REST API migration

- Endpoint changes
- API token permissions
  - Create a new token
  - Edit an existing token
- Chat completions
- Search
- Streaming behavior changes
- Filter format
  - Operator mapping
  - Examples
    - Simple filter
    - Compound filter (AND)
- API references

The AutoRAG API endpoints are the legacy REST API for AI Search. They will continue to work, but all new features and improvements are only available through the new AI Search API endpoints.

Endpoint changes

The legacy AutoRAG API endpoints under `/autorag/rags/` have been replaced by new endpoints under `/ai-search/namespaces/{namespace}/instances/`.

Description	New endpoint	Reference
Chat completions	`/ai-search/namespaces/{namespace}/instances/{id}/chat/completions`	API reference
Search	`/ai-search/namespaces/{namespace}/instances/{id}/search`	API reference

The new API also includes endpoints for instance management, items, and namespace-level search that are not available in the legacy API. For the legacy endpoints, refer to the AutoRAG API reference.

API token permissions

The legacy AutoRAG endpoints used the **AutoRAG** API token permission. The new AI Search endpoints require the **AI Search** permission instead, so update the permissions on the token you use to call the API. We recommend using account API tokens, which are owned by the account rather than a single user, and adding the **AI Search** permission found under **AI & Machine Learning** > **AI Search**.

Create a new token

1. In the Cloudflare dashboard, go to **Manage Account** > **API Tokens**.
2. Select **Create Token**, then start a custom token.
3. Enter a name for the token.
4. Add a permission policy and select **AI & Machine Learning** > **AI Search**, then choose the access level you need. AI Search offers **Read**, **Run**, and **Edit** access.
5. (Optional) Set client IP address filtering and a token expiration.
6. Create the token and copy its value.

Edit an existing token

1. In the Cloudflare dashboard, go to **Manage Account** > **API Tokens**.
2. Select the token you want to update.
3. Add or update a permission policy to include **AI & Machine Learning** > **AI Search** with the access level you need, then save.

For the full token creation flow, refer to Create API token.

Chat completions

How to migrate from the AutoRAG `/ai-search` endpoint to the new `/chat/completions` endpoint:

Before (AutoRAG API):

```
curl -X POST
"https://api.cloudflare.com/client/v4/accounts/<ACCOUNT_ID>/autorag/rags/<INSTANCE_NAME>/ai-
search" \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer <API_TOKEN>" \
  -d '{
    "query": "What is Cloudflare?"
  }'
```

After (AI Search API):

The new API uses the `messages` array format.

```
curl -X POST "https://api.cloudflare.com/client/v4/accounts/<ACCOUNT_ID>/ai-
search/namespaces/default/instances/<INSTANCE_NAME>/chat/completions" \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer <API_TOKEN>" \
  -d '{
    "messages": [
      {
        "content": "What is Cloudflare?",
        "role": "user"
      }
    ]
  }'
```

Search

How to migrate from the AutoRAG `/search` endpoint to the new `/search` endpoint:

Before (AutoRAG API):

```
curl -X POST
"https://api.cloudflare.com/client/v4/accounts/<ACCOUNT_ID>/autorag/rags/<INSTANCE_NAME>/searc
h" \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer <API_TOKEN>" \
  -d '{
    "query": "What is Cloudflare?"
```

```
  }'
```

After (AI Search API):

The new API uses the `messages` array format. The `query` string format is also supported.

```
curl -X POST "https://api.cloudflare.com/client/v4/accounts/<ACCOUNT_ID>/ai-search/namespaces/default/instances/<INSTANCE_NAME>/search" \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer <API_TOKEN>" \
  -d '{
    "messages": [
      {
        "content": "What is Cloudflare?",
        "role": "user"
      }
    ]
  }'
```

Streaming behavior changes

In the old AutoRAG API, when `stream` was set to `true`, you would only receive the streamed response without the retrieved chunks.

In the new AI Search API, streaming responses include the chunks. The retrieved chunks are sent first as a `chunks` event, followed by the streamed response data. This allows you to display the source chunks immediately while streaming the generated response to the user.

Filter format

The new AI Search REST API uses Vectorize-style metadata filtering, which differs from the AutoRAG API format. Filters are now nested under `ai_search_options.retrieval.filters` in the request body. For full documentation of the old format, refer to Metadata filter format (legacy).

Operator mapping

The filter operators have been renamed to use a `$` prefix:

AutoRAG API	AI Search API
`eq`	`$eq` (or implicit)
`ne`	`$ne`
`gt`	`$gt`
`gte`	`$gte`
`lt`	`$lt`
`lte`	`$lte`
	`$in` (new)
	`$nin` (new)

Examples

Simple filter

Filter by a single metadata field using implicit equality:

Before (AutoRAG API):

```
{
  "filters": {
    "type": "eq",
    "key": "folder",
    "value": "customer-a/"
  }
}
```

After (AI Search API):

```
{
  "ai_search_options": {
    "retrieval": {
      "filters": { "folder": "customer-a/" }
    }
  }
}
```

Compound filter (AND)

Combine multiple conditions where all must match:

Before (AutoRAG API):

```
{
  "filters": {
    "type": "and",
    "filters": [
      { "type": "eq", "key": "folder", "value": "customer-a/" },
      { "type": "gte", "key": "timestamp", "value": "1735689600000" }
    ]
  }
}
```

After (AI Search API):

```
{
  "ai_search_options": {
    "retrieval": {
      "filters": {
        "folder": "customer-a/",
        "timestamp": { "$gte": 1735689600 }
      }
    }
  }
}
```

API references

- REST API documentation
- Chat Completions API reference
- Search API reference
- Legacy AutoRAG API reference

#2
2026-10-31 23:44:35 UTC
2026-10-31 23:44:36 UTC