BooksModel.java for search bar to fetch books:

package com.myTraining.core.models;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.annotation.PostConstruct;

import javax.inject.Inject;

import javax.jcr.RepositoryException;

import javax.jcr.Session;

import org.apache.sling.api.resource.Resource;

import org.apache.sling.api.resource.ResourceResolver;

import org.apache.sling.models.annotations.Model;

import org.apache.sling.models.annotations.injectorspecific.Self;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import com.day.cq.search.PredicateGroup;

import com.day.cq.search.Query;

import com.day.cq.search.QueryBuilder;

import com.day.cq.search.result.Hit;

import com.day.cq.search.result.SearchResult;

import com.day.cq.wcm.api.Page;

import com.myTraining.core.beans.BooksList;

@Model(adaptables = Resource.class)

public class BooksModel {

    List<BooksList> bookArray;

    private static final Logger LOGGER=LoggerFactory.getLogger(BooksModel.class);

    @Self

    Resource resource;

    @Inject

    private String rootpath;

    @PostConstruct

    public void init(){

        ResourceResolver resourceResolver=resource.getResourceResolver();

        Session session=resourceResolver.adaptTo(Session.class);

        QueryBuilder builder=resourceResolver.adaptTo(QueryBuilder.class);

        Map<String,String> predicate=new HashMap<>();

        predicate.put("path",rootpath);

        predicate.put("type","cq:Page");

        Query query=null;

        try{

            query=builder.createQuery(PredicateGroup.create(predicate), session);

        }catch(Exception e){

           LOGGER.error("sql exception");

        }

        SearchResult searchResult=query.getResult();

        bookArray=new ArrayList<>();

        for(Hit hit:searchResult.getHits()){

            String path=null;

            BooksList bl=new BooksList();

            try{

                path=hit.getPath();

                Resource bookResource=resourceResolver.getResource(path);

                Page book=bookResource.adaptTo(Page.class);

                String title=book.getTitle();

                String description=book.getDescription();

                bl.setPath(path);

                bl.setTitle(title);

                bl.setDescription(description);

                LOGGER.debug("Title:{}, Description:{}",title,description);

                bookArray.add(bl);

            }catch(RepositoryException e){

                throw new RuntimeException(e);

            }

        }

    }

    public List<BooksList> getBookArray(){

        LOGGER.debug("booklist called");

        return bookArray;

    }

}

Book Title Model:

package com.myTraining.core.models;

import javax.inject.Inject;

import org.apache.sling.api.resource.Resource;

import org.apache.sling.models.annotations.DefaultInjectionStrategy;

import org.apache.sling.models.annotations.Model;

@Model(adaptables = Resource.class,defaultInjectionStrategy=DefaultInjectionStrategy.OPTIONAL)

public class BookPageModel {

@Inject

private String PageTitle;

public String getPageTitle(){

return PageTitle;

}

}

Styling of Pages:

#pageTitle{

padding-left:20px;

}

#home{

padding-top:10px;

a{

color:grey;

text-decoration:none;

}

:hover{

color:black;

font-weight:bold;

text-decoration:underline;

}

}

#aboutme{

padding-top:10px;

a{

color:grey;

text-decoration:none;

}

:hover{

color:black;

font-weight:bold;

text-decoration:underline;

}

}

#contactme{

padding-top:10px;

a{

color:grey;

text-decoration:none;

}

:hover{

color:black;

font-weight:bold;

text-decoration:underline;

}

}

#searchBook{

padding-top:20px;

}

.cmp-container{

background-color:white;

}