

# Tests unitaires de méthodes de la classe Verif

**verifIdEvenement :**

Test unitaire de : *public CodeErreur verifIdEvenement(int unId);*

Résultat :

- null unId est conforme
- CodeErreur.ID\_NEGATIF si unId est inférieur ou égal à 0

Paramètre unId	Résultat CodeErreur
1	null
3	null
0	ID_NEGATIF
-1	ID_NEGATIF
-3	ID_NEGATIF

## **verifNomEvenement :**

Test unitaire de : `public CodeErreur verifNomEvenement (String unNom);`

Initialisation : `this.TAILLE_MAX_NOM_EVENEMENT = 20`

Résultat :

- null si unNom est conforme
- NOM\_ESPACE\_EN\_TROP si unNom commence ou termine par un espace
- NOM\_TROP\_LONG si unNom dépasse la taille maximum
- NOM\_NULL si unNom est null
- NOM\_VIDE si unNom est une chaîne vide

Paramètre <b>unNom</b>	Résultat <b>CodeErreur</b>
"Nom de l'événement" (18 caractères)	null
" espaces avant" (15 caractères)	NOM_ESPACE_EN_TROP
"espaces après " (15 caractères)	NOM_ESPACE_EN_TROP
"Nom d'événement 19c" (19 caractères)	null
"Nom d'événement 20cc" (20 caractères)	null
"Nom d'événement 21ccc" (21 caractères)	NOM_TROP_LONG
null	NOM_NULL
""	NOM_VIDE

## **verifDescriptifEvenement:**

Test unitaire de : `public CodeErreur verifDescriptifEvenement(String unDescriptif);`

Initialisation : `this.TAILLE_MAX_DESCRIPTORIF_EVENEMENT = 20`

Résultat :

- null si unDescriptif est conforme
- DESCRIPTIF\_ESPACE\_EN\_TROP si unDescriptif commence ou termine par un espace
- DESCRIPTIF\_TROP\_LONG si unDescriptif dépasse la taille maximum
- DESCRIPTIF\_NULL si unDescriptif est null

Paramètre <b>unDescriptif</b>	Résultat <b>CodeErreur</b>
"Descriptif" (10 caractères)	null
" "	null
" espaces avant" (15 caractères)	DESCRIPTIF_ESPACE_EN_TROP
"espaces après " (15 caractères)	DESCRIPTIF_ESPACE_EN_TROP
"Descriptif dix-neuf" (19 caractères)	null
"Descriptif vingt car" (20 caractères)	null
"Descriptif vingt et 1" (21 caractères)	DESCRIPTIF_TROP_LONG
null	DESCRIPTIF_NULL

### **verifDateEvenement :**

Test unitaire de : *public CodeErreur verifDateEvenement(Date uneDate);*

#### Résultat :

- null si uneDate est conforme
- DATE\_NULL si uneDate est null
- DATE\_PASSEE si uneDate est une date déjà passée

Paramètre <b>unDescriptif</b>	Résultat <b>CodeErreur</b>
<i>/*25/11/22 19:30*/</i> uneDate = Date(2022-1900, 11, 25, 19, 30);	null
<i>/*Date déjà passée 25/11/21 19:30*/</i> uneDate = Date(2021-1900, 11, 25, 19, 30);	DATE_PASSEE
null	DATE_NULL

### **verifLieuEvenement :**

Test unitaire de : `public CodeErreur setLieu(String unLieu);`

Initialisation : `this.TAILLE_MAX_LIEU_EVENEMENT = 20`

#### Résultat :

- null si unLieu est conforme
- LIEU\_ESPACE\_EN\_TROP si unLieu commence ou termine par un espace
- LIEU\_TROP\_LONG si unLieu dépasse la taille maximum
- LIEU\_NULL si unLieu est null
- LIEU\_VIDE si unLieu est une chaîne vide

Paramètre unLieu	Résultat CodeErreur
"5 rue Paix, Brest"	null
" espaces avant" (15 caractères)	LIEU_ESPACE_EN_TROP
"espaces après " (15 caractères)	LIEU_ESPACE_EN_TROP
"nom du lieu 19 cara" (19 caractères)	null
"un nom de lieu vingt" (20 caractères)	null
"nom du lieu vingtetun" (21 caractères)	LIEU_TROP_LONG
null	LIEU_NULL
" "	LIEU_VIDE

### **verifNbMaxPersonnesEvenement :**

Test unitaire de : *public CodeErreur verifNbMaxPersonnesEvenement (int unNbMaxPersonnes);*

#### Résultat :

- null si unNbMaxPersonnes est conforme
- NB\_MAX\_PERSONNES\_TROP\_PETIT si unNbMaxPersonnes est inférieur ou égal à 2

Paramètre <b>unNbMaxPersonnes</b>	Résultat <b>CodeErreur</b>
30	null
2	null
1	NB_MAX_PERSONNES_TROP_PETIT
0	NB_MAX_PERSONNES_TROP_PETIT
-2	NB_MAX_PERSONNES_TROP_PETIT

### **verifTypeEvenement :**

Test unitaire de : *public CodeErreur verifTypeEvenement (TypeEvenement unType);*

