

# WEBSERVICES REST

## 01/ Présentation Fonctionnelle

- WS Authentification = Récupération d'un jeton
- WS CFN = Utilisation du Coffre-Fort Numérique

## 02/ Tests avec Postman

- Comment paramétrer le client universel Postman pour tester nos webservices

## 03/ Tests avec Swagger UI

- Comment utiliser l'interface Swagger UI pour tester nos webservices

# 01 / PRÉSENTATION FONCTIONNELLE

# Présentation Fonctionnelle

## ❑ Services disponibles :

- **Authentification**
- **Coffr-Fort Numérique (CFN)**

A terme, notre API REST sera enrichie avec tous les services déjà disponibles sur notre API SOAP :

Inscription / Gestion des comptes, Dépôt des déclarations, Suivi des flux, Paramétrage des alertes, Relevés de compte, ...

## ❑ Environnements :

Une description des appels / routes de l'API est disponible :

		RECETTE	PRODUCTION
SwaggerUI		<a href="https://recette-aspone.tessitechno.fr/api/rest/v1/swagger.html">https://recette-aspone.tessitechno.fr/api/rest/v1/swagger.html</a>	<a href="https://www.aspone.fr/api/rest/v1/swagger.html">https://www.aspone.fr/api/rest/v1/swagger.html</a>
JSON	Authentification	<a href="#">swagger_authentication.json</a>	<a href="#">swagger_authentication.json</a>
	CFN	<a href="#">swagger_cfn.json</a>	<a href="#">swagger_cfn.json</a>

Dans la suite du document, les URLs indiquées sont relatives à notre environnement de RECETTE.

➔ **Veillez nous contacter pour disposer d'un compte de TEST : [contact@asp-one.fr](mailto:contact@asp-one.fr)**

# Présentation Fonctionnelle

## ❑ Comptes utilisateurs :

Tous les comptes sont autorisés à utiliser les webservice, un abonnement au CFN est nécessaire afin d'utiliser cette fonctionnalité



### Administrateur

Ne dispose pas son propre coffre-fort numérique, gère seulement les coffres des utilisateurs



*Droits d'administration sur tous les comptes primaires / secondaires*



### Compte(s) primaire(s)

Disposent de leur propre coffre-fort numérique avec un forfait limitant la taille et le nombre de fichiers



*Droits d'administration sur leurs comptes secondaires*



### Compte(s) secondaire(s)

Disposent de leur propre coffre-fort numérique, leur consommation est incluse dans le forfait du compte primaire

# WS Authentication

## ❑ Présentation

La première étape pour utiliser les webservices consiste à récupérer 2 jetons :

- le premier « token » est nécessaire pour appeler les autres webservices, il est valide un temps limité (2h)
- le second « refreshToken » servira uniquement à obtenir un nouveau « token » via un appel différent, il a une date de validité beaucoup plus longue (1 an)

Une fois le jeton invalide, on peut en obtenir un nouveau soit via un nouvel appel du webservice accessToken soit via le webservice refreshToken

## ❑ Appel

2 méthodes existent pour obtenir les jetons, les 2 renvoient le même type de résultat

# WS Authentication

## ✓ Authentication/accessToken

Cette méthode permet d'obtenir un jeton à partir de ses identifiants, il faut appeler en mode POST l'URL :

<https://recette-aspone.tessitechno.fr/api/rest/v1/authentication/accessToken>

Au niveau du header on renseignera la clé « Authorization » avec nos identifiants encodés en base 64.

La clé est formée de la façon suivante :

Base 64 (*NOM\_MARQUE\_BLANCHE:LOGIN:MOT\_DE\_PASSE*)

Exemple : Base 64(ASPONE:compteTest:monMotDePasse)  
= QVNQT05FOmNvbXB0ZVRlc3Q6bW9uTW90RGVQYXNzZQ==

# WS Authentication

- ✓ Authentication/refreshToken

Cette méthode permet d'obtenir un jeton à partir du refreshToken, il faut appeler en mode POST l'URL :

<https://recette-aspone.tessitechno.fr/api/rest/v1/authentication/refreshToken>

Au niveau du header on renseignera la clé « Authorization » avec le refreshToken obtenu lors de l'appel de la méthode précédente.





