***GET ALL AUTORE’S NAME ASSOCIATED WITH A RICETTA***

public List<String> getAllAutori() {

String queryStr = "SELECT {nome AS AutoreDiUnaRicetta} FROM {Autore AS a} JOIN {Ricetta AS r} ON {r.autore= a.id}";

FlexibleSearchQuery fsq = new FlexibleSearchQuery( queryStr );

SearchResult <String> result = getFlexibleSearchService().search( fsq );

List<String> autori = result.getResult();

return autori;

}

***GET ALL RECIPES IN Ricetta TABLE AS OBJECTS***

public List<RicettaModel> getAllRicette() {

String queryStr = "SELECT {PK} FROM {Ricetta} ";

FlexibleSearchQuery fsq = new FlexibleSearchQuery( queryStr );

SearchResult<RicettaModel> result = getFlexibleSearchService().search( fsq );

List<RicettaModel> ricette = result.getResult();

return ricette;

}

***RETURN ALL AUTORI FOR ALL RECIPES’ NAME THAT STARTS WITH “Pasta”***

public List<AutoreModel> getAllAutoriAboutPasta() {

String queryStr = "SELECT {PK} FROM {Autore AS a} JOIN {Ricetta AS r} ON {r.autore= a.id} WHERE {r.nome} LIKE ‘Pasta%’;

FlexibleSearchQuery fsq = new FlexibleSearchQuery( queryStr );

SearchResult<AutoreModel> result = getFlexibleSearchService().search( fsq );

List<AutoreModel> autori = result.getResult();

return autori;}

***RETURN ALL OF INGREDIENTI (AS OBJECT) THAT WAS USED TWICE IN A SPECIFIC RICETTA***

public List<IngredienteModel> getAllIngredientiRidondanti(Integer ridondanza) {

String queryStr = "SELECT {PK} FROM {Ingrediente AS i} JOIN {RecipeEntry AS r} ON {r.ingrediente= i.id} WHERE {r.quantita}> =?ridon;

FlexibleSearchQuery fsq = new FlexibleSearchQuery( queryStr );

fsq.addQueryParameter("ridon", ridondanza);

SearchResult<IngredieteModel> result = getFlexibleSearchService().search( fsq );

List<IngredienteModel> ingredienti = result.getResult();

return ingredienti;

}

***GET NOME AND CALORIE OF ALL RECORD OF RICETTA, AND PUT THEM IN A List<Object>***

public List< List <Object> > getNomeECalorie(){

String queryStr = "SELECT {r.nome AS NomeRicetta}{ r.calorie AS Calorie} FROM {Ricetta AS r}";

FlexibleSearchQuery fsq = new FlexibleSearchQuery( queryStr );

Class[] resultTypesArray = { String.class, Integer.class };

fsq.setResultClassList( Arrays.asList( resultTypesArray ) );

SearchResult < List<Object>> result = getFlexibleSearchService().search(fsq);

List< List <Object> > resultRowList = result.getResult();

return resultRowList;

}