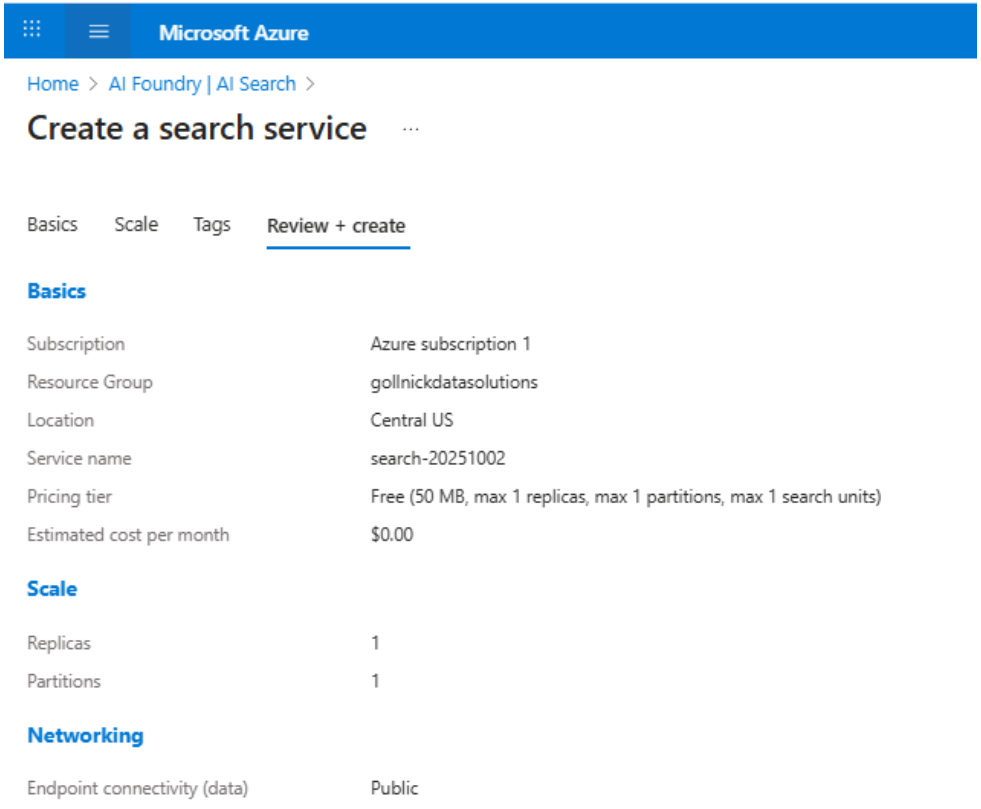
AI Search

gollnickdata.de

AI Search

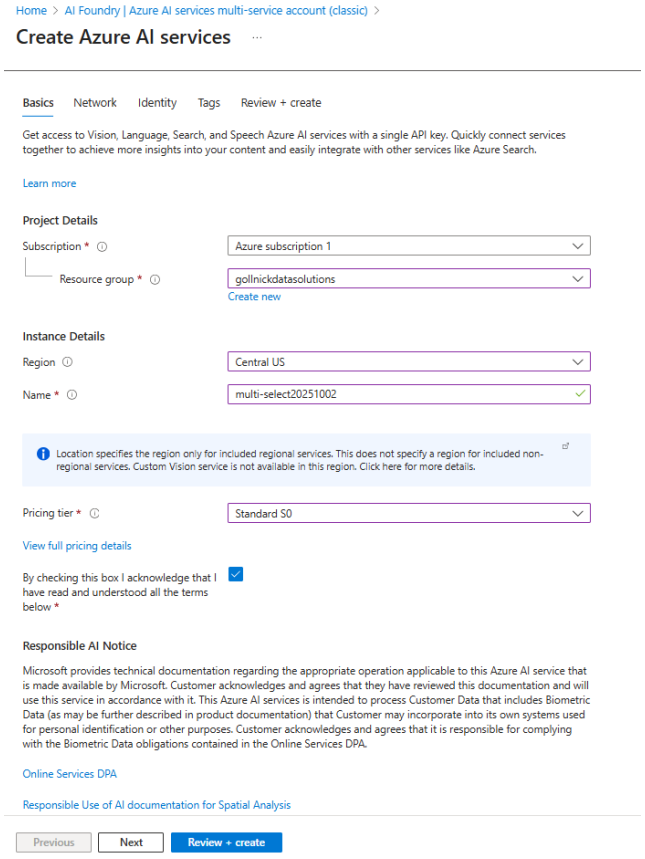
