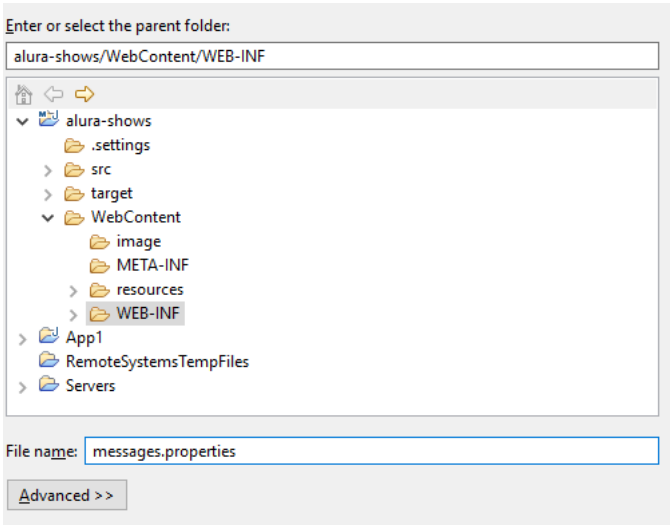
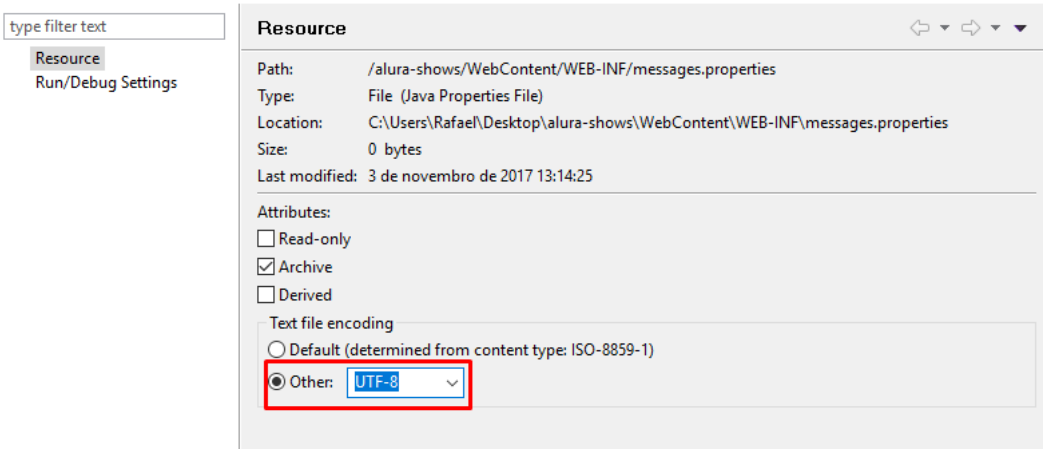
02

Mãos à obra: Disponibilizando a mensagem de erro

Na aula anterior, nós criamos a classe **DepoimentoValidator** com o objetivo de validar se os campos de título e mensagem possuem aberturas ou fechamentos de tag (< ou >), caso isso ocorra, iremos rejeitar o valor e na sequência, iremos mostrar uma mensagem de erro para o usuário. Vamos configurar agora esse arquivo que irá conter essas mensagens o qual chamaremos de **messages.properties** e deverá estar dentro da pasta **WEB-INF** de nossa aplicação. Para isso pressione **CTRL + N** no teclado e crie o arquivo (File) e dê o nome do arquivo como **messages.properties**:



Na sequência, altere esse arquivo para trabalhar no padrão UTF-8, para isso clique com o botão direito do mouse nesse arquivo e clique em **Properties**. Na sequência, na opção **Text File Encoding** coloque **Other** UTF-8:



Feito isso, coloque os mesmos códigos de erro que havíamos configurado na classe DepoimentoValidator com a sua respectiva mensagem personalizada:

```
errors.titulo = O título passado não é válido!
errors.mensagem = A mensagem passada não é válida!
```

Um vez que esse arquivo foi criado, devemos agora avisar o Spring MVC que ele deve usar esse arquivo para buscar as mensagens relativas às chaves que estão configuradas. Vá na classe **CasaDeShowConfiguration** dentro do pacote **infra** e adicione o método:

```
@Bean
    public MessageSource messageSource(){
        ReloadableResourceBundleMessageSource bundle= new ReloadableResourceBundleMessageSource
        bundle.setBasename("/WEB-INF/messages");
        bundle.setDefaultEncoding("UTF-8");
        return bundle;
    }
```

Posteriormente, devemos fazer a validação do Depoimento enviado pelo usuário de acordo com a classe DepoimentoValidator, para isso, vá a classe **DepoimentoController** e coloque o método:

```
@InitBinder
public void InitBinder(WebDataBinder binder){
    binder.setValidator(new DepoimentoValidator());
}
```

Feito isso, vamos no método **enviaMensagem** dentro da classe **DepoimentoController** e coloque a anotação `@Valid` para indicar o processo do disparo de validação. Na sequência, logo depois, adicione o parâmetro `BindingResult result`

```
public String enviaMensagem(@Valid @ModelAttribute(value = "depoimentos") Depoimento depoimento,
    BindingResult result, RedirectAttributes redirect, Model model)
```

Se tivermos erro, vamos retornar para a jsp depoimento para que a mensagem seja visualizada pelo usuário.

```
@RequestMapping(value = "/enviaMensagem", method = RequestMethod.POST)
    public String enviaMensagem(
            @Valid @ModelAttribute(value = "depoimentos") Depoimento depoimento, BindingResult r
            RedirectAttributes redirect, Model model) {
        chamaPostsDoBanco(model);

        if(result.hasErrors()){
            return "depoimento";
        }

        dao.salvaDepoimento(depoimento);
        return "redirect:/depoimento";
    }
```

Para finalizar, voltamos a jsp depoimento e configuramos a tag **form:errors** para que possamos visualizar as mensagens. Coloque essas tags, logo após a label título e mensagem, respectivamente:

```
<form:errors path="titulo" cssClass="error"/>

<form:errors path="mensagem" cssClass="error"/>
```

Agora volte na página de depoimento da aplicação e tente inserir uma tag `<script>` no campo título e mensagem. Qual é o resultado? Você consegue inserir a mensagem?