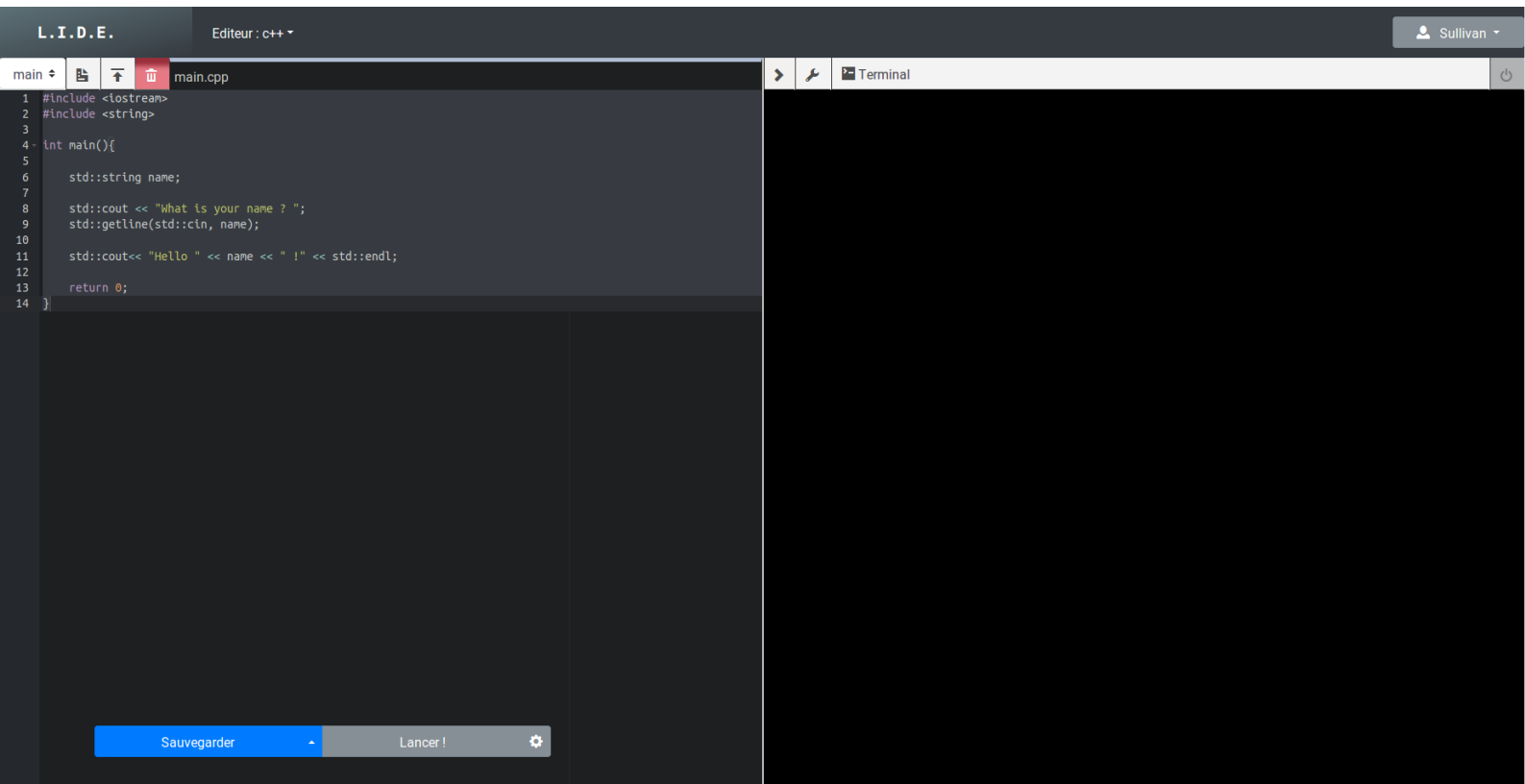


Manuel d'utilisation
Environnement de développement en ligne
L.I.D.E v1.0.0

Contexte

Ce manuel a pour but d’aider les utilisateurs lors de leurs interactions avec l’application en ligne L.I.D.E. qui permet de compiler et d’exécuter le code des étudiants afin de faciliter l’apprentissage des matières informatiques.

Voici la page principale de l’application.



