Item 347
git_comments:

1. camel.apache.org/components/latest/others/main.html[Apache Camel Main] to Camel Quarkus. By adding such extension the programmatic configuration of the Log Component done above, can be done through properties:
2. the configuration of the log component is done programmatically using CDI

git_commits:

1. **summary:** Revisit camel-quarkus bootstrap #1128 (docs)
   message: Revisit camel-quarkus bootstrap #1128 (docs) Co-authored-by: Peter Palaga <ppalaga@redhat.com>

github_issues:

1. **title:** Revisit camel-quarkus bootstrap
   body: **Current State** The **camel-quarkus-core** extension has two main responsibilities: 1. set-up the camel context 2. set-up the camel's main and start it Camel's main support is an optional feature enabled by default that in essence: - takes the camel context from 1 - configure the reactive engine - collect routes - collect main listeners - start the main (which binds routes to the camel context and configure it using properties and other auto configuration steps) With the introduction of the lightweight camel context in camel 3.2 and the work that is planned for for next versions the code may becoming more complex to maintain and the number of option may raise making the configuration harder for user. **Proposal** - **camle-quarkus-core** - configure context (reactive engine, registry, etc) - automatic discovery of routes and bind to the camel-context - provide a default "bootstrap item" that start the camel context <br />The main goal of this extension should be to provide the minimum needed to boot a camel based application which runs the routes available form the classpath / cdi container. This extension won't include a way to configure the context using `camel` properties name-space which is moved to the **camle-quarkus-main** extension. User that need specific component/context configuration can use CDI (we probably need to support the concept of customizer with CDI beans). - **camle-quarkus-main** - provides a alternative bootstrap to use camel-main to further configure the camel context configured by `camle-quarkus-core` and start it <br />Among other things, this extension will include the support for the `camel` properties name-space making it more visible because as today, as example, it is not clear what you can do if you disable main through the `quarkus.camel.main` property but you still have the `camel` properties - **camle-quarkus-main-static** (name is just purely indicative) - provides a alternative bootstrap to use the static camel-main to start the camel context <br />The main goal here is to have a clear separation between default and static mode and have some more freedom to experiment. Eventually it could become the default bootstrap in the future but for the time being I think it could be better to have it in a dedicated extension so it is clear that it is for advanced and concious use.
   label: code-design
2. **title:** Revisit camel-quarkus bootstrap
   body: **Current State** The **camel-quarkus-core** extension has two main responsibilities: 1. set-up the camel context 2. set-up the camel's main and start it Camel's main support is an optional feature enabled by default that in essence: - takes the camel context from 1 - configure the reactive engine - collect routes - collect main listeners - start the main (which binds routes to the camel context and configure it using properties and other auto configuration steps) With the introduction of the lightweight camel context in camel 3.2 and the work that is planned for for next versions the code may becoming more complex to maintain and the number of option may raise making the configuration harder for user. **Proposal** - **camle-quarkus-core** - configure context (reactive engine, registry, etc) - automatic discovery of routes and bind to the camel-context - provide a default "bootstrap item" that start the camel context <br />The main goal of this extension should be to provide the minimum needed to boot a camel based application which runs the routes available form the classpath / cdi container. This extension won't include a way to configure the context using `camel` properties name-space which is moved to the **camle-quarkus-main** extension. User that need specific component/context configuration can use CDI (we probably need to support the concept of customizer with CDI beans). - **camle-quarkus-main** - provides a alternative bootstrap to use camel-main to further configure the camel context configured by `camle-quarkus-core` and start it <br />Among other things, this extension will include the support for the `camel` properties name-space making it more visible because as today, as example, it is not clear what

you can do if you disable main through the `quarkus.camel.main` property but you still have the `camel` properties - **camle-quarkus-main-static** (name is just purely indicative) - provides a alternative bootstrap to use the static camel-main to start the camel context <br />The main goal here is to have a clear separation between default and static mode and have some more freedom to experiment. Eventually it could become the default bootstrap in the future but for the time being I think it could be better to have it in a dedicated extension so it is clear that it is for advanced and concious use.

