Spring MVC:

1. Data bainding: @RequestParam/ ModelAttribute: @InitBinder, BindingResult
2. Bean validation: hibernate-validator, javax-validation (validation-api), @Valid, validator
3. Property file: PropertySourceHolderConfigurer

Spring + JDBC

JdbcTemplate : spring: session/ connection/ tx: autocommit: spring tx

Internationalization: same site be built in more than 1 language

DispatcherServlet: frontController:It looks for ur spring config file and loading ur spring beans into ur spring context

Webapplication:

Root context: created on the startup of ur web application. If we define beans over here, they can be accessed in any child context. The beans are loaded by ContextLoaderListener and beans are specified using context-param.

Child context: DispatcherServlet. If we have more than 1 child context, the beans in 1 child context cannot see the beans in another child context. Bean in child context cannot be accessed in a root context

Class A{}

Class B extends A{} class C extends A{}

**4 stereotype annotations: Spring shud create the object of this class.**

1. @Controller
2. @Service
3. @Repository
4. @Component

@Autowired: inject the dependency

Flyway: flyway-core

1.0: baseline revision

1.1

Checksum

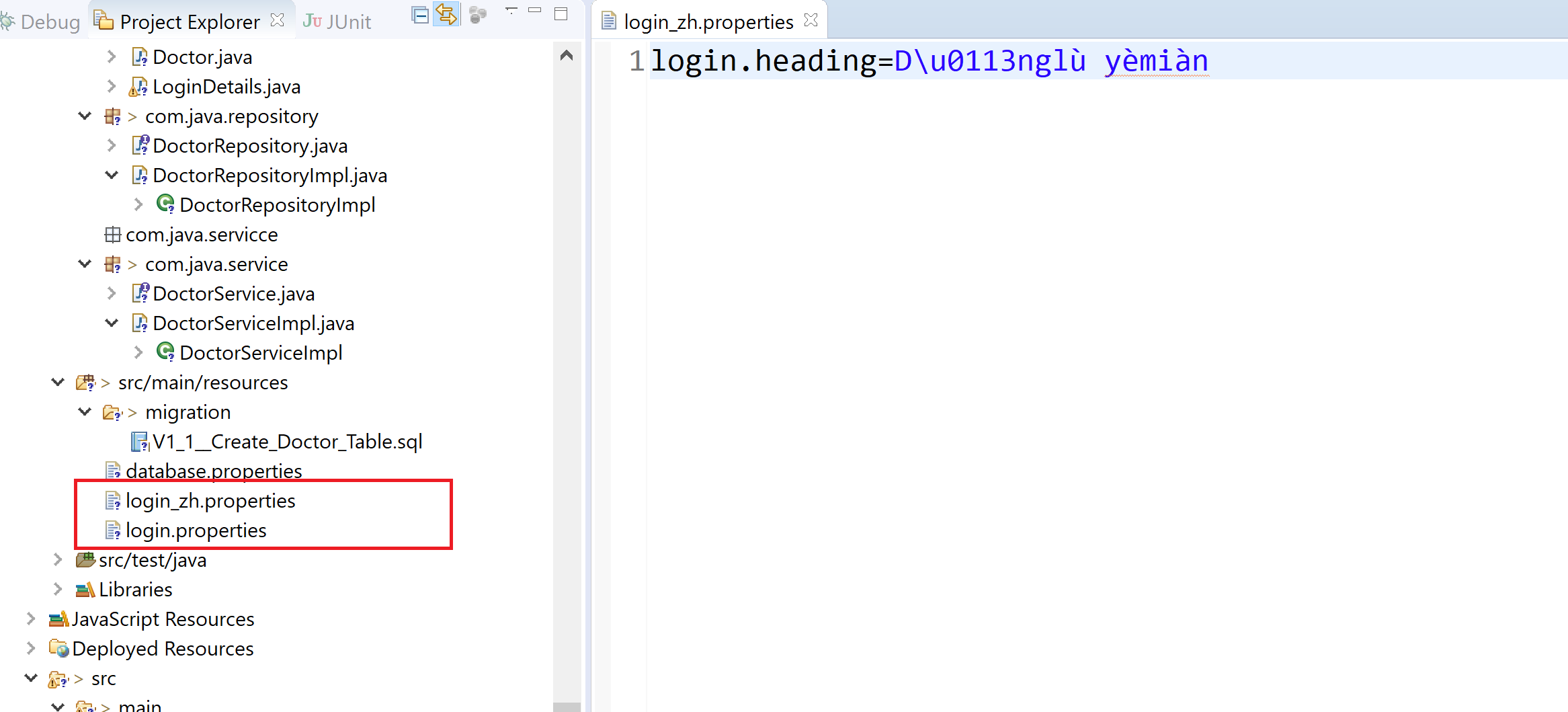
Root context cannot access child context bean

