

Documentation

Organisation du code du projet Symfony

Dans cette documentation seront expliquées les fichiers du projet Symfony qui nous paraissent être les plus importants.

Src

Controller :

Controller	Rôle	Méthodes
TwitterApiController	Envoyer des requêtes à l'API Twitter et rendre les résultats dans un tableau ; Envoyer 8 requêtes à l'API Twitter données rendre les résultats obtenus sous forme d'un unique tableau	getResult() get8Results()
TwitterAPIRequestController	Créer un tableau de 8 dates qui correspondent aux 8 derniers jours à compter de la date actuelle ; Créer les 8 requêtes correspondantes aux tweets adressés à un compte Twitter donné au cours des 8 derniers jours	get8Dates() get8Request()
BullyingController	Détecter et rendre les tweets considérés comme pouvant témoigner de harcèlement parmi un tableau de tweets donné ; Calculer les statistiques brutes associées à ces tweets	getBullying() insultIn()
InsultController	Récupérer les insultes stockées dans la table insults et les rendre sous forme de tableau de string	getInsults() intoStringArray()
MainController	Mettre en relation tous les acteurs avec les algorithmes de détection du cyber-harcèlement. On peut y retrouver les 3 modes de fonctionnement des algorithmes : normal, démo et validation.	generateNaif() generateDemoNaif() validateNaif() generateIA() generateDemoIA()

StatisticsController	Vider la table <i>statistics</i> ; Calculer l'ensemble des attributs d'un objet de la classe <i>Statistics</i> à partir des statistiques brutes qui lui sont déjà attribuées, puis stocker cet objet dans la table <i>statistics</i> ; Calculer la moyenne des attributs des objets de la <i>Statistics</i> déjà générés, puis stocker cet objet dans la table <i>statistics</i> ;	deleteAll() update()
TweetController	Gérer tout ce qui concerne la manipulation d'objets de la classe <i>Tweet</i> , comme l'affichage d'un tweet donné, la suppression d'un ou plusieurs tweets dans la table où ils sont stockés, l'ajout de tweets dans un tableau de tweets sans doublons et la vérification qu'un tweet appartient ou non à un tableau de tweets donné.	show() delete() deleteAll() addTweets() isRegisteredTweet()
TweetDemoController	Récupérer tous les tweets ou scores présents dans la table <i>tweets_demo</i> ; Transformer un tableau d'instance de la classe <i>TweetDemo</i> en un tableau de messages (string) ou de scores (integer) ; Rendre le tableau des messages ou des scores associés à des indices donnés.	getAllTweetsDemo() getAllScores() tweetsIntoArray() scoresIntoArray() getTweetsDemo() getScores()
ViewController	Mettre en relation la plupart des routes du serveur web avec leur template associée	resultView() resultValidationNaifView() homePageView() lawView() contactsView() helpView() simpleAlgorithmView() IAView()
WebRequestController	Créer une nouvelle instance de <i>WebRequest</i> à partir des informations saisies dans le formulaire et la stocker dans la table <i>web_requests</i> ; Récupérer le dernier identifiant Twitter de la victime présumée, saisi par un utilisateur via l'interface utilisateur, dans la table <i>web_requests</i>	new() getLastRecipientId()

Entity :

Entity	Description	Attributs	Méthodes
TwitterAPIConnection	Entité qui contient les informations permettant de se connecter à l'API Twitter : l'url et les tokens d'accès.	settings url	getters setters constructor
TwitterAPIRequest	Entité qui contient les informations relatives à la requête à l'API Twitter (recipientID est l'identifiant du compte Twitter).	NBTweetMax recipientId date	getters setters constructor
Insult	Insultes avec lesquelles on compare les mots contenus dans un tweet pour savoir si ce dernier est insultant ou non	id terms	getTerms()
Statistics	Statistiques réalisées sur le résultat donné par un algorithme de détection du cyberharcèlement	id recipientId date countTweet countBullying pourcentage	getters setters
Tweet	Tweets qui sont le résultat des algorithmes de détection du cyberharcèlement	id date authorId authorName recipientId message url	getters setters
TweetDemo	Tweets qui sont utilisés dans les modes « démo » et « validationNaif » à la place des tweets fournis par l'API Twitter	id message score	getters setters
TwitterAPIExchange	Wrapper qui permet de simplifier l'exécution des requêtes sur Twitter en réunissant les informations. Il construit à partir des informations données la requête à envoyer à l'API.	oauth_access_token oauth_access_token_secret consumer_key consumer_secret postfield	getters setters constructor performRequest buildBaseString buildAuthorizationHeader

	Cette classe a été trouvée sur GIT.	getField oauth url requestMethod httpStatusCode	request getHttpStatusCode
WebRequest	Requêtes faites par l'utilisateur via l'interface utilisateur	id date recipientId	getters setters

Form :

WebRequest : formulaire que l'utilisateur remplit via l'interface utilisateur et dans lequel il spécifie l'identifiant Twitter de la présumée victime de cyber-harcèlement

Public

Img : ce dossier permet de stocker les images utilisées sur le serveur web

Templates

Dossier	Rôle	Vues
tab	Rassembler les vues (à visée informative ou descriptive) des différents onglets du site	contacts help home simpleAlgorithme IA law
result	Afficher tous les tweets générés et leurs statistiques associées ; Afficher également un tweet en particulier et pouvoir le supprimer de la table où il est stocké ; Afficher les statistiques générées pendant le mode "validationNaif" ;	index show_tweet _delete_tweet_form validation_naif
web_request	Afficher le formulaire à remplir par l'utilisateur pour donner l'identifiant Twitter de la présumée victime	_form new

Fichier base.html.twig : contient les réglages concernant la barre de navigation et la banderole du site web

Autres fichiers

.env : c'est dans ce fichier que se trouve entre autres l'URL (locale) de la base de données MySql