



Three-Kid Family Challenge

In this challenge we want to figure out if you have what it takes to be a software engineer for PinkRoccade Local Government. Please read the entire exercise before you start. We estimate this challenge to take 2 hours, but it is up to you to decide how long you spent on the challenge (but please disclose how long you spent when submitting).



The Goal

Your service receives a number of requests containing **person records**.

After **each** record arrives it must decide, *immediately*, whether **any** person now stored satisfies this pattern:

1. Has a partner
2. Has **exactly 3 children** and **all 3** have that same partner listed as mother *or* father
3. **At least one** of those children is **under 18**

If the pattern is satisfied for at least one person, reply with **HTTP 200 OK**; otherwise reply with **HTTP 444 No Response**.



Rules & tool-belt

- **Java 21+** — preferably Spring Boot.
- All filtering in Java — no SQL WHERE tricks.
- **Bidirectional integrity** — if A says B is a child, B must list A as a parent (enforce or auto-repair).
- Include unit & integration tests



Evaluation

Submit a link to your public repo and please make small commits.

Your submission is evaluated on:

1. **Clean code** – we'll look at architecture, tests, and docs
2. **Usage of Git** - we like to see how you work
3. **Performance** - we estimate what your application would do under load



Phase 1 - Input

You'll receive POST requests to `/api/v1/people`

```
{
  "id"      : 42,
  "name"    : "Ada Lovelace",
  "birthDate": "1815-12-10",
  "parent1" : { "id": 1 },
  "parent2" : { "id": 2 },
  "partner"  : { "id": 99 },
  "children" : [{ "id": 77 }, { "id": 78 }]
}
```

- **One** person per request
- IDs may reference parents/partner/children that have never been seen before.
- If we POST the same ID again, treat it as an **update** (replace old data).

Response

- 200 OK – ≥ 1 stored people currently matching
- 444 No Response – no one matches **yet**

The response should consist of all the people currently matching the pattern.
We should respond with everything we know about the matching people (you might not know their name or birthdate yet).

✨ Phase 2 — Purge & ignore-list (Only if you have time)

You'll receive DELETE requests to `/api/v1/people`

```
[
  17, 99, 2137
]
```

you must **delete** those IDs *and* remember them forever: if they show up later, silently ignore them for both storage **and** matching.