GPS online tracking services

1. API endpoints

STT	End point	path	Method	Request data	Response data	ErrorCode
1	Upload "gpx" file	/gpx/upload	Post	MultipartFile: Gpx (XML) file format	Empty	Http_code: 200
2	Listing all gpx	/gpx	Get		Array of Gpx Object DataType: Json	Http_code: 200
3	Get a gpx by id	/gpx/{id}	Get	id: 1,2,3 …	Gpx Object DataType: Json	Http_code: 200
4	Listing latest tracking point	/trkpt/latestTrack	Get		Trkpt Object DataType: Json	Http_code: 200

Upload page: http://localhost:8080/gpx/upload

2. Database structure

tlb_gpx --- tbl_metadata: 1-1

tlb_gpx --- tbl_trkpt: 1-N

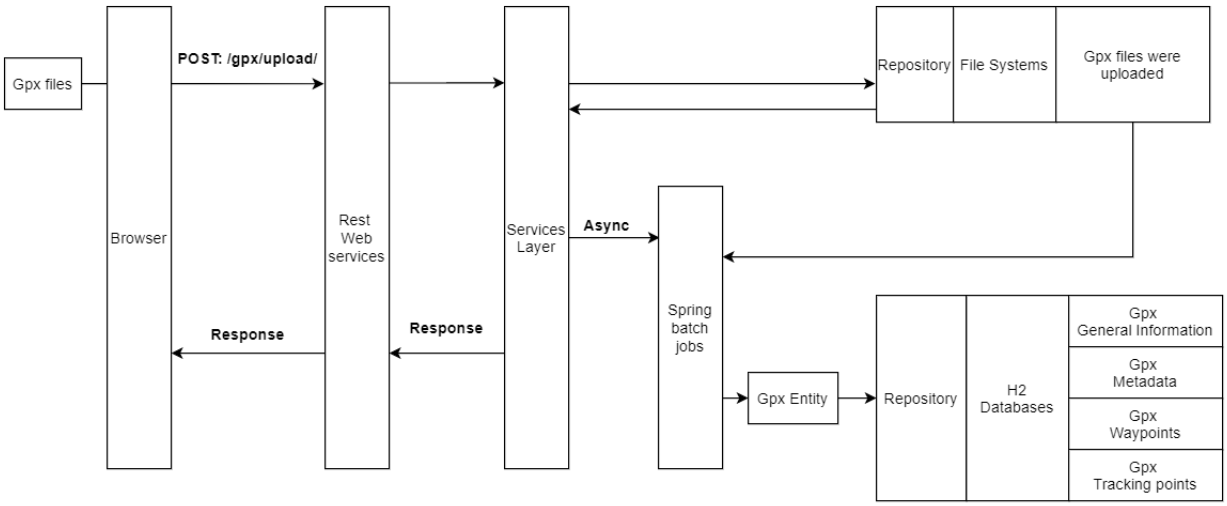
tlb_gpx --- tbl_wpt: 1-N

Note: Finding listing tracking point:

- Finding the largest GpxId then select tracking point with the GpxId found.
- Speed up getting latest tracking point by set B-tree index on gpx field.