3. **title:** Revisit camel-quarkus bootstrap
   body: **Current State** The **camel-quarkus-core** extension has two main responsibilities: 1. set-up the camel context 2. set-up the camel's main and start it Camel's main support is an optional feature enabled by default that in essence: - takes the camel context from 1 - configure the reactive engine - collect routes - collect main listeners - start the main (which binds routes to the camel context and configure it using properties and other auto configuration steps) With the introduction of the lightweight camel context in camel 3.2 and the work that is planned for for next versions the code may becoming more complex to maintain and the number of option may raise making the configuration harder for user. **Proposal** - **camle-quarkus-core** - configure context (reactive engine, registry, etc) - automatic discovery of routes and bind to the camel-context - provide a default "bootstrap item" that start the camel context <br />The main goal of this extension should be to provide the minimum needed to boot a camel based application which runs the routes available form the classpath / cdi container. This extension won't include a way to configure the context using `camel` properties name-space which is moved to the **camle-quarkus-main** extension. User that need specific component/context configuration can use CDI (we probably need to support the concept of customizer with CDI beans). - **camle-quarkus-main** - provides a alternative bootstrap to use camel-main to further configure the camel context configured by `camle-quarkus-core` and start it <br />Among other things, this extension will include the support for the `camel` properties name-space making it more visible because as today, as example, it is not clear what you can do if you disable main through the `quarkus.camel.main` property but you still have the `camel` properties - **camle-quarkus-main-static** (name is just purely indicative) - provides a alternative bootstrap to use the static camel-main to start the camel context <br />The main goal here is to have a clear separation between default and static mode and have some more freedom to experiment. Eventually it could become the default bootstrap in the future but for the time being I think it could be better to have it in a dedicated extension so it is clear that it is for advanced and concious use.
4. **title:** Revisit camel-quarkus bootstrap
   body: **Current State** The **camel-quarkus-core** extension has two main responsibilities: 1. set-up the camel context 2. set-up the camel's main and start it Camel's main support is an optional feature enabled by default that in essence: - takes the camel context from 1 - configure the reactive engine - collect routes - collect main listeners - start the main (which binds routes to the camel context and configure it using properties and other auto configuration steps) With the introduction of the lightweight camel context in camel 3.2 and the work that is planned for for next versions the code may becoming more complex to maintain and the number of option may raise making the configuration harder for user. **Proposal** - **camle-quarkus-core** - configure context (reactive engine, registry, etc) - automatic discovery of routes and bind to the camel-context - provide a default "bootstrap item" that start the camel context <br />The main goal of this extension should be to provide the minimum needed to boot a camel based application which runs the routes available form the classpath / cdi container. This extension won't include a way to configure the context using `camel` properties name-space which is moved to the **camle-quarkus-main** extension. User that need specific component/context configuration can use CDI (we probably need to support the concept of customizer with CDI beans). - **camle-quarkus-main** - provides a alternative bootstrap to use camel-main to further configure the camel context configured by `camle-quarkus-core` and start it <br />Among other things, this extension will include the support for the `camel` properties name-space making it more visible because as today, as example, it is not clear what you can do if you disable main through the `quarkus.camel.main` property but you still have the `camel` properties - **camle-quarkus-main-static** (name is just purely indicative) - provides a alternative bootstrap to use the static camel-main to start the camel context <br />The main goal here is to have a clear separation between default and static mode and have some more freedom to experiment. Eventually it could become the default bootstrap in the future but for the time being I think it could be better to have it in a dedicated extension so it is clear that it is for advanced and concious use.
5. **title:** Revisit camel-quarkus bootstrap
   body: **Current State** The **camel-quarkus-core** extension has two main responsibilities: 1. set-up the camel context 2. set-up the camel's main and start it Camel's main support is an optional feature enabled by default that in essence: - takes the camel context from 1 - configure the reactive engine - collect routes - collect main listeners - start the main (which binds routes to the camel context and

configure it using properties and other auto configuration steps) With the introduction of the lightweight camel context in camel 3.2 and the work that is planned for for next versions the code may becoming more complex to maintain and the number of option may raise making the configuration harder for user. **Proposal** - **camle-quarkus-core** - configure context (reactive engine, registry, etc) - automatic discovery of routes and bind to the camel-context - provide a default "bootstrap item" that start the camel context <br />The main goal of this extension should be to provide the minimum needed to boot a camel based application which runs the routes available form the classpath / cdi container. This extension won't include a way to configure the context using `camel` properties name-space which is moved to the **camle-quarkus-main** extension. User that need specific component/context configuration can use CDI (we probably need to support the concept of customizer with CDI beans). - **camle-quarkus-main** - provides a alternative bootstrap to use camel-main to further configure the camel context configured by `camle-quarkus-core` and start it <br />Among other things, this extension will include the support for the `camel` properties name-space making it more visible because as today, as example, it is not clear what you can do if you disable main through the `quarkus.camel.main` property but you still have the `camel` properties - **camle-quarkus-main-static** (name is just purely indicative) - provides a alternative bootstrap to use the static camel-main to start the camel context <br />The main goal here is to have a clear separation between default and static mode and have some more freedom to experiment. Eventually it could become the default bootstrap in the future but for the time being I think it could be better to have it in a dedicated extension so it is clear that it is for advanced and concious use.

