

**Abstract**

Le présent document explique le fonctionnement du service d'annuaire partagés. Il comporte cinq parties : une à propos des Unités de Données du Protocole (PDU) ; une au sujet de l'étape de connexion du client au serveur ; une concernant les échanges côté utilisateur ; une concernant les échanges côté administrateur et enfin un diagramme de séquence résumant les différents échanges possibles entre le client et le serveur. Chaque partie détaillera les types de requêtes et réponses communiquées entre le client et le serveur ainsi que du serveur au client avec notamment des explications en mode DEBUG. Joint à ce document, les deux niveaux de raffinages des algorithmes client et serveur.

## Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Spécification des unités de données du protocole</b>	<b>5</b>
2.1	Format de l'en-tête	5
2.1.1	typePDU	5
2.1.2	typeRequete	5
2.1.3	typeCible	6
2.1.4	taille	6
2.1.5	codeErreur	6
2.2	Format de la charge utile	6
<b>3</b>	<b>Connexion</b>	<b>7</b>
3.1	Exemple de requête côté interface utilisateur	7
3.2	Exemple de requête côté serveur en mode DEBUG	7
<b>4</b>	<b>Description des échanges pour l'utilisateur</b>	<b>8</b>
4.1	Cas où l'utilisateur consulte son annuaire	8
4.1.1	Exemple de requête côté interface utilisateur	8
4.1.2	Exemple de requête côté serveur en mode DEBUG	8
4.2	Cas où l'utilisateur recherche dans l'annuaire	8
4.2.1	Exemple de requête côté interface utilisateur	9
4.2.2	Exemple de requête côté serveur en mode DEBUG	9
4.3	Cas où l'utilisateur consulte un annuaire distinct	9
4.3.1	Exemple de requête côté interface utilisateur	9
4.3.2	Exemple de requête côté serveur en mode DEBUG	10
4.4	Cas où l'utilisateur ajoute un contact	10
4.4.1	Exemple de requête côté interface utilisateur	10
4.4.2	Exemple de requête côté serveur en mode DEBUG	11
4.5	Cas où l'utilisateur modifie l'annuaire	11
4.5.1	Exemple de requête côté interface utilisateur	11
4.5.2	Exemple de requête côté serveur en mode DEBUG	11
4.6	Cas où l'utilisateur supprime un contact	12
4.6.1	Exemple de requête côté interface utilisateur	12
4.6.2	Exemple de requête côté serveur en mode DEBUG	12
4.7	Cas où l'utilisateur affiche les contacts autorisés à consulter son annuaire	13
4.7.1	Exemple de requête côté interface utilisateur	13
4.7.2	Exemple de requête côté serveur en mode DEBUG	13
4.8	Cas où l'utilisateur autorise à consulter son annuaire	13
4.8.1	Exemple de requête côté interface utilisateur	13
4.8.2	Exemple de requête côté serveur en mode DEBUG	14
4.9	Cas où l'utilisateur se déconnecte	14
4.9.1	Exemple de requête côté interface utilisateur	14
4.9.2	Exemple de requête côté serveur en mode DEBUG	14
<b>5</b>	<b>Description des échanges pour l'administrateur</b>	<b>15</b>
5.1	Cas où l'administrateur ajoute un utilisateur	15
5.1.1	Exemple de requête côté interface utilisateur	15
5.1.2	Exemple de requête côté serveur en mode DEBUG	15
5.2	Cas où l'administrateur supprime un utilisateur	15
5.2.1	Exemple de requête côté interface utilisateur	15
5.2.2	Exemple de requête côté serveur en mode DEBUG	16
5.3	Cas où l'administrateur modifie un utilisateur	16
5.3.1	Exemple de requête côté interface utilisateur	16
5.3.2	Exemple de requête côté serveur en mode DEBUG	16
5.4	Cas où l'administrateur consulte un annuaire distinct	16

5.4.1	Exemple de requête côté interface utilisateur . . . . .	16
5.4.2	Exemple de requête côté serveur en mode DEBUG . . . . .	17
5.5	Cas où l'administrateur affiche les utilisateurs de l'application . . . .	17
5.5.1	Exemple de requête côté interface utilisateur . . . . .	17
5.5.2	Exemple de requête côté serveur en mode DEBUG . . . . .	17
5.6	Cas où l'administrateur se déconnecte . . . . .	17
5.6.1	Exemple de requête côté interface utilisateur . . . . .	17
5.6.2	Exemple de requête côté serveur en mode DEBUG . . . . .	17
<b>6</b>	<b>Exemple d'échanges entre le client et le serveur . . . . .</b>	<b>18</b>
<b>7</b>	<b>Glossaire . . . . .</b>	<b>20</b>
<b>8</b>	<b>Annexes . . . . .</b>	<b>21</b>

## 1. Introduction

Le présent document détaille le fonctionnement d'un service d'annuaire partagé. Le service est basé sur une communication entre un client et un serveur. L'objectif de ce service est d'offrir aux utilisateurs la possibilité de consulter et de maintenir un annuaire propre à chacun. Chaque contact comporte des informations telles que le nom, le prénom et l'adresse email des contacts ajoutés. D'autres données telles que le numéro de téléphone ainsi que l'adresse postale restent des informations facultatives. Chaque utilisateur développe son annuaire en ajoutant ou en supprimant des contacts. Pour réguler les comptes utilisateurs, un administrateur peut ajouter, modifier ou encore supprimer les utilisateurs et a ainsi tous les droits sur l'ensemble des annuaires.