UTILISATEUR

1) Les différents usages

1.1 L'inscription/Connexion

Connexion

Connexion

Vous n'êtes pas inscrit ? [Inscription](#)

Sur la page de connexion, en renseignant son nom d'utilisateur et son mot de passe, il est possible de se connecter. Si vous n'avez pas de compte, il faut alors cliquer sur le lien [Inscription](#).

Inscription

Nom d'utilisateur

Adresse e-mail

Mot de passe

Vérification

Enregistrer

Sur la page d'inscription il faut renseigner son nom d'utilisateur, son adresse e-mail (les adresses autorisées sont paramétrables à l'installation de l'application. Par exemple pour l'université d'Angers, seules les adresses suffixées par @etud.univ-angers.fr ou @univ-angers.fr sont acceptées) et son mot de passe avec une deuxième saisie de vérification.

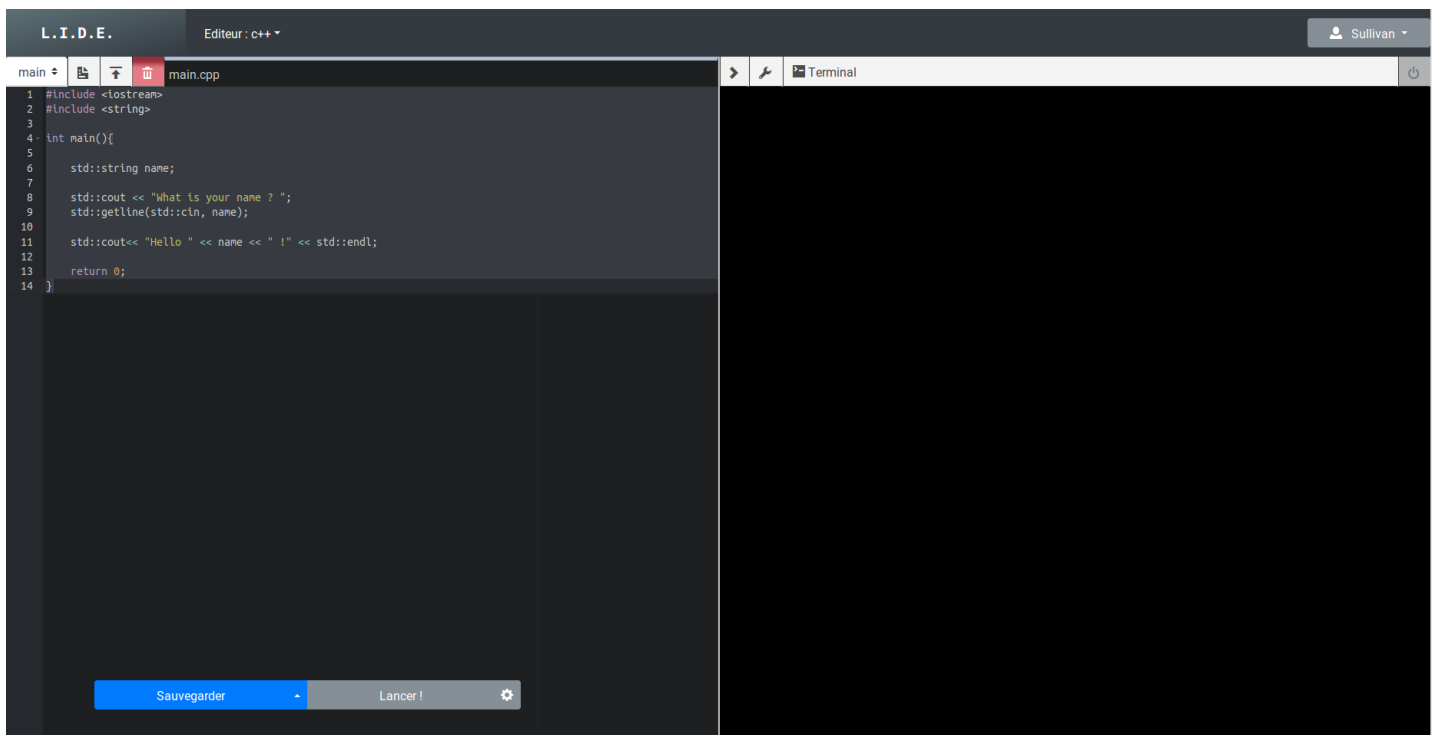
Inscription

Un e-mail a été envoyé à l'adresse PO@etud.univ-angers.fr. Il contient un lien d'activation sur lequel il vous faudra cliquer afin d'activer votre compte.

