

Code Assessment Instructions

We'd like to invite you to do a code assessment, please read these instructions carefully to ensure that you implement everything as expected. after completing the assessment, you can push the project to GitHub or send the source code to us, may need to you explain your approach and choices in the project then you'll give a demo of your app.

This code assessment is not a puzzle or solve the performance issues; it's not meant to be tricky or confusing. it is a simple and typical project just for reviewing your general knowledge in design and developing a real-world RESTful application.

Project Specifications:

The deadline for the project is one week.

The project must be developed by Java8 or later, spring boot, any in-memory database, Junit, Swagger and any other framework or library that is needed for completing the project.

The project must be cover and implement these concepts on a standard scale(not too deep and not too simple, in a balanced way):

- Clean code
- Multi-tier architecture(such as MVC)
- Unit and integration testing
- Logging
- Exception handling
- Documentation

Application Overview:

This application is a single microservice that is responsible for reading and providing the data of posts, comments, and to-do's from a single data source, manipulating and providing these data to other applications by REST API.

Basic Requirements:

1-The application must catch data of the posts and comments and to-do's from these below data sources at the application start-up in a separated thread and persist data into corresponding tables in the local application database:

Posts: <https://jsonplaceholder.typicode.com/posts>

Comments: <https://jsonplaceholder.typicode.com/comments>

ToDo's: <https://jsonplaceholder.typicode.com/todos>

Hint:

- you can find the API samples and guide of each section in the : <https://jsonplaceholder.typicode.com>
- Data catching must be done by a separated thread and error safe, if there is any issue in this operation, app must return a proper and standard response when app API's are called.

2-The application must provide three REST API for Posts, Comments, ToDo's like below:

Posts API's:

GET		Get all posts(with pagination)
GET	/posts/1	Get post by post id
GET	/posts/1/comments	Get comments of specific post by post id
GET	/posts?title=eos	Get all posts that have the "eos" keyword in their title
POST		Create a post
PATCH	/posts/1	Update a post by post id
DELETE	/posts/1	Delete a post by post id

Comments API's:

GET		Get all comments(with pagination)
GET	/comments?postId=1	Get comments of specific post by post id
POST		Create a comments
PATCH	/comments/1	Update a post by post id
DELETE	/comments/1	Delete a post by post id

ToDo's API's:

GET		Get all to-do's
GET	/todos?userId=1&completed=true	Get to-do's of specific user by user id and completed field

Hint:

- You must return the Standard HTTP response&code that existed in java and spring boot for API's.

- You must handle exceptions and return a proper error response when exceptions occurred in every layer of app.
- You must log the CRUD operations(debug or info logs).
- Each service methods must be have a unit test(at the least one unit test for success and failure situation)
- An integration test for data source API is a good idea but it is optional.
- Please document the application API specifications by using the swagger framework.
- Application must have a README file

If you have any questions do not hesitate to contact us, we'll be glad to answer them .

Good luck 😊.