



Documentation utilisateur - Ordogène

Last Project - Darwinners

Client :

Nicolas BORIE

Equipe :

Hugo FEUILLÂTRE

Thibault GIRARD

Yenrick ISKANDAROFF

Gauthier RENARD

Nacer DERGAL



Diffusion

Version	Date	Motif
1.0	02/03/2018	Version Initiale
1.1	06/03/2018	Modification du fichier de configuration Ajustement des commandes de la CLI

Historique des changements

Rôle	Action	Date
L'équipe	Ecriture du document	06/03/2018
L'équipe	Relecture	06/03/2018
Hugo FEUILLÂTRE	Approbation du responsable qualité	06/03/2018
Yenrick ISKANDAROFF	Approbation du chef de projet	06/03/2018

Préambule

Ce document est le manuel utilisateur du projet d'ordonnancement génétique. Ce projet entre dans le cadre de la troisième année de la formation d'ingénieur au sein de l'ESPE. L'équipe Darwinners est constituée de cinq étudiants faisant partie de la filière Informatique et Réseaux.

L'IGM (Institut Gaspard-Monge) est un établissement public de recherche et d'études dans les domaines de l'électronique et de l'informatique. Ils ont besoin d'une démonstration de faisabilité d'un algorithme d'ordonnancement génétique. Il a pour but de minimiser en termes de temps un processus constitué de différents éléments possédant des relations d'utilisation ou de consommation. Ces critères peuvent être très différents et variés selon le contexte.

Dans ce document, nous allons décrire l'installation et l'utilisation du logiciel Ordogène.

Table des matières

Diffusion	1
Historique des changements	1
Préambule	2
Table des matières	3
Glossaire	4
Description d'Ordogène	5
À propos	5
Architecture	5
Configuration requise	6
Représentation d'un processus	7
Concept	7
Sous forme de schéma	8
JSON	9
Utilisation	12
API REST	12
Démarrage	12
Ressources disponibles	13
Vérifier l'identifiant d'un utilisateur	13
Obtenir la liste des calculs lancés par l'utilisateur	13
Obtenir le résultat d'un calcul	14
Obtenir le résultat d'un calcul html	14
Créer un identifiant utilisateur unique	15
Créer un identifiant utilisateur donné	15
Lancer un calcul	16
Arrêter un calcul en cours	17
Supprimer un calcul	17
CLI (Command-Line Interface)	18
Démarrage	18
Commandes	18
Intégrées de base	18
Commandes	18

Glossaire

- **Algorithme génétique** : Modification d'une population basée sur la survie du plus fort, la mutation de génomes et le croisement d'individus.
- **API** : (Application Programming Interface) Une Interface de Programmation Applicative est un ensemble normalisé de classes, méthodes ou de fonctions qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels.
- **CLI** : (Command Line Interface) Interface en Ligne de Commande est une interface homme-machine dans laquelle la communication entre l'utilisateur et l'ordinateur s'effectue en mode texte.
- **Croisement** : Fusion de plusieurs génotypes provenant de différents individus pour créer un ou plusieurs nouveaux individus.
- **Entité** : Équivalent à une ressource dans une entreprise (ex : Machine, employé, stock...).
- **Fitness** : Score d'un individu par rapport à l'équation donnée par l'utilisateur dans le modèle.
- **Génome** : Terme biologique désignant le patrimoine génétique d'une espèce. Dans un algorithme génétique, le génome peut subir des mutations à chaque itération.
- **Mutation** : Modification aléatoire du génome d'un individu particulier.
- **Ordonnancement** : Succession de tâches, avec ou sans contrainte, définies dans le temps.
- **Relation ("flèche")** : Défini les relations entre les entités d'un système (ex : requis, utilise...).
- **Système** : Ensemble de tâches, d'entités et de relations
- **Tâche** : Fonction définie dans le temps, peut consommer et/ou utiliser une entité dans le but de créer une ou plusieurs autres entités.

I. Description d'Ordogène

A. À propos

Ordogène est un logiciel permettant de déterminer, à l'aide d'un algorithme génétique, les actions d'un processus donné par un utilisateur afin de maximiser ou minimiser certaines ressources en un minimum de temps. Ces processus peuvent être très variés, comme le fait de se préparer le matin, l'organisation d'un événement, ou la fabrication de pièces industrielles. Le logiciel fournit après calcul un ordonnancement prenant en compte les paramètres de l'utilisateur et les contraintes du processus.

