

BRANDSPANKING

Markendaalseweg 44 (C104)
4811KC Breda

info@brandspanking.nl
+31-76-7511916
+31-20-21593421

API DOCUMENTATIE

Versie: 02

Datum: 14-06-2021

URL: api.canvas.brandspanking.nl

API key: kEeMVECpYhqHNyTGuvPM

API key (token protected): XgpjNaV0qtws55ZOjsjB

Server key: yh3NY8PoJmkjV-37hjEa



CANVAS API

De Canvas API geeft de mogelijkheid om filmpjes/renderers te maken, render informatie te geven en access tokens te maken. De Canvas API gebruikt een websocket verbinding om communicatie heen en weer te verzorgen of GET/PUT requests met de volgende endpoints voor een enkele aanvraag;

wss://api.canvas.brandspanking.nl/connect/

https://api.canvas.brandspanking.nl/GET

https://api.canvas.brandspanking.nl/PUT

Eventueel is het mogelijk om een extra beveiligings-laag in te schakelen door middel van ip adressen zodat enkel de opgegeven ip adressen toegang hebben om een token te maken. Met een token kan dan 1 API actie uitgevoerd worden binnen X seconden.

Deze documentatie bespreekt 2 test projecten 'example_01' en hetzelfde project 'example_01 beveiligd met een token'. Voorbeelden worden gegeven in JavaScript. Maar voel je vrij om iedere (backend) programmeertaal te gebruiken. Het is verstandig maar niet noodzakelijk om de API key te verbergen voor de gebruiker zodat deze niet zomaar overal renders kan aanmaken, een render/API call brengt uiteindelijk kosten met zich mee.

Example 01

Het filmpje is verzonden om via een webinterface een gebruiker een menu te laten kiezen en wat de klant voor het gekozen menu zou willen betalen. In dit geval kun je kiezen voor een pizza of pasta met muffin of cake als toetje. De gemiddelde keuzes van alle gebruikers werd berekend in de backend en er is keuzen tussen zowel Engels als Nederlandse voice-over die reageert op de hoogte van het ingevoerde bedrag (<10, >10 en >20 euro/dollar) en natuurlijk de gekozen menu-items.

Al deze variabelen kun je nu mee spelen om een eigen filmpje te maken. Om eerst een beetje context te krijgen is het handig een render met de standaard instellingen te maken om te kijken waar de variabelen terecht komen in het filmpje of kijk het meegeleverde filmpje **example_01.mp4**

Gebruik het bijbehorende bestand: **example_01 – V02.html** om de voorbeeld code te bekijken

WEBSOCKET VERBINDING MAKEN

De url om verbinding te maken is: **wss://api.canvas.brandspanking.nl/connect/**

Een voorbeeld van een websocket connectie in javascript, met meestgebruikte events:

```
websocket = new WebSocket('wss://api.canvas.brandspanking.nl/connect/');
websocket.onopen = function() {onOpen()};
websocket.onclose = function() {onClose()};
websocket.onmessage = function(evt) {onMessage(evt)};
websocket.onerror = function(evt) {onError(evt)};
```

Je kunt nu de `websocket.send()` functie gebruiken om data naar de API te versturen, de `onMessage(evt)` functie wordt aangeroepen als er data van de API wordt verstuurd. De verbinding blijft ongeveer 5 minuten open staan, als er binnen deze tijd geen messages worden verstuurd, zal de verbinding worden verbroken.

AUTHENTICATIE

Om te valideren of een API request gemaakt mag worden is een API_KEY nodig, je moet deze in ieder request meegeven, voor de websocket verbinding is dit maar 1x nodig of je kunt een los authentication message versturen, zoals onderstaand voorbeeld. Het is mogelijk de API_KEY te verbergen met tokens, bij het token GET request vind je hier meer informatie over.

```
var jsonData = {
  'type' : 'authentication',
  'api_key': 'kEeMVECpYhqHNYTGuvPM',
};
var stringData = JSON.stringify(jsonData);
websocket.send(stringData);
```

De server geeft een bericht terug met de authenticatie status, bijvoorbeeld:

```
{
  "type" : "authentication",
  "authenticated": true,
}
```

of

```
{
  "type" : "error_authentication",
  "error_id": 11,
  "message" : "Api key not found",
}
```

CREATE RENDER

Na authenticatie kan een json bericht met de onderstaande waarde worden verstuurd om een nieuw filmpje te maken.

