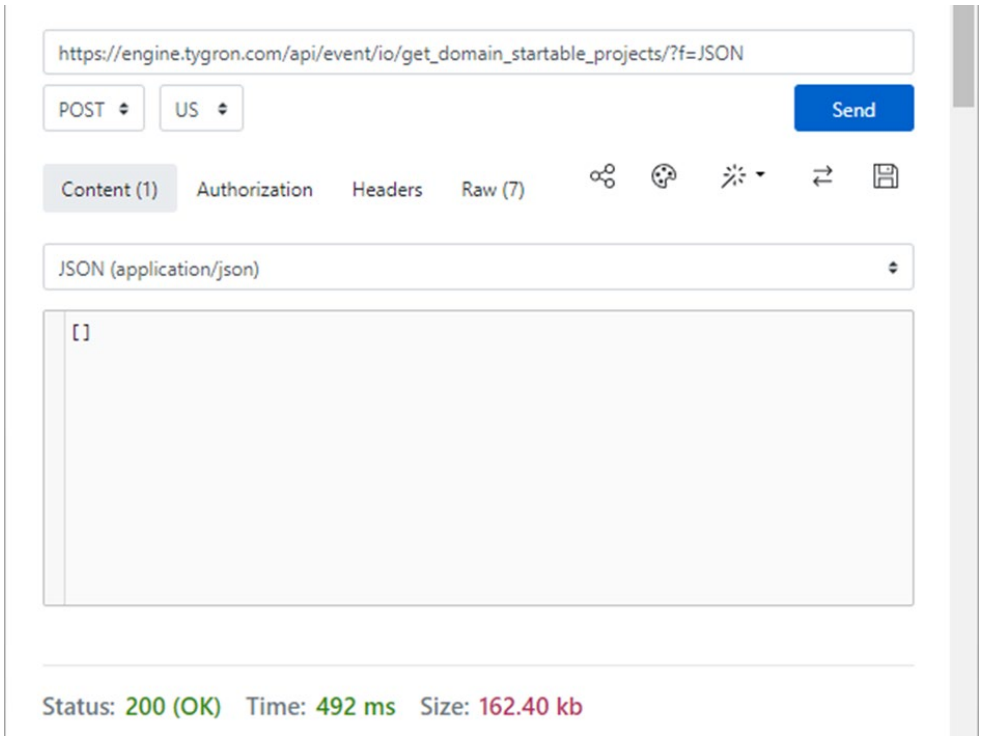
Root API (GET)


https://engine.tygron.com/api/myuser/?f=JSON
GET
US
Send
Content
Authorization
Headers
Raw (3)
Bearer Token
Basic Auth
Custom
No Auth
Username
rudolf@tygron.com
Password
The authorization header will be automatically generated when you send the request. Read more about HTTP Authentication.
Status: 200 (OK)
Time: 271 ms
Size: 0.43 kb

