**List:**

The listAPI will return list of all URLs in JSON format.

The list API takes three Request parameters those are page, size and sortBy and pass those parameters to the getAllUrls method in service. The services there by uses this uses this parameter’s and create a Pageable object that in turn passes as a parameter to findAll method in the UrlRepository that in turn delegate the call to PagingAndSortingRepository which is one of the Spring Data Repository.

**Hierarchy:**

UrlRepository🡪 PagingAndSortingRepository🡪CurdRepository🡪Repository

We are using PagingAndSortingRepository because it by default support Paging and Sorting to our application.

**Helpers get URL:**

The helper method getUrl takes the string URL as input and return’s the Optional<UrlEntity> if the URL is Present else return Optional.empty().

**Store URL:**

The store URL API takes an URL as input and save the URL into the database if the URL is not present else update the URL.

The count API takes one Request parameter that is UrlEntity and call the helper method getUrl and if the URL already exists it will return the existing UrlEntity else it will save the URL into database and return the saved UrlEntity.

**Get:**

The getAPI takes an URL as a parameter and return the unique short key after incrementing the usage count.

The count API takes one Request parameter that is URL as string and call the helper method getUrl and if the URL exists it will return the unique short key of the URL else it will throw RecordNotFoundException.

**Count:**

The countAPI takes an URL as a parameter and should return the latest usage count.

The count API takes one Request parameter that is URL as string and call the helper method getUrl and if the URL exists it will return the usage count of the URL else it will throw RecordNotFoundException.