**Un certains nombres d'appels sont disponibles au niveau des webservices associés au CFN, on va les distinguer en 5 groupes:**

- Contracts**  
Méthodes relatives aux forfaits
- Subscriptions**  
Méthodes relatives aux abonnements
- Documents**  
Méthodes relatives aux documents déposés
- Events**  
Méthodes relatives aux évènements
- Shares**  
Méthodes relatives aux partages des documents

# WS CFN : Contracts

**Il s'agit des méthodes permettant de lister les forfaits CFN disponibles pour le compte connecté.**

*Un compte admin peut gérer les abonnements des comptes qu'il administre.*

*Un compte primaire gère son propre forfait .*

*Un compte secondaire n'a pas de forfait, sa consommation est incluse dans le forfait de son compte primaire.*

## ✓ **cfn/contracts**

Méthode de récupération de la liste des forfaits compatibles avec le compte

On y retrouvera pour chacun la taille du coffre ou le nombre de fichier autorisés

*url [GET] : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/contracts>*

## ✓ **cfn/contracts/{contractId}**

Méthode de récupération du détail d'un forfait

*url [GET] : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/contracts/{contractId}>*

# WS CFN : Subscriptions

**Il s'agit des méthodes permettant de consulter / modifier les abonnements.**

*Un compte admin peut gérer les abonnements des comptes qu'il administre.*

*Un compte primaire gère son propre abonnement et/ou celui de ses comptes secondaires.*

*Un compte secondaire gère seulement son propre abonnement.*

✓ **cfn/subscriptions**

Méthode de récupération de la liste des abonnements en cours avec pour chacun d'eux le nombre de fichiers présents ainsi que la taille totale du coffre, le forfait associé sera également précisé

*url [GET] : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/subscriptions>*

✓ **cfn/subscriptions/{subscriptionId}**

Méthode de récupération du détail d'un abonnement

*url [GET] : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/subscriptions/{subscriptionId}>*

✓ **cfn/subscriptions**

Mise en place d'un abonnement pour le compte géré

*url [POST] : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/subscriptions>*

✓ **cfn/subscriptions/contract**

Changement de forfait lié à un abonnement

*url [PUT] : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/subscriptions/contracts>*

✓ **cfn/subscriptions/state**

Activation / désactivation de l'abonnement au CFN

*url [PUT] : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/subscriptions/state>*

# WS CFN : Documents 1/3

**Il s'agit des méthodes permettant de gérer les documents du CFN.**

*Un compte admin doit préciser le compte dont il souhaite gérer les documents.*

*Un compte primaire gère ses propres documents et/ou ceux de ses comptes secondaires.*

*Un compte secondaire gère seulement ses propres documents.*

✓ **cfn/documents**

Méthode de recherche des documents, pouvant également être utilisée pour obtenir l'identifiant du répertoire racine du CFN

*url [GET] : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/documents>*

✓ **cfn/documents/files**

Méthode de dépôt d'un fichier

*url [POST] : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/documents/files>*

✓ **cfn/documents/files/{documentId}**

Méthode de récupération d'un fichier

*url [GET] : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/documents/files/{documentId}>*

✓ **cfn/documents/files/{documentId}/conservation**

Méthode de modification de la durée de conservation d'un fichier

*url [PUT] : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/documents/files/{documentId}/conservation>*

# WS CFN : Documents 2/3

- ✓ **cfn/documents/folders**

Méthode de création d'un répertoire

*url [POST] : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/documents/folders>*

- ✓ **cfn/documents/folders/{folderId}**

Méthode de parcours d'un répertoire

*url [GET] : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/documents/folders/{folderId}>*

- ✓ **cfn/documents/{documentId}**

Méthode de visualisation du détail d'un document (fichier ou répertoire)

*url [GET] : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/documents/files/{documentId}>*

- ✓ **cfn/documents/{documentId}**

Méthode de suppression d'un document (fichier ou répertoire)

*url [DELETE] : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/documents/files/{documentId}>*

# WS CFN : Documents 3/3

## ✓ **cfn/documents/{documentId}/events**

Récupération de la liste des évènements d'un document (fichier ou répertoire)

url [GET] : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/documents/files/{documentId}/events>

## ✓ **cfn/documents/{documentId}/location**

Méthode de déplacement d'un document (fichier ou répertoire)

url [PUT] : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/documents/files/{documentId}/location>

## ✓ **cfn/documents/{documentId}/shares**

Récupération de la liste des partages d'un document (fichier ou répertoire)

url [GET] : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/documents/files/{documentId}/share>

## ✓ **cfn/documents/{documentId}/shares**

Mise en place d'un partage de document avec un tiers :

- si {documentId} correspond à un fichier, un téléchargement est proposé
- si {documentId} correspond à un répertoire, un dépôt de fichier est demandé dans ce répertoire

url [POST] : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/documents/files/{documentId}/share>

# WS CFN : Events

**Il s'agit des méthodes permettant de gérer les évènements.**

*Un compte admin doit préciser le compte client dont il souhaite visualiser les évènements.*

*Un compte primaire visualise ses propres évènements et/ou ceux de ses comptes secondaires.*

*Un compte secondaire gère seulement ses propres partages.*

✓ **cfn/events**

Méthode de recherche des évènements

url [GET] : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/events>



# WS CFN : Shares

**Il s'agit des méthodes permettant de gérer les partages.**

*Un compte admin doit préciser le compte client dont il souhaite gérer les partages.*

*Un compte primaire gère ses propres partages et/ou ceux de ses comptes secondaires.*

