C4 - Generic MS Code Blueprint (applies to all MS)

Microservice - Code View

Configuration

SecurityConfig

«@Configuration»
SecurityFilterChain security(HttpSecurity):
SFC

OpenApiConfig

«@Configuration»
OpenAPI openAPI(): OpenAPI

GlobalExceptionHandler

«@ControllerAdvice»
- handle(MethodArgumentNotValidException): ResponseEntity<ApiErrorDto>
- handle(RuntimeException): ResponseEntity<ApiErrorDto>

KafkaConfig(if apply)

«@Configuration»
- ProducerFactory<?,?> producerFactory()
- KafkaTemplate<?,?> kafkaTemplate()
- ConsumerFactory<?,?> consumerFactory()
- ConcurrentKafkaListenerContainerFactory<?,?> listenerFactory()

application.yml

spring.datasource.url=jdbc:postgresql://...
spring.jpa.hibernate.default_schema=<schema>
spring.kafka.bootstrap-servers=...

API Client/Gateway

JWT Filter

«OncePerRequestFilter»
doFilterInternal(req,res,chain)

DataSource + default_schema

ExampleController

«@RestController»
- POST /v1/... → create(@RequestBody RequestDto):ResponseEntity<ResponseDto>
- GET /v1/{id} → getById(@PathVariable id): ResponseEntity<ResponseDto>
- GET /v1 → list(@RequestParam ...): ResponseEntity<Page<ResponseDto>>
- PUT/PATCH /v1/{id} → update(id, RequestDto): ResponseEntity<ResponseDto>
- DELETE /v1/{id} → delete(id): ResponseEntity<Void>

ExampleService

«@Service»
- create(dto): ResponseDto
- getById(id): ResponseDto
- list(query): Page<ResponseDto>
- update(id,dto): ResponseDto
- delete(id): void

ExampleRepository

«Spring Data JPA Repository interface»
- save(entity): ExampleEntity
- findById(id): Optional<ExampleEntity>
- findAll(spec/page):Page<ExampleEntity>
- deleteById(id): void

investment_db

schema=<schema>

EventProducer (if apply)

«@Component»
send<Event>(payload): void

EventConsumer (if apply)

«@KafkaListener»
on<Event>(message): void

ExampleEntity

@Table(schema=..., name=...
- id: UUID
- createdAt: Instant
- updatedAt: Instant

RequestDto

ResponseDto

ApiErrorDto

Mapper

«MapStruct @Mapper»
- toModel(dto):ExampleModel
- toDto(model): ResponseDto

ExampleModel

«domain model»
- validate()
- applyBusinessRules()
- calculate()

Kafka Broker

Legend:
Colors → White=internal, DarkGray=configuration, LightGray=application.yml, Yellow=database/Kafka
Lines → Solid=dependency, Dashed=configuration/security
Flow → Controller→Service→Repository→Entity →Database; Controller⬄DTO; Service⬄Mapper⬄Model
Kafka → Service→Producer→Broker; Broker→Consumer

Author: Remus-Ciprian Cotunoaea
Version: 1.0
Date: 09/09/2025
Notation: C4 Container Diagram