

# **Spring 6 and Beyond**

**Что нового в Spring Framework 6?**

Олег Докука, VMware

Алексей Несторов, BeZeroCarbon

## Disclaimer

This presentation may contain product features or functionality that are currently under development.

This overview of new technology represents no commitment from VMware to deliver these features in any generally available product.

Features are subject to change, and must not be included in contracts, purchase orders, or sales agreements of any kind.

Technical feasibility and market demand will affect final delivery.

Pricing and packaging for any new features/functionality/technology discussed or presented, have not been determined.

The information in this presentation is for informational purposes only and may not be incorporated into any contract. There is no commitment or obligation to deliver any items presented herein.

# The next big thing

Основа для экосистемы Spring на следующие N лет

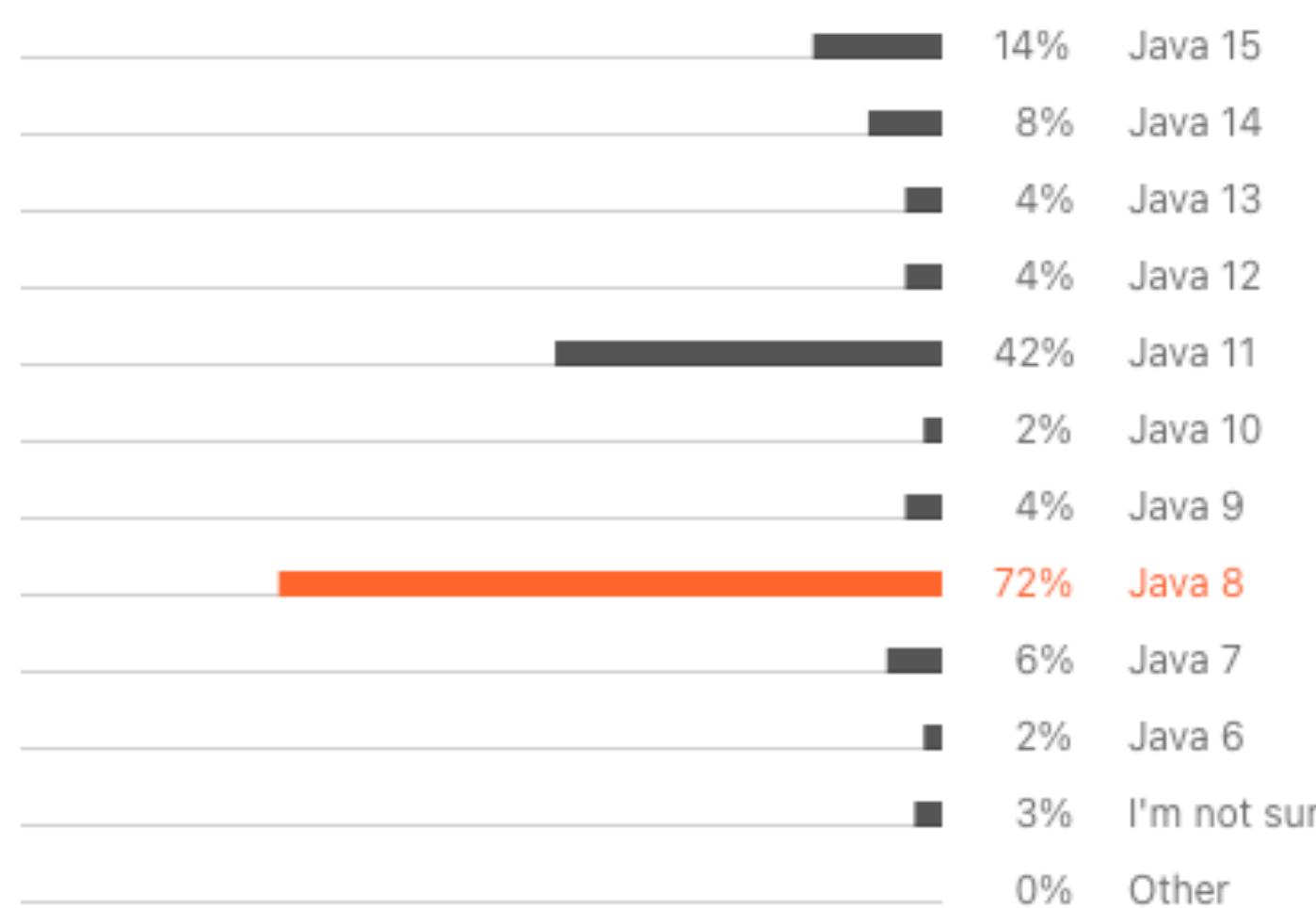
- Spring 6 немного нетипичный релиз, много серьезных изменений
- Новый baseline - JDK 17 😱
- Есть breaking changes
- Возможно, поменяется и release cycle

