

Tchatator413

Un protocole d'échange de tchatator, JSON-based.

- [Fondamentaux](#)
 - [Requêtes](#)
 - [Réponses](#)
- [Rôles](#)
 - [Client](#)
 - [Professionnel](#)
 - [Administrateur](#)
- [Erreurs communes](#)
- [Actions](#)
 - [login](#) : s'authentifier
 - [logout](#) : se déconnecter
 - [whois](#) : rechercher un compte
 - [send](#) : envoyer un message
 - [motd](#) : obtenir les messages reçus non lus
 - [inbox](#) : obtenir les messages reçus
 - [outbox](#) : obtenir les messages envoyés
 - [edit](#) : modifier un message
 - [rm](#) : supprimer un message
 - [block](#) : bloquer un client
 - [unblock](#) : débloquent un client
 - [ban](#) : bannir un client
 - [unban](#) : débannir un client

Fondamentaux

Requêtes

Le serveur reçoit une requête sous la forme d'une liste JSON:

```
[
  {
    "do": "login",
    "with": {
      "api_key": "...",
      "password": "..."
    }
  }
]
```

La liste peut contenir plusieurs actions.

Une forme raccourcie est possible pour les requêtes comprenant une seule action :

```
{
  "do": "login",
  "with": {
    "api_key": "...",
    "password": "..."
  }
}
```

Si la requête est vide [], la réponse est également une liste vide [].

Réponses

Une réponse *réussie* contient uniquement le corps de la réponse sans statut :

```
{
  "body": {
    "token": "token"
  }
}
```

Une réponse d'*erreur* contient un objet error avec un status et éventuellement un détail

```
{
  "error": {
    "status": 403
  }
}
```

La propriété `has_next_page` indique que le résultat est paginé et que le prochain numéro de page est valide (et donc que la page actuelle n'est pas la dernière et elle contient le nombre maximum d'éléments).

Rôles

Client

Aussi appelé membre. Échange avec des professionnels.

Professionnel

Peut échanger avec des clients.

Administrateur

Le seul administrateur du système.

Erreurs communes

Les erreurs sont retournées dans une clé **error** sous la forme suivante :

| Nom | Description | Contenu |
|-----------------------|--------------------------------|--|
| Bad request | Action inconnue | { "status": 400, "message": "description de l'erreur" } |
| Forbidden | Action non autorisée | { "status": 403 } |
| Payload too large | Un des arguments est trop long | { "status": 413, "key": "clé de l'argument trop long", "max_length": integer } |
| Unprocessable content | Invariant enfreint | { "status": 422, "reason": "nom de l'invariant enfreint" } |
| Too many requests | Rate limit atteinte | { "status": 429, "next_request_at": integer } |
| Internal server error | Erreur non spécifiée | { status: 500 } |

Actions

login : s'authentifier

Rôles : tous

| Argument | Type | Description |
|-----------------|---------|--------------|
| api_key | UUID V4 | Clé d'API |
| password | string | Mot de passe |

Crée une session.

Un même utilisateur peut avoir plusieurs sessions. La rate limit s'applique à l'utilisateur lui-même, et non pas indépendamment par session.

Réponse nominale

```
{ "token": 6554564 }
```

Erreurs

| Statut | Raison |
|--------|------------------------|
| 401 | Clé d'API invalide |
| 403 | Mot de passe incorrect |

logout : se déconnecter

Rôles : *tous*

| Argument | Type | Description |
|----------|---------|------------------|
| token | integer | Token de session |

Supprime une session.

Réponse nominale

```
{}
```

Erreurs

| Statut | Raison |
|--------|--------|
| 401 | c |

whois : rechercher un compte

Rôles : *tous*

| Argument | Type | Description |
|----------|------------------------------------|--------------------------------------|
| api_key | UUID V4 | Clé d'API |
| user | Clé de compte (ID, pseudo, e-mail) | Identifie l'utilisateur à rechercher |

Obtient les informations d'un compte à partir d'une de ses clés candidates (ID, pseudo, e-mail).

Réponse nominale

```
{ "user_id": 564, "email": "user@email.org", "last_name": "last name",  
  "first_name": "first name", "display_name": "display name", "kind": 0 }
```

Valeurs de *kind*:

| Valeur | Signification |
|--------|----------------------|
| 0 | Membre |
| 1 | Professionnel privé |
| 2 | Professionnel public |

Erreurs

| Statut | Raison |
|--------|--------------------|
| 401 | Clé d'API invalide |
| 404 | Compte introuvable |

send : envoyer un message

Rôles : *tous*

| Argument | Type | Description |
|----------------|------------------------------------|----------------------------------|
| token | integer | Token de session |
| dest | Clé de compte (ID, pseudo, e-mail) | Identifie le compte destinataire |
| content | string | contenu du message |

Envoie un message.

