SPRING FRAMEWORK

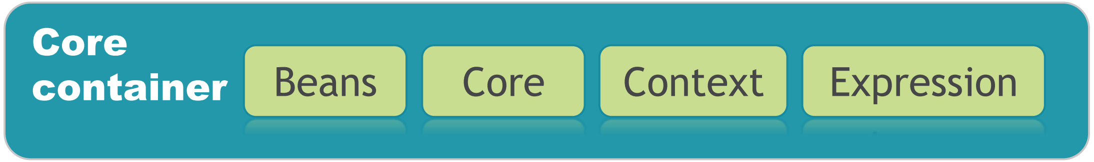
ARCHITECTURE

APRIL, 2015

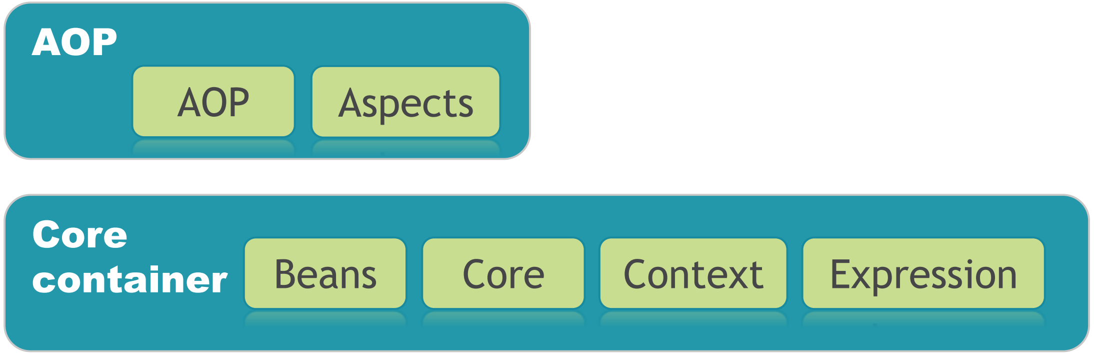
WHAT IS SPRING?

- Lightweight jar libraries
- Container
  - Manages lifecycle of objects
- Framework
  - Classes and utilities for application creation
- Dependency Injection (Inversion of Control)
  - Objects get their dependencies and do not create them
- AOP (aspect oriented programming)
  - Separation of cross-cutting concerns into separate modules