*Un compte secondaire gère seulement ses propres partages.*

✓ **cfn/shares**

Méthode de recherche des partages

url [GET] : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/share>

✓ **cfn/shares/{shareId}**

Méthode de récupération du détail d'un partage

url [GET] : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/share/{shareId}>

02/

# TESTS AVEC POSTMAN

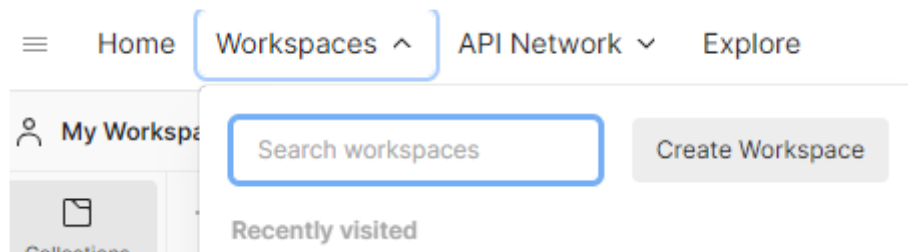
# Tests avec Postman

Postman est un client webservice universel qui permet de tester la plupart des Webservices

- ❑ **Télécharger Postman à l'adresse suivante :**

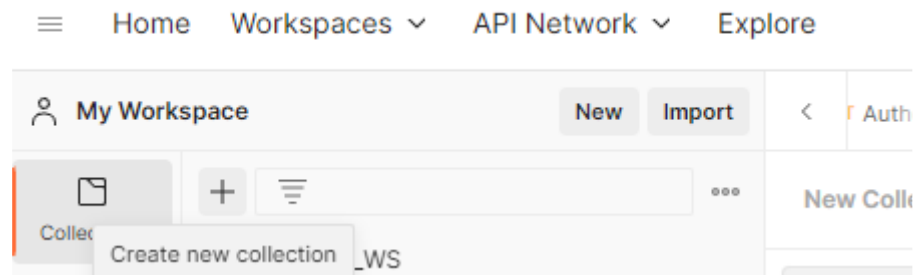
<https://www.postman.com>

- ❑ **Créer un nouveau workspace :**

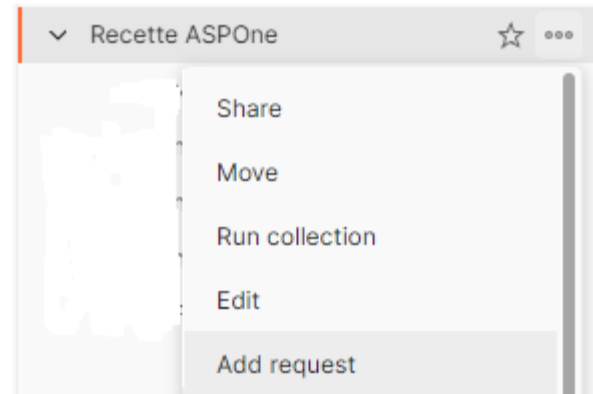


# Tests avec Postman

- ❑ Créer une nouvelle collection

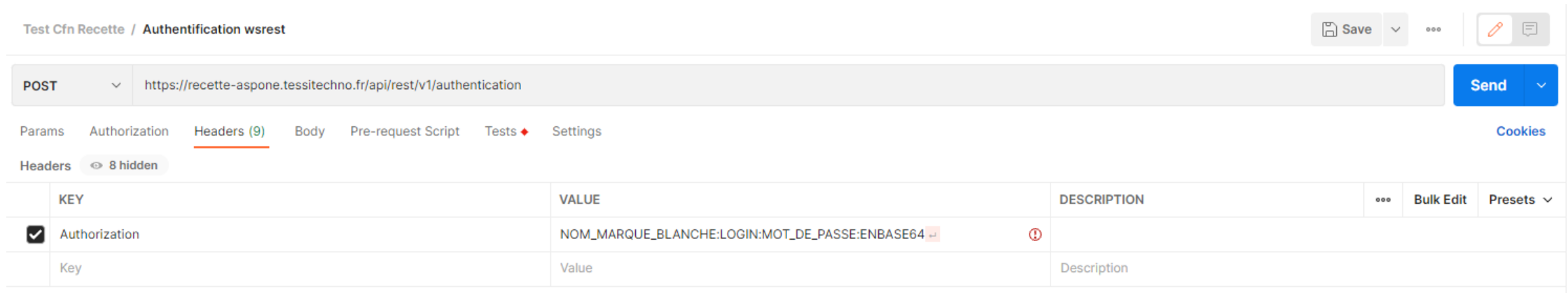


- ❑ Créer une nouvelle requête



# Tests avec Postman : Requête authentification 1/3

- ❑ La 1ère étape consiste à créer une requête d'authentification pour obtenir un jeton



The screenshot shows the Postman interface for a REST client. The request method is set to POST and the URL is `https://recette-aspone.tessitechno.fr/api/rest/v1/authentication`. The 'Headers' tab is selected, showing a table with the following content:

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/> Authorization	NOM_MARQUE_BLANCHE:LOGIN:MOT_DE_PASSE:ENBASE64				
Key	Value	Description			

- Préciser un nom : par exemple « Authentification WS REST »
- Indiquer l'URL : <https://recette-aspone.tessitechno.fr/api/rest/v1/authentication>
- Au niveau du header insérer une clé « Authorization » avec vos identifiants encodés en base 64 :

NOM\_MARQUE\_BLANCHE:LOGIN:MOT\_DE\_PASSE

- Indiquer que c'est une méthode de type « POST »
- Cliquer sur le bouton « Send »

# Tests avec Postman : Requête authentification 2/3

- ❑ La résultat de la requête apparait au niveau de la partie basse de l'écran

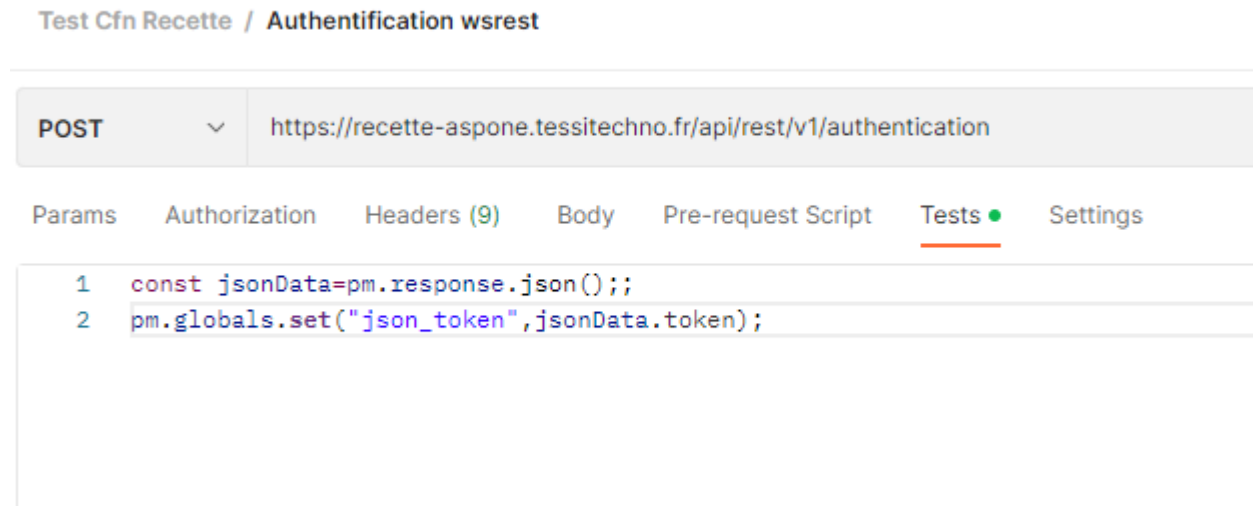


```
Body Cookies (1) Headers (12) Test Results
Status: 200 OK Time: 781 ms Size: 838 B Save Response
Pretty Raw Preview Visualize JSON
1
2 "token": "eyJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJQT1JUQUlMX0FTUE90RSIsImp0aSI6ImVlMDhiOTIxLTllYzctNGY0MS1iOWJkLTQ3NmFlNzM0MGQ2OSIsImh0I6I6MTY2NDI2NzY0MjY0e2NjQyNjY3OTksImkVXNlciI6NzAzNiwibG9naW4iOiJ3c3Jlc3R0ZXN0Iiwicm9sZSI6IkNPTVBURV9QUk1NQ1SRISIm1hcnF1ZUJ5YW5jaGUiOiJBU1BPTkUifQ.vodqq7HdCnoDhH0vsgMkw7LAqjbcLmk8CEv0n1vm-Xw",
3 "expirationDate": "20220927122959"
4
```

- ➔ On retrouve ainsi la valeur du jeton qu'il faudra insérer dans les autres requêtes
- ➔ La date d'expiration est également indiquée, il sera donc nécessaire d'appeler régulièrement cette requête pour régénérer un jeton

# Tests avec Postman : Requête authentification 3/3

- Afin de faciliter l'utilisation du jeton, il est possible de le sauvegarder dans une variable