## 2. Spécification des unités de données du protocole

Pour la réalisation de ce projet, les messages échangés entre le client et le serveur comporteront différents champs qui varient en nombre et en type selon le type de requête. Cette section décrit le format des unités de données du protocole (PDU).

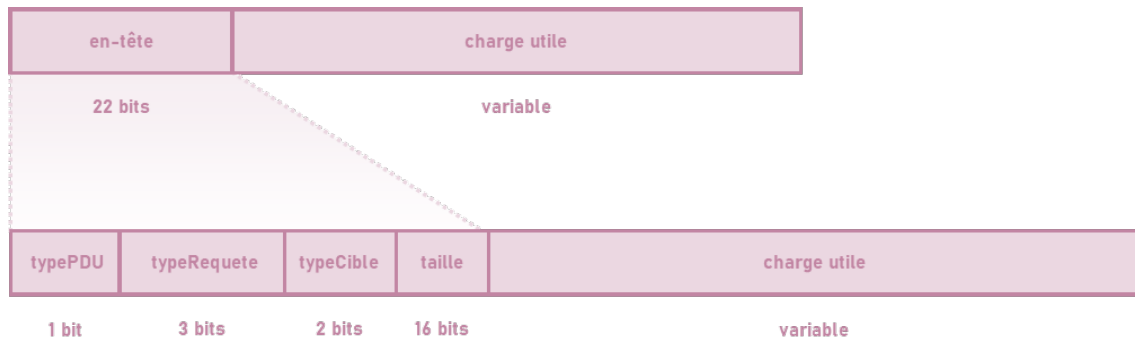


Figure 1: Format d'une unité de données du protocole de type requête

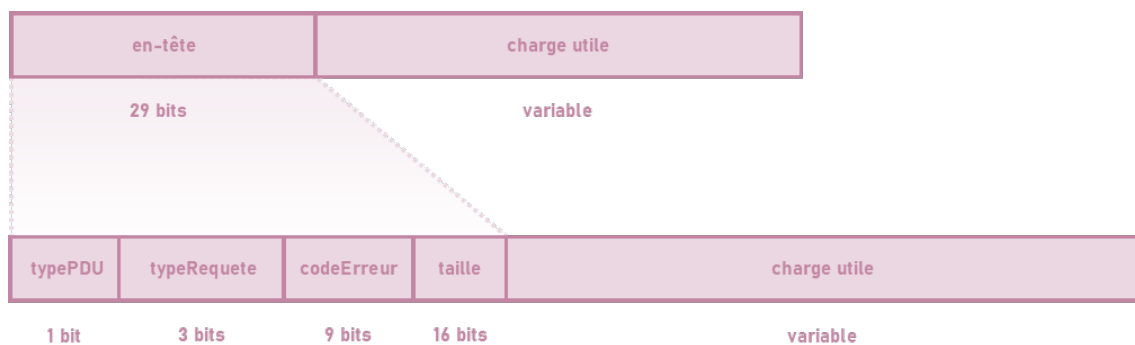


Figure 2: Format d'une unité de données du protocole de type réponse

### 2.1. Format de l'en-tête

#### 2.1.1. typePDU

1 bit. Il s'agit d'un champ de largeur fixe permettant d'indiquer le type de la PDU:  
0 → Request ; 1 → Response.

#### 2.1.2. typeRequete

3 bits. Il s'agit d'un champ de largeur fixe permettant d'indiquer le type de la requête. On en distingue 8, régissant les échanges entre le client et le serveur :

- 000 AUTH
- 001 ADD
- 010 DELETE
- 011 EDIT
- 100 GET
- 101 ALLOW
- 110 SEARCH
- 111 QUIT

### 2.1.3. typeCible

2 bits. Champ spécifique à la requête. Il s'agit d'un champ de largeur fixe permettant d'indiquer le type de l'objet ciblé par la requête. Il en existe trois :

- 01 Utilisateur
- 10 Annuaire
- 11 Contact

On considère que si typeCible vaut 00, on omet sa valeur.

### 2.1.4. taille

16 bits. Champ commun aux deux types d'unités de données du protocole. Il s'agit d'un champ de largeur fixe permettant d'indiquer la longueur de la charge utile (donnée) qui suit.

### 2.1.5. codeErreur

9 bits. Champ spécifique à la réponse. Chaque requête peut générer des codes d'erreurs tels que :

- 200 : Requete traitée avec succès.
- 400 : Identifiants incorrects.
- 401 : Erreur d'émission client.
- 402 : Requête au format incorrect.
- 403 : Erreur client inconnue.
- 501 : Erreur d'émission serveur.
- 502 : Erreur d'ouverture du fichier sur le serveur.
- 503 : Serveur injoignable.

## 2.2. Format de la charge utile

La charge utile est un champ de taille nulle ou plus. Sa taille est indiquée par le champ 'taille' présent dans l'en-tête.

Il contient les informations associées à la requête/réponse. L'envoi de données est en termes de format de JSON.

### 3. Connexion

La connexion nécessite d'avoir un compte utilisateur créé au préalable par l'administrateur. Ceci étant fait, la connexion nécessite de passer par l'étape d'authentification sur le serveur avec un couple identifiant et mot de passe correct.

