

Quarkus stories: How to extend dev services

Aleš Justin, OpenBlend / Red Hat

Dev service?

- ◉ An external service
 - ◉ e.g. database, Kafka broker, ...
 - ◉ BUT used at development NOT prod
- ◉ Usually Testcontainer + Docker image
 - ◉ Extending GenericContainer
 - ◉ Easy to get a "clean slate"
- ◉ Quarkus already has a bunch of them ...
- ◉ Our dev services: Kafka (yeah, not using Quarkus' 😊), VM, S3, ...

The problem?

- ◉ Problem's background
 - ◉ Quarkus app with many dev services
 - ◉ Adding new as we progress with development
 - ◉ Profiles: "test" vs "dev"
 - ◉ Quarkus JUnit support vs hacking CL usage
- ◉ Services lifecycle
 - ◉ Dev services need to stop LAST
 - ◉ "artificial" CDI dependencies 😊

Enter extensions

- Entry into Quarkus' "internals"
 - Eating your own dog food aka features
- Pre-processed "processors" with BuildItems
 - Ordered BuildItems
 - Produces vs Consumes
 - Multi vs single BuildItem
- Deployment vs Runtime modules
 - Only access to Runtime classes
- Recorders
 - Prepare / "record" bytecode steps
 - Access to config classes
 - Performance gain when metadata is involved
- List of capabilities - conditional items, etc

ConfigSource

- ◉ Use custom ConfigSource to start all found DevServices (aka resources)
 - ◉ Old-school META-INF/services mechanism
 - ◉ First call to "getPropertyNames()" or "getValue(name)"
- ◉ Discovery
 - ◉ Initially
 - ◉ As usual META-INF/services
 - ◉ Duplicate work
 - ◉ Now
 - ◉ RuntimeConfigurationSourceBuildItem

Fix stopping?!

- App stop event doesn't work
 - Random order
- How to get rid of dependencies?
 - Quarkus has 2 lifecycles!
 - Luckily CDI is in the inner :-)
 - `ShutdownContext::addLastShutdownTask`

Some code?