Variable	Type	Description	Notes
type	string	request type	for example: create_render
language	string	'en', 'nl'	gebruikte talen
render_input_option1	integer	0 of 1	shot keuze 1, keuze uit 2 shots: pizza/pasta
render_input_option2	integer	0 of 1	shot keuze 2, keuze uit 2 shots: muffin/cake
render_input_amount	integer	0 to 99	prijs (voor de komma)
render_input_amount_float	integer	0 to 99	prijs (na de komma)
render_input_currency	string	'euro', 'dollar'	2 munteenheden
render_stats_percentage1_1	integer	0 to 99	gemiddelde % van klanten welke kiest voor pizza
render_stats_percentage2_1	integer	0 to 99	gemiddelde % van klanten welke kiest voor pasta
render_stats_percentage1_2	integer	0 to 99	gemiddelde % van klanten welke kiest voor muffin
render_stats_percentage2_2	integer	0 to 99	gemiddelde % van klanten welke kiest voor cake

Aanvullend op het voorgaande Javascript voorbeeld, kun je een render maken met de volgende waarde:

```
var jsonData = {  
    'type' : 'create_render',  
    'language' : 'en',  
    'render_input_option1' : 0,  
    'render_input_option2' : 1,  
    'render_input_amount' : 20,  
    'render_input_amount_float' : 50,  
    'render_input_currency' : 'dollar',  
    'render_stats_percentage1_1' : 52,  
    'render_stats_percentage2_1' : 48,  
    'render_stats_percentage1_2' : 48,  
    'render_stats_percentage2_2' : 52,  
    'render_stats_amount_average_float' : 17,  
    'render_stats_amount_average' : 24  
};  
var stringData = JSON.stringify(jsonData);  
websocket.send(stringData);
```

RENDER ACCEPTED

That's it! Als alles goed is ingevoerd komen er vlammen uit de servers en wordt er nu een filmpje gemaakt. De API geeft een bericht terug met de melding 'render_accepted', de gebruiker kan wachten op deze film en zal zo snel mogelijk worden gerenderd. Op de websocket verbinding komt dan een bericht van het type 'render_complete'

Variable	Type	Description	Notes
type	string	return type	for example:render_accepted
video_id	string	uniek id	timestamp_hash-part_unique-id-part

Een voorbeeld van een return message is (Meestal weten we al waar de bestanden komen en wordt de locatie/url hiervan al weergegeven, in het geval van bijvoorbeeld Youtube weten we dat nog niet):

```
{
  "type": "render_accepted",
  "render_time_seconds": 5,
  "location": {
  },
  "video_id": "1570822062_499fcfa7_22904950",
  "que_length": 1,
  "que_time_seconds": 5
}
```

RENDER BUFFERED

In uiterste gevallen wordt de afgesproken capaciteit overschreden, de api call kan dan niet gelijk verwerkt worden en wordt dan opgeslagen om op een minder druk moment alsnog verwerkt te worden. Het is dan niet meer mogelijk om verdere status informatie te ontvangen, met een GET api call is het dan mogelijk de status van de render op te vragen of eventueel kun je de locatie van het video bestand opvragen of het video bestand al weggeschreven is.

```
{
  "type": "render_buffered",
}
```

RENDER COMPLETE

Als de video en eventueel bijbehorende posterframes op de goede locatie staan komt er nog een melding dat de video klaar is: 'render_complete', met bijbehorende locaties. Een voorbeeld hiervan kan zijn:

```
{
  "type": "render_complete",
  "location": {
    "posterframe_button": "https://<url>/1607590232_767ea594_47141319/poster_button_852x362.jpg",
    "posterframe_medium": "https://<url>/1607590232_767ea594_47141319/poster_426x181.jpg",
    "video": "https://<url>/1607590232_767ea594_47141319/video.mp4",
    "posterframe": "https://<url>/1607590232_767ea594_47141319/poster_852x362.jpg",
    "posterframe_button_medium": "https://<url>/1607590232_767ea594_47141319/poster_button_426x181.jpg"
  },
  "video_id": "1570822062_499fcfa7_22904950"
}
```

ERROR!