Create search service



gollnickdata.de

AI Search

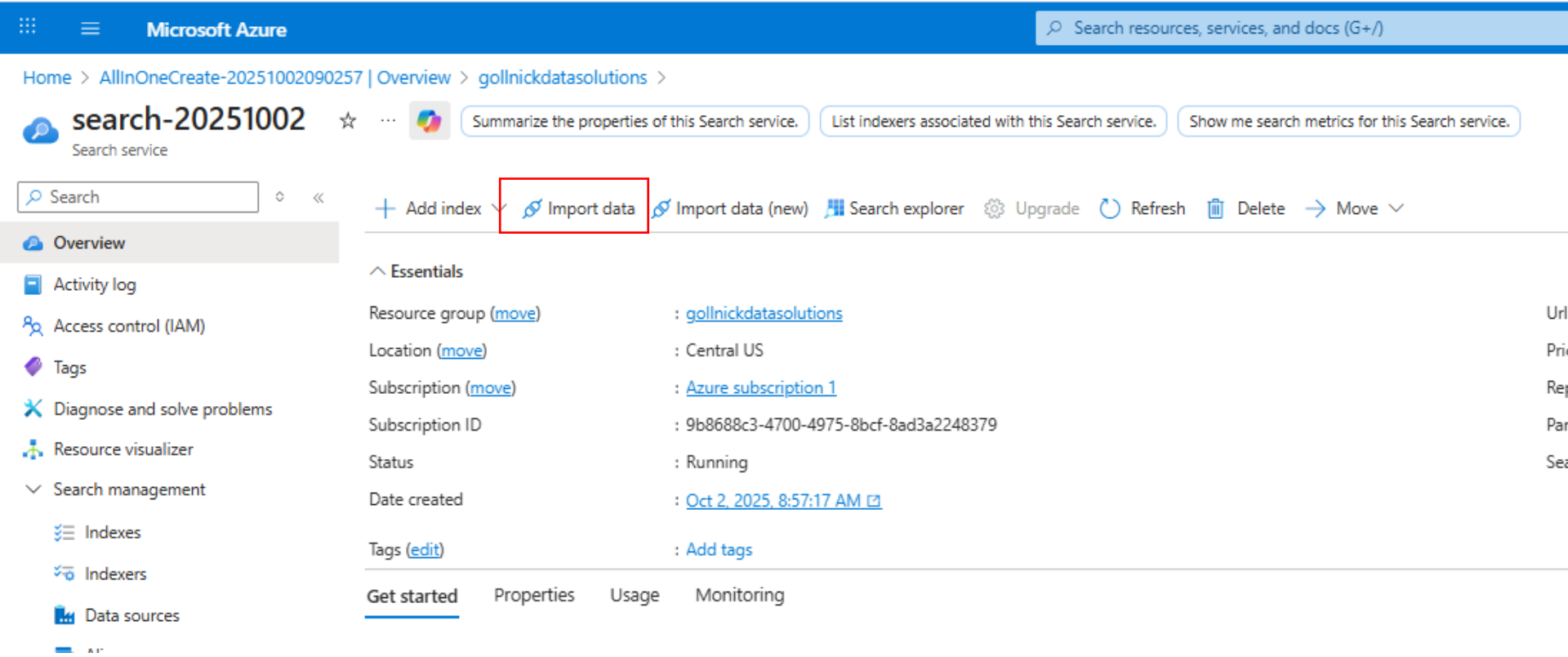
Create AI services multi-service account



