* Dependency Injection
* Loose Coupling

Bean = Instance of an object

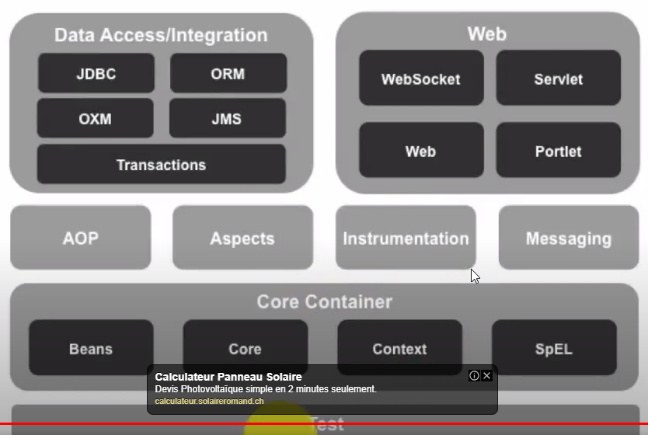
What are the beans? @Component (BinarySearchImpl, BubbleSortAlgorithm)

What are the dependencies of a bean? @Autowired (z.B. sortAlgorithm (Interface) is a dependency of BinarySearchImpl and BubbleSortAlgorithm)

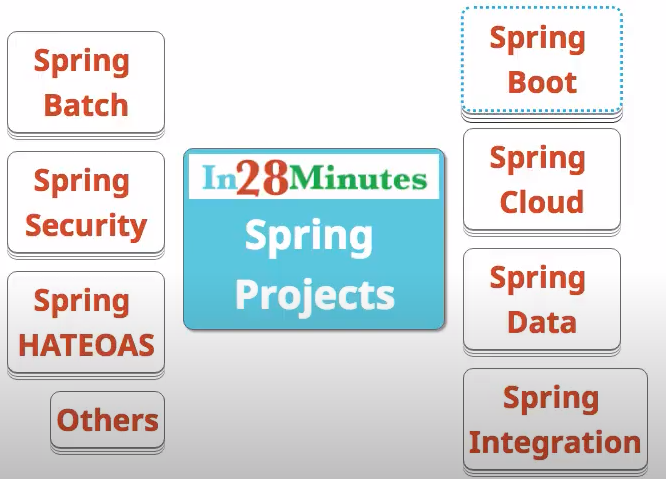
What to search for beans? Springboot would automatically scan the package & subpackages containing the main application class (containing the @SpringBootApplication annotation) for the beans/components.

Application Context is the place where Spring does manage all the beans.

Spring Modules:



Spring Projects:



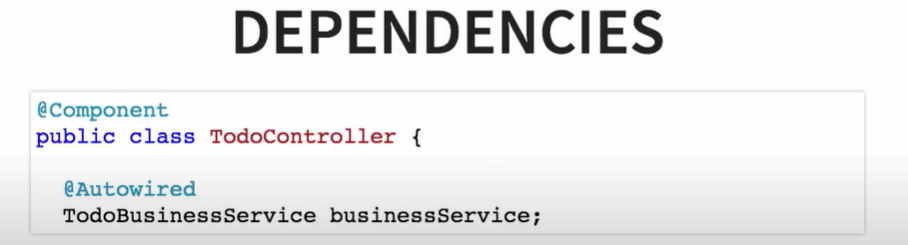
Spring Boot: Makes development of microservices simpler/faster.

**Features (Spring Boot)**

* Create stand-alone Spring applications
* Embed Tomcat, Jetty or Undertow directly (no need to deploy WAR files)
* Provide opinionated 'starter' dependencies to simplify your build configuration
* Automatically configure Spring and 3rd party libraries whenever possible
* Provide production-ready features such as metrics, health checks, and externalized configuration
* Absolutely no code generation and no requirement for XML configuration

**Deep Dive “general”**

We develop business application of several layers (UI/Web, BusinessLogic/Service, Data). In business applications we do have thousands of dependencies.



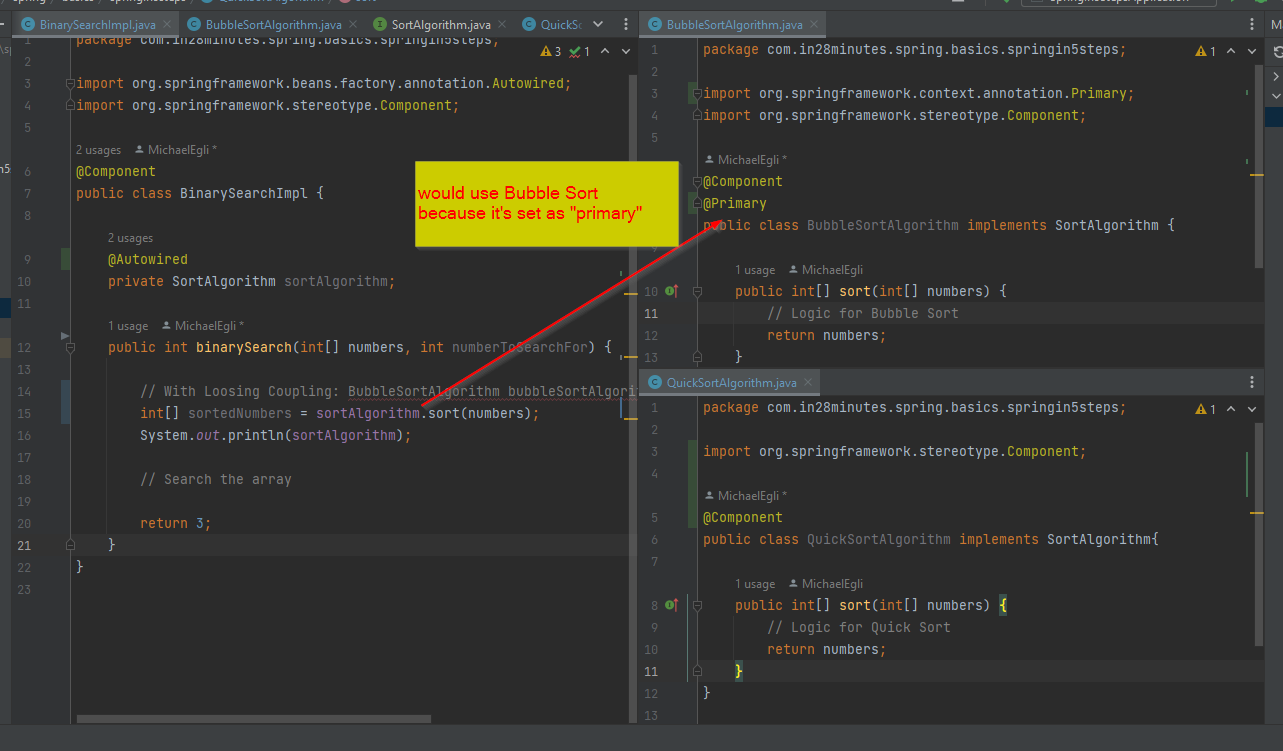
TodoController might be used to render the screen with ToDo Items. It makes use of the businessService to get all the items. TodoBuninessService is a dependency of the TodoController.

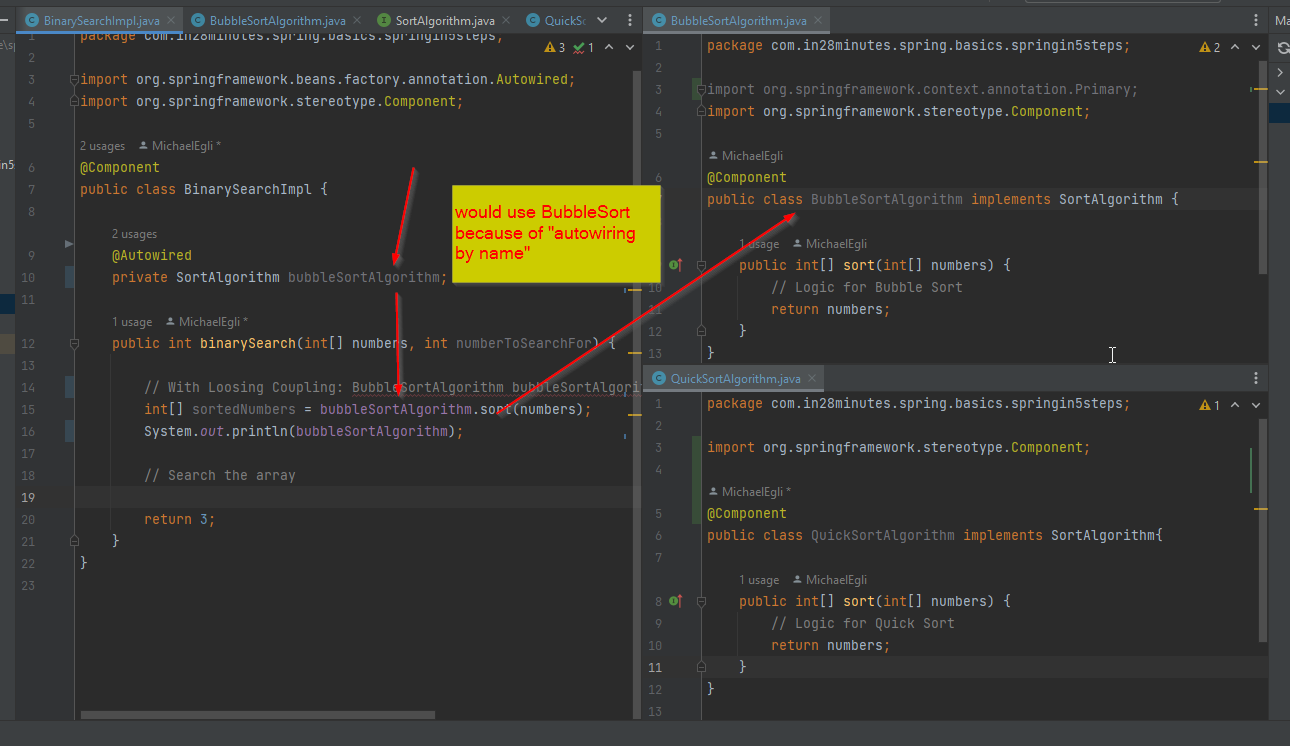
**Deep Dive “Autowiring”**

Options to resolve multiple candidates of dependencies:

* @primary
* Autowiring by name
* By Qualifier

Assume we’ve a component “SortAlgorithm” and two implementations of sorting (BubbleSort & QuickSort). We could use either “@Primary” annotation or “Autowiring by Name ”to choose between“ BubbleSort or QuickSort.

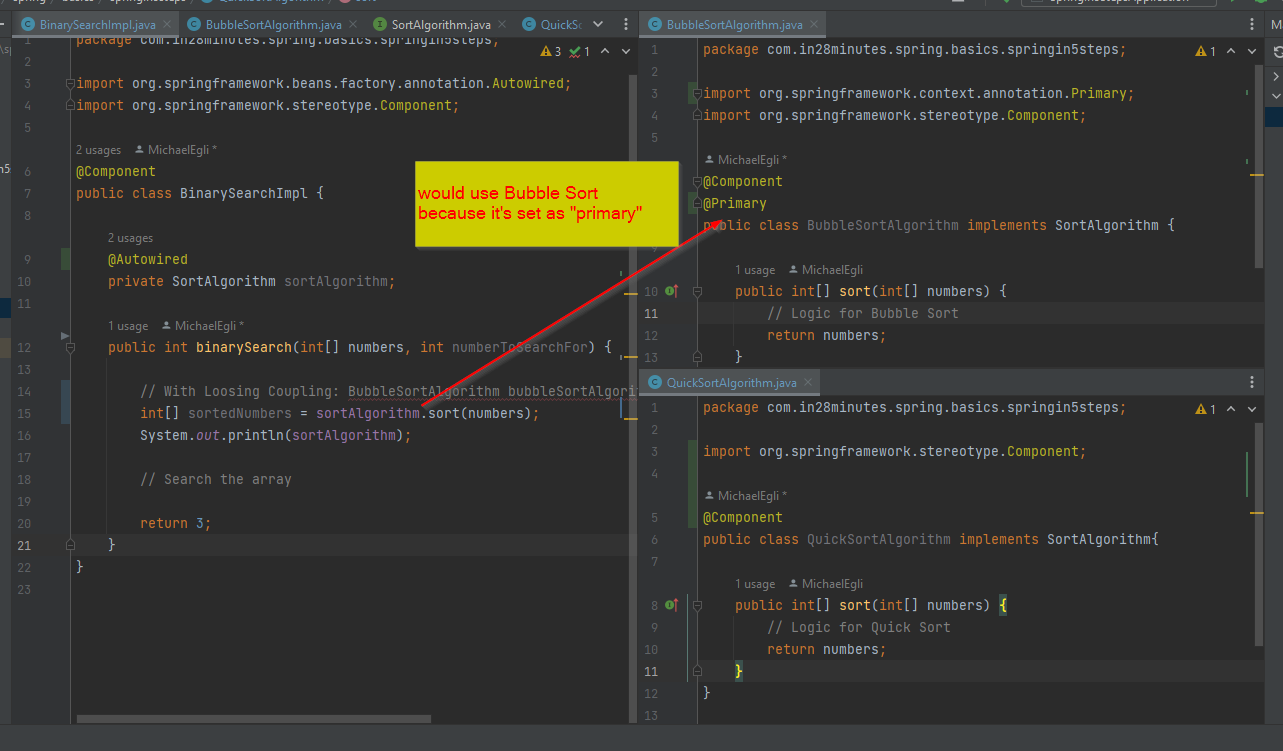




Important: “@Primary” has higher priority than “autowiring by name”.

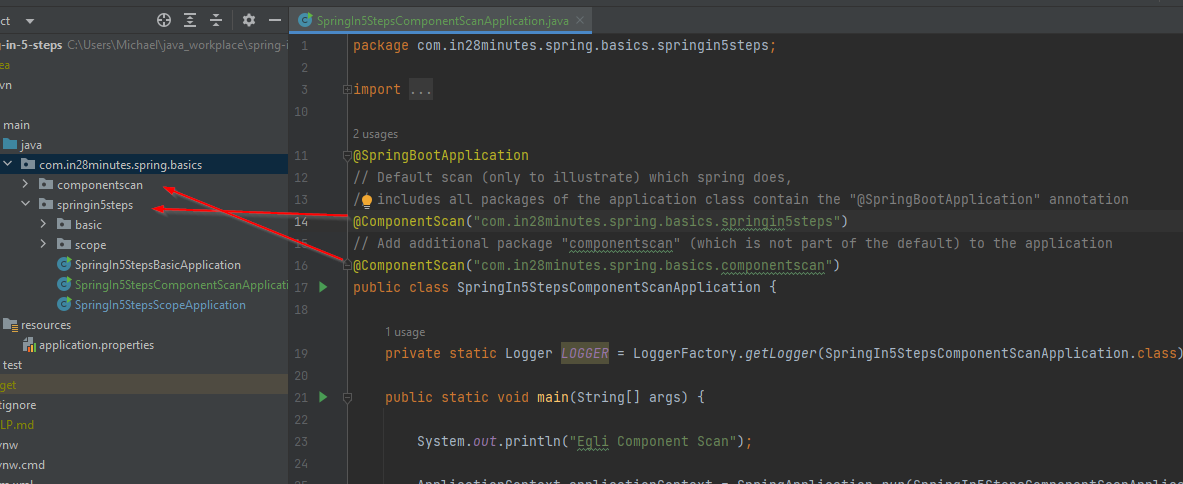
If “@Primary” is set at “BubbleSort” it would still use “BubbleSort” even “QuickSort” has been autowired by name in the “BinaySearchImpl” class in the example above.

“Qualifier” annotation makes sense if you want selectively use a sort algorithm for specific use.



**Component Scan (no bean found exception)**

* Can be used to add additional packages to the spring application



**Lifecycle of a bean**

“@PostConstruct”: is called after the instance of has been created and before any method of the class is executed.

“@PreDestroy”: is called before the bean is removed from the application context by the container

**IOC Container**

Inversion of Control (IOC):

* Software engineering design pattern (not only in Spring context)
* Control of the dependencies moves to Framework (Spring)
* The IOC Container in Spring is the place where IOC is implemented

<https://docs.spring.io/spring-framework/docs/current/reference/html/core.html>

**Application Context**

* Spring recommends to use the application context and not the bean factory
* Is “Bean Factory++” (provides more features than the Bean Factory)
* Provides features which are typically used in enterprise applications   
  - Spring AOP features   
  - l18n capabilities – internationalization f.e. show text, currency or timestamp depending the user country location  
  - WebApplicationContext for web applications

**Bean Factory**

* Provides basic management of beans and wiring mock dependencies
* Can be used by business application to control memory f.e.

@Component: Applies to UI/Web, Service (Business) and Database

@Controller: Only UI/Web

@Service: Only Service (Business)

@Repository: Only Datalevel (Database)

Spring does provide specific exception handling to each layer. @Repository has specific exception handling for jdbc f.e. Furthermore it allows us to develop code which is only applied on the specific application layer (logging on business layer f.e.).