

# Component

- Bean générique

```
@Component
public record Context(List<EventRecord> events, List<PersonRecord> persons) {
    |
    public Context() {
        this(new ArrayList<>(), new ArrayList<>());
        events().add(new EventRecord(UUID.randomUUID(), name: "Mud Day", LocalDate.now().toString()));
        events().add(new EventRecord(UUID.randomUUID(), name: "Vide grenier", LocalDate.now().toString()));

        persons().add(new PersonRecord(UUID.randomUUID(), firstName: "Bill", lastName: "Gates", events.get(0)));
        persons().add(new PersonRecord(UUID.randomUUID(), firstName: "Steve", lastName: "Jobs", events.get(1)));
    }
}
```

# Service

- Composant de gestion
- Exécution du code métier

```
@Service
public class PersonService {

    @Autowired
    private Context context;
    @Autowired
    private PersonRepository personRepository;
    @Autowired
    private EventRepository eventRepository;
    @Value("false")
    private Boolean withContext;

    public void delPerson(UUID id) {
        if (withContext) {
            List<PersonRecord> personsFiltered = context.persons() List<PersonRecord>
                .stream() Stream<PersonRecord>
                .filter(p -> p.id() != id)
                .collect(Collectors.toList());
            context.updatePersons(personsFiltered);
        } else {
            personRepository.deleteById(id);
        }
    }
}
```