

SpringBoot Add Validation Model In Register Form In Spring Boot

Screen

Register

admin

First name

REGISTER

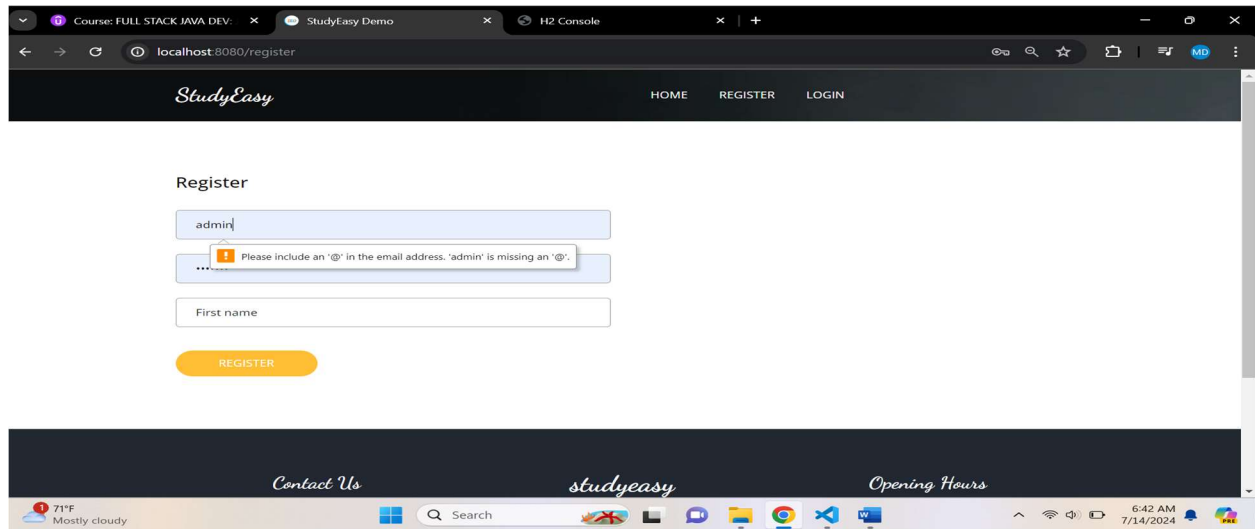
Fixed it

There was an error coming on the button, so fixed it. And When you click on Register button, it will be redirected home page. In Db, it will be added like below.

SELECT * FROM ACCOUNT;

ID	EMAIL	FIRSTNAME	LASTNAME	PASSWORD	ROLE
1	user@user.com	User	lastname	\$2a\$10\$88im.gANspckLkUumfZSR0xso2B2v8ve5VUwebzGvJBGIEdxwQi	ROLE_USER
2	admin@admin.com	Admin	lastname	\$2a\$10\$0cN6Y/m3Hto9f56iBuj.GjQpC3RttqkZoHD.JS6RPJr0QC.	ROLE_ADMIN
3	editor@editor.com	Editor	lastname	\$2a\$10\$0T15XkccDFikRWZ0ivq4CzOw8R6iYK4bgTuTcmP2xhdMpyPp6S	ROLE_EDITOR
4	jag@editor.com	Editor	lastname	\$2a\$10\$VYR2H4EmYqmbz2T166lw nP2ScUF2cy9vD0ALx6BVRw0GjNDq	ROLE_EDITOR
7	jag@gmail.com	null	null	\$2a\$10\$JbEGK3C8ckg7zeimvvt0B0TnrysC24FXtTeTqPmCY2gd0Ny	ROLE_USER

After changing, validation is not a problem like below.



These are known as **Client-Side Validations** i.e. handled by the UI side.

This is a **Monolithic Architecture** where front and back end are combined. But in microservices it is different.

Server-Side Validations

Add a dependency: -

```
<!--  
    https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-  
    starter-validation -->  
<dependency>  
    <groupId>org.springframework.boot</groupId>  
    <artifactId>spring-boot-starter-validation</artifactId>  
    <version>3.3.1</version>  
</dependency>
```

Validations – Email and password

```
@Email(message = "Invalid email")  
@NotEmpty(message = "Email missing")  
private String email;  
  
@NotEmpty(message = "Missing password")  
private String password;
```

In AccountController

```
@PostMapping("/register")
public String register_user(@ModelAttribute Account account){
    accountService.save(account);
    return "redirect:/";
}
```

Update to

```
@PostMapping("/register")
public String register_user(@Valid @ModelAttribute Account account,
BindingResult result){
    if(result.hasErrors()){
        return "account_views/register";
    }
    accountService.save(account);
    return "redirect:/";
}
```

BindingResult – It binds all the error possible in this object

Error Validation Code

```
<div>
    <input id="email" type="text" th:field="*{email}" class="form-control"
placeholder="Email" />
    <p th:if="${#fields.hasErrors('email')}" th:errors="*{email}" class="alert-
danger">Email Error</p>
</div>
```

Output