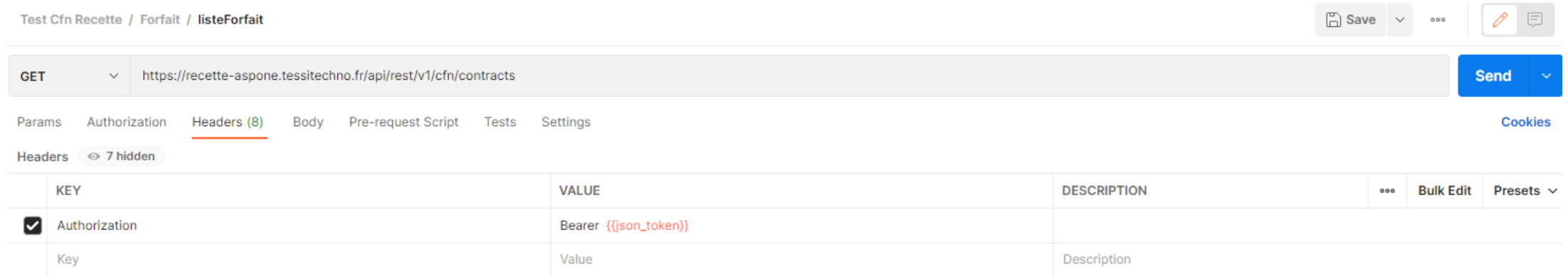
- Au niveau de l'onglet « Tests », renseigner les valeurs suivantes :

```
const jsonData=pm.response.json();
pm.globals.set("json_token",jsonData.token);
```

- A chaque appel de la requête d'authentification, le jeton sera sauvegardé dans la variable « json\_token »

# Tests avec Postman : Exemple d'appel de liste

## ❑ Création d'une requête permettant de lister les forfaits disponibles



The screenshot shows the Postman interface for a REST client. The request is a GET method to the URL `https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/contracts`. The 'Headers' tab is active, showing a table with one header: 'Authorization' with the value 'Bearer {{json\_token}}'. The 'Send' button is visible in the top right corner.

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/> Authorization	Bearer {{json_token}}				
Key	Value	Description			



- Préciser un nom: par exemple « listeForfait »
- Indiquer le type de méthode à utiliser selon ce que vous souhaitez faire : par exemple « GET »
- Indiquer l'URL : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/contracts>
- Au niveau du header, insérer une clé « Authorization » avec le jeton obtenu lors de l'authentification  
➔ On pourra utiliser ici la variable créée précédemment en renseignant : `Bearer {{json_token}}`
- Cliquer sur le bouton « Send »



# Tests avec Postman : Dépôt de fichier 1/3


## ❑ Création d'une requête permettant de déposer un document dans le CFN

Test Cfn Recette / Document / Dépôt

Save ▾ ...  

POST ▾ <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/documents/files> **Send** ▾

Params Authorization Headers (10) Body ● Pre-request Script Tests Settings [Cookies](#)

Headers  8 hidden

	KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets ▾
<input checked="" type="checkbox"/>	Authorization	Bearer {{json_token}}				
<input type="checkbox"/>	Content-Type	multipart/form-data				
	Key	Value	Description			

- Préciser un nom : par exemple « Dépôt »
- Indiquer le type de méthode à utiliser, dans ce cas il s'agit de « POST »
- Indiquer l'URL : <https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/documents/files>
- Au niveau du header, insérer une clé « Authorization » avec le jeton obtenu lors de l'authentification  
➔ On pourra utiliser ici la variable créée précédemment en renseignant : `Bearer {{json_token}}`

# Tests avec Postman : Dépôt de fichier 2/3

Test Cfn Recette / Document / Dépôt



POST ▼ https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/documents/files

Send ▼

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

Cookies

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	folderId	cef38576-f1df-4c10-a318-3986203ff597			
<input checked="" type="checkbox"/>	conservation	24			
<input checked="" type="checkbox"/>	keyword	mot clés			
<input checked="" type="checkbox"/>	fileName	fichierTest.xml			
<input checked="" type="checkbox"/>	content	settings.xml <span>×</span>			
	Key	Value	Description		

- Au niveau de l'onglet « Body »
  - Sélectionner « form-data »
  - Rajouter une clé « folderId » de type « text » en y insérant l'identifiant du répertoire de destination
  - Indiquer la durée de « conservation » souhaitée (3, 6, 9, 12, 18 mois, 2, 5, 10 ans)
  - Préciser éventuellement des mots clés via « keyword » pour faciliter les recherches ultérieures
  - Rajouter une clé « fileName » de type « text » en y précisant le nom du fichier qui devra apparaître sur l'interface
  - Rajouter ensuite « content » de type « file » et sélectionner le fichier à envoyer
- Cliquer sur le bouton « Send »

# Tests avec Postman : Dépôt de fichier 3/3

```
Body Cookies (1) Headers (14) Test Results
Pretty Raw Preview Visualize JSON ↕
1
2 "id": "3e7a2a1b-4097-4ebc-a706-dadd1b9af56e",
3 "parentFolderId": "cef38576-f1df-4c10-a318-3986203ff597",
4 "name": "fichierTest.xml",
5 "type": "FILE",
6 "deposit": "20221003093417",
7 "expiration": "20241003093417",
8 "format": "application/xml",
9 "owner": "asplent",
10 "depositor": "asplent",
11 "size": 24726,
12 "shares": 0,
13 "conservation": 24
14
```

- ➔ Une réponse positive du serveur (code 201) confirme la bonne prise en compte du dépôt
- ➔ L'attribut « id » constitue l'identifiant unique du document déposé

03/

# TESTS AVEC SWAGGER UI

# Tests avec Swagger UI

Swagger UI est une interface Open Source permettant de documenter et de tester nos webservices REST.