Résultat :

- null si unType est conforme
- TYPE\_NULL si unType est null

Paramètre <b>unType</b>	Résultat <b>CodeErreur</b>
TypeEvenement.REPAS	null
null	TYPE_NULL

## verifPseudoMembre

Test unitaire de : `public CodeErreur verifPseudoMembre(String unPseudo)`

Initialisation : `this.TAILLE_MAX_PSEUDO_MEMBRE = 10`

Résultat :

- null si unPseudo est conforme
- PSEUDO\_TROP\_LONG si le pseudo est trop long
- CodeErreur.PSEUDO\_NULL si le pseudo est null
- CodeErreur.PSEUDO\_VIDE si le pseudo est vide
- CodeErreur.PSEUDO\_INVALIDE si le pseudo contient : espace, <, >, |, :, ", \*, ?, /, \

Paramètre unPseudo	Résultat CodeErreur
"Test"	null
"123"	null
"test-"	null
"testé"	null
null	PSEUDO_NULL
""	PSEUDO_VIDE
"un espace"	PSEUDO_INVALIDE
"pseudo-9c" (9 caractères)	null
"pseudo_10c" (10 caractères)	null
"pseudo-onze" (11 caractères)	PSEUDO_TROP_LONG
"<"	PSEUDO_INVALIDE
">"	PSEUDO_INVALIDE
" "	PSEUDO_INVALIDE
":"	PSEUDO_INVALIDE
"'" (guillemet)	PSEUDO_INVALIDE
"*"	PSEUDO_INVALIDE
"?"	PSEUDO_INVALIDE
"/"	PSEUDO_INVALIDE
"\"	PSEUDO_INVALIDE
" " (espace)	PSEUDO_INVALIDE



## verifIdentiteMembre

Test unitaire de : `public CodeErreur verifIdentiteMembre(String designation, int numero)`

### Initialisation :

- `this.TAILLE_MAX_NOM_MEMBRE=10`
- `this.TAILLE_MAX_PRENOM_MEMBRE=10`

### Résultat :

- null si la designation est conforme
- si numero = 0
- CodeErreur.NOM\_TROP\_GRAND si le nom est trop grand
  - CodeErreur.NOM\_NULL si le nom est null
  - CodeErreur.NOM\_VIDE si le nom est vide
  - CodeErreur.NOM\_HORS\_ALPHABET si le nom contient des caractères autres que les lettres de l'alphabet
- si numero != 0
- CodeErreur.PRENOM\_TROP\_GRAND si le nom est trop grand
  - CodeErreur.PRENOM\_NULL si le nom est null
  - CodeErreur.PRENOM\_VIDE si le nom est vide
  - CodeErreur.PRENOM\_HORS\_ALPHABET si le nom contient des caractères autres que les lettres de l'alphabet

Paramètres (designation, numero)	Résultat CodeErreur
("Pierre", 0)	null
("Nom", 0)	null
("NomNeufCa",0) (9 caractères)	null
("NomDixCara",0) (10 caractères)	null
("NomOnzeCara",0) (11 caractères)	NOM_TROP_GRAND
(null, 0)	NOM_NULL
("", 0)	NOM_VIDE
("un espace", 0)	NOM_HORS_ALPHABET
("Accént", 0)	NOM_HORS_ALPHABET
("Chiffre1", 0)	NOM_HORS_ALPHABET
("Pre-nom", 0)	NOM_HORS_ALPHABET
("Pre*nom",0)	NOM_HORS_ALPHABET

("Pierre", 1)	null
("PreNom", 1)	null
("PrenomNeu", 1) (9 caractères)	null
("PrenomDixC", 1) (10 caractères)	null
("PrenomOnzeC", 1) (11 caractères)	PRENOM_TROP_GRAND
(null, 1)	PRENOM_NULL
("", 1)	PRENOM_VIDE
("un espace", 1)	PRENOM_HORS_ALPHABET
("Accént", 1)	PRENOM_HORS_ALPHABET
("Chiffre1", 1)	PRENOM_HORS_ALPHABET
("Pre-nom", 1)	PRENOM_HORS_ALPHABET
("Pre*nom", 1)	PRENOM_HORS_ALPHABET

## verifLieuxMembre

Test unitaire de : `public CodeErreur verifLieuxMembre(String lieu, int numero)`

Initialisation : `this.TAILLE_MAX_LIEU_MEMBRE=10`

Résultat :

- null si le lieu est conforme
- si numero = 0
  - CodeErreur.LIEU\_NULL si le lieu de naissance est null
  - CodeErreur.LIEU\_NOM\_INCORRECT si le nom du lieu de naissance comporte un caractère spécial
  - CodeErreur.LIEU\_TAILLE\_INCORRECTE si le lieu de naissance ne comprend pas le nombre de caractères requis
- si numero != 0
  - CodeErreur.VILLE\_NOM\_NULL si la ville est null
  - CodeErreur.VILLE\_NOM\_INCORRECT si le nom de la ville comporte un caractère spécial
  - CodeErreur.VILLE\_TAILLE\_INCORRECTE si la ville ne comprend pas le nombre de caractères requis

