Thorntail

A Micro Implementation of Eclipse MicroProfile

Michał Szynkiewicz
Senior Software Engineer, Red Hat

Eclipse MicroProfile

THORNTAIL

MICROPROFILE™
OPTIMIZING ENTERPRISE JAVA

- Community-driven **open source** specification for enterprise Java **microservices**
- Industry Collaboration - Red Hat, IBM, Payara, Tomitribe, London Java Community, SouJava, Oracle, Hazelcast, Fujitsu, Lightbend, Microsoft…
- Minimum footprint for Enterprise Java cloud-native services (v2.1) :

JSON-P 1.1	JSON-B 1.0	JWT Auth	Config	OpenAPI	REST Client
CDI 2.0	JAX-RS 2.1	Fault Tolerance	Metrics	Open Tracing	Health Check

THORNTAIL

redhat.

MP State and Roadmap

MicroProfile 1.4

- Config 1.3, Fault Tolerance 1.1, JWT 1.1, OpenTracing 1.1, RestClient 1.1, Metrics 1.1, OpenAPI 1.0, Health Check 1.0,
- and JAX-RS 2.0, CDI 1.2, JSON-P 1.0

MicroProfile 2.0 and 2.1 (Oct 2018)

- Update to CDI 2.0, JAX-RS 2.1, JSON-P 1.1, JSON-B 1.0
- OpenTracing 1.2 (MP 2.1)

MicroProfile 2.2 (Feb 2019):

- Fault Tolerance 1.2, Metrics 2.0, Health Check 1.1, Config 1.4, REST Client 1.2
- Possibly: Reactive Messaging 1.0, Reactive Streams 1.0

Further future: LRA, Concurrency, Service Mesh

MicroProfile Config

- Provides means for configuring an application through properties file, system environment and more
- Values injected via CDI with `@Inject @ConfigProperty(name = “xxx”)` or `ConfigProvider#getConfig().get…`
- Providing configuration
  - Default values come from `META-INF/microprofile-config.properties`
  - They can be overwritten by environment variables
  - Which can be overwritten by Java system properties
  - Custom config sources, custom type converters

MicroProfile Rest Client

- Build REST client based on interface with JAX-RS annotations
- Built on top of JAX-RS 2.0 Client API
  - JAX-RS providers: filters, converters, interceptors, etc can be registered
- `ResponseExceptionMapper` maps response to a client exception
  - If a response is to be translated to a checked exception, `throws` on the JAX-RS method needs to be declared
- Can be asynchronous

```
@Inject
@RestClient
Resource client;
```

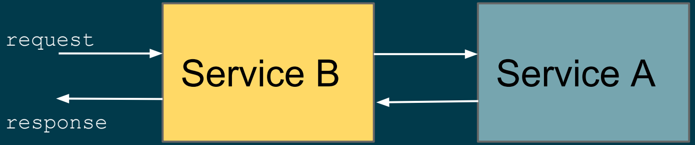
```
RestClientBuilder.newBuilder()
        .baseUrl(url)
        .build(Resource.class)
```

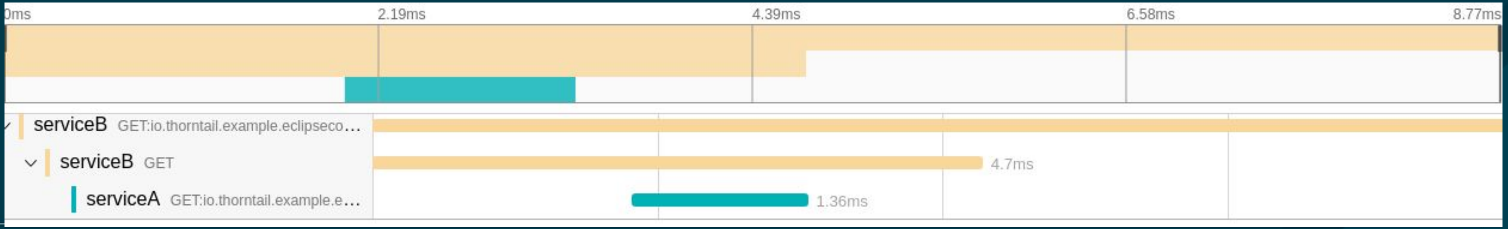
THORNTAIL

redhat.

MicroProfile OpenTracing

- Gives insight on what happens with requests
- Can record all requests or a sample
- Can work with Jaeger, Hawkular, Datadog and more
- Customizable with `@Traced`
- Gotchas:
  - Requires a server-specific dependency
  - Doesn't work out of the box with MP RestClient




0ms
2.19ms
4.39ms
6.58ms
8.77ms
serviceB GET:io.thorntail.example.eclipseco...
serviceB GET
4.7ms
serviceA GET:io.thorntail.example.e...
1.36ms

MicroProfile OpenAPI

- OpenAPI is a standard for REST API documentation, originated from Swagger
- Generate API documentation from code or provide static OpenAPI definition
- Can be leveraged to generate a client or generate an interactive UI to play with API (e.g. with Swagger UI)
- Sensible defaults
- Customizable with annotations
- Additionally, the API description can be configured or filtered programmatically
- Exposed on /openapi

