The Allergy search is implemented using the Hibernate Search library, which integrates Hibernate ORM with the Apache Lucene search engine. This allows for full-text search on entities using a familiar syntax.

**Controller:**

The Spring Boot REST controller provides three endpoints to interact with the Allergys:

1. /Allergys/search: This endpoint performs a full-text search on the Allergy using a search term. The search term is passed as a query parameter, and the search results are returned as a Page of Allergy objects. The Pageable parameter allows clients to specify the page size and offset for the search results. The search is performed using the searchAllergyCode() method of the AllergyService class, which is autowired into the controller.

2. /Allergy/index: This endpoint creates an index for the Allergy. The index is created using the createAllergyIndex() method of the AllergyService class, which is autowired into the controller. The method returns a String message indicating whether the index creation was successful or not.

**Service:**

The AllergyService Java service class provides functionality for searching and indexing Allergy codes. Here's an overview of the methods and their functionality:

1. createAllergyIndex(): This method creates an index for Allergy codes using Hibernate Search. It retrieves an instance of SearchSession using an EntityManager and creates a MassIndexer instance. Then, it sets some options on the MassIndexer and starts the indexing process.

2. searchAllergyCode(): This method searches the indexed Allergy codes using Hibernate Search. It retrieves an instance of SearchSession using an EntityManager and executes a search query on the AllergyCode entity. It searches on the Allergy, damConceptIdDesc, damAlrgnGrpDesc, and allergyDesc fields using the provided search term. It also applies a fuzzy search to cover spelling corrections. It returns a Page object containing the search results.

**Repository:**

The AllergyRepository interface defines a repository that extends the Spring Data JpaRepository interface.

The AllergyRepository interface extends the Spring Data.

**URL’s:**

**POST:** **http://localhost:9191/allergy/index**

**It is not necessary to perform indexing every time a table changes occur. In fact, it is recommended to only perform indexing when necessary, such as when new records are added to the table or when changes are made to the search functionality.**

**Performing indexing too frequently can result in unnecessary resource consumption and slow down the overall performance of the application. It is important to only perform indexing when necessary and to ensure that the indexing process is optimized for performance.**

**GET**: http://localhost:9191/allergy/search?searchTerm=Pertussis&page=0&size=10