# Spring Core

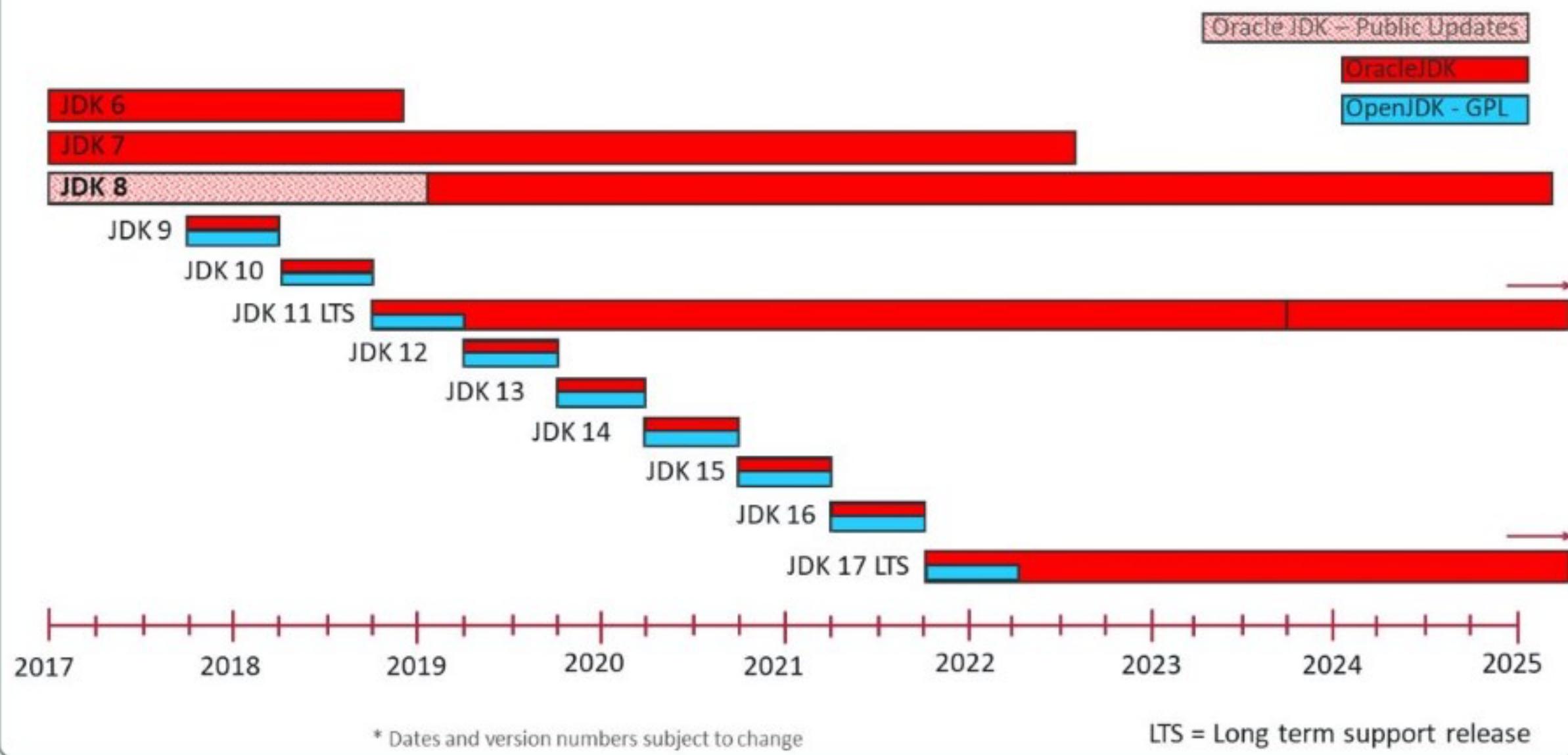
# JDK 17 baseline

An 🐘 in the room

Which versions of Java do you regularly use?