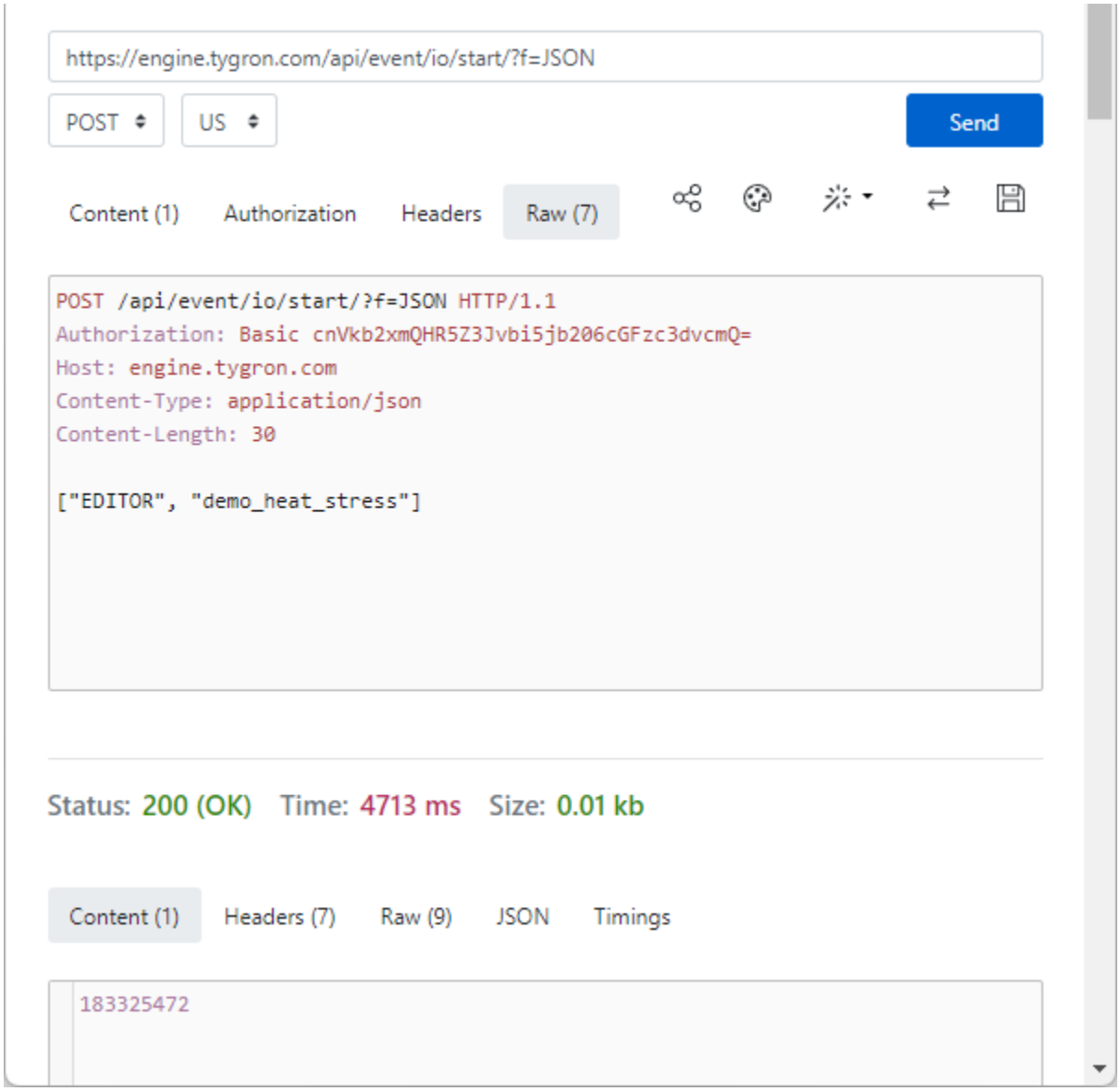

https://engine.tygron.com/api/myuser/?f=JSON
GET
US
Send
Content
Authorization
Headers
Raw (3)
GET /api/myuser/?f=JSON HTTP/1.1
Authorization: Basic cnVkb2xmQHR5Z3Jvbi5jb206cGFzc3dvcmQ=
Host: engine.tygron.com
Status: 200 (OK)
Time: 271 ms
Size: 0.43 kb

Root API (POST) (zonder data)