#### 3.1. Exemple de requête côté interface utilisateur

```
- Bienvenue sur votre service d'annuaire partagés -  
Identifiant : axdls31  
Mot de passe : *****  
Connexion effectuée avec succès.
```

#### 3.2. Exemple de requête côté serveur en mode DEBUG

Pour la connexion, une requête de type AUTH, accompagnée du couple identifiant et mot de passe, est envoyée au serveur. Ce dernier renvoie une confirmation de succès ou d'échec : Code erreur 200 si le couple est correct, 400 sinon.

```
DEBUG : [INFO] Connexion  
DEBUG : [REQUETE] AUTH axdls31 *****  
DEBUG : [CODE ERREUR] 200  
DEBUG : [AFFICHAGE] Connexion effectuée avec succès.
```

## 4. Description des échanges pour l'utilisateur

### 4.1. Cas où l'utilisateur consulte son annuaire

L'utilisateur doit pouvoir lire les données contenues dans son annuaire. Pour se faire, l'utilisateur doit sélectionner 'consulter mon annuaire' en indiquant le chiffre 1.

#### 4.1.1. Exemple de requête côté interface utilisateur

**Axel - Utilisateur**

Sélectionner l'action souhaitée:

- 1 : Consulter l'annuaire
  - 2 : Rechercher dans l'annuaire
  - 3 : Consulter un annuaire distinct
  - 4 : Ajouter un contact
  - 5 : Modifier l'annuaire
  - 6 : Supprimer un contact
  - 7 : Consulter les utilisateurs autorisés à consulter mon annuaire
  - 8 : Autoriser un utilisateur à consulter mon annuaire
  - 9 : Déconnexion
- Choix : 1

Contenu de votre annuaire :

- 1) Admin Admin admin@stri.fr 8098350982 2 Chemin UPSSITECH 31000 TOULOUSE
- 2) Berners Tim tim.berners@email.com 0606060606 2 Rue de UPSSITECH 31000 TOULOUSE
- 3) Naim Amine amine.naim@stri.fr 0101010101 1 Palais de l'Elysée 75000 PARIS

#### 4.1.2. Exemple de requête côté serveur en mode DEBUG

Consulter son annuaire est effectué en envoyant une requête de type GET au serveur.

```
DEBUG : [INFO] Consultation de l'annuaire
DEBUG : [INFO] Lecture de l'annuaire du user courant
DEBUG : [REQUETE] GET axel.delas@stri.fr
DEBUG : [CODE ERREUR] 200
DEBUG : [INFO] Affichage des contacts lus
DEBUG : [AFFICHAGE] Admin Admin admin@stri.fr 8098350982 2 Chemin
UPSSITECH 31000 TOULOUSE
DEBUG : [AFFICHAGE] Berners Tim tim.berners@email.com 0606060606 2 Rue
de UPSSITECH 31000 TOULOUSE
DEBUG : [AFFICHAGE] Naim Amine amine.naim@stri.fr 0101010101 1 Palais
de l'Elysée 75000 PARIS
```

### 4.2. Cas où l'utilisateur recherche dans l'annuaire

L'utilisateur doit pouvoir effectuer une recherche parmi les données présentes dans son annuaire, l'objectif étant de lire spécifiquement un contact. Ceci peut être fait en saisissant le chiffre 2 et ainsi choisir 'Rechercher dans l'annuaire' figurant dans le menu.



#### 4.2.1. Exemple de requête côté interface utilisateur

**Axel - Utilisateur**

Sélectionner l'action souhaitée:

- 1 : Consulter l'annuaire
- 2 : Rechercher dans l'annuaire
- 3 : Consulter un annuaire distinct
- 4 : Ajouter un contact
- 5 : Modifier l'annuaire
- 6 : Supprimer un contact
- 7 : Consulter les utilisateurs autorisés à consulter mon annuaire
- 8 : Autoriser un utilisateur à consulter mon annuaire
- 9 : Déconnexion

Choix : 2

Email du contact : amine.naim@stri.fr

Informations de amine.naim@stri.fr :

Delas Axel axel.delas@stri.fr 01010101 1 Palais de l'Elysée 75000  
PARIS

#### 4.2.2. Exemple de requête côté serveur en mode DEBUG

Rechercher dans l'annuaire un contact spécifique est effectué d'une part grâce à la requête de type GET mais d'une autre part grâce à la requête de type SEARCH.

```
DEBUG : [INFO] Consultation de l'annuaire
DEBUG : [INFO] Lecture de l'annuaire du user courant
DEBUG : [REQUETE] GET axel.delas@stri.fr
DEBUG : [INFO] Recherche du contact amine.naim@stri.fr
DEBUG : [REQUETE] SEARCH amine.naim@stri.fr
DEBUG : [INFO] Lecture du contact amine.naim@stri.fr
DEBUG : [REQUETE] GET amine.naim@stri.fr
DEBUG : [INFO] Affichage des contacts de amine.naim@stri.fr
DEBUG : [CODE ERREUR] 200
DEBUG : [AFFICHAGE] Delas Axel axel.delas@stri.fr 01010101 1 Palais
de l'Elysée 75000 PARIS
```

#### 4.3. Cas où l'utilisateur consulte un annuaire distinct

L'utilisateur doit pouvoir consulter un annuaire distinct appartenant à un utilisateur qui a donné l'autorisation au préalable. Ainsi, l'utilisateur saisie le chiffre 3 afin de choisir 'Consulter un annuaire distinct' figurant dans le menu.