Un mail automatiquement généré sera envoyé à l'adresse mail renseignée afin de confirmer l'inscription.

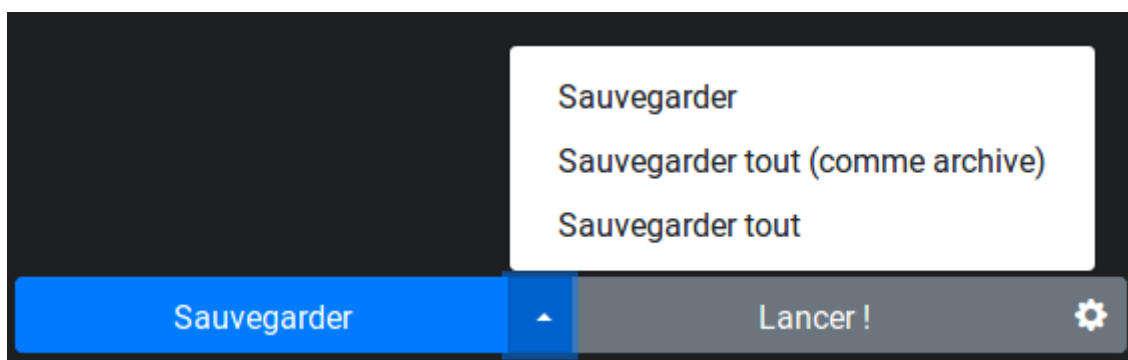
Si jamais le mail de validation n'arrive pas dans la boîte mail, il est aussi possible de demander à un professeur de valider votre compte.

1.2) Page principale, fonctions de base



Une fois connecté, nous sommes redirigé vers la page principale du site. Nous pouvons observer que l'écran est divisé en deux blocs : le bloc de gauche correspond à la zone d'édition où il sera possible d'écrire les lignes du programme et la partie de droite correspond au terminal, c'est ici que le code sera compilé et exécuté après avoir appuyé sur le bouton *Lancer*.

En bas à gauche se trouvent les boutons *Sauvegarder* et *Lancer*. Le bouton *Sauvegarder* permet de sauvegarder le fichier courant affiché à l'écran, de sauvegarder tous les fichiers ouverts ou tout le projet en tant qu'archive suivant l'option choisie à sa droite.



Le bouton *Lancer* quant à lui permet de compiler et d'exécuter le code.

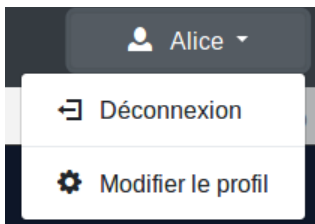
2) Options avancées

2.1 Barre menu



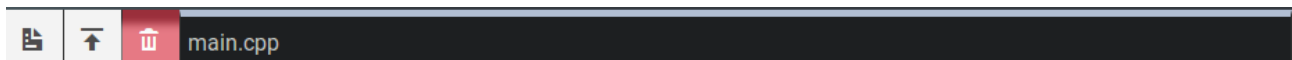
Editeur : c++ ▼

Quelques options plus poussées sont intégrées à l'application telle que l'option de sélection du langage. En effet, en cliquant sur le bouton *Éditeur* il est possible de choisir le langage en fonction des langages disponibles validés par le corps enseignant.



La liste déroulante en haut à droite de l'écran permet de se déconnecter de l'application ou bien de modifier le profil de l'utilisateur.

2.2 Options éditeur



Le bouton permettant de créer un fichier ouvrira une pop-up pour configurer celui-ci.

Nouveau fichier [X]

Entrer le nom du fichier .cpp

Extension du fichier .cpp Voir le modèle associé

☒ Utiliser ce modèle

Créer

Renseigner le nom du fichier, l'extension de celui-ci et s'il doit être généré avec un modèle pré-enregistré de code. Il est possible de consulter ce code en appuyant sur le bouton « *voir le modèle associé* ».



Ce bouton permet d'importer des fichiers. Il ouvrira une pop-up pour les sélectionner.

Importer un fichier [X]

Sélectionner le.s fichier.s à importer

Parcourir... Aucun fichier sélectionné.

