

QUARKUS/[COURSES]

COMPLETE SYLLABUS

Master Production-Grade Microservices
Build Real-Time IoT Platforms

10

MODULES

40+

VIDEOS

10+

SERVICES

START YOUR JOURNEY

QUARKUS ESSENTIALS

 **GOAL:** Ground students in Quarkus before touching the project.



VIDEOS

V0.1 – What is Quarkus?

Quarkus design philosophy, JVM vs native, dev mode, Live Reload, Dev Services.

V0.2 – CDI with Arc

Bean scopes, injection, producers/disposers, lifecycle management.

V0.3 – Config the Quarkus Way

SmallRye Config, @ConfigMapping, build-time vs runtime props, secrets.

V0.4 – RESTEasy Reactive Basics

REST resources, @GET/@POST, JSON-B vs Jackson, exception mappers.

V0.5 – Health & Metrics Quickstart

SmallRye Health, Micrometer, /q/health, /q/metrics endpoints.



LAB

Create hello-world app, add REST endpoint + health check, run in dev mode, watch live reload, inspect metrics.

PROJECT BOOTSTRAP

 **GOAL:** Stand up the monorepo and establish conventions.



VIDEOS

V1.1 – Monorepo Layout

Structure services/, shared/, infrastructure/, tests/ directories.

V1.2 – Quarkus CLI & Extensions

Select extensions for small footprint and native readiness.

V1.3 – Profiles & Secrets

Configure dev/test/prod profiles and safe secret handling.



LAB

Skeleton repo with shared DTOs (positions/zones). Add OpenAPI plugin and verify /q/openapi endpoint.

SIMULATOR SERVICE

 **GOAL:** Master REST Client Reactive, scheduling, config, fault tolerance by building a simulator.



VIDEOS

V2.1 – Creating the Simulator

Extensions: rest-client-reactive-jackson, scheduler, config-yaml.

V2.2 – Config Mapping

@ConfigMapping to bind YAML to typed configs, inject Clock.

V2.3 – Movement Patterns

Random walk, routes (circle/8/patrol), states, speed, noise.

V2.4 – Schedulers & Backoff

@Scheduled producers, retries with SmallRye Fault Tolerance, circuit breaker.

V2.5 – REST Client Reactive

Post positions to Gateway, timeouts, connection pools, JWT headers.

V2.6 – Observability

Counters/timers, health checks, correlation IDs, structured logs.

SIMULATOR SERVICE



VIDEO

V2.7 – Native Build Sanity

Reflection/resource config, build <50 MB images, startup <100ms.

KEY SNIPPETS

```
quarkus create app io.suddo:ipt-simulator \
  --extension='rest-client-reactive-jackson,scheduler,config-yaml'

@ConfigMapping(prefix="simulator")
interface SimConfig {
    List<Pet> pets();
    FloorPlan floorPlan();
}

@Scheduled(every="200ms")
void tick() {
    generator.next()
        .onItem().transformToUni(this::send)
        .subscribe().with(...);
}
```



LAB

Run in dev mode, post to stub endpoint, expose /q/metrics and verify counters. Produce native image and confirm <100ms startup.

HTTP INGEST

 **GOAL:** Master REStEasy Reactive, validation, idempotency, and Kafka production.



VIDEOS

V3.1 – API Gateway Routing

REST Clients for downstream services, timeout/pool tuning, CORS, filters.

V3.2 – Designing Ingest APIs

POST /api/v1/positions, Bean Validation 3.0, problem+json errors.

V3.3 – Idempotency & Logging

Idempotency keys, correlation IDs, log sanitization.

V3.4 – Native Performance

JSON reflection hints, TLS support, slim images.

KEY SNIPPETS

```
quarkus create app io.suddo:ipt-gateway \
  --extension='rest,rest-jackson,rest-client-jackson'

quarkus create app io.suddo:ipt-ingest \
  --extension='resteasy-reactive-jackson,smallrye-reactive-messaging-kafka'
```


KAFKA MESSAGING

 **GOAL:** Wire ingest to Kafka. Learn channels, serializers, error handling, throughput tuning.



VIDEOS

V4.1 – Channels & Config

@Outgoing("raw-positions"), serializers, partitions/keys, config properties.

V4.2 – Producer Throughput

acks, linger.ms, compression, batching, p95/p99 measurement.