- ❑ Sur l'environnement de RECETTE, cette interface est accessible via l'URL ci-dessous :

<https://recette-aspone.tessitechno.fr/api/rest/v1/swagger.html>

## Contrat Interface API REST v1 (7.7.1.0)

[ Base URL: [recette-aspone.tessitechno.fr/api/rest](https://recette-aspone.tessitechno.fr/api/rest) ]  
[/api/rest/v1/swagger.json](#)

### WS/AUTHENTICATION ^

**POST** [/{version}/authentication](#) Obtention d'un jeton ∨

### WS/CFN/CONTRACTS ^

**GET** [/{version}/cfn/contracts](#) Liste des forfaits ∨

**GET** [/{version}/cfn/contracts/{contractId}](#) Détail d'un forfait ∨

### WS/CFN/DOCUMENTS ^

**GET** [/{version}/cfn/documents](#) Recherche documents ∨

**POST** [/{version}/cfn/documents/files](#) Dépôt fichier ∨

**GET** [/{version}/cfn/documents/files/{documentId}](#) Téléchargement fichier ∨

**PUT** [/{version}/cfn/documents/files/{fileId}/conservation](#) Modification conservation fichier ∨

**POST** [/{version}/cfn/documents/folders](#) Creation répertoire ∨

# Tests avec Swagger UI : Authentification 1/3

- ❑ Pour utiliser la méthode d'authentification, déplier le webservice « Authentication » :

The screenshot shows the Swagger UI interface for the authentication endpoint. The endpoint is a POST request to `/{version}/authentication` with the description "Obtention d'un jeton". The method is described as "Méthode de récupération d'un jeton valable 600000 ms".

**Parameters**

Name	Description
Authorization string (header)	Authorization
version * required string (path)	Default value : v1

**Responses**

Response content type: `application/json`

Code	Description
200	Jeton

Example Value | Model

```
{  "token": "string",  "expirationDate": "string"}
```

➔ Cliquer ensuite sur le bouton « Try it out »

# Tests avec Swagger UI : Authentification 2/3

➔ Renseigner le champ « Authorization » avec ses identifiants encodés en base 64 :

BASE64(NOM\_MARQUE\_BLANCHE:LOGIN:MOT\_DE\_PASSE)

## WS/AUTHENTICATION

POST /{version}/authentication Obtention d'un jeton

Méthode de récupération d'un jeton valable 600000 ms

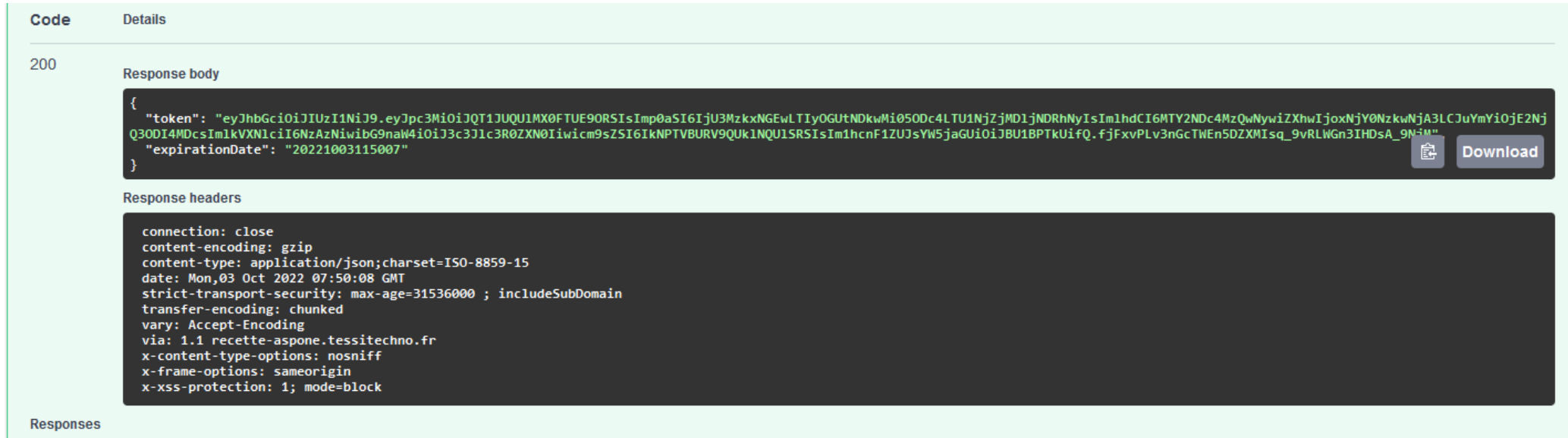
Parameters Cancel

Name	Description
Authorization string (header)	Tk9NX01BUIFVRV9CTEFOQ0hFOkxPR0IOO
version * required string (path)	v1

Execute

➔ Cliquer sur « Execute »

# Tests avec Swagger UI : Authentification 3/3



The screenshot displays the Swagger UI interface for a REST API. It shows a successful 200 response with a JSON body containing a token and an expiration date. The response headers are also visible, including content-type, date, and security-related headers.