##### 4.3.1. Exemple de requête côté interface utilisateur

**Axel - Utilisateur**

Sélectionner l'action souhaitée:

- 1 : Consulter l'annuaire
- 2 : Rechercher dans l'annuaire
- 3 : Consulter un annuaire distinct
- 4 : Ajouter un contact
- 5 : Modifier l'annuaire
- 6 : Supprimer un contact
- 7 : Consulter les utilisateurs autorisés à consulter mon annuaire
- 8 : Autoriser un utilisateur à consulter mon annuaire

```
9 : Déconnexion
Choix : 3

Email de l'utilisateur : amine.naim@stri.fr

Contenu de l'annuaire de Amine Naim :
1) Admin Admin admin@stri.fr 8098350982 2 Chemin UPSSITECH 31000
TOULOUSE
2) Axel Delas axel.delas@stri.fr 0101010101 1 Palais de l'Elysée 75000
PARIS
3) Pierre Beckers pierre.beckers@stri.fr 9275028408 20 Rue du Chêne
31000 TOULOUSE
```

#### 4.3.2. Exemple de requête côté serveur en mode DEBUG

Lire un annuaire distinct est effectué en envoyant une requête de type GET.

```
DEBUG : [INFO] Consultation d'un annuaire distinct
DEBUG : [INFO] Lecture de l'annuaire du user courant
DEBUG : [REQUETE] GET axel.delas@stri.fr
DEBUG : [INFO] Lecture de l'annuaire de amine.naim@stri.fr
DEBUG : [REQUETE] GET amine.naim@stri.fr
DEBUG : [CODE ERREUR] 200
DEBUG : [INFO] Affichage des contacts de amine.naim@stri.fr
DEBUG : [AFFICHAGE] Admin Admin admin@stri.fr 8098350982 2 Chemin
UPSSITECH 31000 TOULOUSE
DEBUG : [AFFICHAGE] Axel Delas axel.delas@stri.fr 0101010101 1 Palais
de l'Elysée 75000 PARIS
DEBUG : [AFFICHAGE] Pierre Beckers pierre.beckers@stri.fr 9275028408
20 Rue du Chêne 31000 TOULOUSE
```

#### 4.4. Cas où l'utilisateur ajoute un contact

L'utilisateur doit pouvoir ajouter un nouveau contact dans son annuaire en renseignant les informations nécessaires. Pour se faire, l'utilisateur doit sélectionner 'Ajouter un contact' en saisissant le chiffre 4.

##### 4.4.1. Exemple de requête côté interface utilisateur

```
Axel - Utilisateur
Sélectionner l'action souhaitée:
1 : Consulter l'annuaire
2 : Rechercher dans l'annuaire
3 : Consulter un annuaire distinct
4 : Ajouter un contact
5 : Modifier l'annuaire
6 : Supprimer un contact
7 : Consulter les utilisateurs autorisés à consulter mon annuaire
8 : Autoriser un utilisateur à consulter mon annuaire
9 : Déconnexion
Choix : 4

Email du nouveau contact : amine.naim@stri.fr
Nouveau contact ajouté avec succès.
```

#### 4.4.2. Exemple de requête côté serveur en mode DEBUG

Ajouter un nouveau contact dans l'annuaire envoie une requête de type ADD au serveur.

```
DEBUG : [INFO] Ajout d'un nouveau contact
DEBUG : [INFO] Ajout du contact amine.naim@stri.fr
DEBUG : [REQUETE] ADD amine.naim@stri.fr
DEBUG : [CODE ERREUR] 200
DEBUG : [AFFICHAGE] Nouveau contact ajouté avec succès.
```

#### 4.5. Cas où l'utilisateur modifie l'annuaire

Ici l'utilisateur doit pouvoir modifier une information appartenant à l'un des contacts faisant parti de l'annuaire. L'utilisateur doit choisir 'Modifier l'annuaire' en saisissant le chiffre 5.

##### 4.5.1. Exemple de requête côté interface utilisateur

```
Axel - Utilisateur
Sélectionner l'action souhaitée:
1 : Consulter l'annuaire
2 : Rechercher dans l'annuaire
3 : Consulter un annuaire distinct
4 : Ajouter un contact
5 : Modifier l'annuaire
6 : Supprimer un contact
7 : Consulter les utilisateurs autorisés à consulter mon annuaire
8 : Autoriser un utilisateur à consulter mon annuaire
9 : Déconnexion
Choix : 5

Contenu de votre annuaire :
1) Admin Admin admin@stri.fr
2) Berners Tim tim.berners@email.com
3) Naim Amine amine.naim@abcd.net

Email du contact à modifier : tim.berners@email.com

Information à modifier : email
Nouvelle information : tim.berners@stri.fr
Contact modifié avec succès.
```

##### 4.5.2. Exemple de requête côté serveur en mode DEBUG

La modification une information associée à un contact effectue plusieurs types de requêtes dont une requête de type GET et une requête de type EDIT.

```
DEBUG : [INFO] Modification d'un contact
DEBUG : [INFO] Lecture de l'annuaire du user courant
DEBUG : [REQUETE] GET axel.delas@stri.fr
DEBUG : [AFFICHAGE] Admin Admin admin@stri.fr
DEBUG : [AFFICHAGE] Berners Tim tim.berners@email.com
DEBUG : [AFFICHAGE] Naim Amine amine.naim@abcd.net
DEBUG : [INFO] Lecture du contact tim.berners@email.com
DEBUG : [INFO] Modification du contact tim.berners@email.com
DEBUG : [REQUETE] EDIT tim.berners@email.com tim.berners@stri.fr
```

```
DEBUG : [CODE ERREUR] 200
DEBUG : [AFFICHAGE] Contact modifié avec succès.
```

#### 4.6. Cas où l'utilisateur supprime un contact

L'utilisateur doit pouvoir supprimer un contact de son annuaire. Ce dernier doit choisir 'Supprimer un contact' figurant dans le menu en saisissant le chiffre 6.

