

# WebGPT

Endpoint : /api/webchat

```
{  
  "query": "string",  
  "temperature": 0.7,  
  "history": [  
    {  
      "role": "string",  
      "content": "string"  
    }  
  ],  
  "moduleName": "WebGPT_15_10_24"  
}
```

▼ Request Payload

[View source](#)

```

▼ {query: "What is time now ?", temperature: 0.5,-}
▼ history: [{role: "system",-}, {role: "user", content: "What is date today ?"},-]
  ▼ 0: {role: "system",-}
    content: "Give the response in concise format unless asked in the query. Give the main points in the response highlighted in bold."
    role: "system"
  ▼ 1: {role: "user", content: "What is date today ?"}
    content: "What is date today ?"
    role: "user"
  ▼ 2: {role: "assistant", content: "Today's date is **Monday, September 15, 2025**."}
    content: "Today's date is **Monday, September 15, 2025**."
    role: "assistant"
moduleName: "WebGPT_15_10_24"
query: "What is time now ?"
temperature: 0.5

```

## Summarizer

Endpoint : /api/webchat

{

```
"query": "string",
"temperature": 0.7,
"history": [
  {
    "role": "string",
    "content": "string"
  }
],
"moduleName": "Summarizer"
}
```

▼ Request Payload

[View source](#)

▼ {query: "What is Ai ?", temperature: 0.7, history: [{role: "system",...}],...}

▼ history: [{role: "system",...}]

▼ 0: {role: "system",...}

content: "Summarize the text with a tone of Generic in bullet format and make it exactly to 1000 words."

role: "system"

moduleName: "Summarizer\_22\_07\_25"

query: "What is Ai ?"

temperature: 0.7

## Translation (Text)

Endpoint : /api/translate

```
{  
  "query": "string",  
  "target_language": "string"  
}
```

I



▲ × Headers Payload EventStream Response >>

▼ Request Payload

[View source](#)

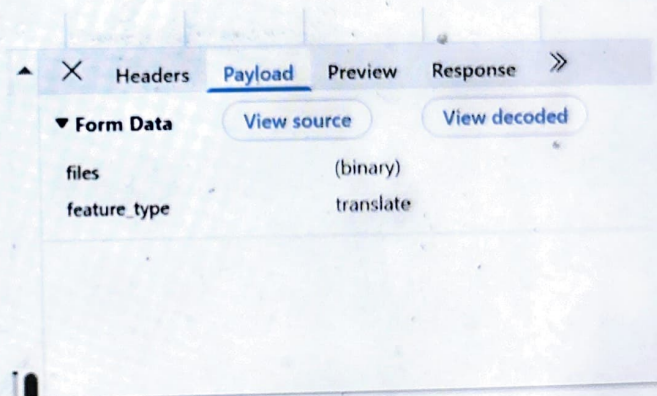
```
{query: "Data please", target_language: "Japanese"}
query: "Data please"
target_language: "Japanese"
```

I

## Translation (Document)

Endpoint : /api/upload

```
{
files : fileinput,
feature_type : "translate"
}
```



Endpoint : /api/docTranslator

```
{
  "toLanguage": "string",
  "filename": "string"
}
```



▼ Request Payload

[view source](#)

```
▼ {toLanguage: "ja", filename: "test.txt"}  
  filename: "test.txt"  
  toLanguage: "ja"
```

## Omniquest

If files need to be uploaded then , two apis will be called

**Endpoint : /api/upload**

```
{  
  files : fileinput,  
  feature_type : "omniquest_docs"  
}
```

FORM DATA

VIEW SOURCE

VIEW REQUEST

files

(binary)

feature\_type

omniquest\_docs

### **Endpoint : /api/status/{job\_id}**


In the header , the job\_id needs to be passed and then we need to check if the parsing indexing is completed or not with polling mechanism every 5 seconds

Different Status will be shown , When status will be completed , it is termed as completed one.

	headers	request	response	initiator	timing
1			{"job_id": "1b149403-aea1-40e3-8330-5671ac0d4b2d", "status": "completed"}		



### Endpoint : /api/omniQuest



```
{
  "query": "string",
  "file_names": [],
  "top_k": 3,
  "temperature": 0.5,
  "feature_type": "string",
  "history": [
    {
      "role": "string",
      "content": "string"
    }
  ]
}
```

```
...
...
▼ {query: "Data please", file_names: [], temperature: 0.7, top_k: 3, feature_type: "omniquest_docs",...}
...
feature_type: "omniquest_docs"
...
file_names: []
...
▼ history: [{role: "system", content: "Hi, Please Give me brief description"}]
...
▼ 0: {role: "system", content: "Hi, Please Give me brief description"}
...
content: "Hi, Please Give me brief description"
...
role: "system"
...
query: "Data please"
...
temperature: 0.7
...
top_k: 3
...
```

## Media Studio

### Image generation

## Image Generation API

**Endpoint : /api/generate-image**

```
{  
  "model": "string",  
  "prompt": "string",  
  "size": "string",  
  "number_of_images": 1,  
  "quality": "high",  
  "addons": "string"  
}
```

## Audio Generation Feature

**AudioGen** – Audio generation API

Payload

```
{  
  "model": "string",  
  "prompt": "string",  
  "language": "string",  
}
```



```
"voice": "string",  
"negative_prompt": "string"  
}
```



### Image Editing Feature

**/api/upload**

```
{  
files : fileinput,  
feature_type : "imgedit"  
}
```

I

**/api/edit-image**

```
{
  "model": "string",
  "prompt": "string",
  "imagename": "string",
  "size": "string",
  "quality": "string",
  "operation_type": "string"
}
```

### **Video Generation**

#### **/api/generate-video**

```
{
  "model": "string",
  "prompt": "string"
}
```

Operation ID will be generated from first API Call will be passed to second one.

#### **/api/video-status**

```
{
  "operation_id": "string",
```