6. **title:** Revisit camel-quarkus bootstrap
   body: **Current State** The **camel-quarkus-core** extension has two main responsibilities: 1. set-up the camel context 2. set-up the camel's main and start it Camel's main support is an optional feature enabled by default that in essence: - takes the camel context from 1 - configure the reactive engine - collect routes - collect main listeners - start the main (which binds routes to the camel context and configure it using properties and other auto configuration steps) With the introduction of the lightweight camel context in camel 3.2 and the work that is planned for for next versions the code may becoming more complex to maintain and the number of option may raise making the configuration harder for user. **Proposal** - **camle-quarkus-core** - configure context (reactive engine, registry, etc) - automatic discovery of routes and bind to the camel-context - provide a default "bootstrap item" that start the camel context <br />The main goal of this extension should be to provide the minimum needed to boot a camel based application which runs the routes available form the classpath / cdi container. This extension won't include a way to configure the context using `camel` properties name-space which is moved to the **camle-quarkus-main** extension. User that need specific component/context configuration can use CDI (we probably need to support the concept of customizer with CDI beans). - **camle-quarkus-main** - provides a alternative bootstrap to use camel-main to further configure the camel context configured by `camle-quarkus-core` and start it <br />Among other things, this extension will include the support for the `camel` properties name-space making it more visible because as today, as example, it is not clear what you can do if you disable main through the `quarkus.camel.main` property but you still have the `camel` properties - **camle-quarkus-main-static** (name is just purely indicative) - provides a alternative bootstrap to use the static camel-main to start the camel context <br />The main goal here is to have a clear separation between default and static mode and have some more freedom to experiment. Eventually it could become the default bootstrap in the future but for the time being I think it could be better to have it in a dedicated extension so it is clear that it is for advanced and concious use.
7. **title:** Revisit camel-quarkus bootstrap
   body: **Current State** The **camel-quarkus-core** extension has two main responsibilities: 1. set-up the camel context 2. set-up the camel's main and start it Camel's main support is an optional feature enabled by default that in essence: - takes the camel context from 1 - configure the reactive engine - collect routes - collect main listeners - start the main (which binds routes to the camel context and configure it using properties and other auto configuration steps) With the introduction of the lightweight camel context in camel 3.2 and the work that is planned for for next versions the code may becoming more complex to maintain and the number of option may raise making the configuration harder for user. **Proposal** - **camle-quarkus-core** - configure context (reactive engine, registry, etc) - automatic discovery of routes and bind to the camel-context - provide a default "bootstrap item" that start the camel context <br />The main goal of this extension should be to provide the minimum needed to boot a camel based application which runs the routes available form the classpath / cdi container. This extension won't include a way to configure the context using `camel` properties name-space which is moved to the **camle-quarkus-main** extension. User that need specific component/context configuration can use CDI (we probably need to support the concept of customizer with CDI beans). - **camle-quarkus-main** - provides a alternative bootstrap to use camel-main to further configure the camel context configured by

`camle-quarkus-core` and start it <br />Among other things, this extension will include the support for the `camel` properties name-space making it more visible because as today, as example, it is not clear what you can do if you disable main through the `quarkus.camel.main` property but you still have the `camel` properties - **camle-quarkus-main-static** (name is just purely indicative) - provides a alternative bootstrap to use the static camel-main to start the camel context <br />The main goal here is to have a clear separation between default and static mode and have some more freedom to experiment. Eventually it could become the default bootstrap in the future but for the time being I think it could be better to have it in a dedicated extension so it is clear that it is for advanced and concious use.

