



neuwO

GETAITOPICS CHEAT SHEET

Simplified technical details of the GetAiTopics endpoint (API and edge API)

ENDPOINTS

POST GetAiTopics	Requires text content in the input. Returns Neuwo's tags, marketing categories and brand safety for the article. Saves new articles in the DB. Takes 5-10 s to process the article.
GET edge/GetAiTopics	Requires article's URL in the input. Returns tags, marketing categories and brand safety for the article. Sends unprocessed URLs into a queue. After the content is analyzed once, the process takes about 30-40 ms to get the results for the same URL. New articles will return 404 for a few seconds while the content is being extracted and analyzed.

POST: GetAiTopics

EXAMPLE REQUESTS IN CURL AND PYTHON

Curl:

```
curl -X 'POST' \  
  'https://m1api.neuwo.ai/GetAiTopics?token=token' \  
-H 'accept: application/json' \  
-H 'Content-Type: application/x-www-form-urlencoded' \  
-d 'content: content' \  
-d 'documentid: documentid' \  
-d 'format: json'
```

Python:

```
import requests  
response = requests.post('https://m1api.neuwo.ai/GetAiTopics',  
  params={'token': 'token'}, data={'format': 'json', 'content': 'content',  
  'documentid': 'documentid'})  
output = response.json()
```

- Token is part of the url - it's given directly as a parameter
- Replace the values of token, content and documentid with your own values

Output:

```
{  
  "tags": [  
    {"URI": uri_value_1, "value": tag_value_1, "score": "0.89342617"},  
    ...,  
    {"URI": uri_value_10, "value": tag_value_10, "score": "0.39074635"}  
  ],  
  "brand_safety": {  
    "BS_score": "1.0",  
    "BS_indication": "yes"  
  },  
  "marketing_categories": {  
    "iab_tier_1": [  
      {"ID": "1-a", "label": label_1, "relevance": "0.81590968"},  
      ...,  
      {"ID": "2-b", "label": label_2, "relevance": "0.65794471"}  
    ],  
    "iab_tier_2": [  
      {"ID": "A-1", "label": label_1, "relevance": "0.68303314"},  
      ...,  
      {"ID": "C-3", "label": label_3, "relevance": "0.42875728"}  
    ]  
  },  
  "smart_tags": [  
    {"ID": smartTagID_1, "name": "name_1"}  
  ]  
}
```

- The 10 Tags are essentially keywords from Neuwo's ontology describing the text
- The 5 Taxonomies describe the iab-1 and iab-2 marketing categories
- In total, we get 2 tier 2 and 3 tier 2 Taxonomies
- SmartTag rules have been predefined by the user. Tags give rise to one SmartTag. For more details see the more comprehensive documentation

POST: GetAiTopics

EXAMPLE REQUEST IN JS

Js:

```
const xhr = new XMLHttpRequest();
var endpoint = 'https://m1api.neuwo.ai/GetAiTopics?token=token';
xhr.open('POST', endpoint);
xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
xhr.onload = () => {
  if (xhr.readyState == 4 && xhr.status == 200) {
    const data = xhr.response;
    console.log(data);
  } else {
    console.log(`Error: ${xhr.status}`);
  }
};
xhr.send('content=content&format=json&documentid=documentid');
```

- Token is part of the endpoint's URL
- Replace the values of token, content and documentid with your own values

Output:

```
{
  "tags": [
    {"URI": "uri_value_1", "value": "tag_value_1", "score": "0.89342617"},
    ...,
    {"URI": "uri_value_10", "value": "tag_value_10", "score": "0.39074635"}
  ],
  "brand_safety": {
    "BS_score": "1.0",
    "BS_indication": "yes"
  },
  "marketing_categories": {
    "iab_tier_1": [
      {"ID": "1-a", "label": "label_1", "relevance": "0.81590968"},
      ...,
      {"ID": "2-b", "label": "label_2", "relevance": "0.65794471"}
    ],
    "iab_tier_2": [
      {"ID": "A-1", "label": "label_1", "relevance": "0.68303314"},
      ...,
      {"ID": "C-3", "label": "label_3", "relevance": "0.42875728"}
    ]
  },
  "smart_tags": [
    {"ID": "smartTagID_1", "name": "name_1"}
  ]
}
```