Pour bien l'utiliser, cela suppose de bien comprendre les termes, les concepts et les méthodes employés dans ce document. Il est fortement conseillé de l'étudier en détail.

Ce manuel utilisateur offre une vue d'ensemble des fonctionnalités de l'application et donne des instructions pour pouvoir les exploiter.

B. Architecture

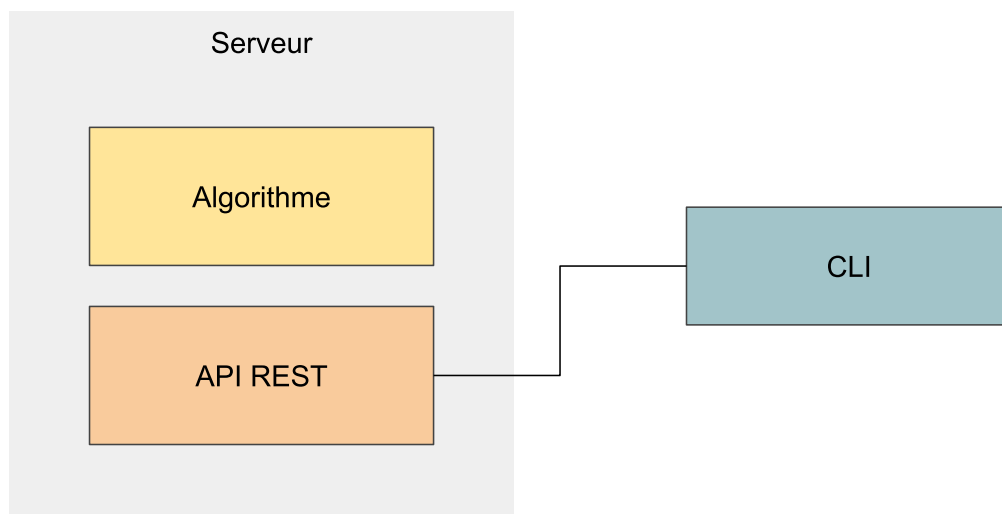


Figure 1 : Architecture d'Ordogène

Ordogène est composé de deux parties :

- La CLI : ce client permet de communiquer avec le serveur à travers son API avec une interface en ligne de commande.
- Le serveur : son rôle est de gérer les utilisateurs et leurs calculs. Il interroge le bloc algorithme en fonction des requêtes reçues par l'API.

C. Configuration requise

La CLI et le serveur Ordogène nécessitent tous les deux d'avoir au minimum Java 8 installé sur le(s) poste(s) désiré(s). Il est possible d'utiliser le serveur sur la même machine que la CLI.