##### 4.6.1. Exemple de requête côté interface utilisateur

```
Amine - Utilisateur
Sélectionner l'action souhaitée:
1 : Consulter l'annuaire
2 : Rechercher dans l'annuaire
3 : Consulter un annuaire distinct
4 : Ajouter un contact
5 : Modifier l'annuaire
6 : Supprimer un contact
7 : Consulter les utilisateurs autorisés à consulter mon annuaire
8 : Autoriser un utilisateur à consulter mon annuaire
9 : Déconnexion
Choix : 6

Contenu de votre annuaire :
1) Admin Admin admin@stri.fr 8098350982 2 Chemin UPSSITECH 31000
TOULOUSE
2) Berners Tim tim.berners@email.com 0606060606 2 Rue de UPSSITECH
31000 TOULOUSE
3) Axel Delas axel.delas@stri.fr 0101010101 1 Palais de l'Elysée 75000
PARIS

Email du contact à supprimer : axel.delas@stri.fr
Contact supprimé avec succès.
```

##### 4.6.2. Exemple de requête côté serveur en mode DEBUG

Supprimer un contact de l'annuaire nécessite d'utiliser plusieurs types de requêtes tels que GET et DELETE.

```
DEBUG : [INFO] Suppression d'un contact
DEBUG : [INFO] Lecture de l'annuaire du user courant
DEBUG : [REQUETE] GET amine.naim@stri.fr
DEBUG : [AFFICHAGE] Admin Admin admin@stri.fr 8098350982 2 Chemin
UPSSITECH 31000 TOULOUSE
DEBUG : [AFFICHAGE] Berners Tim tim.berners@email.com 0606060606 2 Rue
de UPSSITECH 31000 TOULOUSE
DEBUG : [AFFICHAGE] Axel Delas axel.delas@stri.fr 0101010101 1 Palais
de l'Elysée 75000 PARIS
DEBUG : [INFO] Suppression du contact axel.delas@stri.fr
DEBUG : [REQUETE] DELETE axel.delas@stri.fr
DEBUG : [CODE ERREUR] 200
DEBUG : [AFFICHAGE] Contact supprimé avec succès.
```

#### 4.7. Cas où l'utilisateur affiche les contacts autorisés à consulter son annuaire

L'utilisateur doit pouvoir afficher les contacts qu'il a au préalable autorisé à consulter son annuaire. Pour se faire, l'utilisateur choisi 'Consulter les utilisateurs autorisés à consulter mon annuaire' en saisissant le chiffre 7.

##### 4.7.1. Exemple de requête côté interface utilisateur

**Axel - Utilisateur**

Sélectionner l'action souhaitée:

- 1 : Consulter l'annuaire
  - 2 : Rechercher dans l'annuaire
  - 3 : Consulter un annuaire distinct
  - 4 : Ajouter un contact
  - 5 : Modifier l'annuaire
  - 6 : Supprimer un contact
  - 7 : Consulter les utilisateurs autorisés à consulter mon annuaire
  - 8 : Autoriser un utilisateur à consulter mon annuaire
  - 9 : Déconnexion
- Choix : 7

1) tim.berners@email.com

##### 4.7.2. Exemple de requête côté serveur en mode DEBUG

L'affichage des utilisateurs autorisés à consulter l'annuaire est effectuée à travers une requête de type GET au serveur.

```
DEBUG : [INFO] Affichage des users autorisés à consulter l'annuaire
DEBUG : [INFO] Lecture de l'annuaire du user courant
DEBUG : [REQUETE] GET axel.delas@stri.fr
DEBUG : [INFO] Lecture des contacts ayant l'autorisation
DEBUG : [REQUETE] WHO_ALLOW axel.delas@stri.fr
DEBUG : [CODE ERREUR] 200
DEBUG : [AFFICHAGE] tim.berners@email.com
```

#### 4.8. Cas où l'utilisateur autorise à consulter son annuaire

L'utilisateur doit pouvoir autoriser un utilisateur distinct à consulter son annuaire. Pour se faire, l'utilisateur choisi 'Autoriser un utilisateur à consulter mon annuaire' figurant dans le menu en saisissant le chiffre 8.