## New Java Release Timeline\*



<https://www.jetbrains.com/lp/devcosystem-2021/>

<https://www.promatis.de/en/2018/09/30/what-is-behind-the-new-license-model-of-oracle-java/>

# JDK 17 baseline

An  in the room

- В JDK 17 появилось много хорошего
- И за-deprecate-илось тоже
- Strongly Encapsulated JDK internals. Больше никаких –permit-access=all
- Закладывается основа для поддержки Spring Framework на годы вперед
- Поддержка JDK 18, 19 ...

# JPMS

Jigsaw  которого мы все ждали

- Не просто JDK 17, а еще и переход на Java Platform Module System
- Модули для основных компонентов фреймворка
- Дает возможность использовать jlink, jpackage, строгую инкапсуляцию, и вообще все преимущества JPMS
- Возможно, станет поворотным моментом для JPMS

# JPMS

## SPR-13501

### Declare Spring modules with JDK 9 module metadata [SPR-13501] #18079

[New issue](#)

 Open spring-projects-issues opened this issue on 24 Sep 2015 · 14 comments

spring-projects-issues commented on 24 Sep 2015 · edited

Juergen Hoeller opened SPR-13501 and commented

JDK 9's Jigsaw initiative aims to allow for module metadata (`module-info.java`) that can be added to framework and library jars while still keeping them compatible with JDK 8. Let's do this for Spring Framework 5.0's modules as far as feasible. However, we might not be able to express our optional dependency arrangement that way, in which case we might have to resort to an "automatic modules" approach for the more modest purposes of #18289.

**Issue Links:**

-  [Stable module names for Spring Framework jars on JDK 9 module path \[SPR-13716\]](#) #18289 Stable module names for Spring Framework jars on JDK 9 module path ("depends on")
-  [Upgrade core framework build to JDK 17](#) #17778 Upgrade core framework build to JDK 11 ("depends on")
-  [Document Spring recommendations/restrictions for Java 9 module setups \[SPR-14579\]](#) #19148 Document Spring recommendations/restrictions for Java 9 module setups

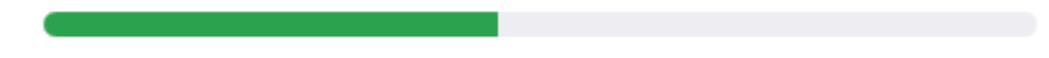
1 votes, 12 watchers

  5

Assignees  
 jhoeller

Labels  
 type: enhancement

Projects  
None yet

Milestone  
 6.0 M1

Linked pull requests  
Successfully merging a pull request may close this issue.

None yet

Notifications  Customize

 [Subscribe](#)

# JPMS

## Больше никакого десктопа

Remove dependency on java.desktop and other modules #26884

New issue

 Open brunoborges opened this issue on 29 Apr · 8 comments

 brunoborges commented on 29 Apr

Spring requires the `java.desktop` module to be present at Java runtimes only so that it can use the classes in the `java.beans` package.

Would be a good start for modernizing Spring apps on a post Java 9+ era of modules if Spring could at least work on slim Java runtimes produced with `jlink`.

Other modules to consider adjusting the dependency are `java.naming`, `java.xml`, `java.sql`, `jdk.*`, `java.instrument`, `java.management`, `java.rmi`, `java.scripting`.

This is not about making Spring compatible with Java SE modules. This is only to allow developers to have smaller Java runtimes created with `jlink`.

In an ideal world, a Java runtime, created with the following `jlink` command should be sufficient to run a Spring Boot Hello World with the Web dependency:

```
$ jlink \
  --add-modules java.base,java.logging \
  --strip-debug \
  --no-man-pages \
  --no-header-files \
  --compress=2 \
  --output /javaruntime
```

  4

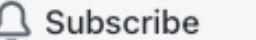
Assignees  
 jhoeller

Labels  
in: core type: enhancement

Projects  
None yet

Milestone  
6.0 M1

Linked pull requests  
Successfully merging a pull request may close this issue.  
None yet

Notifications  
Customize  


You're not receiving notifications from this thread.