II. Représentation d'un processus

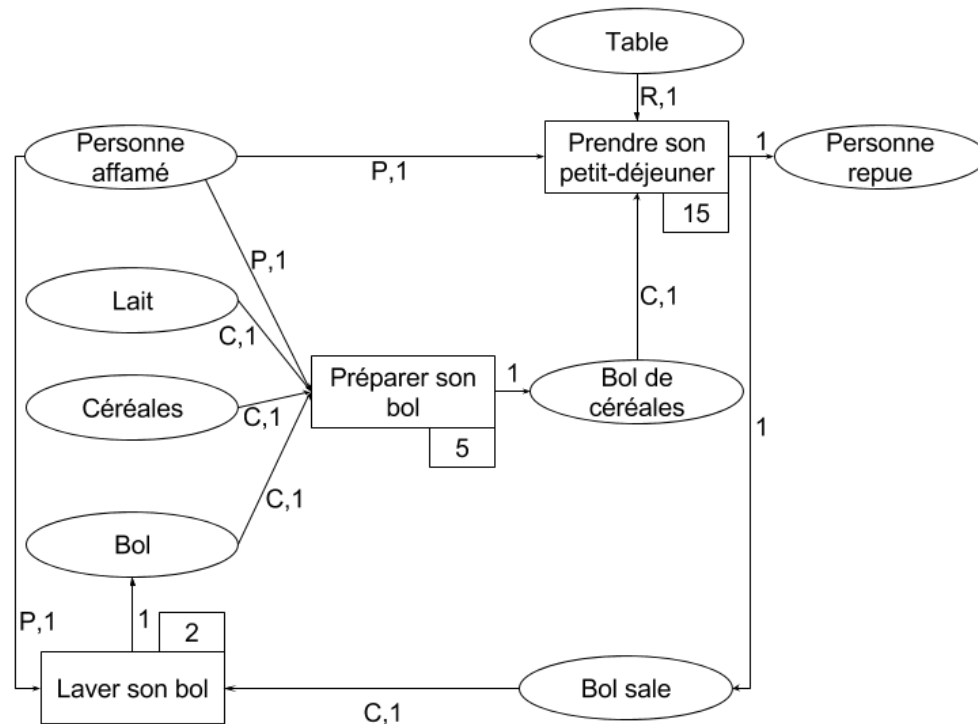
A. Concept

Pour modéliser un processus à optimiser, le logiciel va utiliser quatre notions :

1. Tous les objets du système (processus informatiques, machines, matériaux, employés, ...) seront représentés par ce que l'on appelle une entité.
2. Les actions sont ce qui provoque un changement dans le système. Elles peuvent accepter plusieurs entités que ce soit en entrée ou en sortie. Les seules contraintes qu'elles devront respecter sont les suivantes :
 - Au minimum une entité entrante
 - Au minimum une entité sortante
 - Un temps de réalisation
 - Pour être réalisé, tous les liens entrants sont nécessaires
3. Quatre liens sont autorisés entre une action et ses entités :
 - *consumed* : décrit la consommation de l'entité lors de la réalisation d'une étape. La quantité renseignée sur ce lien est celle nécessaire afin de réaliser l'action.
 - *pending* : décrit l'occupation l'entité lors du changement d'état du système. Durant toute la durée de l'exécution de l'action l'entité n'est plus disponible pour faire une autre action. Le chiffre inscrit sur le lien est la quantité de l'entité qui sera occupée.
 - *required* : décrit la nécessité de posséder l'entité pour réaliser l'action. Cela peut décrire une technologie, un savoir, un prérequis pour un programme, etc. L'entité associée peut être requis pour plusieurs actions faites en parallèle.
 - *produced* : ce lien est utilisé seulement à la sortie des actions. Il représente les entités qui sont produites après la réalisation de l'action. La quantité produite est renseignée directement sur le lien.
4. Le temps est une unité de mesure importante. Lors de l'optimisation d'un processus, il sera l'indicateur déterminant pour différencier deux choix produisant une fitness équivalente. On la représente sous forme de "slot" avec un entier plutôt qu'une unité imposée.

B. Sous forme de schéma

Voici un exemple présentant toutes les notions. Ce schéma représente le processus d'un petit déjeuner.



Dans cet exemple nous pouvons voir l'ensemble des éléments utilisables. Il y a huit entités représentées par des ellipses. Elles possèdent une quantité non-indiquée sur le schéma.

De ces entités sortent des flèches désignant l'utilisation (P pour *pending*), la consommation (C pour *consumed*) ou la nécessité (R pour *required*) afin de réaliser l'action à laquelle elles sont liées.

Pour finir, les actions sont sous forme de rectangle. Le chiffre situé dans un rectangle adjacent à une action est la durée de celle-ci. Pour pouvoir être réalisée, tous les liens entrants sont obligatoires afin de produire les entités sortantes. Le chiffre situé sur les liens entrants est la quantité pour produire et celui sur les liens sortants la quantité produite.

Dans un premier temps, il est recommandé de représenter le processus à modéliser sous forme de schéma car il est plus humainement lisible.

C. JSON

Pour les rendre compréhensible pour le logiciel, les processus devront être décrit dans un document JSON (JavaScript Object Notation) que l'on appelle modèle. Ce dernier doit obligatoirement contenir les attributs suivants :

- **“slots”** : Un entier strictement positif qui détermine la durée maximale d'un individu pour exécuter toutes ses tâches ;
- **“name”** : Une chaîne de caractère correspondant à l'identifiant du modèle ;
- **“exec_time”** : Un entier strictement positif qui fixe pour ce modèle le nombre d'itération maximum que l'algorithme génétique doit effectuer ;
- **“environment”** : Une liste d'objet, contenant les entités de départ.

