Chapter 2: User-Facing Backend

Ticket: Get Comments

‹ Back to the Question

Here is a possible solution:

COPY

```
private Bson buildLookupStage(){
  List<Variable<String>> let = new ArrayList<>();
  let.add(new Variable<String>("id", "$_id"));

  // lookup pipeline
  Bson exprMatch = Document.parse("{'$expr': {'$eq':
['$movie_id', '$$id']}}");

  Bson lookupMatch = Aggregates.match(exprMatch);
  List<Bson> lookUpPipeline = new ArrayList<>();
  // lookup sort stage
  Bson sortLookup = Aggregates.sort(Sorts.descending("date"));

  lookUpPipeline.add(lookupMatch);
  lookUpPipeline.add(sortLookup);
  return Aggregates.lookup("comments", let, lookUpPipeline,
"comments");
}


public Document getMovie(String movieId){

  if (!validIdValue(movieId)) {
   return null;
 }

 List<Bson> pipeline = new ArrayList<>();
 // match stage to find movie
 Bson match = Aggregates.match(Filters.eq("_id", new
ObjectId(movieId)));
 pipeline.add(match);

 // comments lookup stage
 Bson lookup = buildLookupStage();
 if(lookup != null) {
   pipeline.add(lookup);
 }

 Document movie = moviesCollection.aggregate(pipeline)
         .batchSize(1)
         .iterator().tryNext();
 return movie;
}
```

Proceed to next section