- The 10 Tags are essentially keywords from Neuwo's ontology describing the text
- The 5 Taxonomies describe the iab-1 and iab-2 marketing categories
- In total, we get 2 tier 2 and 3 tier 2 Taxonomies
- SmartTag rules have been predefined by the user. Tags give rise to one SmartTag. For more details see the more comprehensive documentation

GET: edge/GetAiTopics

EXAMPLE REQUESTS IN CURL AND PYTHON

Curl:

```
curl -X 'GET' \  
  'https://m1api.neuwo.ai/edge/GetAiTopics?token=token&url=url' \  
-H 'accept: application/json' \  
-H 'Origin: magazine.com'
```

Python:

```
import requests  
response = requests.get('https://m1api.neuwo.ai/edge/GetAiTopics',  
  params={'token': 'Your token'}, 'url': 'Your Url'}, headers={'Origin': 'Your  
  domain/publication name', 'accept': 'application/json'})  
output = response.json()
```

- All input values are part of the url - they are given directly as parameters
- In the place of url=url you'd have the real article url
- The edge call needs to have origin defined as a header

Output:

```
{  
  "tags": [  
    {"URI": uri_value_1, "value": tag_value_1, "score": "0.89342617"},  
    ...,  
    {"URI": uri_value_10, "value": tag_value_10, "score": "0.39074635"}  
  ],  
  "brand_safety": {  
    "BS_score": "1.0",  
    "BS_indication": "yes"  
  },  
  "marketing_categories": {  
    "iab_tier_1": [  
      {"ID": "1-a", "label": label_1, "relevance": "0.81590968"},  
      ...,  
      {"ID": "2-b", "label": label_2, "relevance": "0.65794471"}  
    ],  
    "iab_tier_2": [  
      {"ID": "A-1", "label": label_1, "relevance": "0.68303314"},  
      ...,  
      {"ID": "C-3", "label": label_3, "relevance": "0.42875728"}  
    ]  
  },  
  "smart_tags": [  
    {"ID": smartTagID_1, "name": "name_1"}  
  ]  
}
```

- The 10 Tags are essentially keywords from Neuwo's ontology describing the text
- The 5 Taxonomies describe the iab-1 and iab-2 marketing categories
- In total, we get 2 tier 2 and 3 tier 2 Taxonomies
- SmartTag rules have been predefined by the user. Tags give rise to one SmartTag. For more details see the more comprehensive documentation

GET: edge/GetAiTopics EXAMPLE REQUEST IN JS

Js:

```
const xhr = new XMLHttpRequest();
var endpoint =
'https://m1api.neuwo.ai/edge/GetAiTopics?token=token&url=url';
xhr.open('GET', endpoint);
xhr.setRequestHeader('accept', 'application/json');
xhr.send();
xhr.responseType = 'json';
xhr.onload = () => {
  if (xhr.readyState == 4 && xhr.status == 200) {
    const data = xhr.response;
    console.log(data);
  } else {
    console.log(`Error: ${xhr.status}`);
  }
};
```

- Both article url and token are part of the endpoint's URL
- In the place of url=url you'd have the real article url
- The edge call needs to have origin defined as a header which is usually added automatically by your browser

Output:

```
{
  "tags": [
    {"URI": "uri_value_1", "value": "tag_value_1", "score": "0.89342617"},
    ...,
    {"URI": "uri_value_10", "value": "tag_value_10", "score": "0.39074635"}
  ],
  "brand_safety": {
    "BS_score": "1.0",
    "BS_indication": "yes"
  },
  "marketing_categories": {
    "iab_tier_1": [
      {"ID": "1-a", "label": "label_1", "relevance": "0.81590968"},
      ...,
      {"ID": "2-b", "label": "label_2", "relevance": "0.65794471"}
    ],
    "iab_tier_2": [
      {"ID": "A-1", "label": "label_1", "relevance": "0.68303314"},
      ...,
      {"ID": "C-3", "label": "label_3", "relevance": "0.42875728"}
    ]
  },
  "smart_tags": [
    {"ID": "smartTagID_1", "name": "name_1"}
  ]
}
```

- The 10 Tags are essentially keywords from Neuwo's ontology describing the text
- The 5 Taxonomies describe the iab-1 and iab-2 marketing categories
- In total, we get 2 tier 2 and 3 tier 2 Taxonomies
- SmartTag rules have been predefined by the user. Tags give rise to one SmartTag. For more details see the more comprehensive documentation