Pour chaque entité :

- **“name”** : Une chaîne de caractère identifiant l'entité ;
- **“quantity”** : Un entier positif ou nul définissant la quantité de cette entité.
- **“actions”** : Une liste d'objet contenant les actions du processus :
 - **“name”** : Une chaîne de caractère identifiant l'action ;
 - **“time”** : Un entier strictement positif correspondant au nombre de slots que prend l'action pour se terminer ;
 - **“input”** : Une liste d'objet contenant les entités en entrée de l'action :
 - **“name”** : Une chaîne de caractère correspondant au nom de l'entité ;
 - **“quantity”** : Un entier strictement positif correspondant à la quantité à utiliser ;
 - **“relation”** : Un caractère parmi les suivants : « p » pour pending, « r » pour require et « c » pour consume.
 - **“output”** : Une liste d'objet contenant les entités en sortie de l'action :
 - **“name”** : Une chaîne de caractère qui correspond au nom de l'entité ;
 - **“quantity”** : Un entier qui correspond à la quantité produite.
- **“fitness”** : Une liste d'objet contenant les entités à prendre en compte pour l'étape :
 - **“type”** : Une chaîne de caractère définissant la tendance de la fonction de fitness pour ce modèle parmi « min », « max » et « value » ;

- **“value”** : si le type est “value” ce champ doit apparaître avec comme valeur un entier ;
- **“operands”** : Une liste d’objet contenant l’ensemble des entités à prendre en compte pour la fitness :
 - **“name”** : Une chaîne de caractère correspondant au nom de l’entité concerné ;
 - **“coef”** : Un entier, détermine le coefficient concernant cette entité pour le calcul de la fitness d’un individu.

Attention ! Toutes les entités utilisées dans le modèle doivent être présente dans l’environnement de départ (avec des quantités nulles si nécessaire).

Exemple de modèle au format JSON :

```
{
  "name" : "petit-dej.json",
  "slots" : 60,
  "exec_time" : 1000,
  "environment" : [
    {"name" : "Personne affamée", "quantity" : 1},
    {"name" : "Céréales", "quantity" : 1},
    {"name" : "Bol", "quantity" : 0},
    {"name" : "Bol sale", "quantity" : 1},
    {"name" : "Personne repue", "quantity" : 0},
    {"name" : "Bol de céréales", "quantity" : 0},
    {"name" : "Lait", "quantity" : 1}
  ],
  "actions" : [
    {
      "name" : "Préparer son bol", "time" : 5,
      "input" : [
        { "name" : "Personne affamée", "quantity" : 1, "relation" : "p" },
        { "name" : "Lait", "quantity" : 1, "relation" : "c" },
        { "name" : "Céréales", "quantity" : 1, "relation" : "c" },
        { "name" : "Bol", "quantity" : 1, "relation" : "c" }
      ],
      "output" : [
        { "name" : "Bol de céréales", "quantity" : 1 }
      ]
    },
    {
      "name" : "Prendre son petit dej", "time" : 15,
      "input" : [
        { "name" : "Personne affamée", "quantity" : 1, "relation" : "c" },
        { "name" : "Bol de céréales", "quantity" : 1, "relation" : "c" }
      ],
      "output" : [
        { "name" : "Personne repue", "quantity" : 1 },
        { "name" : "Bol sale", "quantity" : 1 }
      ]
    }
  ]
}
```

```
    ]
  },
  {"name" : "Laver son bol", "time" : 2,
   "input" : [
     { "name" : "Personne affamée", "quantity" : 1, "relation" : "p" },
     { "name" : "Bol sale", "quantity" : 1, "relation" : "c" }
   ],
   "output" : [
     { "name" : "Bol", "quantity" : 1 }
   ]
  }
],
"fitness" : {
  "type" : "max",
  "operands" : [
    { "name" : "Personne repue", "coef" : 100 }
  ]
}
```

III. Utilisation

A. API REST

1. Démarrage

L'API REST s'exécute dans un terminal. Pour la démarrer, il faut se placer dans le dossier de l'archive JAR «ordogene-api-<version>.jar» puis exécuter la commande :

```
java -jar ordogene-api-<version>.jar --config=<path> -port=<port>
```

Pour java 9 :

```
java --add-modules java.xml.bind -jar ordogene-api-<version>.jar  
--config=<path> -port=<port>
```

L'argument **--config <path>** correspond au fichier de configuration de l'API rest.

L'argument **--port <port>** correspond au port d'écoute de l'API rest. Il est optionnel et par défaut la valeur du port d'écoute est 8888.