8. **title:** Revisit camel-quarkus bootstrap
   body: **Current State** The **camel-quarkus-core** extension has two main responsibilities: 1. set-up the camel context 2. set-up the camel's main and start it Camel's main support is an optional feature enabled by default that in essence: - takes the camel context from 1 - configure the reactive engine - collect routes - collect main listeners - start the main (which binds routes to the camel context and configure it using properties and other auto configuration steps) With the introduction of the lightweight camel context in camel 3.2 and the work that is planned for for next versions the code may becoming more complex to maintain and the number of option may raise making the configuration harder for user. **Proposal** - **camle-quarkus-core** - configure context (reactive engine, registry, etc) - automatic discovery of routes and bind to the camel-context - provide a default "bootstrap item" that start the camel context <br />The main goal of this extension should be to provide the minimum needed to boot a camel based application which runs the routes available form the classpath / cdi container. This extension won't include a way to configure the context using `camel` properties name-space which is moved to the **camle-quarkus-main** extension. User that need specific component/context configuration can use CDI (we probably need to support the concept of customizer with CDI beans). - **camle-quarkus-main** - provides a alternative bootstrap to use camel-main to further configure the camel context configured by `camle-quarkus-core` and start it <br />Among other things, this extension will include the support for the `camel` properties name-space making it more visible because as today, as example, it is not clear what you can do if you disable main through the `quarkus.camel.main` property but you still have the `camel` properties - **camle-quarkus-main-static** (name is just purely indicative) - provides a alternative bootstrap to use the static camel-main to start the camel context <br />The main goal here is to have a clear separation between default and static mode and have some more freedom to experiment. Eventually it could become the default bootstrap in the future but for the time being I think it could be better to have it in a dedicated extension so it is clear that it is for advanced and concious use.
9. **title:** Revisit camel-quarkus bootstrap
   body: **Current State** The **camel-quarkus-core** extension has two main responsibilities: 1. set-up the camel context 2. set-up the camel's main and start it Camel's main support is an optional feature enabled by default that in essence: - takes the camel context from 1 - configure the reactive engine - collect routes - collect main listeners - start the main (which binds routes to the camel context and configure it using properties and other auto configuration steps) With the introduction of the lightweight camel context in camel 3.2 and the work that is planned for for next versions the code may becoming more complex to maintain and the number of option may raise making the configuration harder for user. **Proposal** - **camle-quarkus-core** - configure context (reactive engine, registry, etc) - automatic discovery of routes and bind to the camel-context - provide a default "bootstrap item" that start the camel context <br />The main goal of this extension should be to provide the minimum needed to boot a camel based application which runs the routes available form the classpath / cdi container. This extension won't include a way to configure the context using `camel` properties name-space which is moved to the **camle-quarkus-main** extension. User that need specific component/context configuration can use CDI (we probably need to support the concept of customizer with CDI beans). - **camle-quarkus-main** - provides a alternative bootstrap to use camel-main to further configure the camel context configured by `camle-quarkus-core` and start it <br />Among other things, this extension will include the support for the `camel` properties name-space making it more visible because as today, as example, it is not clear what you can do if you disable main through the `quarkus.camel.main` property but you still have the `camel` properties - **camle-quarkus-main-static** (name is just purely indicative) - provides a alternative bootstrap to use the static camel-main to start the camel context <br />The main goal here is to have a clear separation between default and static mode and have some more freedom to experiment. Eventually it could become the default bootstrap in the future but for the time being I think it could be better to have it in a dedicated extension so it is clear that it is for advanced and concious use.
   label: code-design
10. **title:** Revisit camel-quarkus bootstrap

body: **Current State** The **camel-quarkus-core** extension has two main responsibilities: 1. set-up the camel context 2. set-up the camel's main and start it Camel's main support is an optional feature enabled by default that in essence: - takes the camel context from 1 - configure the reactive engine - collect routes - collect main listeners - start the main (which binds routes to the camel context and configure it using properties and other auto configuration steps) With the introduction of the lightweight camel context in camel 3.2 and the work that is planned for for next versions the code may becoming more complex to maintain and the number of option may raise making the configuration harder for user. **Proposal** - **camle-quarkus-core** - configure context (reactive engine, registry, etc) - automatic discovery of routes and bind to the camel-context - provide a default "bootstrap item" that start the camel context <br />The main goal of this extension should be to provide the minimum needed to boot a camel based application which runs the routes available form the classpath / cdi container. This extension won't include a way to configure the context using `camel` properties name-space which is moved to the **camle-quarkus-main** extension. User that need specific component/context configuration can use CDI (we probably need to support the concept of customizer with CDI beans). - **camle-quarkus-main** - provides a alternative bootstrap to use camel-main to further configure the camel context configured by `camle-quarkus-core` and start it <br />Among other things, this extension will include the support for the `camel` properties name-space making it more visible because as today, as example, it is not clear what you can do if you disable main through the `quarkus.camel.main` property but you still have the `camel` properties - **camle-quarkus-main-static** (name is just purely indicative) - provides a alternative bootstrap to use the static camel-main to start the camel context <br />The main goal here is to have a clear separation between default and static mode and have some more freedom to experiment. Eventually it could become the default bootstrap in the future but for the time being I think it could be better to have it in a dedicated extension so it is clear that it is for advanced and concious use.
label: code-design