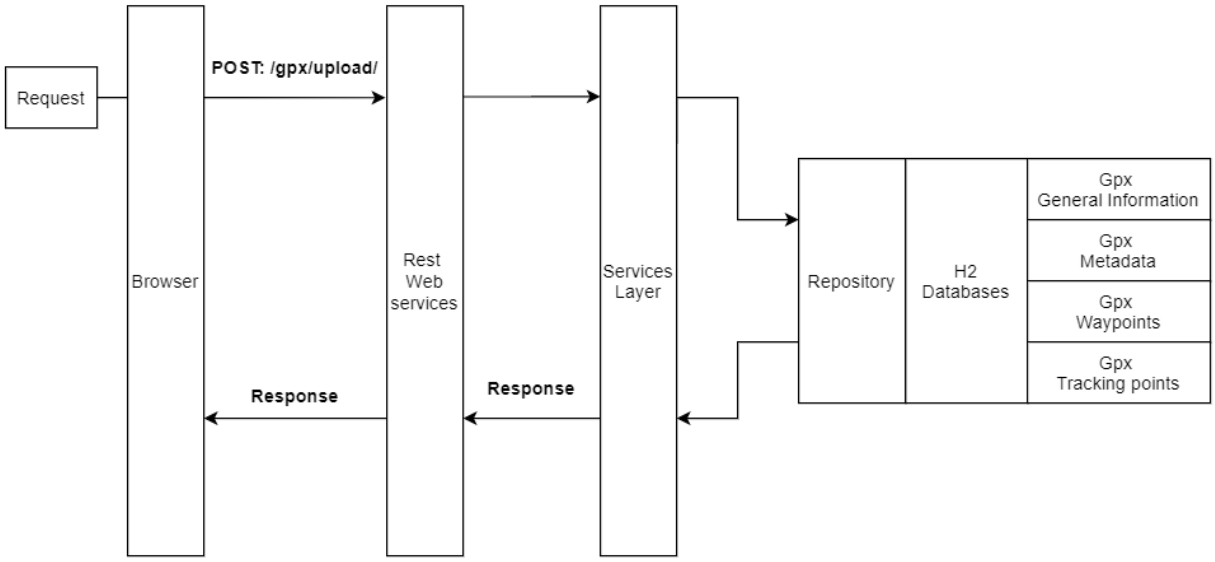
3. Workflows

a. Upload Gpx file



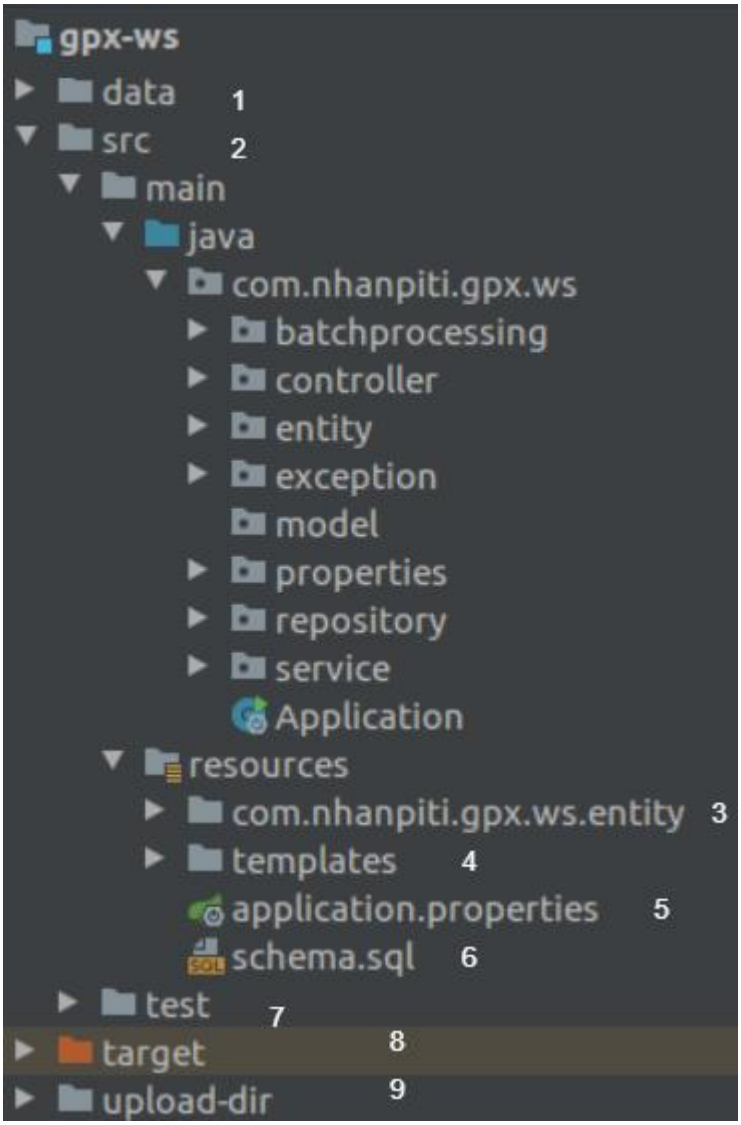
Note: File systems will be deleted every time the service restarts.

b. Taking Gpx info / latest tracking point



Note: Speedup getting Gpx info / latest tracking point by caching them in internal cache (Guava cache / Spring cache) or server cache (Redis, Halzecast…) and eviction when Gpx info was changed.

4. Project Structure



1 – Sample / test data

2 – Main class folder

3 – Xml.blind properties

4 – Upload page html

5 – App properties

6 – Datebase schema

7 – Test class folder (1 logic test, 2 integration test)

8 – Build result folder

9 – Gpx file uploaded

JDK: 11

MainApp: Application.class

Project command: maven(mvn) run / test / verify.