Internationalisation des exceptions

# Démonstration 7 du module 4

|  |
| --- |
| Les objectifs de cette démonstration :   * Configuration de l’internationalisation (I18N) * Création des fichiers de properties par langues * Mise en place sur les validations * Et la classe AppExceptionHandler |

Contexte

* Nous continuons notre application représentant une partie du cœur de métier de l’ENI Ecole.

Déroulement

# Configuration de l’encodage UTF-8 dans application.properties

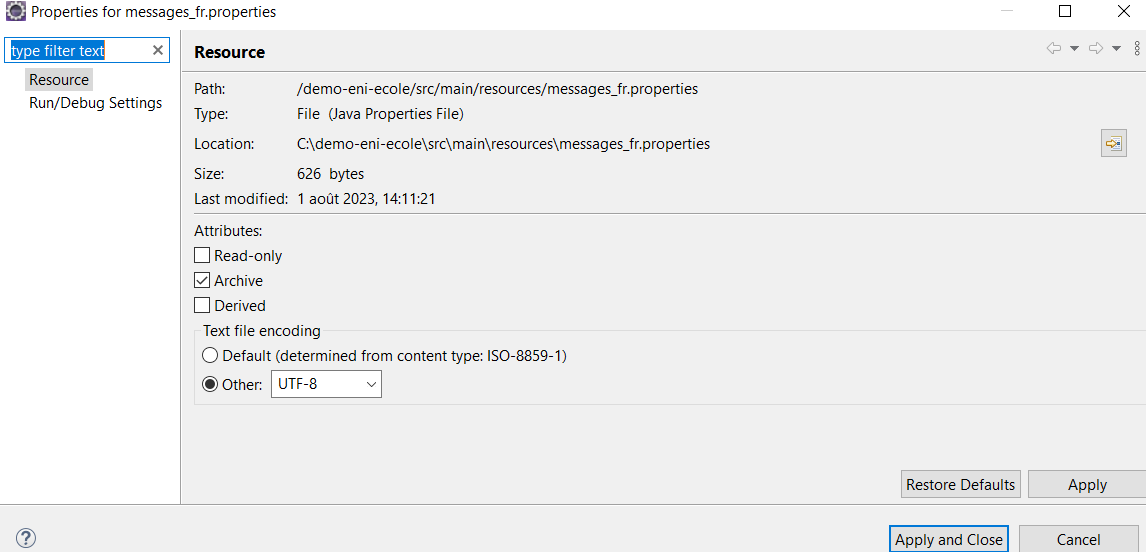
#i18n

spring.messages.encoding=UTF-8

* Si vous voulez un autre nom que messages pour les fichiers de properties
  + Ou un sous répertoire
  + Vous devez spécifier : **spring.messages.basename**=i18n/messages\_errors

# Fichiers messages.properties

* Dans ressources, vous trouverez les 3 fichiers de properties
* Copier les dans src/main/ressources
* Attention, celui en \_fr doit être compilé en UTF-8
  + Faire : clic droit 🡪 properties



# Utiliser les clefs dans Employe

* Pour chaque annotation de validation, il existe une clef dans le fichier de messages.properties
* Répercutez-les

**package** fr.eni.demo.bo;

**import** jakarta.persistence.\*;

**import** jakarta.validation.constraints.\*;

**import** lombok.\*;

**import** lombok.experimental.SuperBuilder;

@NoArgsConstructor

@AllArgsConstructor

@Getter

@Setter

@EqualsAndHashCode(of = { "immatriculation" })

@ToString

@SuperBuilder

@Entity

@Table(name = "EMPLOYEE")

@Inheritance(strategy = InheritanceType.***JOINED***)

**public** **class** Employe {

…

@ManyToOne

@JoinColumn(name = "CIVILITY\_ID")

@NotNull(message = "{employee.civility.error}")

**private** Civilite civilite;

…

@Column(name = "LAST\_NAME", nullable = **false**, length = 90)

@NotBlank(message = "{employee.ln.blank-error}")

@Size(max = 90, message = "{employee.ln.size-error}")

**private** String nom;

@Column(name = "FIRST\_NAME", nullable = **false**, length = 150)

@NotBlank(message = "{employee.fn.blank-error}")

@Size(max = 150, message = "{employee.fn.size-error}")

**private** String prenom;

@Column(nullable = **false**, unique = **true**, length = 255)

@NotBlank(message = "{employee.email.blank-error}")

@Size(max = 255, message = "{employee.email.size-error}")

**private** String email;

@Column(name = "EMPLOYEE\_REGISTRATION", nullable = **false**, unique = **true**, length = 100)

@NotBlank(message = "{employee.registration.blank-error}")

@Size(max = 100, message = "{employee.registration.size-error}")

**private** String immatriculation;

…

@OneToOne(cascade = CascadeType.***ALL***, orphanRemoval = **true**, fetch = FetchType.***EAGER***)

