



Graceful shutdown w Springu czyli prawidłowe zamykanie aplikacji

Jarosław Czekalski - Dev Engineer
ING Bank Śląski S.A.

Maj 2022





Jarosław Czekalski ING Bank Śląski S.A.

Programista Java, niegdyś hobbystycznie, obecnie profesjonalnie. Entuzjasta open source. Odpoczynku od komputera szuka w wycieczkach górskich (najchętniej z gitarą), grze w brydża, w szachy.



Agenda spotkania

- Java a shutdown
- Standard Spring Boot
- Jak wpiąć się do Springa
- Interesujące przypadki: Hazelcast, TaskExecutor
- Otoczka, o której trzeba pamiętać
- Sesja Q&A

Java a shutdown



Java a shutdown

- `java.lang.Runtime.addShutdownHook`



Java a shutdown

Sposoby zainicjowania zamknięcia aplikacji

- Przesłać do niej sygnał SIGTERM (tylko Linux)
- Wywołać *System.exit(n)* z wnętrza aplikacji, na przykład na skutek wywołania endpointu
- Zakończyć usługę Windows (wrapper powinien to wspierać)
- Zakończyć aplikację z poziomu środowiska programistycznego (np. IntelliJ)
- Zatrzymać prawidłowo skonfigurowany kontener dockerowy
- Ctrl-C w terminalu



Beauty and the Geek Poland



Źródło: pudelek.pl, dostęp 20.05.2022

Standard Spring Boot



Standard Spring Boot

- Domyślne zachowanie
- Standardowe zachowanie web serwera
- Standardowe zamykanie aplikacji Spring Boot



Domyślnie w Tomcacie

`executorTerminationTimeoutMillis`

- The time that the private internal executor will wait for request processing threads to terminate before continuing with the process of stopping the connector. If not set, the default is 5000 (5 seconds).

<https://tomcat.apache.org/tomcat-9.0-doc/config/http.html>



Standard Spring Boot

Definiujemy property:

`server.shutdown=graceful`

`spring.lifecycle.timeout-per-shutdown-phase=35s`

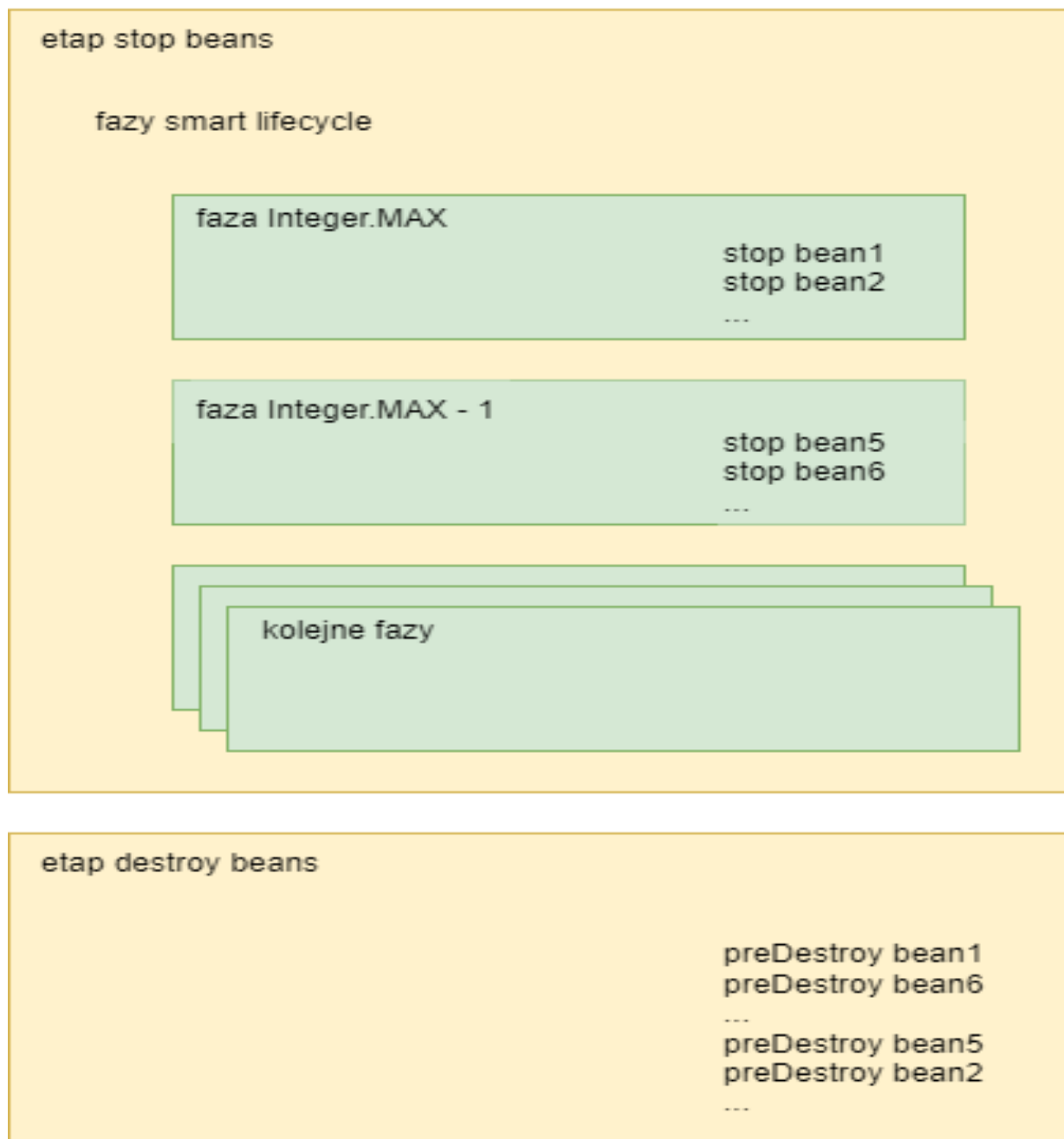


Góra lodowa

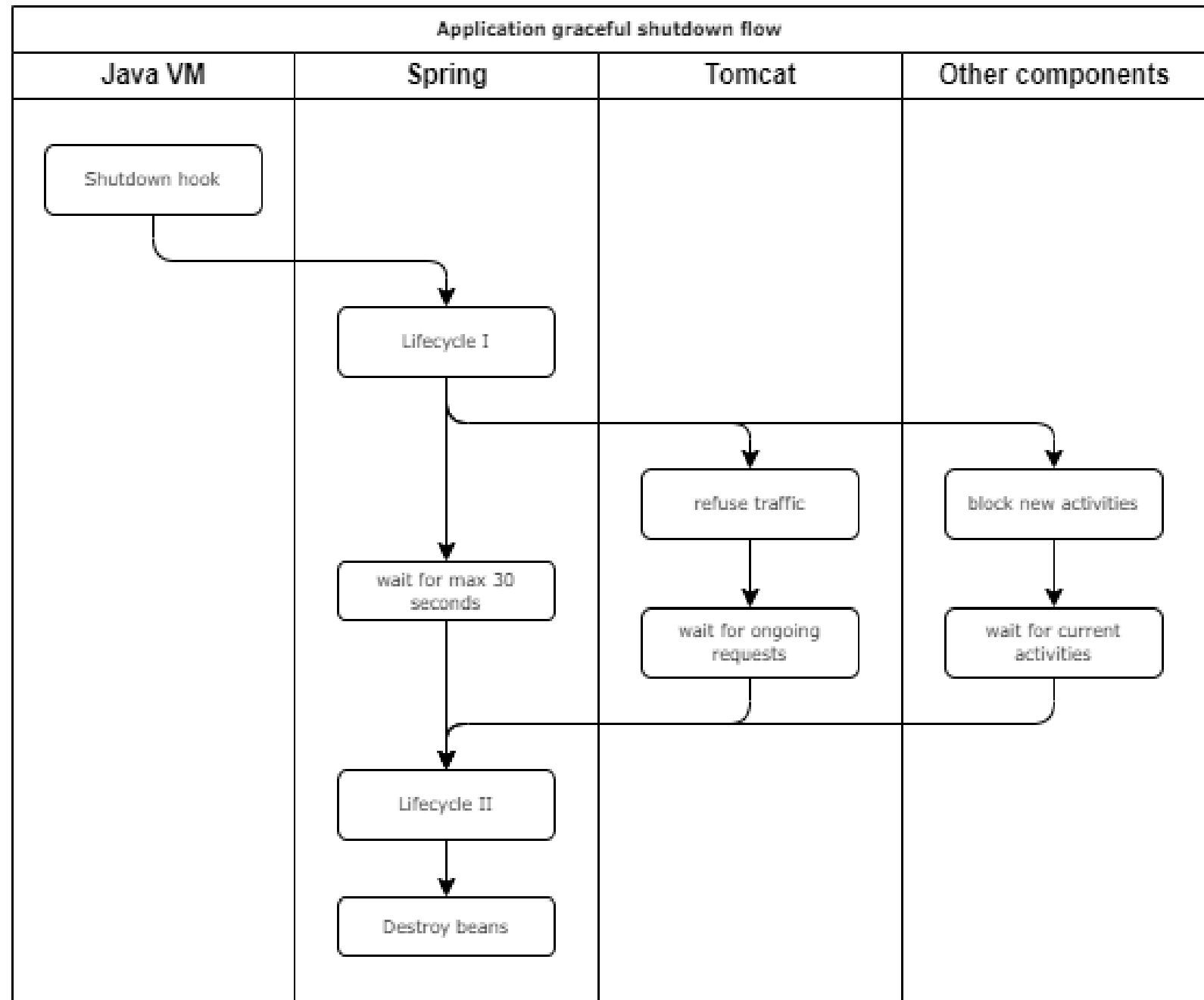


Źródło: wiadomości.dziennik.pl. dostęp 20.05.2022

Standard Spring Boot



Standard Spring Boot



Źródło: materiały wewnętrzne autora

Jak wpiąć się do Springa



Jak wpiąć się do Springa

- Implementując SmartLifecycle
- Napisać helpera