##### 4.8.1. Exemple de requête côté interface utilisateur

**Axel - Utilisateur**

Sélectionner l'action souhaitée:

- 1 : Consulter l'annuaire
- 2 : Rechercher dans l'annuaire
- 3 : Consulter un annuaire distinct
- 4 : Ajouter un contact
- 5 : Modifier l'annuaire
- 6 : Supprimer un contact
- 7 : Consulter les utilisateurs autorisés à consulter mon annuaire
- 8 : Autoriser un utilisateur à consulter mon annuaire
- 9 : Déconnexion

```
Choix : 8
```

```
Adresse email de l'utilisateur à autoriser : amine.naim@stri.fr  
Autorisation ajoutée avec succès.
```

#### 4.8.2. Exemple de requête côté serveur en mode DEBUG

Autoriser un utilisateur à consulter son annuaire effectue d'une part une requête de type GET et d'une autre part une requête ALLOW.

```
DEBUG : [INFO] Autorisation d'un user à consulter l'annuaire  
DEBUG : [INFO] Lecture de l'annuaire du user courant  
DEBUG : [REQUETE] GET axel.delas@stri.fr  
DEBUG : [INFO] Ajout de l'autorisation à amine.naim@stri.fr  
DEBUG : [REQUETE] ALLOW amine.naim@stri.fr  
DEBUG : [CODE ERREUR] 200  
DEBUG : [AFFICHAGE] Autorisation ajoutée avec succès.
```

#### 4.9. Cas où l'utilisateur se déconnecte

L'utilisateur doit pouvoir se déconnecter du serveur de l'application qui permet de mettre fin à la connexion entre le client et le serveur. Pour se faire, l'utilisateur doit saisir le chiffre 10 et ainsi choisir 'Déconnexion' dans le menu.

##### 4.9.1. Exemple de requête côté interface utilisateur

###### Axel - Utilisateur

Sélectionner l'action souhaitée:

```
1 : Consulter l'annuaire  
2 : Rechercher dans l'annuaire  
3 : Consulter un annuaire distinct  
4 : Ajouter un contact  
5 : Modifier l'annuaire  
6 : Supprimer un contact  
7 : Consulter les utilisateurs autorisés à consulter mon annuaire  
8 : Autoriser un utilisateur à consulter mon annuaire  
9 : Déconnexion  
Choix : 9
```

Déconnexion effectuée avec succès.

##### 4.9.2. Exemple de requête côté serveur en mode DEBUG

La déconnexion se fait grâce à l'envoi d'une requête de type QUIT au serveur.

```
DEBUG : [INFO] Déconnexion  
DEBUG : [INFO] Déconnexion du user courant  
DEBUG : [REQUETE] QUIT axel.delas@stri.fr  
DEBUG : [CODE ERREUR] 200  
DEBUG : [AFFICHAGE] Déconnexion effectuée avec succès.
```

## 5. Description des échanges pour l'administrateur

### 5.1. Cas où l'administrateur ajoute un utilisateur

L'administrateur doit pouvoir ajouter un nouvel utilisateur au sein du service d'annuaire partagés. A l'ajout de ce dernier, un annuaire lui ait créé et associé. Pour se faire, l'administrateur doit saisir le chiffre 1 et ainsi choisir 'Ajouter un utilisateur' figurant dans le menu

#### 5.1.1. Exemple de requête côté interface utilisateur

```
Axel - Administrateur
Sélectionner l'action souhaitée:
1 : Ajouter un utilisateur
2 : Supprimer un utilisateur
3 : Modifier un utilisateur
4 : Consulter un annuaire distinct
5 : Afficher les utilisateurs de l'application
6 : Déconnexion
Choix : 1

Informations du nouvel utilisateur :
Nom : Delas
Prénom : Axel
Email : axel.delas@stri.fr
Numéro de tel : 0101010101
Adresse postale : 1 Palais de l'Elysée 75000 PARIS
Nouvel utilisateur crée avec succès.
```

#### 5.1.2. Exemple de requête côté serveur en mode DEBUG

Ajouter un nouvel utilisateur dans le service est effectué en envoyant une requête de type ADD au serveur.

```
DEBUG : [INFO] Ajout d'un nouveau user
DEBUG : [INFO] Ajout de axel.delas@stri.fr
DEBUG : [REQUETE] ADD Delas Axel axel.delas@stri.fr 0101010101 1 Rue
de l'Elysée 75000 PARIS
DEBUG : [CODE ERREUR] 200
DEBUG : [AFFICHAGE] Nouvel utilisateur crée avec succès
```

### 5.2. Cas où l'administrateur supprime un utilisateur

L'administrateur doit pouvoir supprimer un utilisateur au sein du service d'annuaire partagés. De plus, en supprimant le compte de ce dernier, son annuaire partagé est également supprimé. Pour se faire, l'administrateur doit saisir le chiffre 2 afin de choisir 'Supprimer un utilisateur' figurant dans le menu

#### 5.2.1. Exemple de requête côté interface utilisateur

```
Axel - Administrateur
Sélectionner l'action souhaitée:
1 : Ajouter un utilisateur
2 : Supprimer un utilisateur
3 : Modifier un utilisateur
4 : Consulter un annuaire distinct
```

```
5 : Afficher les utilisateurs de l'application
6 : Déconnexion
Choix : 2

Utilisateurs de l'application :
1) amine.naim@stri.fr
2) pierre.beckers@stri.fr
3) tim.berners@stri.fr
4) andrei.markov@stri.fr

Email de l'utilisateur à supprimer : axel.delas@stri.fr
Utilisateur supprimé avec succès.
```