7 participants

# Deprecations

## Но только возможные

- Drop RPC-style remoting: Hessian, HTTP Invoker, JMS Invoker, JAX-WS
- SecurityManager
- EJB
- Joda-Time support removal
- RxJava 1/2
- FactoryBean
- Autowire через сеттеры (наконец-то!)
- XML Configuration 😱 (**not possible at all**)

# Spring Native

# Spring Native

- 0.11.x - скорей всего последний релиз
- Native goes in Spring Boot 3
- AOT engine goes into Spring Core 6.0

# resources/META-INF/spring.factories

# Logging Systems

```
org.springframework.boot.logging.LoggingSystemFactory=\
org.springframework.boot.logging.logback.LogbackLoggingSystem.Factory,\
org.springframework.boot.logging.log4j2.Log4J2LoggingSystem.Factory,\
org.springframework.boot.logging.java.JavaLoggingSystem.Factory
```

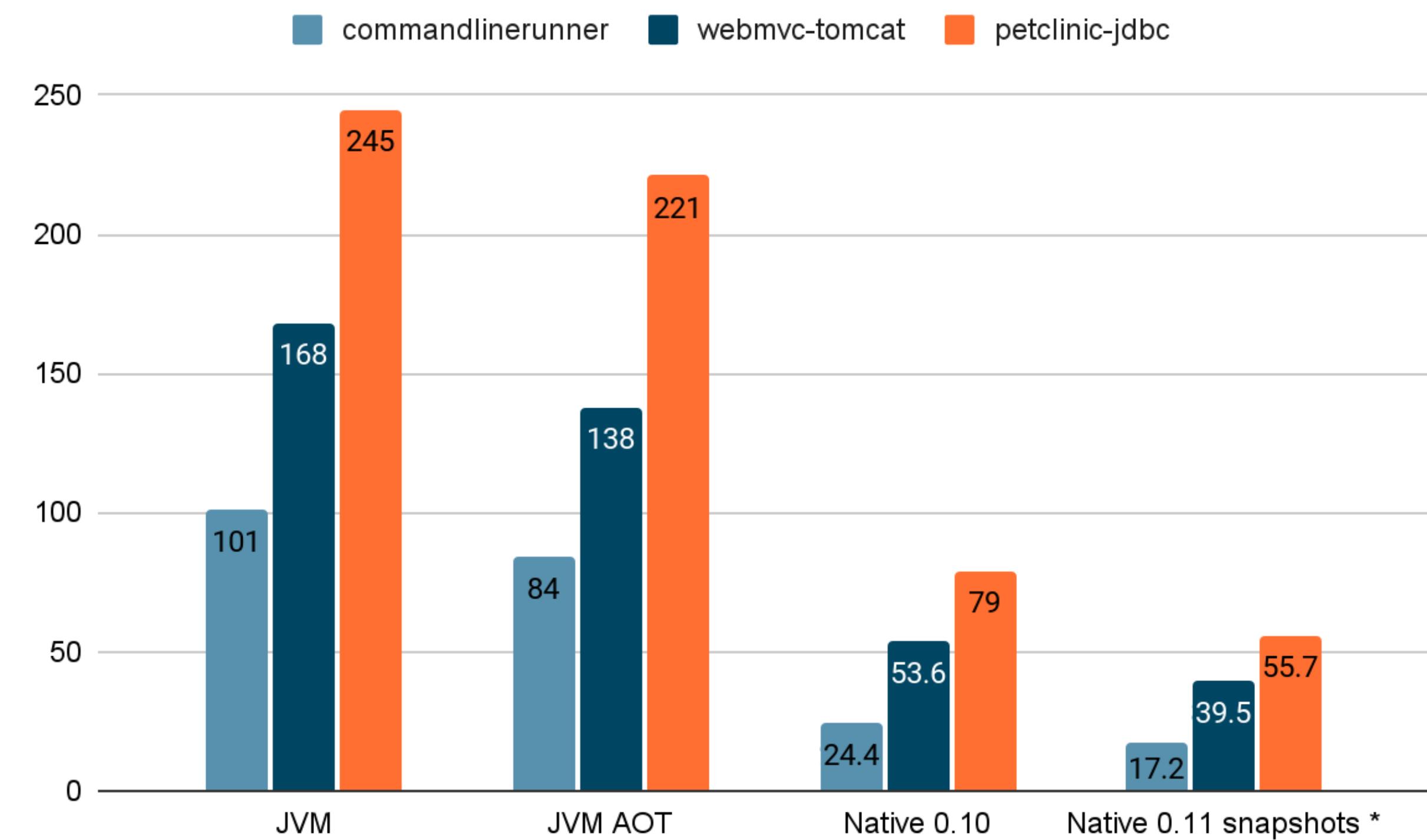
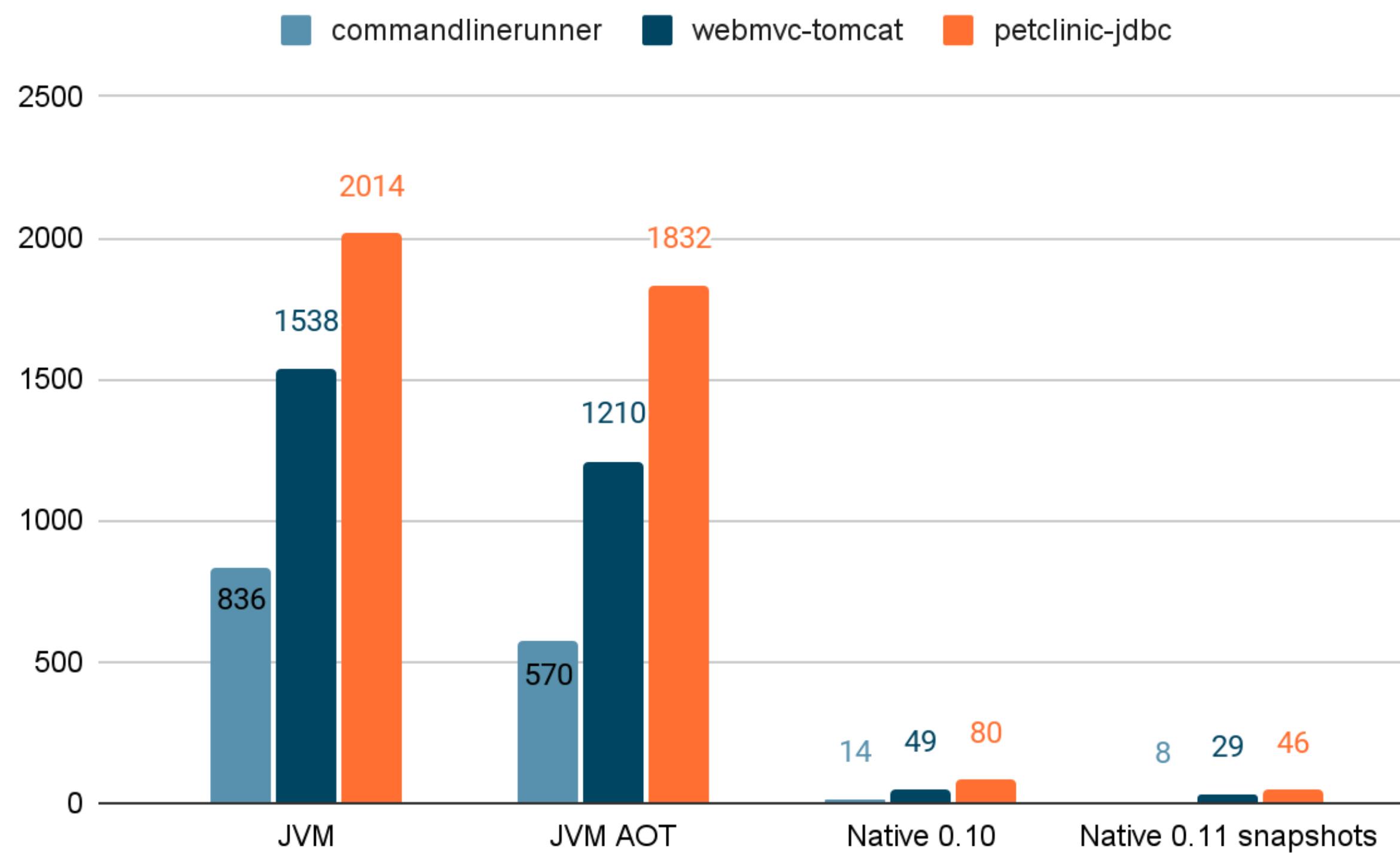
# PropertySource Loaders

```
org.springframework.boot.env.PropertySourceLoader=\
org.springframework.boot.env.PropertiesPropertySourceLoader,\
org.springframework.boot.env.YamlPropertySourceLoader
```

```
public abstract class StaticSpringFactories {  
    public static MultiValueMap<Class, Supplier<Object>> factories = new LinkedMultiValueMap();  
  
    static {  
        factories.add(org.springframework.boot.logging.LoggingSystemFactory.class,  
            () -> new org.springframework.boot.logging.logback.LogbackLoggingSystem.Factory());  