11. **title:** Revisit camel-quarkus bootstrap
    body: **Current State** The **camel-quarkus-core** extension has two main responsibilities: 1. set-up the camel context 2. set-up the camel's main and start it Camel's main support is an optional feature enabled by default that in essence: - takes the camel context from 1 - configure the reactive engine - collect routes - collect main listeners - start the main (which binds routes to the camel context and configure it using properties and other auto configuration steps) With the introduction of the lightweight camel context in camel 3.2 and the work that is planned for for next versions the code may becoming more complex to maintain and the number of option may raise making the configuration harder for user. **Proposal** - **camle-quarkus-core** - configure context (reactive engine, registry, etc) - automatic discovery of routes and bind to the camel-context - provide a default "bootstrap item" that start the camel context <br />The main goal of this extension should be to provide the minimum needed to boot a camel based application which runs the routes available form the classpath / cdi container. This extension won't include a way to configure the context using `camel` properties name-space which is moved to the **camle-quarkus-main** extension. User that need specific component/context configuration can use CDI (we probably need to support the concept of customizer with CDI beans). - **camle-quarkus-main** - provides a alternative bootstrap to use camel-main to further configure the camel context configured by `camle-quarkus-core` and start it <br />Among other things, this extension will include the support for the `camel` properties name-space making it more visible because as today, as example, it is not clear what you can do if you disable main through the `quarkus.camel.main` property but you still have the `camel` properties - **camle-quarkus-main-static** (name is just purely indicative) - provides a alternative bootstrap to use the static camel-main to start the camel context <br />The main goal here is to have a clear separation between default and static mode and have some more freedom to experiment. Eventually it could become the default bootstrap in the future but for the time being I think it could be better to have it in a dedicated extension so it is clear that it is for advanced and concious use.
12. **title:** Revisit camel-quarkus bootstrap
    body: **Current State** The **camel-quarkus-core** extension has two main responsibilities: 1. set-up the camel context 2. set-up the camel's main and start it Camel's main support is an optional feature enabled by default that in essence: - takes the camel context from 1 - configure the reactive engine - collect routes - collect main listeners - start the main (which binds routes to the camel context and configure it using properties and other auto configuration steps) With the introduction of the lightweight camel context in camel 3.2 and the work that is planned for for next versions the code may becoming more complex to maintain and the number of option may raise making the configuration harder for user. **Proposal** - **camle-quarkus-core** - configure context (reactive engine, registry, etc) - automatic discovery of routes and bind to the camel-context - provide a default "bootstrap item" that start the camel context <br />The main goal of this extension should be to provide the minimum needed to boot a camel

based application which runs the routes available form the classpath / cdi container. This extension won't include a way to configure the context using `camel` properties name-space which is moved to the **camle-quarkus-main** extension. User that need specific component/context configuration can use CDI (we probably need to support the concept of customizer with CDI beans). - **camle-quarkus-main** - provides a alternative bootstrap to use camel-main to further configure the camel context configured by `camle-quarkus-core` and start it <br />Among other things, this extension will include the support for the `camel` properties name-space making it more visible because as today, as example, it is not clear what you can do if you disable main through the `quarkus.camel.main` property but you still have the `camel` properties - **camle-quarkus-main-static** (name is just purely indicative) - provides a alternative bootstrap to use the static camel-main to start the camel context <br />The main goal here is to have a clear separation between default and static mode and have some more freedom to experiment. Eventually it could become the default bootstrap in the future but for the time being I think it could be better to have it in a dedicated extension so it is clear that it is for advanced and concious use.