@JoinColumn(name = "ADDRESS\_ID")

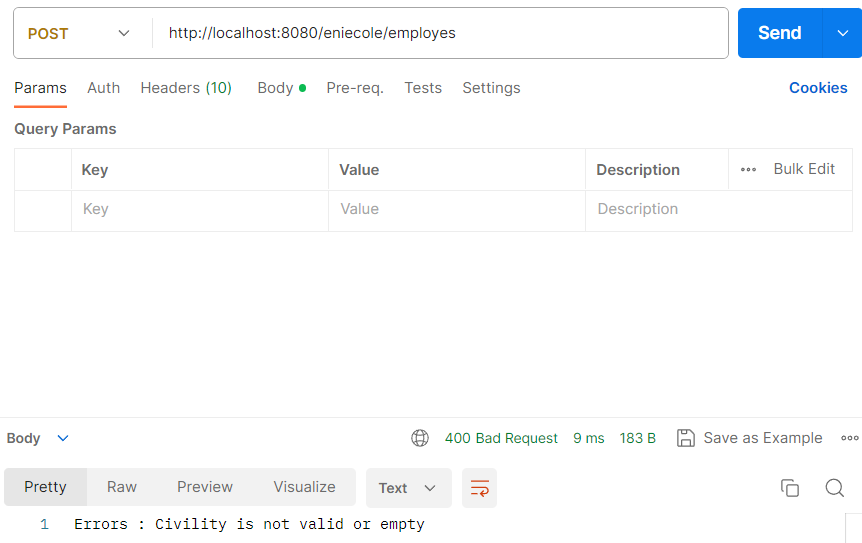
@NotNull(message = "{employee.address.error}")

**private** Adresse adresse;

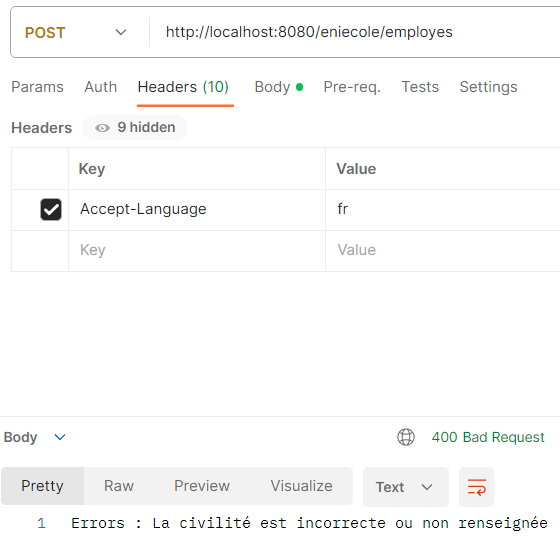
}

## Exécution avec Postman

* Testons sans civilité :



* Le statut de la réponse (400) – Bad Request
* Il affiche le message par défaut en anglais du fichier messages.properties
* Ajouter dans l’entête la langue en fr :



* Le statut de la réponse (400) – Bad Request
* Il affiche bien le message en Français

# Utiliser les clefs dans un contrôleur : AppExceptionHandler

* Injection du bean MessageSource
  + Utilisation de l’annotation Lombok : @AllArgsConstructor
* Injection de Locale dans la méthode de gestion des exceptions : MethodArgumentNotValidException
  + Récupération de la clef : notvalidexception pour avoir la partie « Errors : » selon la langue

**package** fr.eni.demo.exception;

**import** java.util.Locale;

**import** org.springframework.context.MessageSource;

**import** org.springframework.http.\*;

**import** org.springframework.web.bind.MethodArgumentNotValidException;

**import** org.springframework.web.bind.annotation.\*;

**import** lombok.AllArgsConstructor;

@ControllerAdvice

@AllArgsConstructor

**public** **class** AppExceptionHandler {

**private** **final** MessageSource messageSource;

…

// Gestion des exceptions de validation - MethodArgumentNotValidException

@ExceptionHandler(value = { MethodArgumentNotValidException.**class** })

**public** ResponseEntity<String> capturerMethodArgumentNotValidException(MethodArgumentNotValidException ex,

**Locale locale**) {

**final** String titreMsg = messageSource.getMessage("notvalidexception", **null**, locale);

String message = ex

.getFieldErrors()

.stream()

.map(e -> e.getDefaultMessage())

.reduce(**titreMsg,** String::concat);

**return** ResponseEntity.*status*(HttpStatus.***BAD\_REQUEST***).body(message);

}

}

## Exécution avec Postman

* Testons sans civilité :

Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement

* Le statut de la réponse (400) – Bad Request
* Il affiche les 2 messages par défaut en anglais du fichier messages.properties
  + Errors(s) :
  + Civility is not valid or empty
* Ajouter dans l’entête la langue en fr :

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

* Le statut de la réponse (400) – Bad Request
* Il affiche bien les 2 messages en Français