**Internationalization:**

<bean class=*"org.springframework.context.support.ResourceBundleMessageSource"* id=*"messageSource"*>

<property name=*"basename"* value=*"login"*></property>

</bean>

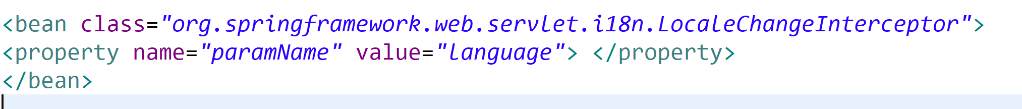


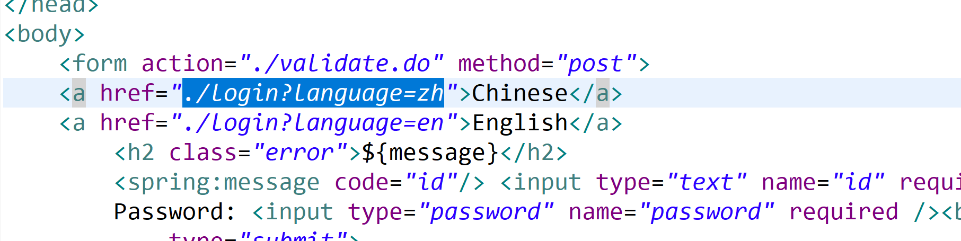
It will search for a locale based on accept-language which is set in the header for a request.

But in an mvc application, we need a link to set this value/locale.

We need to set this locale as a parameter and tell spring to look for this parameter in the request rather than search for accept-language header.

LocaleChangeInterceptor: It will look for locale information in a parameter instead of accept-language header

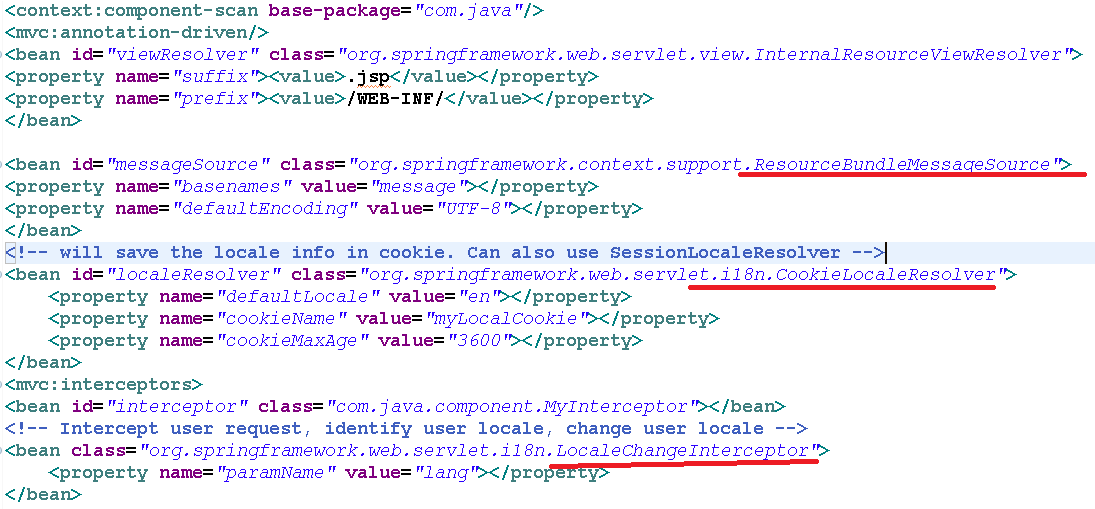




This change should be saved in the session:

SessionLocaleResolver: To add this parameter in the session

CookieLocaleResolver: add it as a cookie: so for subsequent request it would be going as a cookie value from the browser



But we don’t prefer cookies as it may be disabled on some browsers.