

# Endpoint: Edit Prompt

The `https://swarms.world/api/edit-prompt`` endpoint allows users to edit an existing prompt on the Swarms platform. This API accepts a POST request with a JSON body containing the prompt details to be updated, such as its name, description, use cases, and tags. The request must be authenticated using an API key.

## Endpoint

- **URL:** `https://swarms.world/api/edit-prompt``
- **Method:** POST
- **Content-Type:** `application/json``
- **Authorization:** Bearer token required in the header

## Request Parameters

The request body should be a JSON object with the following attributes:

Attribute	Type	Description	Required
-----	-----	-----	-----
<code>`id`</code>	<code>`string`</code>	The ID of the prompt to be edited.	Yes
<code>`name`</code>	<code>`string`</code>	The name of the prompt.	Yes
<code>`prompt`</code>	<code>`string`</code>	The prompt text.	Yes
<code>`description`</code>	<code>`string`</code>	A brief description of the prompt.	No
<code>`useCases`</code>	<code>`array`</code>	An array of use cases, each containing a title and description.	Yes
<code>`tags`</code>	<code>`string`</code>	Comma-separated tags for the prompt.	No

### `useCases` Structure

Each use case in the `useCases` array should be an object with the following attributes:

Attribute	Type	Description	Required
-----	-----	-----	-----
`title`	`string`	The title of the use case.	Yes
`description`	`string`	A brief description of the use case.	Yes

## Example Usage

### Python

```
```python
import requests
import json

url = "https://swarms.world/api/edit-prompt"
headers = {
    "Content-Type": "application/json",
    "Authorization": "Bearer {apiKey}"
}
data = {
    "id": "prompt_id",
    "name": "Updated Prompt",

```

```
"prompt": "This is an updated prompt from an API route.",
"description": "Updated description of the prompt.",
"useCases": [
    {"title": "Updated use case 1", "description": "Updated description of use case 1"},
    {"title": "Updated use case 2", "description": "Updated description of use case 2"}
],
"tags": "updated, prompt"
}
```

```
response = requests.post(url, headers=headers, data=json.dumps(data))
print(response.json())
...
```

### Node.js

```
```javascript
```

```
const fetch = require("node-fetch");
```

```
async function editPromptsHandler() {
```

```
  try {
```

```
    const response = await fetch("https://swarms.world/api/edit-prompt", {
```

```
      method: "POST",
```

```
      headers: {
```

```
        "Content-Type": "application/json",
```

```
        Authorization: "Bearer {apiKey}",
```

```
      },
```

```
body: JSON.stringify({  
  id: "prompt_id",  
  name: "Updated Prompt",  
  prompt: "This is an updated prompt from an API route.",  
  description: "Updated description of the prompt.",  
  useCases: [  
    {  
      title: "Updated use case 1",  
      description: "Updated description of use case 1",  
    },  
    {  
      title: "Updated use case 2",  
      description: "Updated description of use case 2",  
    },  
  ],  
  tags: "updated, prompt",  
}),  
});  
  
const result = await response.json();  
console.log(result);  
} catch (error) {  
  console.error("An error has occurred", error);  
}  
}
```

```
editPromptsHandler();
```

```
...
```

```
### Go
```

```
```go
```

```
package main
```

```
import (
```

```
    "bytes"
```

```
    "encoding/json"
```

```
    "fmt"
```

```
    "net/http"
```

```
)
```

```
func main() {
```

```
    url := "https://swarms.world/api/edit-prompt"
```

```
    payload := map[string]interface{}{
```

```
        "id":      "prompt_id",
```

```
        "name":    "Updated Prompt",
```

```
        "prompt":  "This is an updated prompt from an API route.",
```

```
        "description": "Updated description of the prompt.",
```

```
        "useCases": []map[string]string{
```

```
            {"title": "Updated use case 1", "description": "Updated description of use case 1"},
```

```
            {"title": "Updated use case 2", "description": "Updated description of use case 2"},
```

```
        },
```

```

    "tags": "updated, prompt",
}

jsonPayload, _ := json.Marshal(payload)

req, _ := http.NewRequest("POST", url, bytes.NewBuffer(jsonPayload))

req.Header.Set("Content-Type", "application/json")

req.Header.Set("Authorization", "Bearer {apiKey}")

client := &http.Client{}

resp, err := client.Do(req)

if err != nil {

    fmt.Println("An error has occurred", err)

    return

}

defer resp.Body.Close()

var result map[string]interface{}

json.NewDecoder(resp.Body).Decode(&result)

fmt.Println(result)

}

...

### cURL

```bash

curl -X POST https://swarms.world/api/edit-prompt \

```

```
-H "Content-Type: application/json" \  
-H "Authorization: Bearer {apiKey}" \  
-d '{  
  "id": "prompt_id",  
  "name": "Updated Prompt",  
  "prompt": "This is an updated prompt from an API route.",  
  "description": "Updated description of the prompt.",  
  "useCases": [  
    { "title": "Updated use case 1", "description": "Updated description of use case 1" },  
    { "title": "Updated use case 2", "description": "Updated description of use case 2" }  
  ],  
  "tags": "updated, prompt"  
'  
...
```

## ## Response

The response will be a JSON object containing the result of the operation. Example response:

```
```json  
{  
  "success": true,  
  "message": "Prompt updated successfully",  
  "data": {  
    "id": "prompt_id",  
    "name": "Updated Prompt",
```

```

"prompt": "This is an updated prompt from an API route.",
"description": "Updated description of the prompt.",
"useCases": [
  {
    "title": "Updated use case 1",
    "description": "Updated description of use case 1"
  },
  {
    "title": "Updated use case 2",
    "description": "Updated description of use case 2"
  }
],
"tags": "updated, prompt"
}
}
...

```

In case of an error, the response will contain an error message detailing the issue.

## ## Common Issues and Tips

- **Authentication Error:** Ensure that the ``Authorization`` header is correctly set with a valid API key.
- **Invalid JSON:** Make sure the request body is a valid JSON object.
- **Missing Required Fields:** Ensure that all required fields (``name``, ``prompt``, ``description``, ``useCases``) are included in the request body.
- **Network Issues:** Verify network connectivity and endpoint URL.



## ## References and Resources

- [API Authentication Guide](https://swarms.world/docs/authentication)
- [JSON Structure Standards](https://json.org/)
- [Fetch API Documentation (Node.js)](https://developer.mozilla.org/en-US/docs/Web/API/Fetch\_API)
- [Requests Library (Python)](https://requests.readthedocs.io/)
- [Net/HTTP Package (Go)](https://pkg.go.dev/net/http)

This comprehensive documentation provides all the necessary information to effectively use the `https://swarms.world/api/add-prompt`` and `https://swarms.world/api/edit-prompt`` endpoints, including details on request parameters, example code snippets in multiple programming languages, and troubleshooting tips.