Le fichier de configuration est au format JSON, il contient l'information suivante :

```
{  
  "ApplicationPath" : "<chemin d'enregistrement des fichiers résultats>",  
  "ResultSaveInterval": <intervalle de temps entier en seconde pour les résultats  
intermédiaires> ,  
  "MaxComputationThreads": <nombre entier de thread maximum disponible pour lancer les  
calculs> ,  
  "IndividualNumberByGeneration": <nombre entier d'individu par génération> ,  
  "ProbabilityToStopIndividualCreation": <nombre à virgule représentant la probabilité d'arrêter  
la création d'individus>  
}
```

Après le lancement de l'API, elle est disponible à l'adresse suivante :

```
http://<adresseServeur>:<port>/ordogene
```

2. Ressources disponibles

Vérifier l'identifiant d'un utilisateur

Méthode HTTP : GET

Format de l'url : http://<adresseServeur>:<port>/ordogene/<userID>

Description : Cette requête vérifie que l'utilisateur <userID> existe.

Réponse :

Situation	Retour de l'API
Tout se passe correctement	code 200
L'utilisateur est vide	code 400
L'utilisateur n'existe pas	code 404

Obtenir la liste des calculs lancés par l'utilisateur

Méthode HTTP : GET

Format de l'url : http://<adresseServeur>:<port>/ordogene/<userID>/calculations

Description : Cette requête permet d'obtenir tous les calculs qui ont été lancés par l'utilisateur <userID>.

Réponse : La réponse est un tableau au format JSON avec dans chaque objet : le cid, le nom du calcul, date, la fitness, le statut du calcul, le nombre d'itérations effectuées, le numéro d'itération de la dernière sauvegarde, le nombre d'itérations maximales et la date.

Situation	Retour de l'API
Tout se passe correctement	code 200 avec un body contenant toutes les informations
L'utilisateur est vide	code 400
L'utilisateur n'existe pas	code 404

Obtenir le résultat d'un calcul

Méthode HTTP : GET

Format de l'url : http://<adresseServeur>:<port>/ordogene/<userID>/calculations/<cid>

Description : Cette requête sert à obtenir le résultat d'un calcul.

Réponse : La réponse est une image encodé en base64 correspondant au meilleur ordonnancement calculé.

Situation	Retour de l'API
Tout se passe correctement	code 200 avec un body contenant l'image représentant le meilleur individu
L'utilisateur est vide	code 400
L'utilisateur n'existe pas	code 404
Le calcul n'existe pas	code 404
Le résultat du calcul n'existe pas	code 404

Obtenir le résultat d'un calcul html

Méthode HTTP : GET

Format de l'url : http://<adresseServeur>:<port>/ordogene/<userID>/calculations/<cid>/html

Description : Cette requête sert à obtenir le résultat d'un calcul.

Réponse : La réponse est un fichier encodé en base64 correspondant au meilleur ordonnancement calculé au format HTML.

Situation	Retour de l'API
Tout se passe correctement	code 200 avec un body contenant l'image HTML représentant le meilleur individu
L'utilisateur est vide	code 400
L'utilisateur n'existe pas	code 404
Le calcul n'existe pas	code 404
Le résultat du calcul n'existe pas	code 404

Créer un identifiant utilisateur unique

Méthode HTTP : PUT

Format de l'url : http://<adresseServeur>:<port>/ordogene/

Corps de la requête : Vide.

Description : Cette requête permet de créer un identifiant unique.

Réponse : La réponse correspond à l'identifiant généré aléatoirement.

Situation	Retour de l'API
Tout se passe correctement	code 200 avec un body contenant un nouvel identifiant
Impossibilité de créer l'utilisateur	code 500

Créer un identifiant utilisateur donné

Méthode HTTP : PUT

Format de l'url : http://<adresseServeur>:<port>/ordogene/<userID>

Corps de la requête : Vide.

Description : Cette requête permet de créer un identifiant en lui fournissant l'identifiant.

Réponse : La réponse correspond à l'identifiant passé en paramètre.

Situation	Retour de l'API
Tout se passe correctement	code 200 avec un body contenant un nouvel identifiant
L'utilisateur est vide	code 400
Impossibilité de créer l'utilisateur	code 500

Lancer un calcul

Méthode HTTP : PUT

Format de l'url : http://<adresseServeur>:<port>/ordogene/<userID>/calculations

Corps de la requête : Le corps de la requête doit être le modèle à optimiser au format JSON.