13. **title:** Revisit camel-quarkus bootstrap
    body: **Current State** The **camel-quarkus-core** extension has two main responsibilities: 1. set-up the camel context 2. set-up the camel's main and start it Camel's main support is an optional feature enabled by default that in essence: - takes the camel context from 1 - configure the reactive engine - collect routes - collect main listeners - start the main (which binds routes to the camel context and configure it using properties and other auto configuration steps) With the introduction of the lightweight camel context in camel 3.2 and the work that is planned for for next versions the code may becoming more complex to maintain and the number of option may raise making the configuration harder for user. **Proposal** - **camle-quarkus-core** - configure context (reactive engine, registry, etc) - automatic discovery of routes and bind to the camel-context - provide a default "bootstrap item" that start the camel context <br />The main goal of this extension should be to provide the minimum needed to boot a camel based application which runs the routes available form the classpath / cdi container. This extension won't include a way to configure the context using `camel` properties name-space which is moved to the **camle-quarkus-main** extension. User that need specific component/context configuration can use CDI (we probably need to support the concept of customizer with CDI beans). - **camle-quarkus-main** - provides a alternative bootstrap to use camel-main to further configure the camel context configured by `camle-quarkus-core` and start it <br />Among other things, this extension will include the support for the `camel` properties name-space making it more visible because as today, as example, it is not clear what you can do if you disable main through the `quarkus.camel.main` property but you still have the `camel` properties - **camle-quarkus-main-static** (name is just purely indicative) - provides a alternative bootstrap to use the static camel-main to start the camel context <br />The main goal here is to have a clear separation between default and static mode and have some more freedom to experiment. Eventually it could become the default bootstrap in the future but for the time being I think it could be better to have it in a dedicated extension so it is clear that it is for advanced and concious use.
14. **title:** Revisit camel-quarkus bootstrap
    body: **Current State** The **camel-quarkus-core** extension has two main responsibilities: 1. set-up the camel context 2. set-up the camel's main and start it Camel's main support is an optional feature enabled by default that in essence: - takes the camel context from 1 - configure the reactive engine - collect routes - collect main listeners - start the main (which binds routes to the camel context and configure it using properties and other auto configuration steps) With the introduction of the lightweight camel context in camel 3.2 and the work that is planned for for next versions the code may becoming more complex to maintain and the number of option may raise making the configuration harder for user. **Proposal** - **camle-quarkus-core** - configure context (reactive engine, registry, etc) - automatic discovery of routes and bind to the camel-context - provide a default "bootstrap item" that start the camel context <br />The main goal of this extension should be to provide the minimum needed to boot a camel based application which runs the routes available form the classpath / cdi container. This extension won't include a way to configure the context using `camel` properties name-space which is moved to the **camle-quarkus-main** extension. User that need specific component/context configuration can use CDI (we probably need to support the concept of customizer with CDI beans). - **camle-quarkus-main** - provides a alternative bootstrap to use camel-main to further configure the camel context configured by `camle-quarkus-core` and start it <br />Among other things, this extension will include the support for the `camel` properties name-space making it more visible because as today, as example, it is not clear what you can do if you disable main through the `quarkus.camel.main` property but you still have the `camel` properties - **camle-quarkus-main-static** (name is just purely indicative) - provides a alternative bootstrap to use the static camel-main to start the camel context <br />The main goal here is to have a clear separation between default and static mode and have some more freedom to experiment. Eventually

it could become the default bootstrap in the future but for the time being I think it could be better to have it in a dedicated extension so it is clear that it is for advanced and concious use.