Réponse nominale

```
{ "msg_id": 465 }
```

Erreurs

| Statut | Raison |
|--------|--------------------------------------|
| 401 | l'argument token est invalide |
| 403 | utilisateur actuel bloqué ou banni |
| 404 | Compte dest introuvable |

Invariants

| Nom | Description |
|----------------------------|---|
| no_send_self | Le destinataire est différent de l'émetteur |
| client_send_pro | Si le token appartient à un client, le destinataire est un professionnel |
| pro_responds_client | Si le token appartient à un professionnel, le destinataire est un client lui ayant déjà envoyé un message |

motd : obtenir les messages reçus non lus

Rôles : *tous*

| Argument | Type | Description |
|----------|------|-------------|
|----------|------|-------------|

| Argument | Type | Description |
|----------|---------|------------------|
| token | integer | Token de session |

Obtient la liste des messages non lus, ordonnées par date d'envoi (plus ancien au plus récent).

Réponse nominale

Liste de messages

```
{
  "body": [
    {
      "msg_id": 55,
      "sent_at": 1736355387,
      "content": "Bonjour. j'ai une question.",
      "sender": 17,
      "recipient": 3
    },
    {
      "msg_id": 56,
      "sent_at": 1736355397,
      "content": "Bonjour. j'ai une question aussi (je suis pas la même
personne).",
      "sender": 16,
      "recipient": 3
    }
  ]
}
```

Erreurs

| Statut | Raison |
|--------|-------------------------------|
| 401 | l'argument token est invalide |

Problèmes possibles

- Si il y a des milliers de messages non lus, la réponse pourrait être énorme. Implémenter une pagination.
- Il y a des informations redondantes (read, recipient). Pour l'instant on les garde par souci de simplicité mais si il y a des problèmes de performance on les enlèvera.

inbox : obtenir les messages reçus

Rôles : tous

| Argument | Type | Description |
|----------|------|-------------|
|----------|------|-------------|

| Argument | Type | Description |
|----------|---------|--|
| token | integer | Token de session |
| page | integer | Numéro de page (1-based). Optionnel (1 par défaut) |

Obtient l'historique des messages reçus, avec pagination.

Réponse nominale

Liste de messages

```
{
  "has_next_page": true,
  "body": [
    {
      "msg_id": 55,
      "sent_at": 1736355387,
      "read_at": 1736355987,
      "read": true,
      "content": "Bonjour. j'ai une question.",
      "sender": 17,
      "recipient": 3
    },
    {
      "msg_id": 56,
      "sent_at": 1736355397,
      "content": "Bonjour. j'ai une question aussi (je suis pas la même
personne).",
      "sender": 16,
      "recipient": 3
    }
  ]
  // more...
}
```

Erreurs

| Statut | Raison |
|--------|-------------------------------|
| 401 | l'argument token est invalide |
| 404 | Numéro de page invalide |

Problèmes possibles

- Information redondante : recipient (elle est donnée par l'expéditeur de la requête)

outbox : obtenir les messages envoyés

Rôles : tous

| Argument | Type | Description |
|----------|---------|--|
| token | integer | Token de session |
| page | integer | Numéro de page (1-based). Optionnel (1 par défaut) |

Obtient l'historique des messages envoyés, avec pagination.

Réponse nominale

Liste de messages

```
{
  "has_next_page": true,
  "body": [
    {
      "msg_id": 57,
      "sent_at": 1736355487,
      "modified_at": 1736355487,
      "content": "Bonjour, quelle est votre question?",
      "sender": 3,
      "recipient": 17
    },
    {
      "msg_id": 58,
      "sent_at": 1736355497,
      "read_at": 1736356097,
      "content": "Bonjour, quelle est votre question? (je vous crois)",
      "sender": 3,
      "recipient": 16
    }
  ]
  // more...
}
```

Erreurs

| Statut | Raison |
|--------|-------------------------------|
| 401 | l'argument token est invalide |
| 404 | Numéro de page invalide |

Problèmes possibles

- Information redondante : sender (elle est donnée par l'expéditeur de la requête)

edit : modifier un message

Rôles : *tous*

| Argument | Type | Description |
|--------------------------|---------|----------------------------|
| <code>token</code> | integer | Token de session |
| <code>msg_id</code> | integer | ID du message à modifier |
| <code>new_content</code> | string | Nouveau contenu du message |

Modifie un message.

