**Code System Search For CPT Documentation**

The CPT (Current Procedural Terminology) code system 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 CPT code system:

1. /cpt/search: This endpoint performs a full-text search on the Cpt\_Short code system using a search term. The search term is passed as a query parameter, and the search results are returned as a Page of CptCode objects. The Pageable parameter allows clients to specify the page size and offset for the search results. The search is performed using the searchCptCode() method of the CptCodeSearchService class, which is autowired into the controller.

2. /cpt/index: This endpoint creates an index for the Cpt\_Short code system. The index is created using the createCptIndex() method of the CptCodeSearchService class, which is autowired into the controller. The method returns a String message indicating whether the index creation was successful or not.

3. /cpt/category: This endpoint retrieves a hierarchical category of CPT codes, given a code. The code is passed as a query parameter, and the grouping is returned as a list of CptCodeGrouping objects. The Grouping is generated using the getCPTCategoryGrouping() method of the CptCodeSearchService class, which is autowired into the controller.

**Service:**

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

1. createCptIndex(): This method creates an index for CPT 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. searchCptCode(): This method searches the indexed CPT codes using Hibernate Search. It retrieves an instance of SearchSession using an EntityManager and executes a search query on the CptCode entity. It searches on the code and shortName 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.

3. getCPTCategoryGrouping(): This method retrieves a list of CPT code grouping nodes (majorcategory and minorcategory) for the given code using the CptCodeSearchRepository. It returns the list of nodes.

Overall, this service class provides functionality for searching and indexing CPT codes using Hibernate Search and retrieving a list of CPT code Grouping nodes.

**Repository:**

The CptCodeSearchRepository interface defines a repository that extends the Spring Data JpaRepository interface. The repository provides a method getCPTCategoryGrouping() that retrieves a list of CptCodeGrouping objects by calling a stored procedure getCPTCategoryGrouping in the database.

The @Query annotation is used to specify the SQL query to be executed by the getCPTCategoryGrouping () method.

The code placeholder in the query is replaced with the value of the code parameter when the method is called.

The CptCodeSearchRepository interface extends the Spring Data.

**URL’s:**

**POST:**

<http://localhost:9191/cpt/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/cpt/search?searchTerm=nose&page=5&size=10>

**GET:**

<http://localhost:9191/cptcode/searchwithctg?searchTerm=00100>