#### 5.2.2. Exemple de requête côté serveur en mode DEBUG

Supprimer un utilisateur de l'application se fait en envoyant une requête GET puis une requête DELETE.

```
DEBUG : [INFO] Suppression d'un user
DEBUG : [INFO] Lecture des users de l'application
DEBUG : [REQUETE] GET_ALL admin@stri.fr
DEBUG : [CODE ERREUR] 200
DEBUG : [AFFICHAGE] amine.naim@stri.fr
DEBUG : [AFFICHAGE] pierre.beckers@stri.fr
DEBUG : [AFFICHAGE] tim.berners@stri.fr
DEBUG : [AFFICHAGE] andrei.markov@stri.fr
DEBUG : [INFO] Suppression du user axel.delas@stri.net
DEBUG : [REQUETE] DELETE axel.delas@stri.fr
DEBUG : [CODE ERREUR] 200
DEBUG : [AFFICHAGE] Utilisateur supprimé avec succès.
```

### 5.3. Cas où l'administrateur modifie un utilisateur

L'administrateur doit pouvoir modifier un utilisateur. Ceci se fait de manière similaire quant à l'utilisateur. De plus, l'administrateur doit saisir le chiffre 3 afin de choisir 'Modifier un utilisateur' dans le menu

#### 5.3.1. Exemple de requête côté interface utilisateur

Voir 4.5.1

#### 5.3.2. Exemple de requête côté serveur en mode DEBUG

Voir 4.5.2

### 5.4. Cas où l'administrateur consulte un annuaire distinct

L'administrateur doit pouvoir consulter un annuaire distinct. Ceci se fait de manière similaire quant à l'utilisateur à l'exception que l'administrateur n'a pas besoin d'autorisation au préalable. Ce dernier doit saisir le chiffre 3 pour choisir 'Modifier un utilisateur' figurant dans le menu

#### 5.4.1. Exemple de requête côté interface utilisateur

Voir 4.3.1



#### 5.4.2. Exemple de requête côté serveur en mode DEBUG

Voir 4.3.2

### 5.5. Cas où l'administrateur affiche les utilisateurs de l'application

L'administrateur doit pouvoir afficher les utilisateurs faisant parti de l'application. Pour se faire, l'administrateur doit saisir le chiffre 5 pour choisir 'Afficher les utilisateurs de l'application' figurant dans le menu.

#### 5.5.1. Exemple de requête côté interface utilisateur

```
Axel - Administrateur
Sélectionner l'action souhaitée:
1 : Ajouter un utilisateur
2 : Supprimer un utilisateur
3 : Modifier un utilisateur
4 : Consulter un annuaire distinct
5 : Afficher les utilisateurs de l'application
6 : Déconnexion
Choix : 5

Utilisateurs de l'application :
1) amine.naim@stri.fr
2) pierre.beckers@stri.fr
3) tim.berners@stri.fr
4) andrei.markov@stri.fr
```

#### 5.5.2. Exemple de requête côté serveur en mode DEBUG

```
DEBUG : [INFO] L'admin affiche les users de l'application
DEBUG : [INFO] Lecture des users de l'application
DEBUG : [REQUETE] GET_ALL admin@stri.fr
DEBUG : [CODE ERREUR] 200
DEBUG : [AFFICHAGE] amine.naim@stri.fr
DEBUG : [AFFICHAGE] pierre.beckers@stri.fr
DEBUG : [AFFICHAGE] tim.berners@stri.fr
DEBUG : [AFFICHAGE] andrei.markov@stri.fr
```

### 5.6. Cas où l'administrateur se déconnecte

L'administrateur doit pouvoir se déconnecter du serveur de l'application. Ceci se fait de manière similaire quant à l'utilisateur à l'exception que l'administrateur doit saisir le chiffre 6 pour choisir 'Déconnexion' figurant dans le menu.

#### 5.6.1. Exemple de requête côté interface utilisateur

Voir 4.9.1

#### 5.6.2. Exemple de requête côté serveur en mode DEBUG

Voir 4.9.2

6. Exemple d'échanges entre le client et le serveur

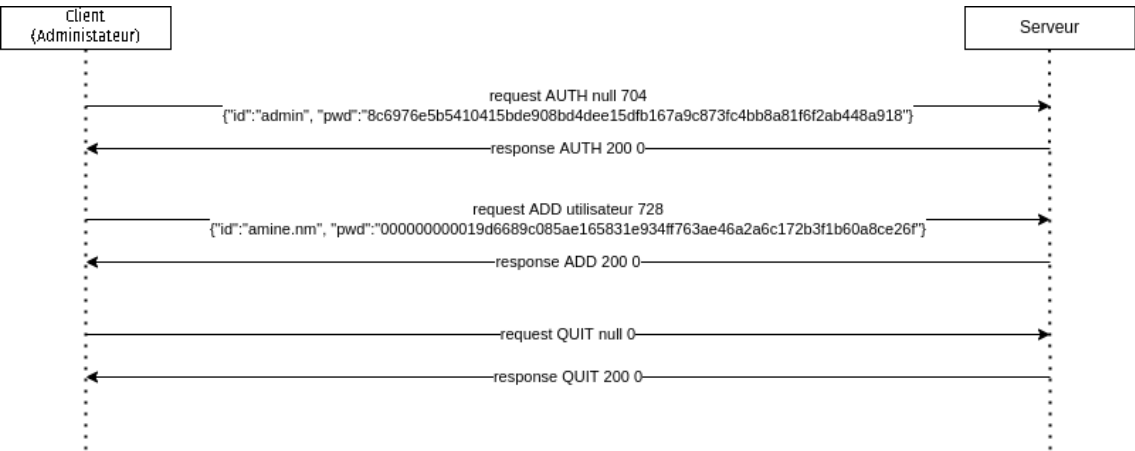


Figure 3: Exemple sans erreur d'échanges entre l'administrateur et le serveur pour une création d'utilisateur