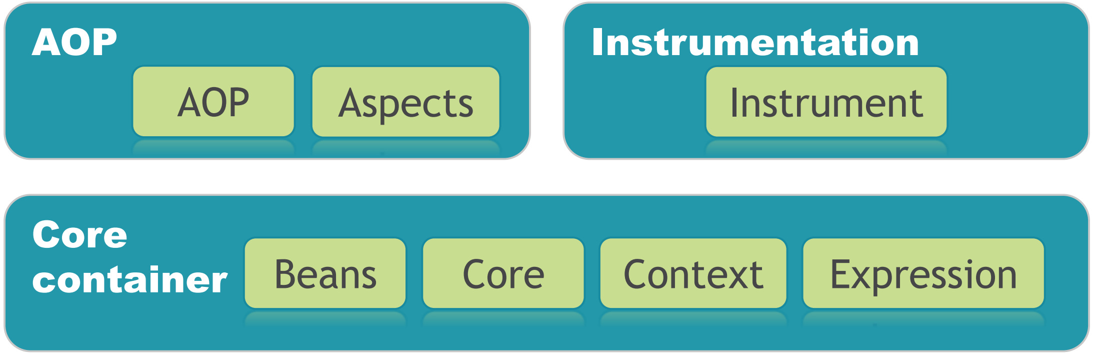
SPRING MODULES



SPRING MODULES



SPRING MODULES



AOP

AOP

Aspects

Instrumentation

Instrument

Core container

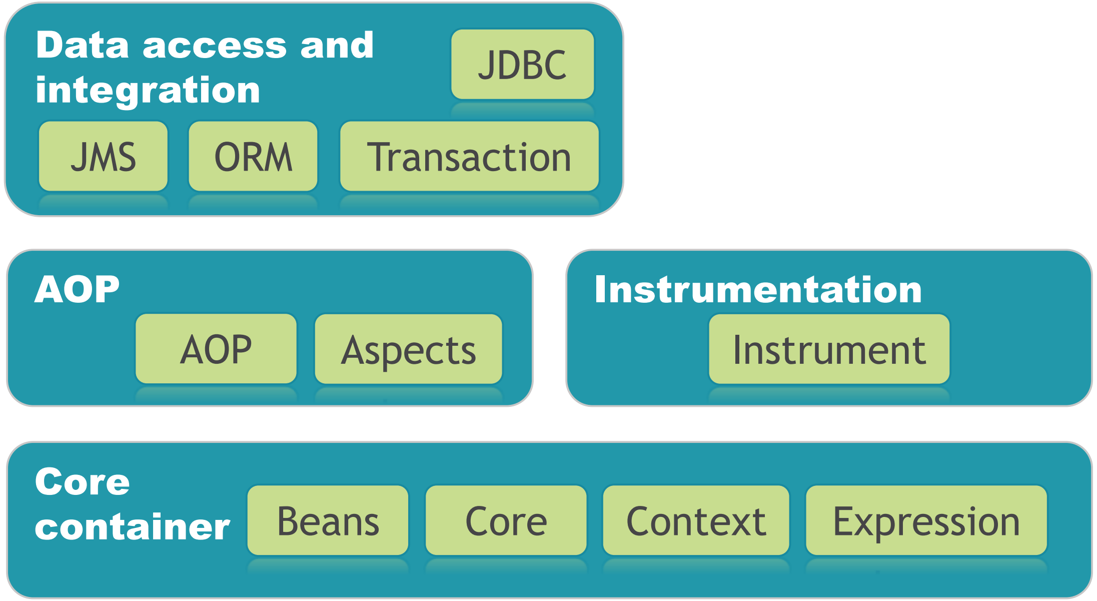
Beans

Core

Context

Expression

SPRING MODULES



Data access and integration

JDBC

JMS

ORM

Transaction

AOP

AOP

Aspects

Instrumentation

Instrument

Core container

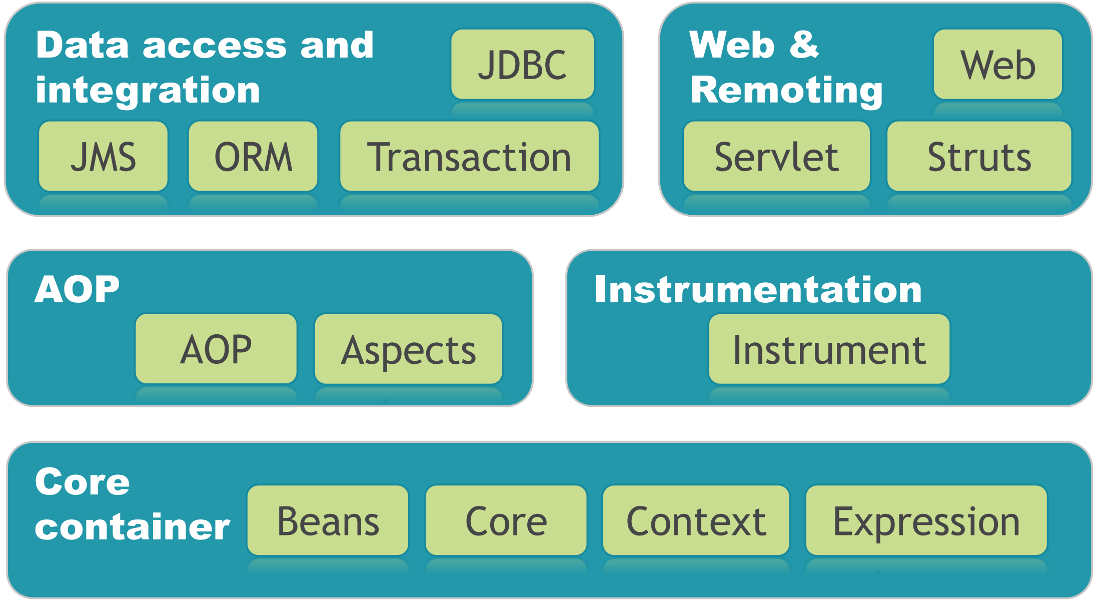
Beans

Core

Context

Expression

SPRING MODULES



Data access and integration

JDBC

JMS

ORM

Transaction

Web & Remoting

Web

Servlet

Struts

AOP

AOP

Aspects

Instrumentation

Instrument

Core container

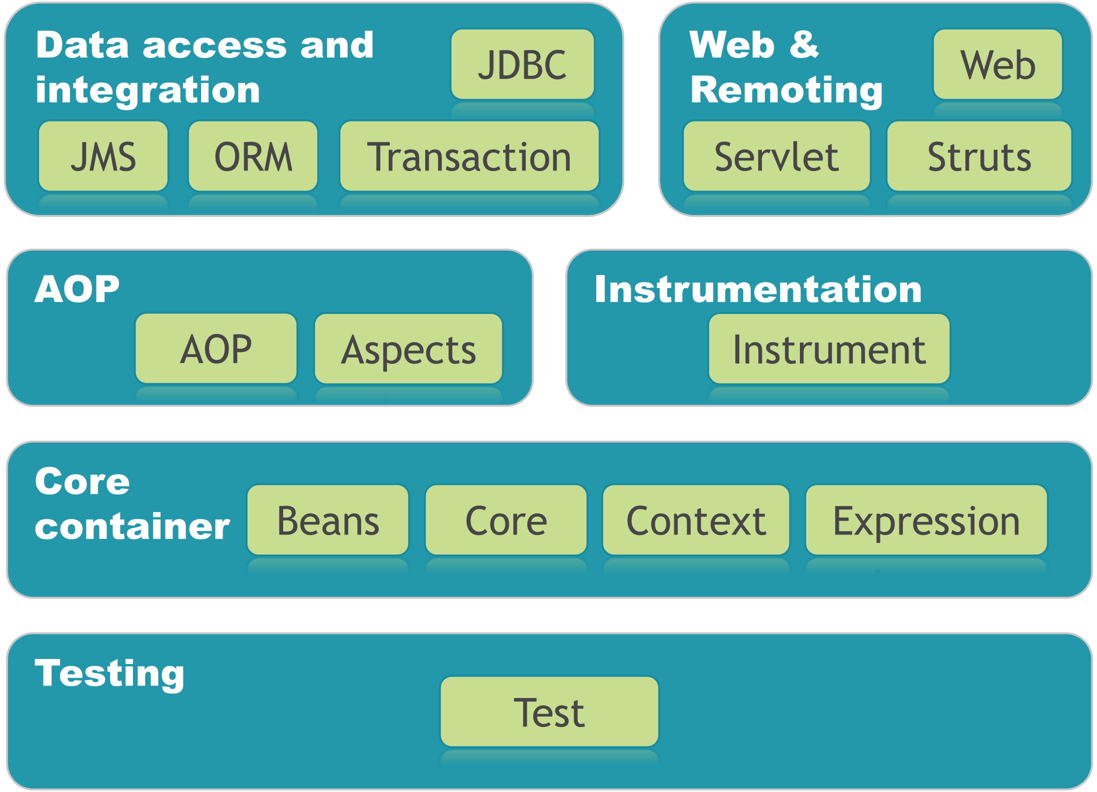
Beans

Core

Context

Expression

SPRING MODULES



SPRING MODULES

DIY

- Create Maven Project
  - **groupId**: com.epam.spring
  - **artifactId**: com.epam.spring.core
- In pom.xml add dependencies for Spring modules:
  - spring-context
  - spring-context-support
  - spring-tx
  - spring-jdbc

SPRING MODULES

POM.XML

DIY

```
<properties>
    <spring.ver>3.2.13.RELEASE</spring.ver>
</properties>

<dependencies>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>${spring.ver}</version>
    </dependency>

   ...
</dependencies>
```

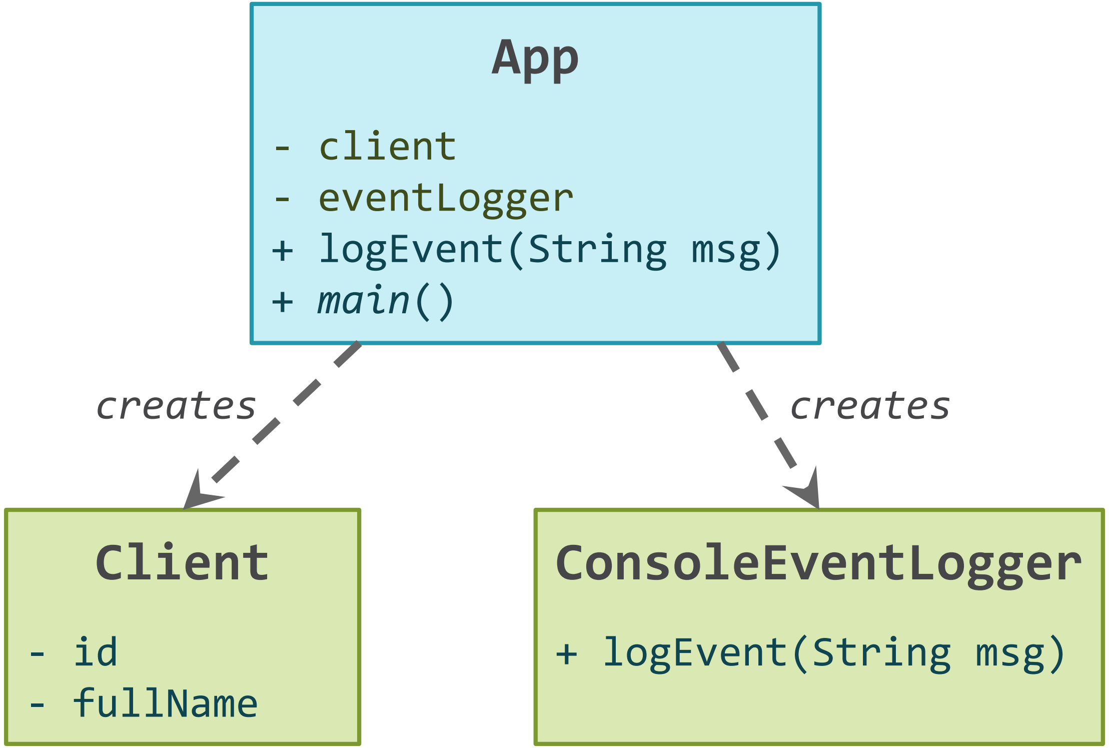
spring-context-support

spring-tx

spring-jdbc

EXAMPLE

DIY



App
- client
- eventLogger
+ logEvent(String msg)
+ *main*()
creates
creates
Client
- id
- fullName
ConsoleEventLogger
+ logEvent(String msg)

EXAMPLE

DIY

```java
public static void main(String[] args) {
    App app = new App();

    app.client = new Client("1", "John Smith");
    app.eventLogger = new ConsoleEventLogger();

    app.logEvent("Some event for user 1");
}

private void logEvent(String msg) {
    String message = msg.replaceAll(
        client.getId(), client.getFullName());
    eventLogger.logEvent(message);
```

APPLICATION PROBLEMS

APPLICATION PROBLEMS

- Modification is problematic
  - Data inside code

APPLICATION PROBLEMS

- Modification is problematic
  - Data inside code

- Scaling is impossible
  - Logger is created in single instance

APPLICATION PROBLEMS

- Modification is problematic
  - Data inside code

- Scaling is impossible
  - Logger is created in single instance

- Testing is hard
  - Unit test for App.logEvent() method indirectly tests ConsoleEventLogger.logEvent()

APPLICATION PROBLEMS

- Modification is problematic
  - Data inside code
- Scaling is impossible
  - Logger is created in single instance
- Testing is hard
  - Unit test for App.logEvent() method indirectly tests ConsoleEventLogger.logEvent()

HIGH / TIGHT COUPLING

THANK YOU

YURIY_TKACH@EPAM.COM

APRIL, 2015