MicroProfile JWT Authentication

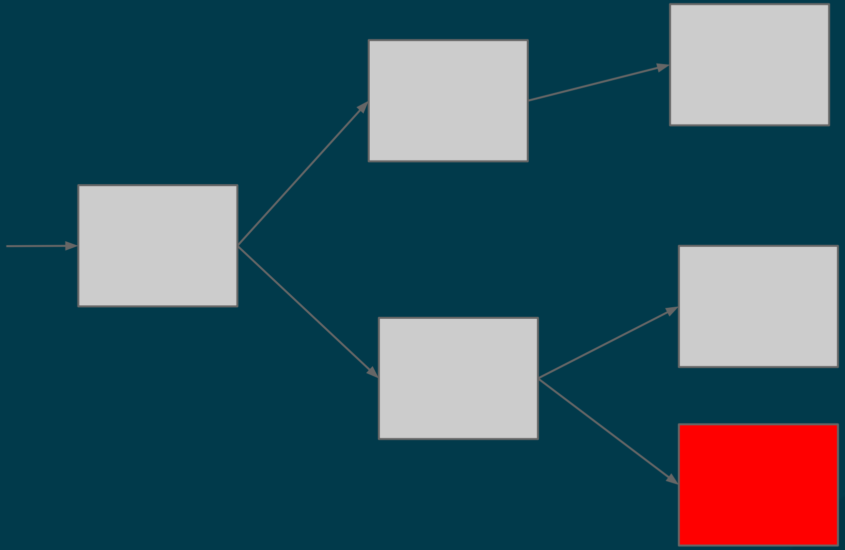
- JSON Web Token Authentication
  - Authenticate once and pass a signed token to all services
  - Can be verified without a call to auth server
  - Can carry additional information
- Use Common Security Annotations to secure your app
  - `@RolesAllowed`, `@PermitAll`, `@DenyAll`
- `@Inject` whole token or some of its fields (`@Claims`)
- Use JAX-RS SecurityContext to check for roles or get a hold of the token/principal

MicroProfile Metrics

- Provides means for monitoring applications
- Three metrics scopes:
  - `/metrics/base` - basic metrics, most need to be provided by all vendors, e.g. used heap, thread count, GC stats, number of loaded classes
  - `/metrics/vendor` - vendor-specific metrics;
  - `/metrics/application` - business-logic metrics
- JSON or Prometheus format
- The exposed data can be consumed e.g. by Prometheus

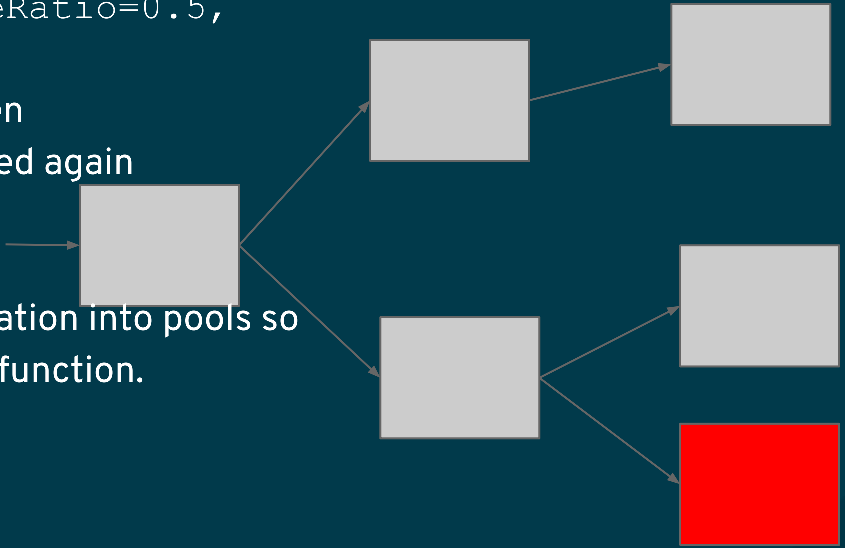
MicroProfile Fault Tolerance

- `@Timeout(500)`
  - Interrupt if possible or throw `@TimeoutException` if not
- `@Retry`
  - maxRetries
  - delay, jitter
  - maxDuration
  - retryOn, abortOn
- `@Fallback`



MicroProfile Fault Tolerance

- `@CircuitBreaker(successThreshold=4, requestVolumeThreshold=4,failureRatio=0.5, delay=500)`
    - Delay - when the circuit gets half open
    - successThreshold - when it gets closed again
    - Open if 50% of last 4 requests failed
- `@Asynchronous`
- `@Bulkhead` isolates elements of an application into pools so that if one fails, the others will continue to function.



MicroProfile Health

- Standardized way of providing status of the application
- Custom health checks can be added
- Can be leveraged by service mesh

Thorntail

- An application generator, pick dependencies that you need and pack with your code into an uber-jar
- Red Hat's Eclipse MicroProfile implementation
  - Currently supports MicroProfile 1.3 via SmallRye
- Yaml configuration, helpers for testing

Demo

THORNTAIL

Don’t miss

Cloud-native development with MicroProfile on Kubernetes tomorrow at 10:45

THORNTAIL

redhat.

Links

Demo: http://bit.ly/microprofile-with-thorntail

http://thorntail.io https://groups.google.com/forum/#!forum/thorntail

#thorntail on FreeNode

http://smallrye.io https://github.com/smallrye/

http://microprofile.io https://groups.google.com/forum/#!forum/microprofile

MicroProfile data sheet http://bit.ly/MP-ebook MicroProfile calendar

@mszynkiewicz