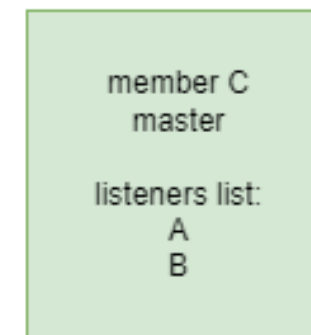
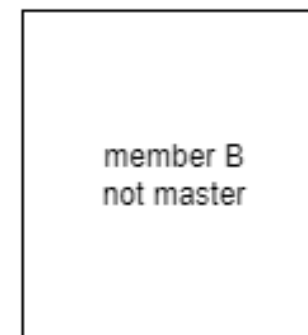
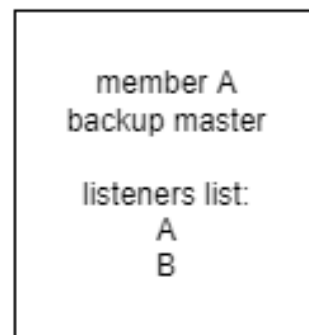
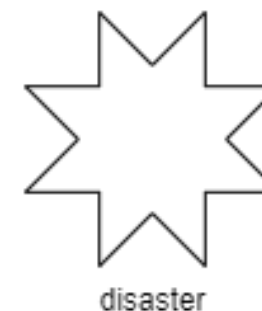
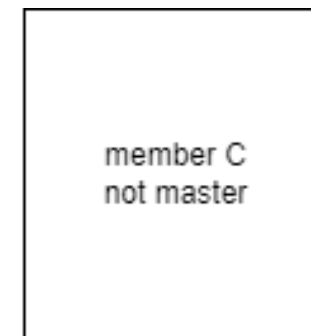
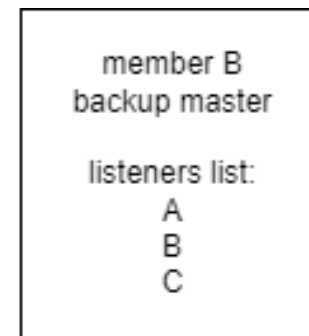
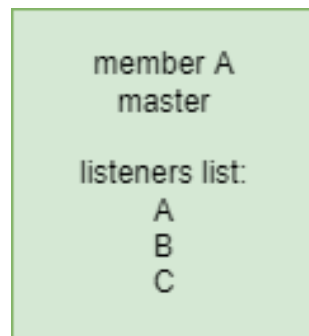


Interesujące przypadki: Hazelcast, TaskExecutor



Zagadka Hazelcasta

- [hazelcast/issues/20054](https://github.com/hazelcast/hazelcast/issues/20054)