**Code**    **Details**

200

**Response body**

```
{
  "token": "eyJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJQT1JUQU1MX0FTUE90RSIsImp0aSI6IjU3MzZkxNGEwLTIyOGUtNDkwMi05ODc4LTU1NjZjMD1jNDRhNyIsIm1hdCI6MTY2NDc4MzQwNywiZXhwIjoxNjY0NzkwNjA3LCJmYmYiOiJlE2NjQ3ODI4MDcsImkVXN1ciI6NzAzNiwiYm9naW4iOiJ3c3Jlc3R0ZXN0Iiwicm9sZSI6IkpNTVBURV9QUk1NQ1SR5Im1hcnF1ZUJsYW5jaGUiOiJBU1BPTkUifQ.fjFzvPLv3nGcTWEn5DZXMIsq_9vRLWgn3IHDsA_9N-iM",
  "expirationDate": "20221003115007"
}
```

**Response headers**

```
connection: close
content-encoding: gzip
content-type: application/json;charset=ISO-8859-15
date: Mon,03 Oct 2022 07:50:08 GMT
strict-transport-security: max-age=31536000 ; includeSubDomain
transfer-encoding: chunked
vary: Accept-Encoding
via: 1.1 recette-aspone.tessitechno.fr
x-content-type-options: nosniff
x-frame-options: sameorigin
x-xss-protection: 1; mode=block
```

**Responses**

➔ Le jeton retourné par le serveur (permettant d'utiliser les autres méthodes de l'API) apparaît dans la section « Responses »



# Tests avec Swagger UI : recherche 1/3

- ❑ Pour rechercher un document, déplier le webservice « Documents » :

WS/CFN/DOCUMENTS ^

GET /{version}/cfn/documents Recherche documents ^

Méthode de recherche de documents (fichiers ou répertoires)

Parameters Try it out

Name	Description
<b>Authorization</b> * required	
string (header)	Jeton de connexion <i>Example</i> : Bearer [valeurDuJetonTransmis]
	<input type="text" value="Bearer [valeurDuJetonTransmis]"/>
login string (header)	Login du compte pour lequel on souhaite obtenir les documents (obligatoire pour Admin)
	<input type="text" value="login"/>
<b>version</b> * required	
string (path)	Numéro de version des WebServices <i>Default value</i> : v1 <i>Example</i> : v1
	<input type="text" value="v1"/>
nbDayExpired integer(\$int32) (query)	<input type="text" value="nbDayExpired"/>
dateBegin string (query)	Date de depot - DEBUT : format :yyyyMMddHHmmss
	<input type="text" value="dateBegin"/>
dateEnd	

➔ Cliquer ensuite sur le bouton « Try it out »

# Tests avec Swagger UI : recherche 2/3

- ➔ Renseigner le champ « Authorization » avec le jeton retourné par la méthode d'authentification ainsi que les critères de recherche souhaités :

dans l'exemple ci-dessous, le but est de rechercher tous les dossiers :