        factories.add(org.springframework.boot.logging.LoggingSystemFactory.class,  
            () -> new org.springframework.boot.logging.log4j2.Log4J2LoggingSystem.Factory());  
        factories.add(org.springframework.boot.logging.LoggingSystemFactory.class,  
            () -> new org.springframework.boot.logging.java.JavaLoggingSystem.Factory());  
        factories.add(org.springframework.boot.env.PropertySourceLoader.class,  
            () -> new org.springframework.boot.env.PropertiesPropertySourceLoader());  
    }  
}
```

```
class ContextBootstrapInitializer implements ApplicationContextInitializer<GenericApplicationContext> {  
    @Override  
    public void initialize(GenericApplicationContext context) {  
        context.getDefaultListableBeanFactory()  
            .setAutowireCandidateResolver(new ContextAnnotationAutowireCandidateResolver());  
  
        BeanDefinitionRegistrar.of("commandlinerunnerApplication", CommandlinerunnerApplication.class)  
            .instanceSupplier(CommandlinerunnerApplication::new).register(context);  
        BeanDefinitionRegistrar.of("CLR", CLR.class).instanceSupplier(CLR::new).register(context);  
        BeanDefinitionRegistrar.of("propertySourcesPlaceholderConfigurer",  
            PropertySourcesPlaceholderConfigurer.class).instanceSupplier(() ->  
            PropertyPlaceholderAutoConfiguration.propertySourcesPlaceholderConfigurer()).register(context);  
        // ...  
    }  
}
```

