

**High level overview:**

1. LoadInputQuery.java is the main class that accepts api and movie paramters
2. The input is validated using LoadInputQueryValidatorimpl
3. Then based on the type – configured in applicationContext.xml and in API.java, the user request is passed on to the corresponding handler i.e either InputQueryMovieHandlerImpl or InputQueryMusichandlerImpl.
4. The handlers then use ApliCallServiceImpl to call the corresponding api and obtain results in JSON format
5. InputQueryMovieHandlerImpl uses the HelperApiHandler to get all the movie related information.
6. The response from ApliCallServiceImpl is then printed to console.

**Improvements:**

1. I used [www.jsonschema2pojo.org](http://www.jsonschema2pojo.org) to generate models from JSON. So they need to be refactored to move all the common fields to an Abstract model.
2. By accomplishing #1 the printThResult method in both the handlers can be moved to AbstractQueryhandler there by having only one copy of printThResult
3. More research can be done to find the movie api that gives all the information there by reducing the call to HelperApiHandler.
4. More Junit tests can be created to cover all the methods and to increase the code coverage.
5. Mockito could have been used
6. Factory pattern could have been used rather than configuring spring for deciding which handler to use.
7. Java 8 lamda could have been used.
8. -Dmovie accepts both movie and music. It is better to use and handle the parameter music when querying music

**Note** :- Due to time constraint I am unable to incorporate the improvements from #1 to #8