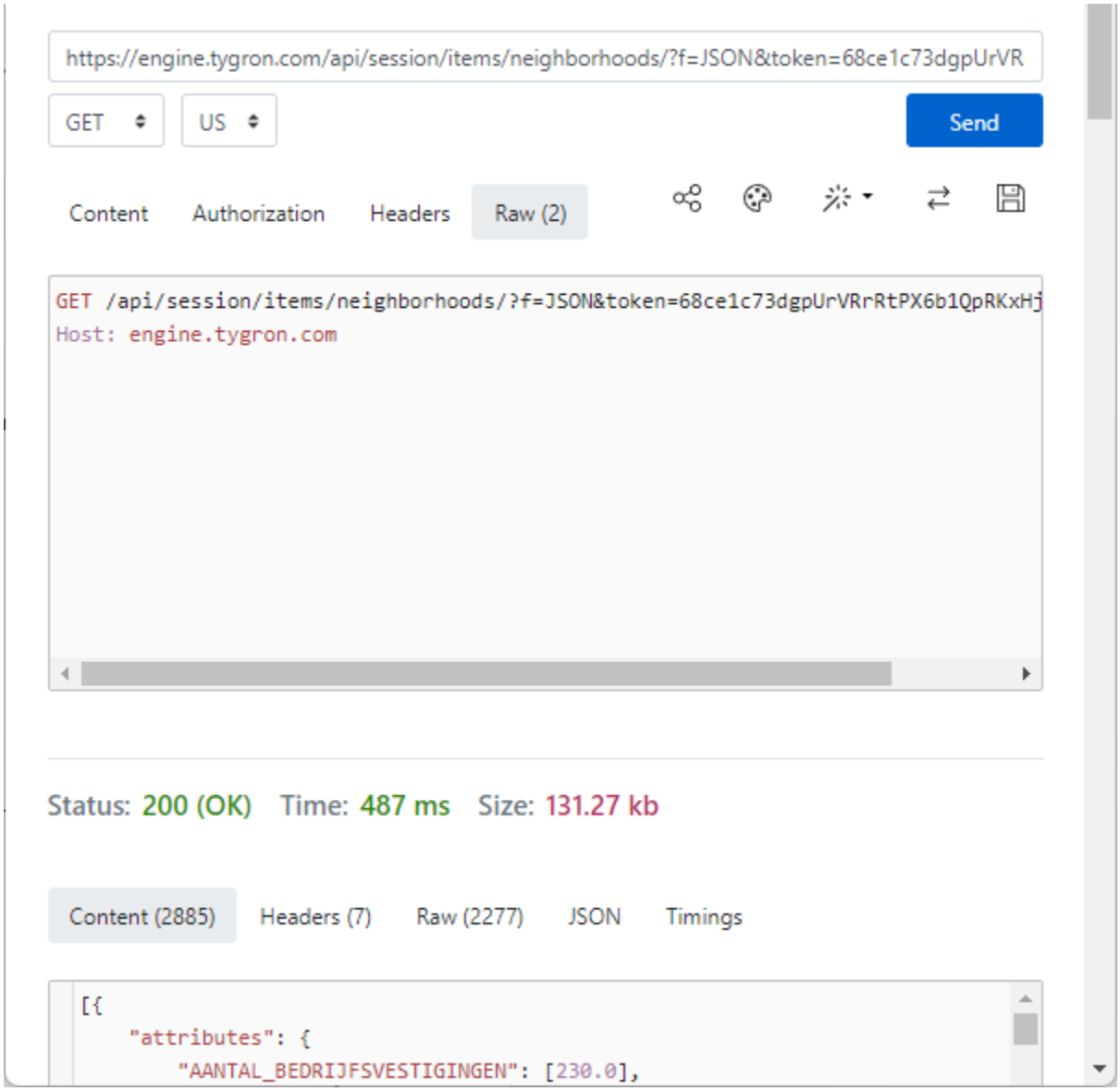




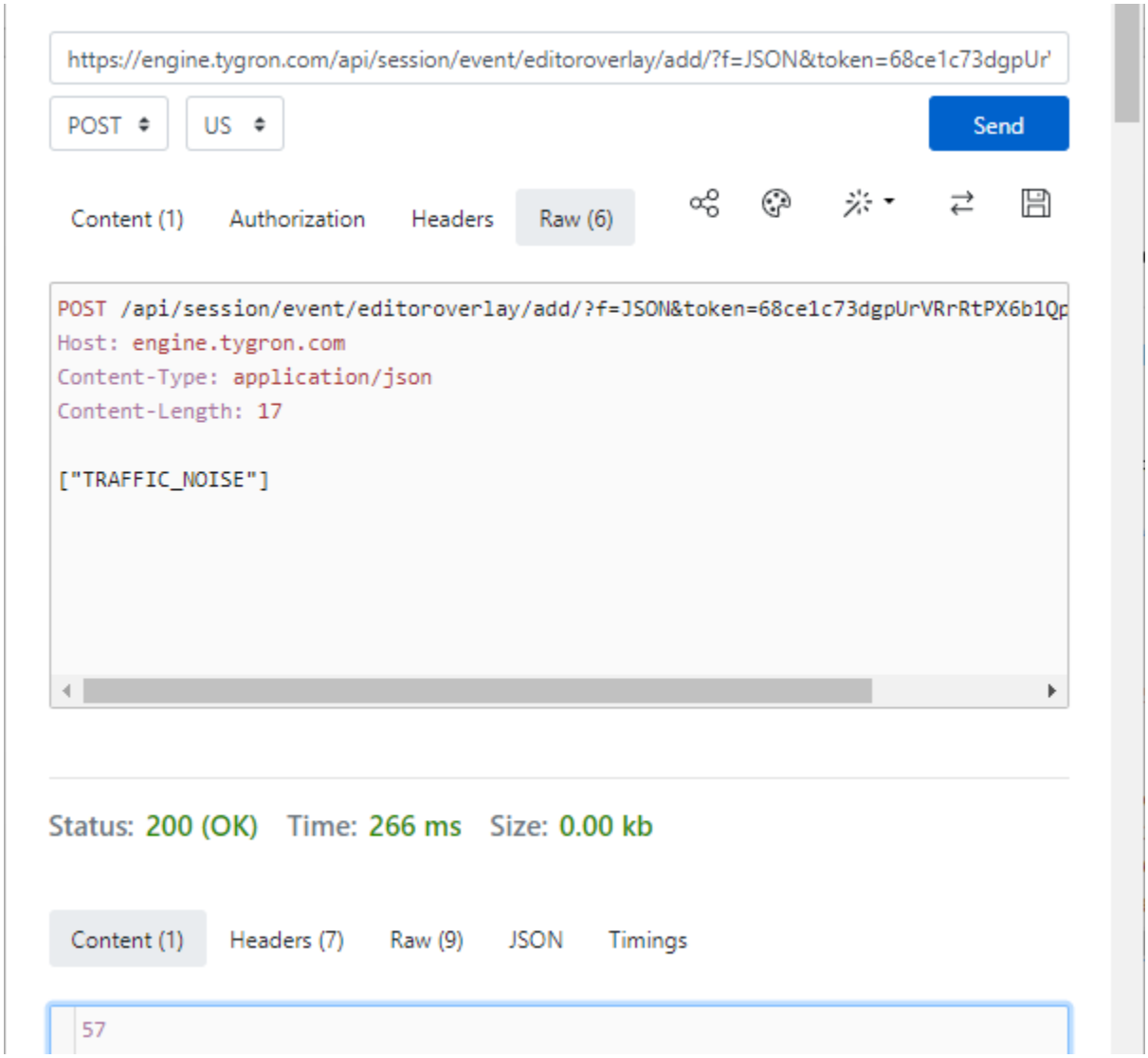
Root API (POST) (met data)



Session API (GET)



Session API (POST)



Overige interessante endpoints

Doel	Method	Url (en data)	Antwoord
Server		https://engine.tygron.com	
Session endpoint	GET	/api/session/?token=...	HTML
WMS	GET	/web/wms?REQUEST=GetCapabilities&token=…	XML
WFS	GET	/api/session/wfs?token=...	XML
Project locatie	GET	/api/session/location/?f=JSON&token=…	JSON
GPU Clusters	GET	/api/session/gpu/clusters/?f=JSON&token=…	JSON
Overzicht Web viewers	GET	/api/session/viewers.html?token=	HTML

Extra links en referenties

API Tutorial

Wil je meer weten over werken met de API, andere werkprocessen oefenen of alle stappen nog een keer begeleid uitproberen? Zie de beschikbare API Tutorial en bijbehorende documentatie om zelfstandig aan de slag te gaan:

https://support.tygron.com/wiki/API_tutorial

PythonSDK op Github

Ben je er klaar voor om het Tygron Platform volautomatisch in je werkproces op te nemen? De Python SDK biedt je een handvat en voorbeelden om zo snel mogelijk aan de slag te gaan. De SDK open-source en direct te verkrijgen vanaf GitHub:

https://github.com/Tygron/python-tygronsdk

Blogs

https://www.tygron.com/nl/blog/2022/11/17/register-duurzame-leefomgeving-gelanceerd-door-nlgreenlabel/

https://www.tygron.com/nl/blog/2023/08/21/standardizing-flood-defenses-using-a-supercomputer/

https://www.tygron.com/nl/blog/2022/04/19/rivm-lanceert-koppeling-groene-baten-planner-met-tygron/

Feedback

Je feedback op deze sessie wordt zeer op prijs gesteld!