Paramètre (lieu, numero)	Résultat CodeErreur
("Lieu", 0)	null
("LieuNeufC", 0) (9 caractères)	null
("LieuDixCar", 0) (10 caractères)	null
("LieuOnzeCar", 0) (11 caractères)	LIEU_TAILLE_INCORRECTE
("Accént", 0)	LIEU_NOM_INCORRECT
(null, 0)	LIEU_NULL
("un espace", 0)	null
("chiffre1", 0)	LIEU_NOM_INCORRECT
("", 0)	LIEU_TAILLE_INCORRECTE
("un,lieu",0)	LIEU_NOM_INCORRECT
("Ville", 1)	null
("VilleNeuf", 1) (9 caractères)	null
("VilleDixCa", 1) (10 caractères)	null

("VilleOnzeCa", 1) ( <i>11 caractères</i> )	VILLE_TAILLE_INCORRECTE
("Accént", 1)	VILLE_NOM_INCORRECT
(null, 1)	VILLE_NULL
("un espace", 1)	null
("chiffre1", 1)	VILLE_NOM_INCORRECT
("", 1)	VILLE_TAILLE_INCORRECTE
("une,ville",1)	VILLE_NOM_INCORRECT

## verifDateMembre

Test unitaire de : `public CodeErreur verifDateMembre(String uneDateNaissance)`

Résultat :

- null si la date de naissance est conforme
- Code\_Erreur.DATE\_NULL si la date est null
- CodeErreur.DATE\_IMPOSSIBLE si la date de naissance est plus grande que la date courante
- CodeErreur.DATE\_AGE\_IMPOSSIBLE si l'âge du membre est supérieur à 130 ans

Paramètre <b>uneDateNaissance</b>	Résultat <b>CodeErreur</b>
<code>/*01/01/2001 01:01*/ uneDate = Date(2001-1900, 1, 1, 1, 1);</code>	null
<code>/*Date plus grande que la date actuelle 01/01/2050 01:01 */  uneDate = Date(2023-1900, 1, 1, 1, 1);</code>	DATE_IMPOSSIBLE
<code>/*Age &gt; 130 ans 01/01/1890 01:01 */  uneDate = Date(1890-1900, 1, 1, 1, 1);</code>	DATE_AGE_IMPOSSIBLE
null	DATE_NULL

## verifMailMembre

Test unitaire de : `public CodeErreur verifMailMembre(String unMail)`

Initialisation : `this.TAILLE_MAX_MAIL_MEMBRE=20`

Résultat :

- null si la définition du mail est un succès
- CodeErreur.MAIL\_NULL le mail est null
- CodeErreur.MAIL\_VIDE si le mail est vide
- CodeErreur.MAIL\_TROP\_LONG si le mail est trop long
- CodeErreur.MAIL\_INVALIDE si le mail ne correspond pas à une adresse mail

Paramètre unMail	Résultat CodeErreur
"tof.bo@example.com"	null
"tof@example.me.org"	null
"tof@example.com"	null
null	MAIL_NULL
""	MAIL_VIDE
"to19@example.me.org" (19 caractères)	null
"tof20@example.me.org" (20 caractères)	null
"tof.21@example.me.org" (21 caractères)	MAIL_TROP_LONG
"tof@.com"	MAIL_INVALIDE
"tof@exemple"	MAIL_INVALIDE
"tof.example.com"	MAIL_INVALIDE
"tof..bob@example.com"	MAIL_INVALIDE
"@example.me.org"	MAIL_INVALIDE
"tof#@exemple.me.org"	MAIL_INVALIDE

## setMotDePasse

Test unitaire de : `public CodeErreur setMotDePasse(String unMotDePasse)`

Initialisation : `this.TAILLE_MAX_MDP_MEMBRE = 10;`

Résultat :

- null si le mot de passe est conforme
- CodeErreur.MDP\_NULL si le mot de passe est null
- CodeErreur.MDP\_VIDE si le mot de passe n'est composé d'aucun caractère
- CodeErreur.MDP\_TAILLE\_INCORRECTE si le mot de passe est trop court ou trop long
- CodeErreur.MDP\_INCORRECT si le mot de passe n'est pas assez sécurisant (manque majuscule, minuscule, chiffre et/ou caractère spécial)

Paramètre unMotDePasse	Résultat CodeErreur
"Mdp1!a"	null
"mdp1!a"	MDP_INCORRECT (pas de majuscule)
"Mdp!aa"	MDP_INCORRECT (pas de chiffre)
"MDP1!A"	MDP_INCORRECT (pas de minuscule)
"Mdp1aa"	MDP_INCORRECT (pas de caractère spécial)
"mdpaaa"	MDP_INCORRECT (que des minuscules)
null	MDP_NULL
""	MDP_TAILLE_INCORRECTE
"Mdp5!" (5 caractères)	MDP_TAILLE_INCORRECTE
"Mdp6c!" (6 caractères)	null
"Mdp9cara!" (9 caractères)	null
"Mdp10cara!" (10 caractères)	null
"Mdp11cara!!" (11 caractères)	MDP_TAILLE_INCORRECTE