Documentation for ElasticsearchService NuGet Package (Version 1.0.2)

Overview

The `ElasticsearchService` class provides a simple and flexible way to interact with Elasticsearch in your .NET applications. It includes methods for indexing, searching, updating, and deleting documents.

Installation

To install the `ElasticsearchService` package, run the following command in the Package Manager Console:

```
Install-Package ElasticsearchService -Version 1.0.2
```

Usage

1. Initialize the Service

First, you need to create an instance of `ElasticsearchService`. This requires an instance of `IElasticClient` and the name of the index you want to work with.

```
using atikapps;

var settings = new ConnectionSettings(new Uri("http://localhost:9200"))
    .DefaultIndex("your-index-name");

var client = new ElasticClient(settings);
var elasticsearchService = new ElasticsearchService(client, "your-index-
name");
```

2. Creating an Index (If Not Already Exists)

You can create an index if it doesn't already exist by calling the `CreateIndexIfNotExists` method.

```
await elasticsearchService.CreateIndexIfNotExists("your-index-name");
```

3. Indexing Documents

Index a Single Document

You can add or update a single document using the `AddOrUpdate` method.

```
var document = new { Id = Guid.NewGuid(), Name = "Sample Document" };
await elasticsearchService.AddOrUpdate(document, document.Id);
```

Index Multiple Documents

To index a bulk of documents, use the `AddOrUpdateBulk` method.

```
var documents = new List<object>
{
    new { Id = Guid.NewGuid(), Name = "Document 1" },
    new { Id = Guid.NewGuid(), Name = "Document 2" }
};

await elasticsearchService.AddOrUpdateBulk(documents);
```

4. Retrieving Documents

Get a Single Document by Key

Retrieve a document by its key using the `Get` method.

```
var document = await elasticsearchService.Get<object>("document-key");
```

Get All Documents

To retrieve all documents in the current index, use the `GetAll` method.

```
var documents = await elasticsearchService.GetAll<object>();
```

5. Querying Documents

You can perform custom queries using the `Query` method. Pass a `QueryContainer` to define the query.

```
var predicate = new TermQuery { Field = "fieldName", Value = "value" };
var results = await elasticsearchService.Query<object>(predicate);
```

6. Deleting Documents

Delete a Single Document by Key

You can remove a document using its key with the `Remove` method.

```
bool success = await elasticsearchService.Remove<object>("document-key");
```

Delete All Documents

To delete all documents in the current index, use the `RemoveAll` method.

```
long deletedCount = await elasticsearchService.RemoveAll<object>();
```

Delete Documents by Text Match

You can delete documents that match specific text in a field using `RemoveByTextMatch`.

```
long deletedCount = await
elasticsearchService.RemoveByTextMatch<object>("fieldName", "text");
```

Method Summary

- **Index(string indexName)**: Sets the index name and returns the `ElasticsearchService` instance for method chaining.
- **SetIndex(string indexName)**: Sets the index name without returning the service instance.
- **CreateIndexIfNotExists(string indexName)**: Creates an index if it does not already exist.
- **AddOrUpdateBulk<T>(IEnumerable<T> documents)**: Adds or updates a bulk of documents in the current index.
- **AddOrUpdate<T>(T document, Guid id)**: Adds or updates a single document in the current index.
- **Get<T>(string key)**: Retrieves a single document by its key from the current index.
- **GetAll<T>()**: Retrieves all documents from the current index.
- **Query<T>(QueryContainer predicate)**: Queries documents based on a custom predicate.
- **Remove<T>(string key)**: Removes a single document by its key from the current index.
- **RemoveAll<T>()**: Removes all documents from the current index.
- **RemoveByTextMatch<T>(string fieldName, string text)**: Removes documents that match a specific text in a specific field.

Example

Here is a complete example of using the `ElasticsearchService`:

```
using atikapps;

var settings = new ConnectionSettings(new Uri("http://localhost:9200"))
    .DefaultIndex("your-index-name");

var client = new ElasticClient(settings);
var elasticsearchService = new ElasticsearchService(client, "your-index-
name");

// Ensure index exists
await elasticsearchService.CreateIndexIfNotExists("your-index-name");

// Index a document
var document = new { Id = Guid.NewGuid(), Name = "Sample Document" };
await elasticsearchService.AddOrUpdate(document, document.Id);

// Retrieve the document
var retrievedDocument = await
elasticsearchService.Get<object>(document.Id.ToString());

// Query documents
var predicate = new TermQuery { Field = "Name", Value = "Sample Document" };
var results = await elasticsearchService.Query<object>(predicate);

// Delete the document
await elasticsearchService.Remove<object>(document.Id.ToString());
```

This documentation should help you get started with the `ElasticsearchService` package. For any issues or further details, refer to the source code or contact the package author.