Poczekajmy na coś

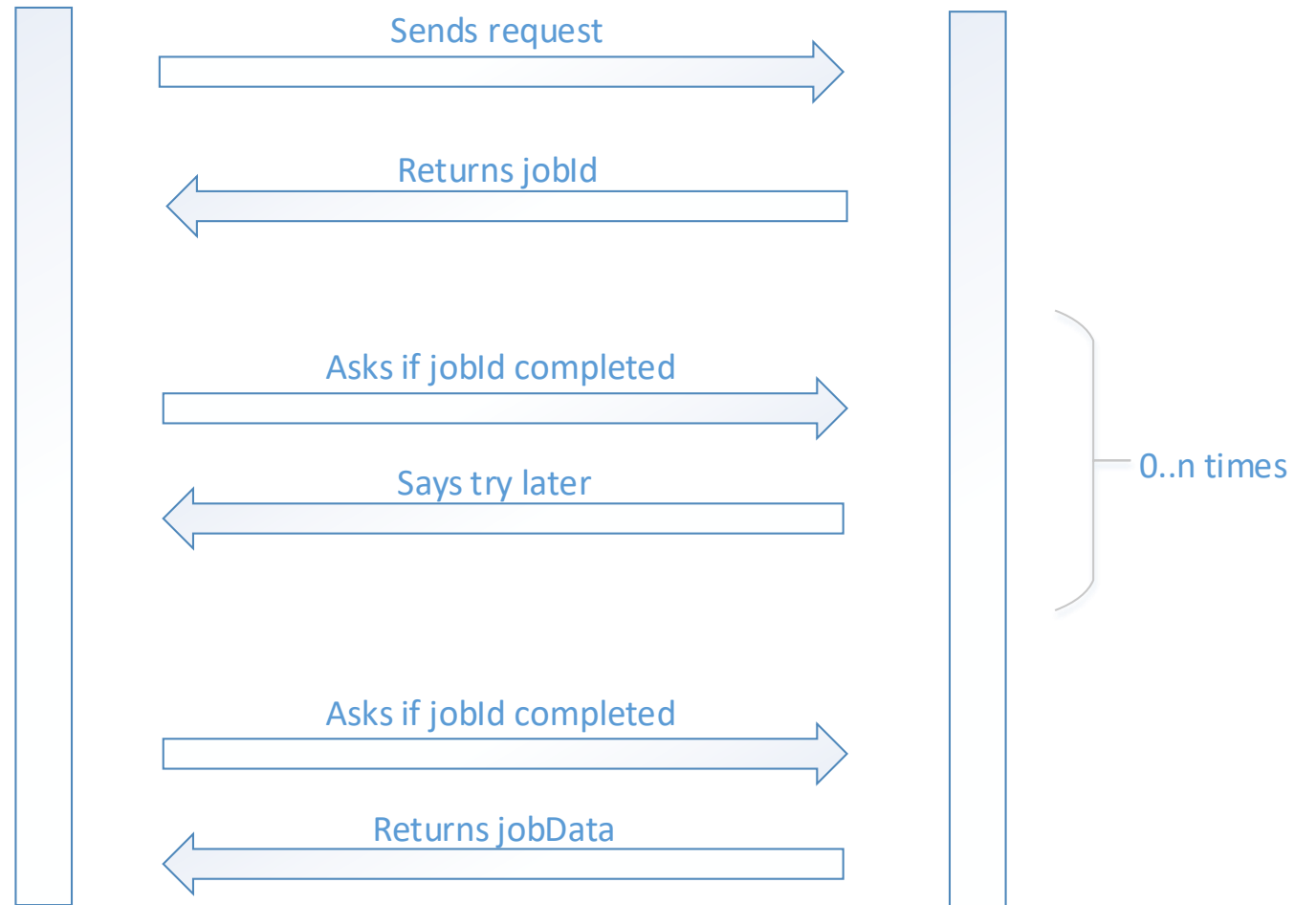
- Mamy jakiś indywidualny proces
- Nie chcemy wyłączać aplikacji, jeżeli proces jest aktywny
- Przykład bardziej półkowny: TaskExecutor



Api asynchroniczne



Server



Źródło: materiały wewnętrzne autora

Otoczka, o której trzeba pamiętać



Otoczka

- Windows wrapper
- Docker



Dockerfile

```
ENTRYPOINT ["/home/app/entrypoint.sh"]
```

```
#!/bin/bash
```

```
java -jar application.jar
```

```
#!/bin/bash
```

```
exec java -jar application.jar
```



Podsumowanie



Podsumowanie

- Inwentaryzacja hooków
- Aktywacja springa
- Indywidualne implementacje
- Spięcie ze środowiskiem uruchomieniowym



Dziękuję za uwagę!

Czas na pytania