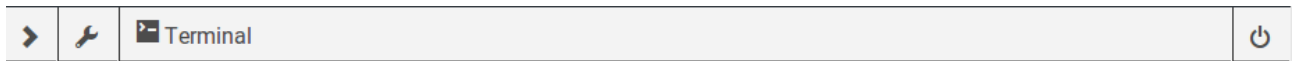
Importer

L'importation des fichiers présents sur votre ordinateur vers l'application est permise. En appuyant sur le bouton « *parcourir* », il est possible d'aller chercher ses propres fichiers.



Un clic sur le bouton « *supprimer* » permet la suppression du fichier courant affiché à l'écran.

2.3 Options terminal



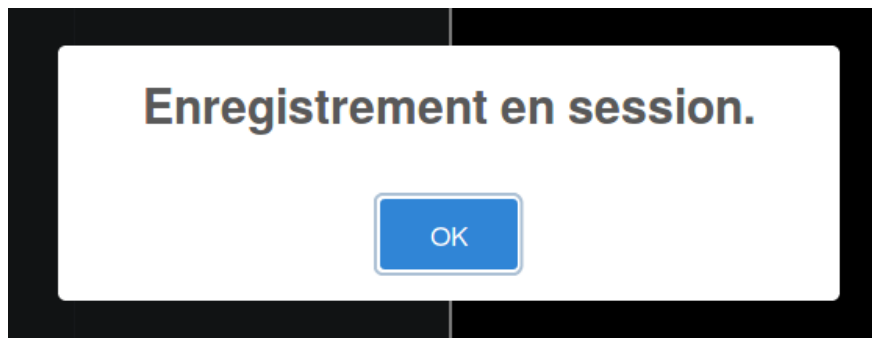
Si pendant le développement, la console prend trop de place et dérange l'utilisateur, il est possible de la réduire en appuyant sur ce bouton.



La possibilité d'interrompre l'exécution est permise en appuyant sur le bouton.

2.4 Options générales

La combinaison de touches Ctrl+S permet l'enregistrement en session du code, ce qui permet en cas de page qui se ferme ou de reconnexion sur la page de récupérer le code précédemment écrit.



De plus, des options d'affichage sont aussi disponibles si les couleurs du terminal ou de l'éditeur ne sont pas aux goûts de l'utilisateur.



Ce bouton là permet de paramétrer l'affichage de l'éditeur.

Options

Thème de l'éditeur

tomorrow_night

Thème de la console

sombre

12

Cancel

Ok

Il est donc possible via ces options de changer la couleur de l'éditeur ainsi que celle de la console. De plus si la taille de police ne convient pas à l'utilisateur celle-ci est paramétrable.

3) Les options de compilation



En appuyant sur le rouage qui se trouve à côté de *Lancer*, il est possible d'accéder aux options de compilation.

La première option proposée est « *Compilation uniquement* ». Si cette option est cochée le code sera seulement compilé et aucune exécution ne sera effectuée.

« *Gestion des entrées* », cette option propose trois interactions différentes :

- *Aucune* : les interactions avec l'utilisateur sont désactivées, le code s'exécutera et ne prendra pas en compte les entrées que l'utilisateur est censé fournir.
- *Interactive* : ici les interactions avec l'utilisateur sont autorisées. S'il lui est demandé de renseigner une valeur dans le terminal, le programme attendra que l'utilisateur le fasse avant de continuer son exécution.
- *Texte* : ici un champ apparaîtra juste en dessous et l'utilisateur pourra automatiser les interactions en les renseignant dans le champ.

Les options de compilation permettent de renseigner des options qui pourraient être nécessaires lors de la compilation du programme tel que `-O2` en `c++`.

