

# POST jobs/seed\_async | Experimental API for AI services

---

**API** [useapi.net/docs/api-v2/post-jobs-seed\\_async](https://useapi.net/docs/api-v2/post-jobs-seed_async)



1. [Midjourney API v2](#)
2. **POST** jobs/seed\_async

## Retrieve the [seed](#).

---

October 7, 2024 (August 15, 2025)

## Table of contents

---

1. [Request Headers](#)
2. [Request Body](#)
3. [Responses](#)
4. [Model](#)
5. [Examples](#)
6. [Try It](#)

Retrieve the [seed](#) and **four separate upscaled** images for the following completed jobs:

- [jobs/imagine](#)
- [jobs/button](#)
- [jobs/blend](#)

Please note that video generation jobs are not supported.

| **[https://api.useapi.net/v2/jobs/seed\\_async](https://api.useapi.net/v2/jobs/seed_async)**

### Request Headers

Authorization: Bearer {API token}  
Content-Type: application/json

**API token** is **required**, see [Setup useapi.net](#) for details.

### Request Body

```
{
  "jobid": "jobid",
  "replyUrl": "Place your call back URL here",
  "replyRef": "Place your reference id here"
}
```

- **jobid** is **required**, **jobid** of successfully completed (**status** set to [completed](#)) [jobs/imagine](#), [jobs/button](#) or [jobs/blend](#) job.
- **replyUrl** is optional, if not provided value from [useapi.net account](#) will be used.  
Place here your callback URL. API will call the provided **replyUrl** once job completed or failed.  
Maximum length 1024 characters.  
We recommend using sites like [webhook.site](#) to test callback URL functionality.
- **replyRef** is optional, place here your reference id which will be stored and returned along with this job response / result.  
Maximum length 1024 characters.

#### Responses

- [200](#)
- [400](#)
- [401](#)
- [402](#)
- [404](#)
- [502](#)
- [596](#)

## 200 OK

Use returned `jobid` to [retrieve job status and results](#).

If you specify the optional parameter [replyUrl](#) the API will call the provided `replyUrl` with task progress updates until the task is complete or fails.

```
{
  "jobid": "<jobid>",
  "verb": "seed_async",
  "status": "started",
  "created": "2023-09-09T20:19:57.073Z",
  "updated": "2023-09-09T20:19:58.536Z",
  "parentJobId": "<jobid>",
  "discord": "<ABC...secured...xyz>",
  "channel": "<Discord channel id>",
  "replyUrl": "https://webhook.site/abc",
  "replyRef": "<your optional reference id>",
  "code": 200
}
```

If the `seed` was already retrieved for the provided `jobid` API will return the original job along with the seed as shown below. You can extract seed value from field `seed.value` and generated images from `seed.attachments` array.

```

{
  "jobid": "<jobid>",
  "verb": "imagine|button|blend",
  "status": "completed",
  "created": "2023-09-09T02:04:49.667Z",
  "updated": "2023-09-09T02:04:53.490Z",
  "content": "***Steampunk cat --s 750 --v 5.2** - <@Discord user id> (fast)",
  "attachments": [
    {
      "id": "<id>",
      "url": "<image url>",
      "proxy_url": "<original image proxy url>",
      "width": 2048,
      "height": 2048,
      "size": 1234567,
      "placeholder_version": 1,
      "placeholder": "<placeholder>",
      "filename": "<image file name>",
      "content_type": "<image content-type>"
    }
  ],
  "seed": {
    "content": "***<prompt>**\n**Job ID**: <Job ID>n**seed** <seed>",
    "value": 123456789,
    "attachments": [
      {
        "id": "<id 1>",
        "url": "<image 1 url>",
        "proxy_url": "<image 1 proxy url>",
        "width": 1024,
        "height": 1024,
        "size": 12345,
        "placeholder_version": 1,
        "placeholder": "<placeholder 1>",
        "filename": "<image 1 file name>",
        "content_type": "<image 1 content-type>"
      },
      {
        "id": "<id 2> ...the rest of JSON object omitted for brevity...",
      },
      {
        "id": "<id 3> ...the rest of JSON object omitted for brevity...",
      },
      {
        "id": "<id 4> ...the rest of JSON object omitted for brevity...",
      }
    ]
  },
  "prompt": "Steampunk cat",
  "discord": "<ABC...secured...xyz>",
  "channel": "<Discord channel id>",
  "maxJobs": 3,

```

```
"replyUrl": "https://webhook.site/abc",  
"replyRef": "<your optional reference id>",  
"messageId": "<Discord message id>",  
"timestamp": "2023-09-09T02:04:51.926000+00:00",  
"code": 200  
}
```

## Model

```
{ // TypeScript, all fields are optional
  jobId: string, // Use returned jobId value to retrieve job status and results
  parentJobId: string,
  verb: 'seed_async' | 'imagine' | 'button' | 'blend',
  status: 'started' | 'failed',
  seed: {
    value: number, // Parsed seed number
    content: string // Original content message returned by Midjourney for a seed
  },
  request: {
    attachments: { // Seed images
      id: string,
      url: string,
      proxy_url: string,
      width: number,
      height: number,
      size: number,
      placeholder_version: number,
      placeholder: string,
      filename: string,
      content_type: string[]
    },
    created: string, // YYYY-MM-DDTHH:mm:ss.sssZ, ISO 8601, UTC
    updated: string, // YYYY-MM-DDTHH:mm:ss.sssZ, ISO 8601, UTC
    discord: string, // Provided for debugging purposes only, contains the first 3 and
    // the last 3 characters of the original value
    channel: string,
    replyUrl: string,
    replyRef: string,
    messageId: string,
    content: string, // Contains message generated by Midjourney reflecting current
    // generation parameters and progress
    attachments: {
      id: string,
      url: string,
      proxy_url: string,
      width: number,
      height: number,
      size: number,
      placeholder_version: number,
      placeholder: string,
      filename: string,
      content_type: string[]
    },
    timestamp: string,
    error: string,
    errorDetails: string,
    executingJobs: string[],
    code: number
  }
}
```

## Examples

- [Curl](#)
- [JavaScript](#)
- [Python](#)

```
curl -H "Accept: application/json" \  
      -H "Content-Type: application/json" \  
      -H "Authorization: Bearer ..." \  
      -X POST https://api.useapi.net/v2/jobs/seed_async \  
      -d '{"jobid": "..."}'
```

## Try It