V4.3 – Errors & DLQ

Nacks, retries/backoff, dead-letter topics, poison messages.

V4.4 – Tracing & Metrics

Trace context propagation, consumer lag in Grafana, Kafka client metrics.



LAB

Push 10k msg/s from simulator, verify topic keys/compression, break Kafka to observe DLQ behavior.

STATEFUL PROCESSING

 **GOAL:** Build stateful consumer with Caffeine caches, zone sync, enrichment, alerts.



VIDEOS

V5.1 – Project Setup

@Incoming raw-positions/zone-updates, @Outgoing enriched/alerts.

V5.2 – Caffeine Caching

Pet LRU + TTL, zone cache warm-up on startup, cache metrics dashboards.

V5.3 – Enrichment Pipelines

Velocity/direction, point-in-polygon, dwell time, hysteresis for alerts.

V5.4 – Multi-Topic Output

Emit enriched + alerts, ordering and idempotence concerns.



LAB

Warm zone cache from Zone Service, trigger danger-zone breach, observe alert emission to zone-alerts topic.

MONGODB WRITER

 **GOAL:** Reactive Mongo client, time-series collections, buffered batch writes.



VIDEOS

V6.1 – Reactive Client & POJOs

Codecs/POJOs, projections, time-series with timeField/metaField.

V6.2 – Batch Writes

Buffer with Multi.buffer(), insertMany, graceful shutdown flush.

V6.3 – Indexing & Retention

TTL indexes, {petId, timestamp}, write patterns for read optimization.



LAB

Sustain 100k positions/sec into Mongo with batching. Track batch size and write latency metrics.



PERFORMANCE TARGET

By Module 6: 100K events/sec ingestion, in-memory enrichment, MongoDB persistence—all with sub-100ms p99 latency!

REAL-TIME STREAMS

 **GOAL:** Expose live event streams via SSE and WebSockets with backpressure handling.



VIDEOS

V7.1 – WebSockets in Quarkus

Endpoint lifecycle, session registry, ping/pong, filter by petId, MessagePack.

V7.2 – SSE for Browsers

text/event-stream, reconnect, keep-alive, event types for positions/alerts.

V7.3 – Backpressure

Slow clients: drop vs buffer, quotas/limits, fan-out approaches.



REAL-TIME MAGIC

Watch your web UI come alive with sub-second position updates. This is where backend work pays off with stunning visualizations!

OBSERVABILITY

 **GOAL:** Complete visibility from simulator to database with production dashboards.



VIDEOS

V8.1 – Metrics Everywhere

Micrometer counters/timers, percentiles/exemplars, consistent naming, /q/metrics.

V8.2 – Distributed Tracing

OpenTelemetry integration, manual spans, baggage/trace context, Jaeger visualization.

V8.3 – Structured Logging

JSON logs, correlation IDs, log levels per package, ELK integration.



LAB

Build Grafana dashboard: ingestion rate, processing latency p50/p95/p99, cache hit/miss, consumer lag, MongoDB throughput, WebSocket connections, JVM metrics.



PRODUCTION CONFIDENCE

Full observability = 10x faster debugging. This module alone is worth the entire course price!

NATIVE & CI/CD

 **GOAL:** Production-ready native images, minimal containers, automated security scanning.



VIDEOS

V9.1 – Mandrel 25 & Native Tips

Build-time init, reflection/resources config, native TLS, shrink to <50MB.

V9.2 – Docker Multi-Stage

Micro runtime images, distroless base, UPX pros/cons, security best practices.

V9.3 – CI Pipeline

Cache native builds, test matrix, Trivy scanning, Kubernetes deployment.



LAB

Build native images for all services. Verify <100ms startup and memory targets. Deploy complete system to Kubernetes.



FINAL METRICS

Startup: <100ms • **Memory:** 30-80MB RSS • **Image:** <50MB • **Throughput:** 100K/sec • **Latency:** p99 <100ms

READY TO MASTER QUARKUS?



WHAT YOU'LL BUILD

10 microservices • Event-driven architecture • 100K events/sec •
Sub-100ms latency • Native images <50MB • Real-time WebSockets •
Production observability



CAREER IMPACT

Build a production-grade IoT platform processing millions of
events per day—the exact skills companies pay \$150K+ for.

ENROLL TODAY

quarkus.courses/enroll

Questions?

Contact Instructor Leonid Herasimau
leonid@suddo.io