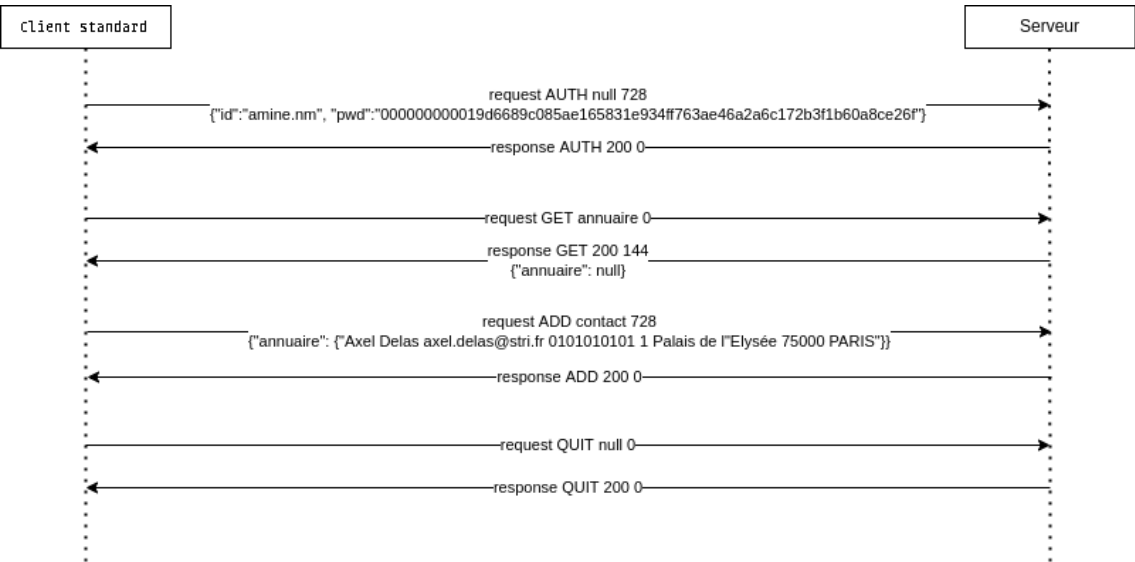


Figure 4: Exemple sans erreur d'échanges entre l'utilisateur et le serveur pour ajouter un contact dans l'annuaire

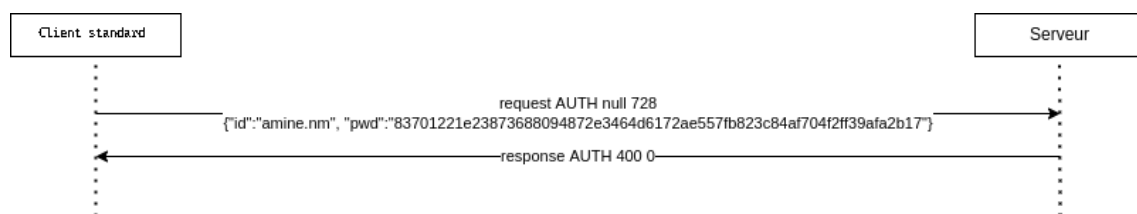


Figure 5: Exemple d'authentification avec des identifiants incorrects

## 7. Glossaire

**ADD** Requête indiquant l'ajout de données dans le fichier de données. 5

**administrateur** Profil au sein de l'application qui a pour rôle d'administrer les utilisateurs : création, modification, suppression. 7

**ALLOW** Requête permettant d'ajouter à un utilisateur spécifié les droits d'accès en lecture à l'annuaire du client émettant la requête. 5

**AUTH** Requête permettant au serveur d'identifier si le client est éligible à l'accès à une ressource. 5

**charge utile** Partie du paquet de données transmis qui contient l'information à transmettre. (en anglais payload). 2, 6

**DELETE** Requête indiquant la suppression d'une donnée dans le fichier de données. 5

**EDIT** Requête indiquant la modification d'une donnée dans le fichier de données. 5

**GET** Requête permettant de demander une donnée d'une ressource spécifiée. 5

**PDU** Unité de données du protocole (Protocol Data Unit). 5

**QUIT** Requête indiquant la demande de déconnexion de l'utilisateur émetteur. 5

**requête** Sollicitation d'un service par une entité pour faire une activité. 5

**réponse** Réponse à un événement. 1, 5

**SEARCH** Requête permettant de chercher une donnée parmi une ressource spécifiée. 5

**utilisateur** Profil au sein de l'application qui est consommateur de l'application. Un annuaire lui est fourni à la création de son compte qu'il pourra développer en ajoutant, modifiant, supprimant des contacts. . 7

## 8. Annexes

Premier Raffinage, Client > Disponible dans `raffinage1/premier_raffinage_client.txt`

Premier raffinage, Serveur > Disponible dans `raffinage1/premier_raffinage_serveur.txt`

Second raffinage, Client > Disponible dans `raffinage2/second_raffinage_client.txt`

Second raffinage, Serveur > Disponible dans `raffinage2/second_raffinage_serveur.txt`