Het is altijd mogelijk dat er ergens in het systeem iets fout gaat, of dat een doorgestuurde waarde niet helemaal juist is. In dat geval wordt er een error bericht verstuurd vanuit de API

Variable	Type	Description	Notes
type	string	return type	error
message	string	string	
user_error	array/list	array/list	Deze verschijnt alleen bij een input fout (20)
error_id	string	3 characters	

Een voorbeeld van een foutmelding van gebruikers input:

```
{
  "type" : "error",
  "error_id" : "20",
  "message" : "Please check the 'user_error' field for a list of errors",
  "user_error": [{
    "error_id": "100"
    "message" : "Missing the option: name"
  },
  {
    "error_id": "101"
    "message" : "name: is not a string"
  }
]}
```

Mogelijke fouten:

- 1 – Het JSON bericht zelf kan niet gelezen worden
- 2 – Dit type request is niet ondersteund of bestaat niet
- 11 – De API key is niet juist
- 20 – Een of meerdere ingevoerde waarde wordt niet geaccepteerd door de API
- 100 – En hoger zijn custom fouten en verschillen per API key,

RENDER INFO

Soms wil je weten of een render al klaar is of waar deze ook al weer staat, met dit bericht kun je alle info van een render achterhalen

Variable	Type	Description	Notes
type	string	request_type	render_info
video_id	string	string	video_id uit het render_accepted bericht

Een voorbeeld van een foutmelding

```
{
  "type": "error",
  "error_id": 7,
  "message": "The video_id is invalid"
}
```

Een voorbeeld van een foutmelding

```
{
  "type": "render_info"
  "video_id": "1604677675_4998735e_34523254",
  "project_name": "brandspanking_example_01",
  "language": "en",
  "location": {
    "video": "https://storage.../1604677675_4998735e_34523254/video.mp4",
    "posterframe": "https://stor..1604677675_4998735e_34523254/poster_852x362.jpg",
    "posterframe_button": "https://...8735e_34523254/poster_button_852x362.jpg",
    "posterframe_medium": "https://...4998735e_34523254/poster_426x181.jpg",
    "posterframe_button_medium": "https://..._34523254/poster_button_426x181.jpg"
  },
  "date_created": "2020-11-06 16:47:55",
  "date_rendering": "2020-11-06 16:50:03",
  "date_processed": "2020-11-06 16:50:09",
  "status": 1,
}
```

Mogelijke render status:

-1 = Het is fout gegaan

0 = Deze render is onderweg

1 = Klaar en kan getoond worden

2 = Render staat in de buffer en zal als het minder druk is verwerkt worden

GET RENDER INFO

Alle mogelijkheden via websockets, kun je ook met een normale GET of PUT request maken, met de volgende URI krijg je alle info over render: 1607082722_5beba9d6_13333320

https://api.canvas.brandspanking.nl/GET?api_key=kEeMVECpYhqHNyTGuvPM&video_id=1607082722_5beba9d6_13333320

PUT RENDER

Alle mogelijkheden via websockets, kun je ook met een normale GET of PUT request maken, met de volgende URI kun je een Json body meesturen en een render maken, je krijgt dan een error of `render_accepted` zoals in de websocket versie. Vergeet niet het veld `api_key` op te nemen in de verstuurde json zodat deze gevalideerd kan worden.

<https://api.canvas.brandspanking.nl/PUT>

Een andere mogelijkheid is om alle variabelen in de get string mee te geven, zoals bijvoorbeeld:

https://api.canvas.brandspanking.nl/PUT?api_key=1607082722_5beba9d6_13333320&language=en&render_input.....

Dit is eigenlijk alleen de bedoeling als er geen andere mogelijkheid is met een project waar enkel een naam in hoeft.

TOKENS

Om de API te beschermen is het mogelijk om met de server in de backend een token op te halen met de volgende GET request:

https://api.canvas.brandspanking.nl/GET/token?api_key=XgpjNaV0qtws55ZOJsjB&server_key=yh3NY8PoJmkjV-37hjEa

Variable	Type	Description	Notes
api_key	string	string	
server_key	string	string	

De token kan doorgegeven worden aan de frontend, met de token kan een authentication request worden gemaakt gebruik dan 'token' in plaats van 'api_key' in de json. Een token verloopt na een gezette tijd, deze kunnen we per project bespreken. Het token request kunnen we ook beveiligen met een ip check van de servers die de token aanvraag gaan doen.