# Startup time & Memory Footprint



**Even smaller numbers for Spring 6**

# **Spring Observability**

<https://github.com/spring-projects-experimental/spring-observability>

# **Spring Cloud Sleuth**

# Spring Observability

- standard SPI for tracing and telemetry (micrometer like model)
- each module can use SPI to report telemetry
- different integrations can be supported

# Spring Web

# Spring Web

- Поддержка ошибок по RFC-7807 ([https://datatracker.ietf.org/doc/html/rf7807](https://datatracker.ietf.org/doc/html/rfc7807))
- Review MVC vs Boot style of configs (@EnableWebMvc vs Boot auto-configuration)
- @EnableWebMvc, @EnableWebFlux, но нет @EnableWebFn?

# **WebMVC**

# Jakarta EE 9

Support for Jakarta EE 9 (annotations and interfaces in jakarta.\* namespace)

#25354

New issue

Closed zenbones opened this issue on 2 Jul 2020 · 13 comments

zenbones commented on 2 Jul 2020

What is the timing for the switch to the jakarta.\* namespaced dependencies in JEE9? Is there a version I can use or build that's made that change?

29

jhoeller self-assigned this on 2 Jul 2020

spring-projects-issues added the `status: waiting-for-triage` label on 2 Jul 2020

Assignees  
jhoeller

Labels  
`type: dependency-upgrade`

Projects  
None yet

Milestone  
6.0 M1

<https://github.com/spring-projects/spring-framework/issues/25354>

# Jakarta EE 9

- Это breaking change, поменялся namespace  
`javax.*` – `jakarta.*`
- Вызвано ~~адвокатами Oracle~~ изменениями в API
- Зато будут доступны Tomcat 10 и Jetty 11
- Основа для будущих EE версий (Jakarta EE 9)

# Hibernate 6

- Hibernate Validator 7.x
  - Такой же, как Validator 6.x, но с jakarta.\* неймспейсом
- Hibernate ORM 6.x

Уже хотя бы есть альфа.

# **WebFlux**

# Spring WebFlux

- HttpClient from JDK11
- Rework Threading model for non-Reactive cases
- ThreadLocal <-> Reactor Context interop

# Project Reactor

# Reactor Core

- 4.0 is planned (<https://github.com/reactor/reactor/issues/703>) (<https://github.com/reactor/reactor/issues/705>)
- Performance Improvements
- Unification of Concurrency Patterns
- Codebase cleanup
- Better API research

# Reactor Netty

- 2.0??
- Netty 5
- QUIC support

# Spring Kotlin

# Spring Kotlin

- K2-Compiler - 3rd party compilers plugin support
- Better AOT via kotlin-metadata-jvm
- Spring Fu?
- Kotlin Compose

# Beyond

- Spring 6 это фундамент для развития фреймворка на следующие N-цать лет
- Цикл релизов, возможно, изменится на полугодовой, как и у Boot

# Когда?

- M1 - Nov/Dec 2021
- M2
- M3
- M4
- RC1 - July 2022
- RC2
- RC3
- GA - Oct 2022

<https://github.com/spring-projects/spring-framework/issues?q=milestone%3A%226.0+M1%22>

# Feedback

- Как дать обратную связь команде
- Все еще не окончательно, многое может измениться

**Спасибо!**