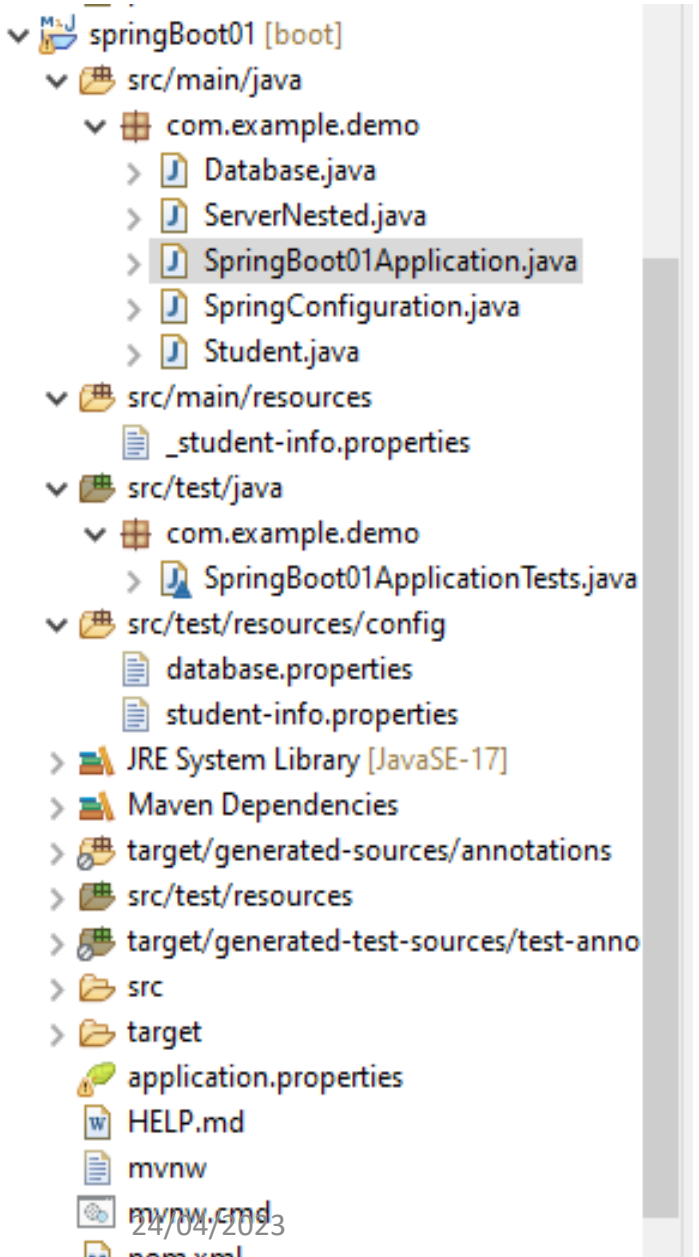
EXPLORING DEPENDING INJECTION

springboot01

SpringBoot01 test dependency injection




```java
import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

//@EnableConfigurationProperties(Database.class)

@SpringBootApplication
public class SpringBoot01Application implements CommandLineRunner {

    @Autowired
    private Database database;

    @Autowired
    private Student student;

    @Value("ABCS")
    private String stringValue;

    @Autowired
    private List<String> listOfValues;

    private List<String> listOfValuesBySetter = new ArrayList<>();

    @Value("#{${valuesMap}}")
    private Map<String, Integer> valuesMap;

    @Value("#{${valuesMap}.key1}")
    private String valueMapKey1;

    @Value("#{${unknownMap : {key1: '11', key2: '22'}}}")
    private Map<String, Integer> unknownMap;
```

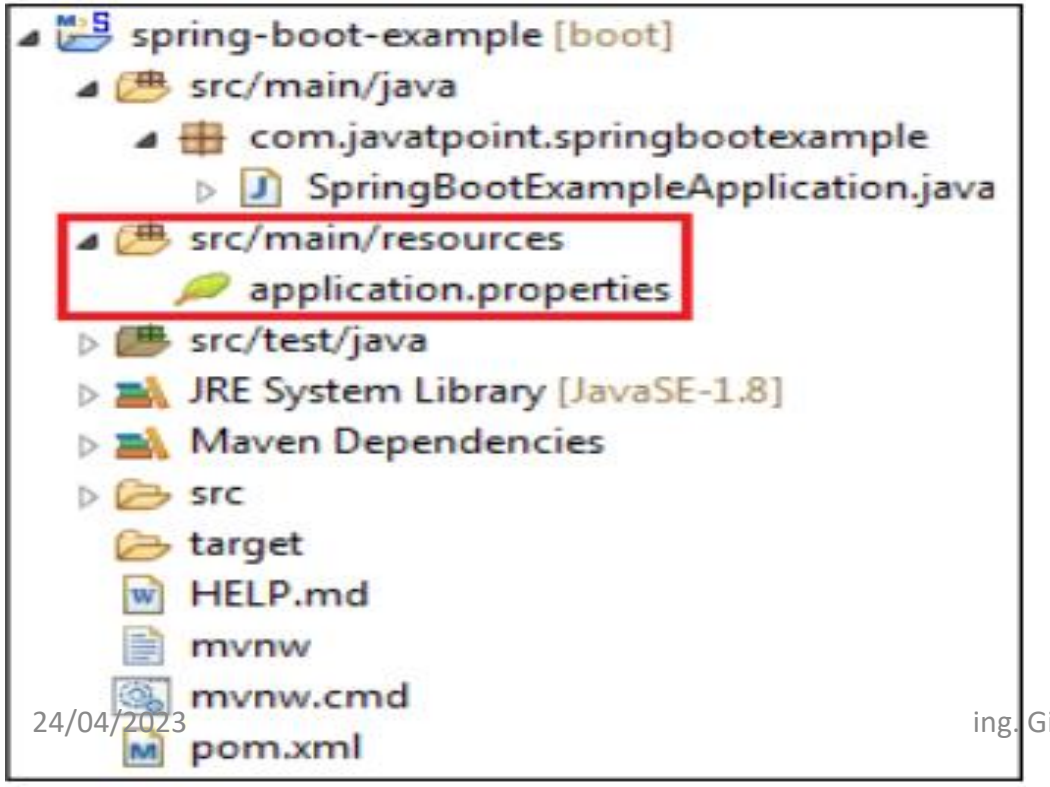
Spring Boot Gestione Dipendenze

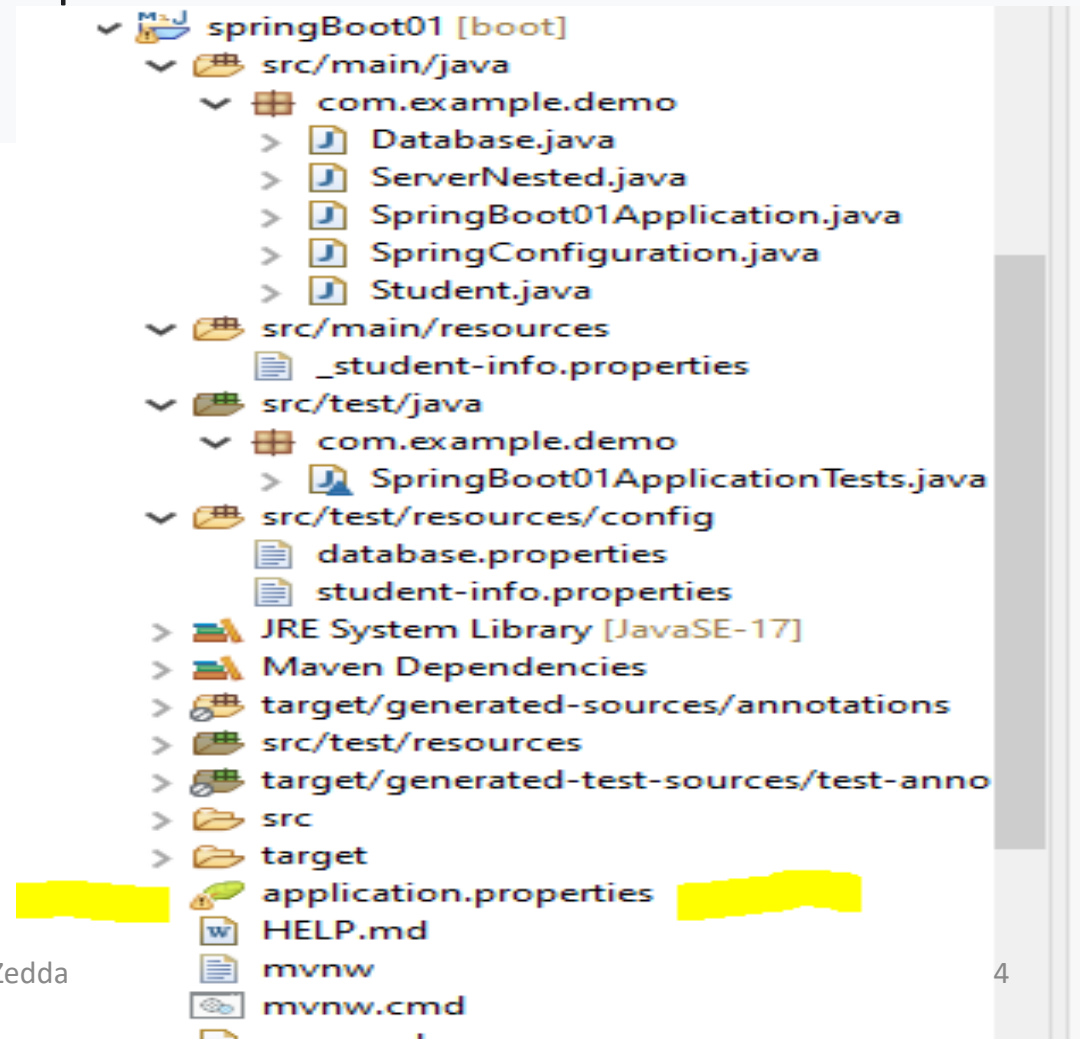
- Spring Boot gestisce automaticamente le dipendenze e la configurazione e ogni versione di Spring Boot fornisce un elenco di dipendenze supportate, gestite da Maven
- Non è necessario specificare la versione delle dipendenze nella nostra configurazione, Spring Boot si gestisce da solo.
- Spring Boot aggiorna automaticamente tutte le dipendenze in modo coerente quando aggiorniamo la versione di Spring Boot, specificata in un'unica posizione, semplificando il passaggio da una versione all'altra
- Spring Boot garantisce che tutte le versioni delle librerie sono compatibili
- Nel POM di Maven **spring-boot-starter-parent** eredita automaticamente le funzionalità di base

```
5    <parent>
6        <groupId>org.springframework.boot</groupId>
7        <artifactId>spring-boot-starter-parent</artifactId>
8        <version>3.0.5</version>
9        <relativePath/> <!-- lookup parent from repository -->
10   </parent>
```

Spring Boot Application Properties

- Spring Boot Framework viene fornito con un meccanismo integrato per la configurazione dell'applicazione attraverso un file chiamato **application.properties**.
- Si trova all'interno della cartella **src/main/resources**, come mostrato nella figura seguente ma possono anche essere definiti file custom multipli di properties.
- Se il file non viene trovato nella cartella predisposta, viene individuato nella root del progetto





Spring Boot Application Properties

- Esempio di **application.properties**.

```
#configuring application name
spring.application.name = demoApplication
#configuring port
server.port = 8081
```

- Abbiamo configurato il **nome** e la **porta** dell'applicazione. La porta 8081 indica che l'applicazione viene eseguita sulla porta **8081**.

Spring Boot Categorie di Properties

- Ci sono **16** categorie di Spring Boot Properties

1. Core Properties
2. Cache Properties
3. Mail Properties
4. JSON Properties
5. Data Properties
6. Transaction Properties
7. Data Migration Properties
8. Integration Properties
9. Web Properties
10. Templating Properties
11. Server Properties
12. Security Properties
13. RSocket Properties
14. Actuator Properties
15. DevTools Properties
16. Testing Properties

Spring Boot Tabella Properties

Property	Default Values	Description
Debug	false	It enables debug logs.
spring.application.name		It is used to set the application name.
spring.application.admin.enabled	false	It is used to enable admin features of the application.
spring.config.name	application	It is used to set config file name.
spring.config.location		It is used to config the file name.
server.port	8080	Configures the HTTP server port
server.servlet.context-path		It configures the context path of the application.
logging.file.path		It configures the location of the log file.
spring.banner.charset	UTF-8	Banner file encoding.

Spring Boot Tabella Properties

Property	Default Values	Description
spring.banner.location	classpath:banner.txt	It is used to set banner file location.
logging.file		It is used to set log file name. For example, data.log.
spring.application.index		It is used to set application index.
spring.application.name		It is used to set the application name.
spring.application.admin.enabled	false	It is used to enable admin features for the application.
spring.config.location		It is used to config the file locations.
spring.config.name	application	It is used to set config the file name.
spring.mail.default-encoding	UTF-8	It is used to set default MimeMessage encoding.

Spring Boot Tabella Properties

Property	Default Values	Description
spring.mail.host		It is used to set SMTP server host. For example, smtp.example.com.
spring.mail.password		It is used to set login password of the SMTP server.
spring.mail.port		It is used to set SMTP server port.
spring.mail.test-connection	false	It is used to test that the mail server is available on startup.
spring.mail.username		It is used to set login user of the SMTP server.
spring.main.sources		It is used to set sources for the application.
server.address		It is used to set network address to which the server should bind to.
server.connection-timeout		It is used to set time in milliseconds that connectors will wait for another HTTP request before closing the connection.
server.context-path		It is used to set context path of the application.

Spring Boot Tabella Properties

Property	Default Values	Description
server.server-header		It is used for the Server response header (no header is sent if empty)
server.servlet-path	/	It is used to set path of the main dispatcher servlet
server.ssl.enabled		It is used to enable SSL support.
spring.http.multipart.enabled	True	It is used to enable support of multi-part uploads.
spring.servlet.multipart.max-file-size	1MB	It is used to set max file size.
spring.mvc.async.request-timeout		It is used to set time in milliseconds.
spring.mvc.date-format		It is used to set date format. For example, dd/MM/yyyy.
spring.mvc.locale		It is used to set locale for the application.

Spring Boot Tabella Properties

Property	Default Values	Description
spring.social.facebook.app-id		It is used to set application's Facebook App ID.
spring.social.linkedin.app-id		It is used to set application's LinkedIn App ID.
spring.social.twitter.app-id		It is used to set application's Twitter App ID.
security.basic.authorize-mode	role	It is used to set security authorize mode to apply.
security.basic.enabled	true	It is used to enable basic authentication.
Spring.test.database.replace	any	Type of existing DataSource to replace.
Spring.test.mockmvc.print	default	MVC Print option
spring.freemaker.content-type	text/html	Content Type value
server.server-header		Value to use for the server response header.

Spring Boot Tabella Properties

Property	Default Values	Description
spring.security.filter.dispatcher-type	async, error, request	Security filter chain dispatcher types.
spring.security.filter.order	-100	Security filter chain order.
spring.security.oauth2.client.registration.*		OAuth client registrations.
spring.security.oauth2.client.provider.*		OAuth provider details.

Spring Boot Starters

- **Spring Boot** fornisce una serie di **starter** integrati che consentono di aggiungere jar nel classpath e rendono lo sviluppo più semplice e rapido.
- Gli **Spring Boot Starter** sono i **descrittori di dipendenza**.
- In Spring Boot Framework, tutti gli starter seguono uno schema di denominazione simile: **spring-boot-starter-***, dove * denota un particolare tipo di applicazione.
  Ad esempio, se vogliamo utilizzare Spring e JPA per l'accesso al database, dobbiamo includere la dipendenza **spring-boot-starter-data-jpa** nel file **pom.xml** del progetto.
- Spring-boot-starter è riservato per artifact Spring Boot ufficiali e Spring Boot Framework fornisce gli starter di avvio dell'applicazione nel gruppo **org.springframework.boot**.
- Ci sono tre tipologie di Starter
  - Third-Party Starters
  - Spring Boot Framework Starters
  - Spring Boot Technical Starters

Spring Boot Framework Starters

Name	Description
spring-boot-starter-thymeleaf	It is used to build MVC web applications using Thymeleaf views.
spring-boot-starter-data-couchbase	It is used for the Couchbase document-oriented database and Spring Data Couchbase.
spring-boot-starter-artemis	It is used for JMS messaging using Apache Artemis.
spring-boot-starter-web-services	It is used for Spring Web Services.
spring-boot-starter-mail	It is used to support Java Mail and Spring Framework's email sending.
spring-boot-starter-data-redis	It is used for Redis key-value data store with Spring Data Redis and the Jedis client.
spring-boot-starter-web	It is used for building the web application, including RESTful applications using Spring MVC. It uses Tomcat as the default embedded container.
spring-boot-starter-data-gemfire	It is used to GemFire distributed data store and Spring Data GemFire.
spring-boot-starter-activemq	It is used in JMS messaging using Apache ActiveMQ.

Spring Boot Framework Starters

Name	Description
spring-boot-starter-integration	It is used for Spring Integration.
spring-boot-starter-test	It is used to test Spring Boot applications with libraries, including JUnit, Hamcrest, and Mockito.
spring-boot-starter-jdbc	It is used for JDBC with the Tomcat JDBC connection pool.
spring-boot-starter-mobile	It is used for building web applications using Spring Mobile.
spring-boot-starter-validation	It is used for Java Bean Validation with Hibernate Validator.
spring-boot-starter-hateoas	It is used to build a hypermedia-based RESTful web application with Spring MVC and Spring HATEOAS.
spring-boot-starter-jersey	It is used to build RESTful web applications using JAX-RS and Jersey. An alternative to spring-boot-starter-web.
spring-boot-starter-data-neo4j	It is used for the Neo4j graph database and Spring Data Neo4j.
spring-boot-starter-data-ldap	It is used for Spring Data LDAP.
spring-boot-starter-websocket	It is used for building the WebSocket applications. It uses Spring Framework's WebSocket support.

Spring Boot Framework Starters

Name	Description
spring-boot-starter-aop	It is used for aspect-oriented programming with Spring AOP and AspectJ.
spring-boot-starter-amqp	It is used for Spring AMQP and Rabbit MQ.
spring-boot-starter-data-cassandra	It is used for Cassandra distributed database and Spring Data Cassandra.
spring-boot-starter-social-facebook	It is used for Spring Social Facebook.
spring-boot-starter-jta-atomikos	It is used for JTA transactions using Atomikos.
spring-boot-starter-security	It is used for Spring Security.
spring-boot-starter-mustache	It is used for building MVC web applications using Mustache views.
spring-boot-starter-data-jpa	It is used for Spring Data JPA with Hibernate.
spring-boot-starter	It is used for core starter, including auto-configuration support, logging, and YAML.
spring-boot-starter-groovy-templates	It is used for building MVC web applications using Groovy

Spring Boot Production Starters

Name	Description
spring-boot-starter-actuator	It is used for Spring Boot's Actuator that provides production-ready features to help you monitor and manage your application.
spring-boot-starter-remote-shell	It is used for the CRaSH remote shell to monitor and manage your application over SSH. Deprecated since 1.5.

Spring Boot Technical Starters

Name	Description
spring-boot-starter-undertow	It is used for Undertow as the embedded servlet container. An alternative to spring-boot-starter-tomcat.
spring-boot-starter-jetty	It is used for Jetty as the embedded servlet container. An alternative to spring-boot-starter-tomcat.
spring-boot-starter-logging	It is used for logging using Logback. Default logging starter.
spring-boot-starter-tomcat	It is used for Tomcat as the embedded servlet container. Default servlet container starter used by spring-boot-starter-web.
spring-boot-starter-log4j2	It is used for Log4j2 for logging. An alternative to spring-boot-starter-logging.

Spring Boot Starter Parent

- Spring-boot-starter-parent è uno starter del progetto e fornisce configurazioni predefinite per tutte le applicazioni.
- Viene utilizzato internamente da tutte le dipendenze.
- Tutti i progetti Spring Boot usano spring-boot-starter-parent da cui ereditare tutte le dipendenze base, nel file pom.xml.

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http:
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://m
    <modelVersion>4.0.0</modelVersion>

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.0.5</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>

    <groupId>com.example</groupId>
    <artifactId>springIOBasic</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>springIOBasic</name>
    <description>Demo project for Spring Boot</description>
```

Spring Boot Starter Web

Ci sono due important caratteristiche di **spring-boot-starter-web**:

- Compatibilità per web development
- Autoconfigurazione

- Starter of Spring Web utilizza Spring MVC, REST e Tomcat come server incorporato predefinito.
- La singola dipendenza spring-boot-starter-web ingloba in modo transitivo tutte le dipendenze relative allo sviluppo web.

```
5⊖      <dependency>
6           <groupId>org.springframework.boot</groupId>
7           <artifactId>spring-boot-starter-web</artifactId>
8           <version>3.0.5</version>
9       </dependency>
```