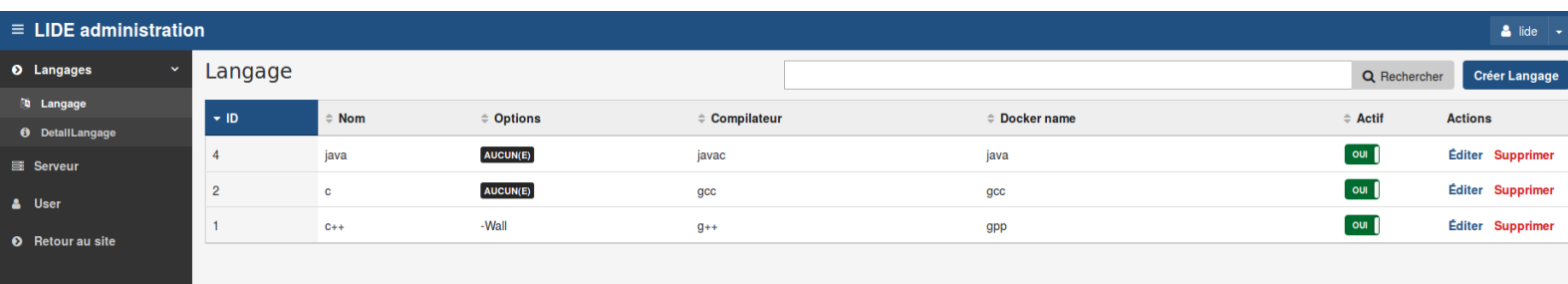
L'option *Paramètres de lancement* correspond aux arguments du programme.

Les *Fichiers additionnels* quant à eux permettent à l'utilisateur de renseigner un fichier avec lequel le programme doit interagir.

Appuyer sur le bouton « *Lancer* » permet de valider les options et lancer la compilation.

ADMINISTRATION

Une partie du site est réservée à l'administration, les administrateurs ont la possibilité via des pages spécifiques de gérer la base de données.



The screenshot shows the 'LIDE administration' interface. On the left is a sidebar with navigation links: 'Langages', 'Langage', 'DetailLangage', 'Serveur', 'User', and 'Retour au site'. The main content area is titled 'Langage' and contains a table with the following data:

ID	Nom	Options	Compilateur	Docker name	Actif	Actions
4	java	AUCUN(E)	javac	java	OUI	Éditer Supprimer
2	c	AUCUN(E)	gcc	gcc	OUI	Éditer Supprimer
1	c++	-Wall	g++	gpp	OUI	Éditer Supprimer

At the top right of the table is a search bar labeled 'Rechercher' and a button labeled 'Créer Langage'.

1) Les options des différentes tables

Il y a actuellement quatre tables avec lesquelles l'administrateur peut interagir avec différentes options disponibles :

- Langage : l'ajout, la modification, la suppression et la désactivation des langages
- DetailLangage : l'ajout, la modification, la suppression et la désactivation des extensions
- Serveur : l'ajout, la modification, la suppression et la désactivation des serveurs.
- Utilisateur : la modification, suppression et la désactivation d'un utilisateur est autorisé. L'ajout n'est pas autorisé, il faut passer par l'inscription.

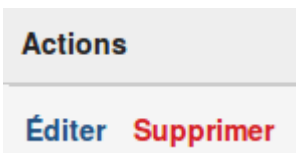


Permet l'activation ou la désactivation d'un élément des tables de la base :

- Oui : l'élément est activé.
- Non : l'élément est désactivé.



Ce bouton-ci permet la création d'un langage dans la table correspondante. Un bouton semblable est présent pour chacune des autres tables.



Dans « actions », la possibilité de modifier ou de supprimer un élément de la table est permise.

2) Ajout d'un nouveau langage

- Création du Dockerfile :
 - Afin d'ajouter un nouveau langage à l'application il faut dans un premier temps créer un dockerfile sur le serveur d'exécution.
 - Il est important de nommer l'image d'un nom distinct des autres. Lors de sa construction via la ligne de commande « docker build --tag nom . » dans le dossier contenant le dockerfile.
- Création du langage :
 - Dans la table Langage de l'application,
 - Il faut nommer le langage
 - Préciser le compilateur de celui-ci,
 - Renseigner le dockerfile correspondant au langage pour pouvoir s'en servir en cas de besoin sur d'autres serveurs
 - Ajouter le nom utilisé lors de la création de l'image docker sur le serveur pour pouvoir y accéder

- Ajouter le script permettant l'exécution et la compilation du langage en automatique
 - Pour finir, activé ce langage si l'image a bien été construite.
- c) Ajout des extensions :
- Dans la table DetailLangage :
 - Il faut renseigner l'extension (.cpp, .java ...)
 - Indiquer le modèle par défaut
 - Choisir l'ordre de l'extension par rapport aux autres extensions du langage
 - Relier l'extension avec le langage précédemment créé.
- .