



Veebirakenduse projekt (IT10302)

Individual assignment 1

Siim Rebane

Individual assignment 1

- Deadline: 5th week Sunday 23:59
- Individual work, done in your **iti0302-2026** repo
 - Feature branches and merge requests for everything
 - Merge requests made before your CI pipeline exists (weeks 1–2) are fine without it
 - After you have submitted this assignment, a green pipeline before merging is expected going forward
- Using LLMs is allowed
 - You have to understand everything that you submit
- No separate defense for the individual assignment
 - Questions can be asked about your homework in the final defense
 - If you cannot explain your solution in the defense, the assignment grade may be reduced or nullified
 - Theoretical questions can be asked about this subject in the test

Task

- Build a small REST application and get it running on **your server**
- Storage: in-memory (a list or a map) is enough — **no database required**
- The application must run in **Docker** on your server
- A **GitLab CI pipeline** must build **and deploy** the application to your server on every push — and be green
- Your resource depends on the **last digit of your student code** (next slide)

Your variant

Last digit of your student code (the digit before the letters), e.g. 22345**7**IAIB → resource `teams` :

Digit	Resource	Extra field
0	<code>books</code>	<code>pageCount</code>
1	<code>movies</code>	<code>durationMinutes</code>
2	<code>songs</code>	<code>lengthSeconds</code>
3	<code>games</code>	<code>playerCount</code>
4	<code>recipes</code>	<code>cookingTimeMinutes</code>
5	<code>courses</code>	<code>credits</code>
6	<code>plants</code>	<code>wateringIntervalDays</code>
7	<code>teams</code>	<code>memberCount</code>
8	<code>events</code>	<code>capacity</code>
9	<code>gadgets</code>	<code>batteryLifeHours</code>

Required endpoints

- `GET /api/{resource}` → list of all, 200
- `GET /api/{resource}/{id}` → 200, or 404 if it doesn't exist
- `POST /api/{resource}` → body without id, returns **201** with the created object (server assigns the id)
- `DELETE /api/{resource}/{id}` → **204**, or 404 if it doesn't exist
- `GET /api/{resource}?name=x` → list filtered by name, 200
- `GET /api/version` → `{ "commit": "abc1234" }` — the git commit SHA the running app was built from (we wire this up together in the week 3 lab; until then a fallback like `{ "commit": "local" }` is fine when running locally)
- URLs must follow REST conventions — plural nouns, no verbs

DTO

- Fields: `id` (number), `name` (string) and your variant's extra field (number)
- Use correct JSON types — numbers must not be strings
- Example for digit 7 (`teams` / `memberCount`):

```
{ "id": 1, "name": "Meeskond A", "memberCount": 3 }
```

Deliverable

- Link to your **iti0302-2026** repo
 - Git history matters: feature branches and merge requests, green pipeline
- URL where the application answers on your server
- Add **siimrebane** as a member of the repo with the **Maintainer** role (needed for checking)
- CI pipeline visible and green in GitLab
- The check is done against **exactly** the endpoints listed here
 - If `/api/team/5` answers instead of `/api/teams/5`, the check fails
 - Wrong status codes fail the check

How this is checked

- After the deadline, a script checks every submission the same way:
 - i. Reads `/api/version` on your URL
 - ii. Pushes a small commit to your repo
 - iii. Waits up to 15 minutes for `/api/version` to show the new commit
 - iv. Calls exactly the endpoints on the previous slides and checks paths, status codes and JSON types
- If your server is down or the pipeline is red at check time, the submission fails — the normal late rules apply
- You can run the same checks yourself before submitting — test your own URL
- Passing the automated check is **required, but not sufficient** — the repo, git history and merge requests are reviewed by hand