Réponse nominale

Erreurs

| Statut | Raison |
|--------|--|
| 401 | l'argument <code>token</code> est invalide |
| 403 | utilisateur actuel bloqué ou banni |
| 404 | Message introuvable |

Invariants

| Nom | Description |
|-----------------------|---|
| <code>owns_msg</code> | L'utilisateur actuel est soit l'administrateur, soit l'émetteur du message. |

`rm` : supprimer un message

Rôles : *tous*

| Argument | Type | Description |
|---------------------|---------|--------------------------|
| <code>token</code> | integer | Token de session |
| <code>msg_id</code> | integer | ID du message à modifier |

Supprime un message.

Réponse nominale

Erreurs

| Statut | Raison |
|--------|--|
| 401 | l'argument <code>token</code> est invalide |
| 404 | Message introuvable |

Invariants

| Nom | Description |
|-----------------------|---|
| <code>owns_msg</code> | L'utilisateur actuel est soit l'administrateur, soit l'émetteur du message. |

`block` : bloquer un client

Rôles : administrateur, professionnel

| Argument | Type | Description |
|--------------------|--|--|
| <code>token</code> | integer | Token de session |
| <code>user</code> | Clé de compte utilisateur (ID, pseudo, e-mail) | Identifie le client à bloquer (la cible) |

Bloque un client pendant une durée limitée.

Si l'utilisateur actuel est un professionnel, empêche la cible d'envoyer ou de modifier tout message s'adressant à celui-ci.

Si l'utilisateur actuel est l'administrateur, empêche la cible d'envoyer ou de modifier tout message.

Réponse nominale

| |
|-----------------|
| <code>{}</code> |
|-----------------|

Erreurs

| Statut | Raison |
|--------|--|
| 401 | l'argument <code>token</code> est invalide |
| 404 | Utilisateur introuvable |

Invariants

| Nom | Description |
|---------------------------------|---|
| <code>target_is_client</code> | La cible est un client |
| <code>target_not_blocked</code> | La cible n'est pas déjà bloquée par cet utilisateur |

unlock: débloquent un client

Rôles : administrateur, professionnel

| Argument | Type | Description |
|----------|--|---|
| token | integer | Token de session |
| user | Clé de compte utilisateur (ID, pseudo, e-mail) | Identifie le client à débloquent (la cible) |

Débloquent un client avant l'expiration de son blocage.

Réponse nominale

```
{}
```

Erreurs

| Statut | Raison |
|--------|--------------------------------------|
| 401 | l'argument token est invalide |
| 404 | Utilisateur introuvable |

Invariants

| Nom | Description |
|----------------|--|
| target_blocked | Si l'utilisateur actuel est un professionnel, la cible a été bloquée par celui-ci. Cela signifie que l'administrateur peut intervenir sur les blocages d'un professionnel, mais pas les autres professionnels. |

ban : bannir un client

Rôles : administrateur, professionnel

| Argument | Type | Description |
|----------|--|---|
| token | integer | Token de session |
| user | Clé de compte utilisateur (ID, pseudo, e-mail) | Identifie le client à bannir (la cible) |

Bannit un client.

Si l'utilisateur actuel est un professionnel, empêche la cible d'envoyer ou de modifier tout message s'adressant à celui-ci.

Si l'utilisateur actuel est l'administrateur, empêche la cible d'envoyer ou de modifier tout message.

Réponse nominale

Erreurs

| Statut | Raison |
|--------|--|
| 401 | l'argument <code>token</code> est invalide |
| 404 | Utilisateur introuvable |

Invariants

| Nom | Description |
|--------------------------------|--|
| <code>target_is_client</code> | La cible est un client |
| <code>target_not_banned</code> | La cible n'a pas déjà été bannie par cet utilisateur |

`unban` : débannir un client

Rôles : administrateur, professionnel

| Argument | Type | Description |
|--------------------|--|---|
| <code>token</code> | integer | Token de session |
| <code>user</code> | Clé de compte utilisateur (ID, pseudo, e-mail) | Identifie le client à débannir (la cible) |

Débannit un client.

Réponse nominale

Erreurs

| Statut | Raison |
|--------|--|
| 401 | l'argument <code>token</code> est invalide |
| 404 | Utilisateur introuvable |

Invariants

| Nom | Description |
|-----|-------------|
|-----|-------------|

| Nom | Description |
|---------------|---|
| target_banned | Si l'utilisateur actuel est un professionnel, la cible a été bannie par celui-ci. Cela signifie que l'administrateur peut intervenir sur les blocages d'un professionnel, mais pas les autres professionnels. |