15. **title:** Revisit camel-quarkus bootstrap
    body: **Current State** The **camel-quarkus-core** extension has two main responsibilities: 1. set-up the camel context 2. set-up the camel's main and start it Camel's main support is an optional feature enabled by default that in essence: - takes the camel context from 1 - configure the reactive engine - collect routes - collect main listeners - start the main (which binds routes to the camel context and configure it using properties and other auto configuration steps) With the introduction of the lightweight camel context in camel 3.2 and the work that is planned for for next versions the code may becoming more complex to maintain and the number of option may raise making the configuration harder for user. **Proposal** - **camle-quarkus-core** - configure context (reactive engine, registry, etc) - automatic discovery of routes and bind to the camel-context - provide a default "bootstrap item" that start the camel context <br />The main goal of this extension should be to provide the minimum needed to boot a camel based application which runs the routes available form the classpath / cdi container. This extension won't include a way to configure the context using `camel` properties name-space which is moved to the **camle-quarkus-main** extension. User that need specific component/context configuration can use CDI (we probably need to support the concept of customizer with CDI beans). - **camle-quarkus-main** - provides a alternative bootstrap to use camel-main to further configure the camel context configured by `camle-quarkus-core` and start it <br />Among other things, this extension will include the support for the `camel` properties name-space making it more visible because as today, as example, it is not clear what you can do if you disable main through the `quarkus.camel.main` property but you still have the `camel` properties - **camle-quarkus-main-static** (name is just purely indicative) - provides a alternative bootstrap to use the static camel-main to start the camel context <br />The main goal here is to have a clear separation between default and static mode and have some more freedom to experiment. Eventually it could become the default bootstrap in the future but for the time being I think it could be better to have it in a dedicated extension so it is clear that it is for advanced and concious use.
16. **title:** Revisit camel-quarkus bootstrap
    body: **Current State** The **camel-quarkus-core** extension has two main responsibilities: 1. set-up the camel context 2. set-up the camel's main and start it Camel's main support is an optional feature enabled by default that in essence: - takes the camel context from 1 - configure the reactive engine - collect routes - collect main listeners - start the main (which binds routes to the camel context and configure it using properties and other auto configuration steps) With the introduction of the lightweight camel context in camel 3.2 and the work that is planned for for next versions the code may becoming more complex to maintain and the number of option may raise making the configuration harder for user. **Proposal** - **camle-quarkus-core** - configure context (reactive engine, registry, etc) - automatic discovery of routes and bind to the camel-context - provide a default "bootstrap item" that start the camel context <br />The main goal of this extension should be to provide the minimum needed to boot a camel based application which runs the routes available form the classpath / cdi container. This extension won't include a way to configure the context using `camel` properties name-space which is moved to the **camle-quarkus-main** extension. User that need specific component/context configuration can use CDI (we probably need to support the concept of customizer with CDI beans). - **camle-quarkus-main** - provides a alternative bootstrap to use camel-main to further configure the camel context configured by `camle-quarkus-core` and start it <br />Among other things, this extension will include the support for the `camel` properties name-space making it more visible because as today, as example, it is not clear what you can do if you disable main through the `quarkus.camel.main` property but you still have the `camel` properties - **camle-quarkus-main-static** (name is just purely indicative) - provides a alternative bootstrap to use the static camel-main to start the camel context <br />The main goal here is to have a clear separation between default and static mode and have some more freedom to experiment. Eventually it could become the default bootstrap in the future but for the time being I think it could be better to have it in a dedicated extension so it is clear that it is for advanced and concious use.
17. **title:** Revisit camel-quarkus bootstrap
    body: **Current State** The **camel-quarkus-core** extension has two main responsibilities: 1. set-up the camel context 2. set-up the camel's main and start it Camel's main support is an optional feature enabled by default that in essence: - takes the camel context from 1 - configure the reactive engine - collect routes - collect main listeners - start the main (which binds routes to the camel context and configure it using properties and other auto configuration steps) With the introduction of the lightweight camel context in camel 3.2 and the work that is planned for for next versions the code may becoming more complex to maintain and the number of option may raise making the configuration harder for user. **Proposal** - **camle-quarkus-core** - configure context (reactive engine, registry, etc) - automatic

discovery of routes and bind to the camel-context - provide a default "bootstrap item" that start the camel context <br />The main goal of this extension should be to provide the minimum needed to boot a camel based application which runs the routes available form the classpath / cdi container. This extension won't include a way to configure the context using `camel` properties name-space which is moved to the **camle-quarkus-main** extension. User that need specific component/context configuration can use CDI (we probably need to support the concept of customizer with CDI beans). - **camle-quarkus-main** - provides a alternative bootstrap to use camel-main to further configure the camel context configured by `camle-quarkus-core` and start it <br />Among other things, this extension will include the support for the `camel` properties name-space making it more visible because as today, as example, it is not clear what you can do if you disable main through the `quarkus.camel.main` property but you still have the `camel` properties - **camle-quarkus-main-static** (name is just purely indicative) - provides a alternative bootstrap to use the static camel-main to start the camel context <br />The main goal here is to have a clear separation between default and static mode and have some more freedom to experiment. Eventually it could become the default bootstrap in the future but for the time being I think it could be better to have it in a dedicated extension so it is clear that it is for advanced and concious use.

github_issues_comments:

1. **body:** <details> <summary>An older comment by me that got obsolete by Luca's update https://github.com/apache/camel-quarkus/issues/1128#issuecomment-618407713. Click to expand</summary> > * embedded (core) > > * routes are added to the camel context directly ... by the user > * context need to be programmatically configured ... by the user > * main > * routes and configuration classes are managed by camel-main > * support for command line > * support for configuration through properties > * static (aka lightweight) > * routes are added to the camel context directly By whom? > * some limitation may happen > * context need to be programmatically configured ... by the user? Which aspects of the context for example? > We can then provide main and static behaviour through extensions so depending of the end user need we may achieve different optimization or behaviours without cluttering the code too much. My standpoint is the same as in the past when we were introducing the main: If 80+% of Camel Quarkus users will be supposed to add dependency X, then the dependency should be present by default and the users do not need to know about the fine tuning unless they know what they do. Requiring a decision from the end users will cause unnecessary cognitive load and confusion. Having said that I vote for configuring core vs. main via application.properties rather than via dependencies. My very naive initial attitude to main vs. static is that we should have only one of them, probably only static. "some limitation may happen" could be presented as limitations of Camel Quarkus rather than a limitation of the single "static" mode. If ppl. need more flexibility they should be advised to choose another Camel platform. </details>
   label: code-design
2. I think you get this wrong as my proposal is to revisit the boostrap in general so if you don't have main as extension, then camel will still start the camel context and will still auto discovery routes as it does today, what it will lacks is some bits that are supported by camel main which you cna leverage by adding the main extension. static initit cane become the default only if it does not break other runtimes i.e. camel-k so I'd say the default should be what would for for the majority and you should opt for the static and more optimized options if you are conscious about what you are doing.
3. > I think you get this wrong as my proposal Could you please edit your proposal and clearly split * Current state * Problems with the current state * Proposed state * Pros/cons of the proposed state
4. @ppalaga updated the issue
5. > provide a default "bootstrap item" that start the camel context So this is the same as the CamelMain.start() currently ? > User that need specific component/context configuration can use CDI (we probably need to support the concept of customizer with CDI beans). How can we support this customizer with CDI beans ? provide the BeanConfigurationBuildItem in the user's application code ? I just introduce the ```quarkus.camel.bootstrap.enabled``` option and the user can disable the bootstrap in the extension and will do the configuration and boot the camel context by themselves. Do you think it make sense ? or we have to provide the mandatory default bootstrap ?
6. > > provide a default "bootstrap item" that start the camel context > > So this is the same as the CamelMain.start() currently ? Except it does not use camel-main > > > User that need specific component/context configuration can use CDI (we probably need to support the concept of customizer with CDI beans). > mh, probably we don't need it as we fire CDI events so one can listen and react to them. > How can we support this customizer with CDI beans ? provide the BeanConfigurationBuildItem in the user's application code ? > > I just introduce the `quarkus.camel.bootstrap.enabled` option and the

user can disable the bootstrap in the extension and will do the configuration and boot the camel context by themselves. Do you think it make sense ? or we have to provide the mandatory default bootstrap ? The default bootstrapper should be marked as optional so developers can provide an alternative one if they want a custom boostrap logic

7. > `mh, probably we don't need it as we fire CDI events so one can listen and react to them. This looks like to add a similar CdiEventNotifier in camel context during the building time ?
8. yeah, something like that but we need to think a little about it
9. **body:** Thanks for the update, @lburgazzoli , it makes much more sense to me now. Could you please confirm/deny/explain the following? 1. According to your proposal, our non-core extensions will depend on `camel-quarkus-core` (rather than `camel-quarkus-main`), right? 2. Our existing itests should in theory either (a) work with the proposed changes as long as they do not use `camel.*` properties or (b) work with the proposed changes if the `camel-quarkus-main` dependency is added. 3. Which of the three options ("core only", "main", "main-static") is the one that will meet the needs of 80% of users? 4. Can we make core-deployment smart enough to decide whether main or main-static is needed and add one of those as necessary?
   label: code-design
10. **body:** > 1. According to your proposal, our non-core extensions will depend on `camel-quarkus-core` (rather than `camel-quarkus-main`), right? correct > 2. Our existing itests should in theory either (a) work with the proposed changes as long as they do not use `camel.*` properties or (b) work with the proposed changes if the `camel-quarkus-main` dependency is added. correct > 3. Which of the three options ("core only", "main", "main-static") is the one that will meet the needs of 80% of users? this is very hard to answer as there's many way to use camel, using main to create a standalone camel application is certainly one option but you can also use some little camel from [smallrye-reactive-messaging](https://smallrye.io/smallrye-reactive-messaging/smallrye-reactive-messaging/2/camel/camel.html) or [mutiny](https://issues.apache.org/jira/browse/CAMEL-14381) or when you need some small integration capabilities in your application. I think that nowadays it is likely common to require some integration capabilities so I believe having a core that helps setting up things but give you freedom to i.e. provide your own main is the most sensible choice we can make. > 4. Can we make core-deployment smart enough to decide whether main or main-static is needed and add one of those as necessary? I think there have been some discussion on quarkus side to have a way to add additional artefacts depending on some conditions but I'd rather start with something simple and less magic and see how it goes.
    label: code-design
11. @ppalaga are the info here good enough to start working on the implementation ?
12. One more question: `main-static` will have no support for the `camel.*` properties?
13. Additional note, there is an [issue](https://issues.apache.org/jira/browse/CAMEL-14600) about move component auto configuration as part of standard component life-cycle which would make revisiting the bootstrap process even more appealing as configuring components/languages/etc with be available out of the box with less overhead
14. @lburgazzoli is there any update on this issue ?
15. I'm workign on a big refactor :)

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

jira_issues_comments: