# Usage Document Of Password Validation Service:

The project consists of a password validation spring boot starter and an example project.

1.  Password-validation-service-springboot-example : https://github.com/skazaruddin/password-validation-service-springboot-example
2.  A spring boot starter project.
    https://github.com/skazaruddin/password-validator-spring-boot-starter

    **The password-validator-library module contains the password service implementation classes and unit tests.**

3.  **The example project** : https://github.com/skazaruddin/password-validation-service-springboot-example
    **contains the service integration and the integration tests.**

    1.  **Configuring password-service-starter in Spring Boot Project**

        Build the starter project using 'mvn clean install'

        https://github.com/skazaruddin/password-validator-spring-boot-starter

        The example project uses the dependency :

        ```
        <dependency>
                <groupId>io.azar</groupId>
                <artifactId>spring-boot-starter-password-validator</artifactId>
                <version>0.0.1-SNAPSHOT</version>
        </dependency>
        ```

        Easily autowire in any component of your project.Below is an attached example.

        ```
        @Controller
        @RequestMapping("/api/")
        public class PasswordValidationController {

                @Autowired
                private ValidationService validationService;

                @PostMapping("/v1/validatePassword")
                @ResponseBody
                public ResponseEntity<String> validatePassword(@RequestBody PasswordDTO password) {

                        this.validationService.validatePassword(password.getPassword());
                        return new ResponseEntity<String>("Password passed compliance test successfully.",
        HttpStatus.OK);
                }

                @ExceptionHandler({ PasswordComplainceFailureException.class })
                protected ResponseEntity<String>
        handlePasswordLengthViolationException(PasswordComplainceFailureException ex) {
                        String error = ex.getMessage();
                        return new ResponseEntity<String>(error, HttpStatus.INTERNAL_SERVER_ERROR);
                }

        }
        ```

## 2. Configuring Password Service in Spring Project

1. Import the starter project https://github.com/skazaruddin/password-validator-spring-boot-starter and do **mvn clean install** to generate artifacts in Maven Local Repository.
2. Add the dependency in your project to use Validation Service

   ```xml
   <dependency>
           <groupId>io.azar</groupId>
           <artifactId>password-validator-library</artifactId>
           <version>0.0.1-SNAPSHOT</version>
   </dependency>
   ```
3. Create a configuration class to create bean of service

   ```java
   import org.springframework.context.annotation.Bean;
   import org.springframework.context.annotation.Configuration;

   import io.azar.pservice.service.DefaultValidationServiceImpl;
   import io.azar.pservice.service.ValidationService;

   @Configuration
   public class PasswordValidatorConfig {

           @Bean
           public ValidationService validatorService() {
                   return new DefaultValidationServiceImpl();
           }

   }
   ```
4. Make sure to do a component scan of the created configuration class.

## 3. Using the password Validation Service

```java
@Controller
@RequestMapping("/api/")
public class PasswordValidationController {

        @Autowired
        private ValidationService validationService;

        @PostMapping("/v1/validatePassword")
        @ResponseBody
        public ResponseEntity<String> validatePassword(@RequestBody PasswordDTO password) {

                this.validationService.validatePassword(password.getPassword());
                return new ResponseEntity<String>("Password passed compliance test successfully.", HttpStatus.OK);
        }

        @ExceptionHandler({ PasswordComplainceFailureException.class })
        protected ResponseEntity<String>
handlePasswordLengthViolationException(PasswordComplainceFailureException ex) {
                String error = ex.getMessage();
                return new ResponseEntity<String>(error, HttpStatus.INTERNAL_SERVER_ERROR);
        }

}
```

- The **ValidationService** can be autowired in any spring component.
- Then **validatePassword(String password) throws PasswordComplainceFailureException**

  The **validatePassword(String password)** can be called to validate password.
  If there will be any compliance error then
  **PasswordComplainceFailureException.class** will be thrown.