# Test task for a backend developer

## Summary

A web service (HTTP REST API) to work with tasks.

## Glossary

*A Task* is an object with name, description, last modification date, and a unique identifier.

## Details

Operations to be provided by the web service:

* Creating a task.
* Getting a task by its identifier. In the response we get a complete description of the task.
* Change task data by its identifier. In the response we get an updated full description of the task.
  + You cannot change task id.
  + You cannot delete task attributes.
  + All changes to tasks must occur atomically. That is, if we change the *name* of the task, then we should not get an intermediate state during concurrent reading.
* Deleting a task. We can delete the task by its identifier.
* Getting a list of tasks. In the response we get a list of all tasks sorted by modification date.

## Requirements

* The API must conform to the REST architecture.
* Do only the server side, you don’t need to do visualization.
* It should be a Spring Boot application.
* Maven or Gradle should be used as a build tool.
* Data should only be stored in memory. You can use any classes of the standard Java library to organize storage. It is not allowed to use any external repositories and databases.
* At least 30% of the code should be covered by tests (preferably the presence of both unit and integration tests).
* Submit sources via a public git repository.

## Complication

All complications are optional — you may or may not do them. You can pick any of them.

### 1. Pagination Implement a pagination mechanism for a list of tasks, which by default will produce 10 elements, but with the ability to specify a different number (up to 500).

### 2. SQL database You need to create another storage implementation that works with SQL databases. You can use any database in-memory implementation (for example, H2).

The choice of storage implementation (native or in-memory SQL) should be determined in the server configuration and does not require changes in the code.