Description : Cette requête permet de lancer un calcul.

Réponse : La réponse est le cid associé au calcul.

Situation	Retour de l'API
Tout se passe correctement	code 200 avec un body contenant le cid.
L'utilisateur est vide	code 400
Le corps de la requête est vide	code 400
L'utilisateur n'existe pas	code 404
Le nombre de calcul lancé simultanément est atteint	code 503
Le modèle JSON est incorrect	code 400
Problème interne du serveur de l'API	code 500

Arrêter un calcul en cours

Méthode HTTP : POST

Format de l'url : http://<adresseServeur>:<port>/ordogene/<userID>/calculations/<cid>

Corps de la requête : Vide.

Description : Cette requête arrête le calcul correspondant au cid.

Réponse : La réponse est OK dans le cas où le calcul est arrêté.

Situation	Retour de l'API
Tout se passe correctement	code 200
L'utilisateur est vide	code 400
L'utilisateur n'existe pas	code 404
Le calcul n'est pas en cours	code 400
Le calcul n'existe pas	code 404

Supprimer un calcul

Méthode HTTP : DELETE

Format de l'url : http://<adresseServeur>:<port>/ordogene/<userID>/calculations/<cid>

Description : Cette requête permet la suppression d'un calcul.

Réponse : La réponse est OK si la suppression s'est déroulée sans problème.

Situation	Retour de l'API
Tout se passe correctement	code 200
L'utilisateur est vide	code 400
L'utilisateur n'existe pas	code 404
Le calcul n'existe pas	code 404
Le calcul est en cours	code 400
Le calcul ne peut être supprimé	code 500

B. CLI (Command-Line Interface)

1. Démarrage

La CLI s'exécute dans un terminal. Pour la démarrer, il faut se placer dans le dossier de l'archive JAR «ordogene-cli-<version>.jar» puis exécuter la commande :

```
java -jar ordogene-cli-<version>.jar --url=<server_url>
```

Pour java 9 :

```
java --add-modules java.xml.bind -jar ordogene-cli-0.0.1-SNAPSHOT.jar  
--url=<server_url>
```

L'argument **--url** doit correspondre à l'adresse d'un serveur API REST Ordogène.

2. Commandes

Intégrées de base

- **clear** : Efface la CLI.
- **exit, quit** : Quitte la CLI.
- **help** : Affiche les commandes disponibles
- **help <Command>** : Affiche l'aide de la commande.
- **script** : Lit et exécute un fichier de commandes.
- **stacktrace** : Affiche la stacktrace de la dernière erreur.

Commandes

- **launch-calculation** : Démarre un calcul à partir d'un modèle

```
launch-calculation [-m] file
```

argument :

- (Obligatoire) **-m** ou **--model** file : Le chemin d'un fichier modèle JSON.

- **list-calculations** : Liste les calculs.

```
list-calculation
```

- **remove-calculation** : Supprime un calcul.

```
remove-calculation [-cid] int
```

argument :

- (Obligatoire) `-cid` ou `--calculationid` int : L'identifiant du calcul.

- **result-calculation** : Récupère le résultat d'un calcul.

```
result-calculation [-cid] int [-d] file [--force] [--html]
```

arguments :

- (Obligatoire) `-cid` ou `--calculationid` int : L'identifiant du calcul.
- (Obligatoire) `-d` ou `--destination` file : Chemin de destination du résultat.
- (Optionnel) `--force` default = false : Forcer l'écrasement à l'emplacement du résultat.
- (Optionnel) `--html` : Enregistre le résultat en HTML.

- **stop-calculation** : Arrête un calcul.

```
stop-calculation [-cid] int
```

argument :

- (Obligatoire) `-cid` ou `--calculationid` int : L'identifiant du calcul.

- **xdg-open/open/start** : Ouvrir un fichier.

```
xdg-open/open/start [-i] file
```

argument :

- (Obligatoire) `-i` ou `--file` file : Nom du fichier.

- **whoami** : Affiche le nom d'utilisateur.

```
whoami
```

Il n'est pas obligatoire de recopier le nom des arguments à chaque appel d'une commande. Il est possible de les écrire directement, ils seront alors lus dans l'ordre fourni dans ce manuel.