gollnickdata.de

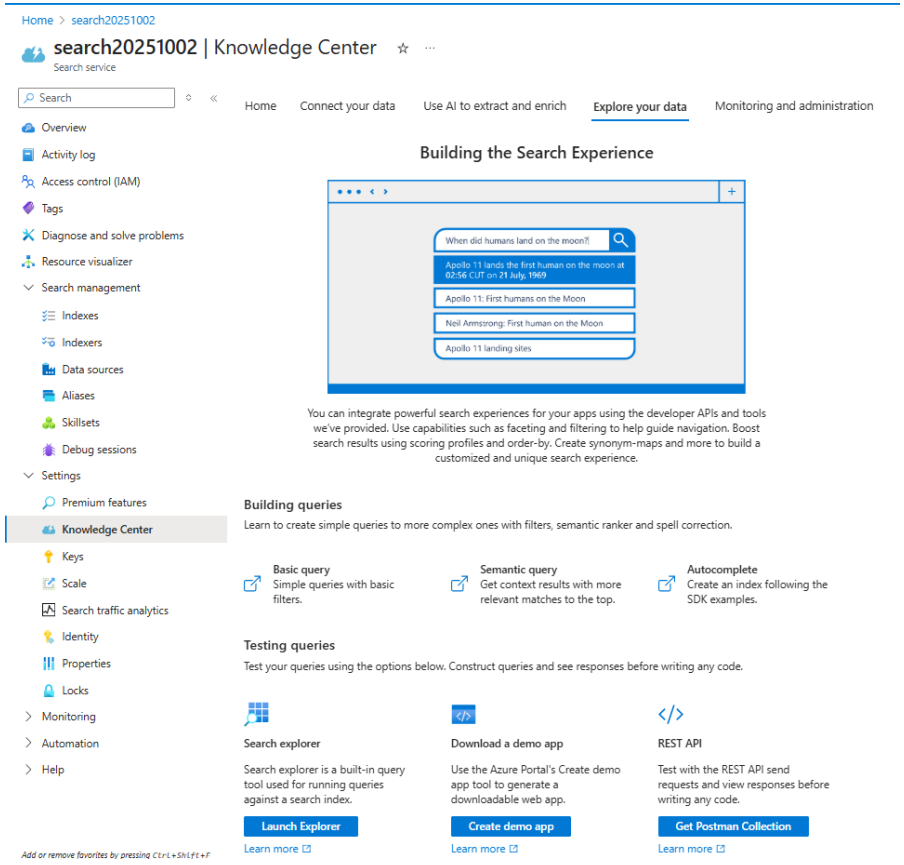
AI Search

Add Data



AI Search

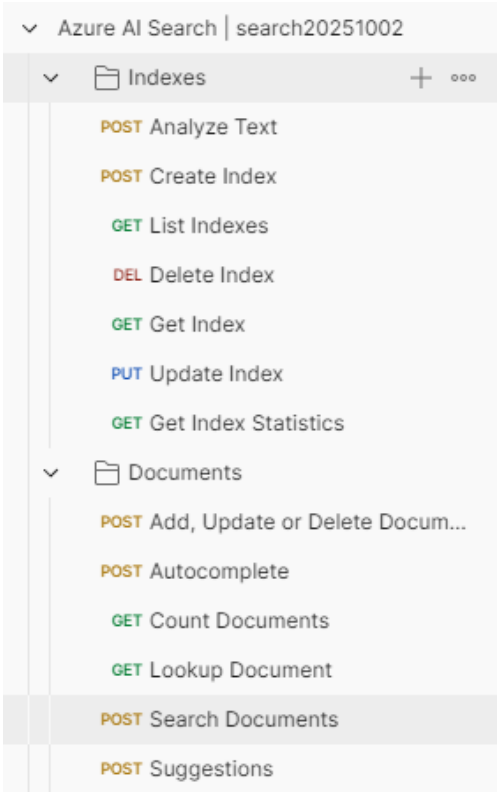
Postman



gollnickdata.de

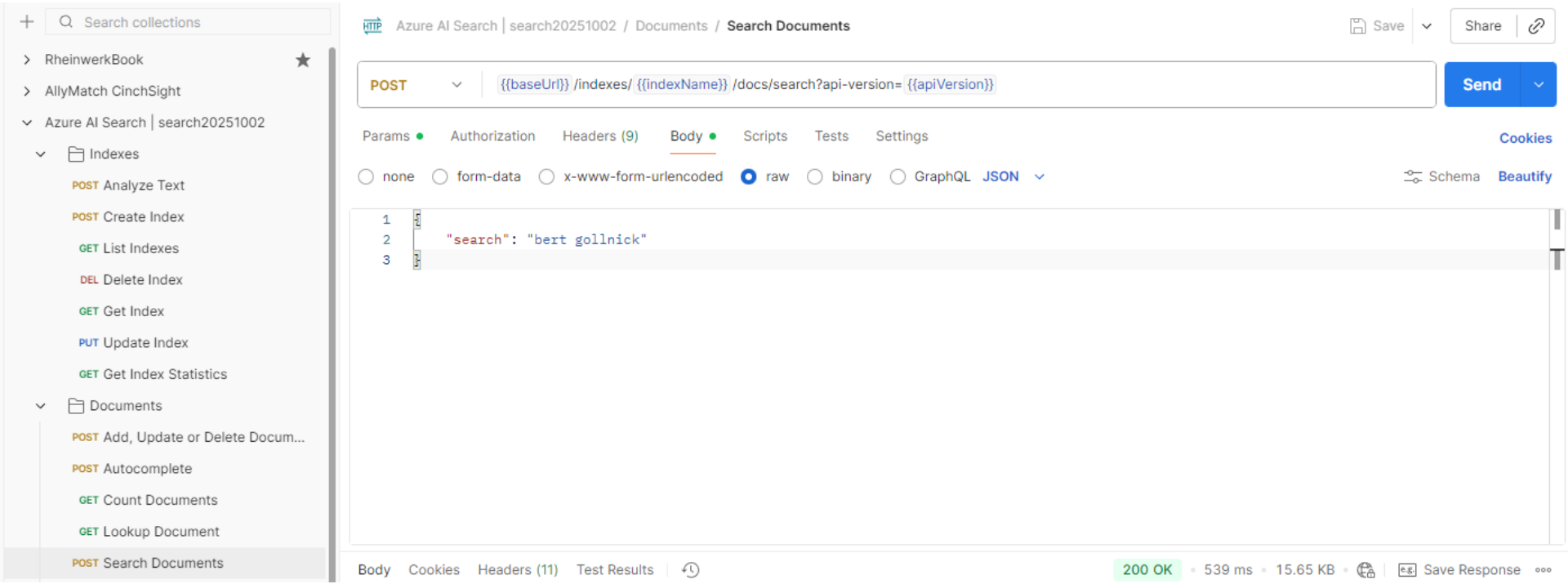
AI Search

Postman



AI Search

Postman



gollnickdata.de

AI Search

Postman

1. 🔍 Query Expression Parameters

These parameters define *what* and *how* you search.

Parameter Key	Description	Example Value
`search`	The **core query string** for full-text search. Use Lucene or simple syntax.	`"luxury hotel near airport"`
`queryType`	Specifies the query parser. Defaults to `"simple"`. Use `"full"` for Lucene syntax, or `"semantic"` for semantic ranking.	`"full"` or `"semantic"`
`searchMode`	Defines whether *any* or *all* terms must match. Defaults to `"any"`.	`"all"`
`searchFields`	A **comma-separated list of fields** to restrict the search to. By default, it searches all `searchable` fields.	`"title, description, tags"`
`filter`	An **OData $filter expression** to apply a Boolean filter *before* ranking. Use this for exact matches, ranges, or geospatial queries.	`"category eq 'Electronics' and price gt 500"`
`vectorQueries`	Defines a list of vector search queries (required for **Hybrid Search**). This object contains the embedding vector (`values`) or text (`text`) to be vectorized.	`[ { "vector": [...], "k": 5 } ]`

2. 📋 Result Shaping Parameters

These parameters control the appearance, quantity, and order of the results.

Parameter Key	Description	Example Value
`select`	A **comma-separated list of fields** to return in the search results. Only `retrievable` fields are included.	`"id, name, price"`
`orderby`	A **comma-separated list of OData $orderby expressions** for sorting. The default is ranking by relevance score (`@search.score`).	`"price desc, rating asc"`
