Agents API Documentation

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

Endpoint: Add Agent

- **URL:** `https://swarms.world/api/add-agent`

- **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
`name`	`string`	The name of the agent.	Yes
`agent`	`string`	The agent text.	Yes
`description`	`string`	A brief description of the agent.	Yes
`language`	`string`	The agent's syntax language with a default of python	No
`useCases`	`array`	An array of use cases, each containing a title and description.	

Yes |

| `requirements` | `array` | An array of requirements, each containing a package name and installation. | Yes |

| `tags` | `string` | Comma-separated tags for the agent. | Yes |

`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

`requirements` Structure

Each requirement in the `requirements` array should be an object with the following attributes:

Attribute	Type	Description	Required
`package`	`string`	The name of the package.	Yes
`installation`	`string`	Installation command for the package	Yes

Example Usage

Python

```python
import requests

import json

import os




url = "https://swarms.world/api/add-agent"




headers = {

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

  "Authorization": f"Bearer {os.getenv("SWARMS_API_KEY")}"

}




data = {

  "name": "Example Agent",

  "agent": "This is an example agent from an API route.",

  "description": "Description of the agent.",

  "language": "python",

  "useCases": [

    {"title": "Use case 1", "description": "Description of use case 1"},

    {"title": "Use case 2", "description": "Description of use case 2"}

  ],

  "requirements": [

    {"package": "pip", "installation": "pip install"},

    {"package": "pip3", "installation": "pip3 install"}
```

```
  ],
    "tags": "example, agent"
}


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

Node.js

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

async function addAgentHandler() {
  try {
    const response = await fetch("https://swarms.world/api/add-agent", {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
        Authorization: "Bearer {apiKey}",
      },
      body: JSON.stringify({
        name: "Example Agent",
        agent: "This is an example agent from an API route.",
        description: "Description of the agent.",
        language: "python",
```

```
      useCases: [

        { title: "Use case 1", description: "Description of use case 1" },

        { title: "Use case 2", description: "Description of use case 2" },

      ],

      requirements: [

        { package: "pip", installation: "pip install" },

        { package: "pip3", installation: "pip3 install" },

      ],

      tags: "example, agent",

    }),

  });


    const result = await response.json();

    console.log(result);

  } catch (error) {

    console.error("An error has occurred", error);

  }

}


addAgentHandler();
```

Go

```go
package main
```

```go
import (

    "bytes"

    "encoding/json"

    "fmt"

    "net/http"

)


func main() {

    url := "https://swarms.world/api/add-agent"

    payload := map[string]interface{}{

        "name":        "Example Agent",

        "agent":       "This is an example agent from an API route.",

        "description": "Description of the agent.",

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

            {"title": "Use case 1", "description": "Description of use case 1"},

            {"title": "Use case 2", "description": "Description of use case 2"},

        },

        "requirements": []map[string]string{

            {"package": "pip", "installation": "pip install"},

            {"package": "pip3", "installation": "pip3 install"}

        },

        "tags": "example, agent",

    }

    jsonPayload, _ := json.Marshal(payload)
```

```go
    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/add-agent \

-H "Content-Type: application/json" \

-H "Authorization: Bearer {apiKey}" \

-d '{

    "name": "Example Agent",
```

```
    "agent": "This is an example agent from an API route.",

    "description": "Description of the agent.",

    "language": "python",

    "useCases": [

        { title: "Use case 1", description: "Description of use case 1" },

        { title: "Use case 2", description: "Description of use case 2" },

    ],

    "requirements": [

        { package: "pip", installation: "pip install" },

        { package: "pip3", installation: "pip3 install" },

    ],

    "tags": "example, agent",

}'
```

Response

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

```json
{

  "success": true,

  "message": "Agent added successfully",

  "data": {

    "id": "agent_id",

    "name": "Example Agent",
```

```
    "agent": "This is an example agent from an API route.",

    "description": "Description of the agent.",

    "language": "python",

    "useCases": [

      { "title": "Use case 1", "description": "Description of use case 1" },

      { "title": "Use case 2", "description": "Description of use case 2" }

    ],

    "requirements": [

      { "package": "pip", "installation": "pip install" },

      { "package": "pip3", "installation": "pip3 install" }

    ],

    "tags": "example, agent"

  }

}
```