**WS/CFN/DOCUMENTS** ^

GET `/{version}/cfn/documents` Recherche documents ^

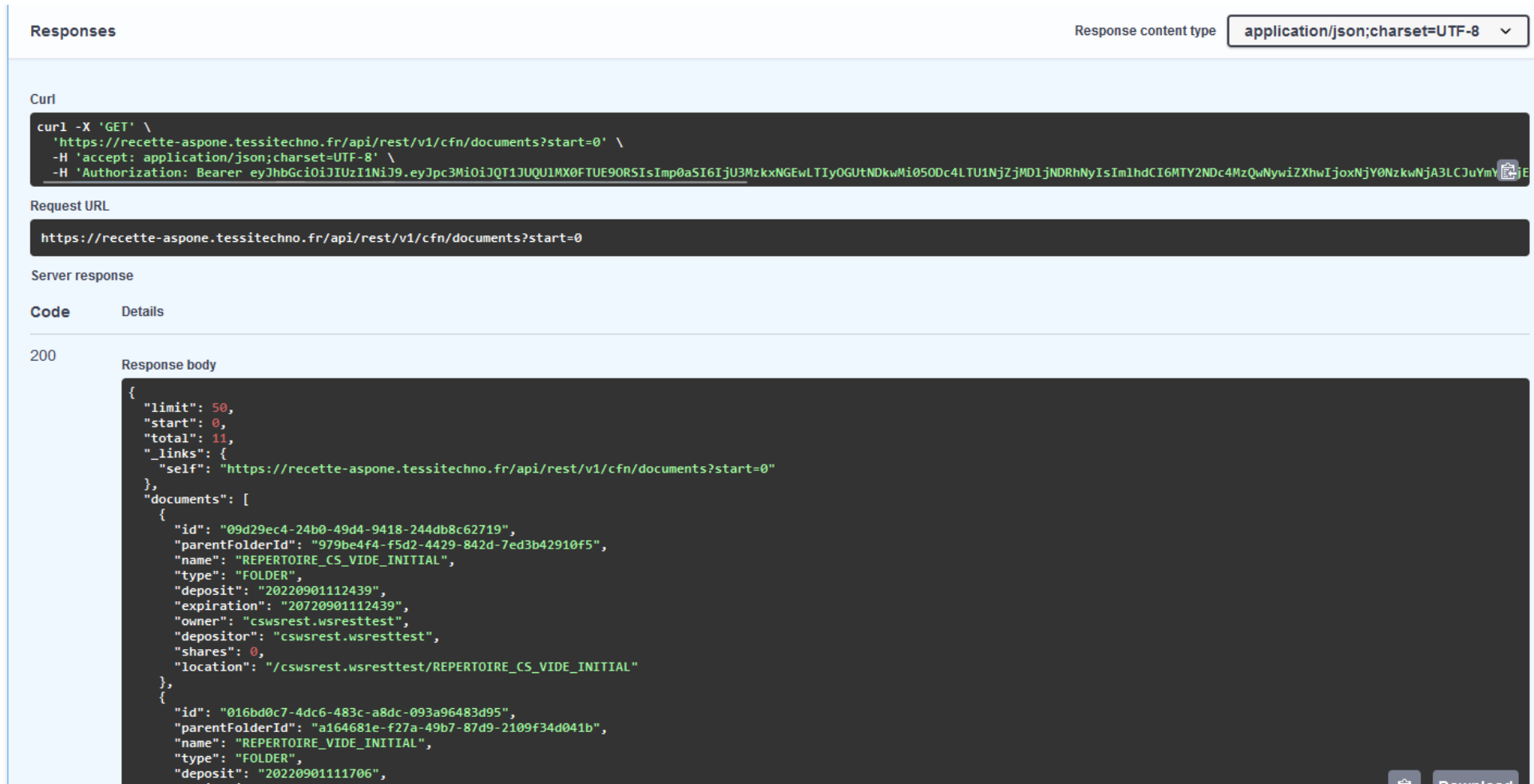
Méthode de recherche de documents (fichiers ou répertoires)

**Parameters** Cancel

Name	Description
<b>Authorization</b> * required string (header)	Jeton de connexion
login string (header)	Login du compte pour lequel on souhaite obtenir les documents (obligatoire pour Admin)
type string (query)	Type de document : FOLDER, FILE ou vide

- ➔ Cliquer sur « Execute »

# Tests avec Swagger UI : recherche 3/3



Responses Response content type `application/json;charset=UTF-8`

Curl

```
curl -X 'GET' \
  'https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/documents?start=0' \
  -H 'accept: application/json;charset=UTF-8' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJQT1JUQU1MX0FUE90RS1sImp0aSI6IjU3MzZkxkNGEwLTlYOGU0tNDkwMi05ODc4LTU1NjZjMD1jNDRhNyIsIm1hdCI6MTY2NDc4MzQwNywiZXRhIjoxNjY0NzkwNjA3LCJmYm94IjE
```

Request URL

```
https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/documents?start=0
```

Server response

Code	Details
200	<p>Response body</p> <pre>{   "limit": 50,   "start": 0,   "total": 11,   "_links": {     "self": "https://recette-aspone.tessitechno.fr/api/rest/v1/cfn/documents?start=0"   },   "documents": [     {       "id": "09d29ec4-24b0-49d4-9418-244db8c62719",       "parentFolderId": "979be4f4-f5d2-4429-842d-7ed3b42910f5",       "name": "REPertoire_CS_VIDE_INITIAL",       "type": "FOLDER",       "deposit": "20220901112439",       "expiration": "20720901112439",       "owner": "cswsrest.wsresttest",       "depositor": "cswsrest.wsresttest",       "shares": 0,       "location": "/cswsrest.wsresttest/REPertoire_CS_VIDE_INITIAL"     },     {       "id": "016bd0c7-4dc6-483c-a8dc-093a96483d95",       "parentFolderId": "a164681e-f27a-49b7-87d9-2109f34d041b",       "name": "REPertoire_VIDE_INITIAL",       "type": "FOLDER",       "deposit": "20220901111706",       "expiration": "20720901111706",       "owner": "cswsrest.wsresttest",       "depositor": "cswsrest.wsresttest",       "shares": 0,       "location": "/cswsrest.wsresttest/REPertoire_VIDE_INITIAL"     }   ] }</pre>

➔ La réponse du serveur retourne la liste des documents qui correspondent aux critères précisés, dans notre exemple il s'agit de la liste des dossiers.

# tessi