`top`	The **maximum number of results** to return in the current request (for pagination). Defaults to 50.	`10`
`skip`	The **number of search results to skip** (for pagination).	`20`
`count`	If `true`, the response includes the total number of documents that match the query (`@odata.count`).	`true`
`highlight`	A **comma-separated list of fields** for hit highlighting.	`"description"`

AI Search

Postman

3. 🧠 Advanced AI (Semantic) Parameters

These are used when `queryType` is set to `"semantic"` to enable **Semantic Ranking** and advanced content extraction.

Parameter Key	Description	Example Value
`semanticConfiguration`	The name of the **semantic configuration** defined in your index.	`"my-semantic-config"`
`answers`	Specifies whether to extract a single, concise answer from the most relevant documents.	`"extractive
`captions`	Specifies whether to return the most representative passages (captions) from the documents.	`"extractive
`scoringProfile`	The name of a **custom scoring profile** defined in your index to influence the relevance score.	`"product-promoter"`

AI Search

Postman

JSON

```
{
    "search": "latest durable waterproof hiking boots",
    "queryType": "semantic",
    "semanticConfiguration": "default-semantic-config",
    "select": "ProductID, Name, Description, Price",
    "filter": "Price lt 300",
    "orderby": "@search.score desc",
    "top": 5,
    "skip": 0,
    "count": true,
    "answers": "extractive|full",
    "captions": "extractive|highlight-and-hit"
}
```

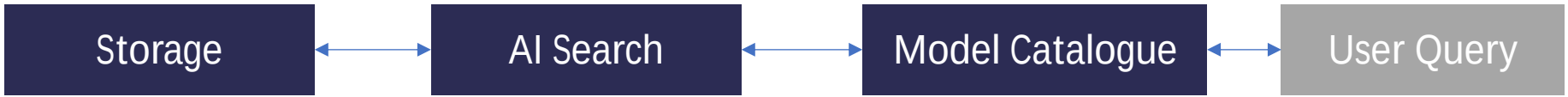
AI Search

Exercise

Exercise

Intelligent Knowledge Portal

- Allows colleagues to chat with complex internal documents



- Holds documents
- Indexing,
- Vectorization
- RAG