Intégration des solution via Swagger

CAP ADRESSE – 2 RUE DE MALLEVILLE, 95880 ENGHIEN LES BAINS - Tél. 01.84.17.99.99
SARL capital social 100 000 euros – RCS Pontoise n°500 283 601 – Code NAF 5829C
N°TVA INTRA COMMUNAUTAIRE FR 79500283601

SOMMAIRE

CAP ADRESSE – 2 RUE DE MALLEVILLE, 95880 ENGHIEN LES BAINS - Tél. 01.84.17.99.99
SARL capital social 100 000 euros – RCS Pontoise n°500 283 601 – Code NAF 5829C
N°TVA INTRA COMMUNAUTAIRE FR 79500283601

PRESENTATION

Afin d'éditer ou de créer une application en Swagger, il faut utiliser l'outil en ligne disponible à cette adresse :https://editor.swagger.io/.

Pour réaliser des tests de l'application, il est nécessaire d'avoir au minimum l'un des deux, soit NodeJS ou serveur local PHP.

1. FICHIER YAML

Exemple type (monfichier.yaml) :

```
swagger: "2.0"
info:
  title: "Nom de la solution ex: CAP Email"
  description: "description du service."
  version: "1.0.0"
  contact:
    email: "support@capadresse.com"
host: "url du service ex:dev.capadresse.com:7003"
schemes:
  - "http"
  - "https"
paths:
  /NomDeLaFonction:
    get:
      summary: "descriptif succinct de la fonction"
      description: "descriptif complet de la fonction."
      consumes:
      - "application/json"
      produces:
      - "application/json"
      parameters:
      - name: "request"
        in: "query"
        required: true
        type: "string"
        enum:
          - "NomDeLaFonction"
        default: " NomDeLaFonction"
    responses:
        200:
          description: "Succès"
          schema:
            $ref: "#/definitions/NomDeLaFonctionResponse"
definitions:
  NomDeLaFonctionResponse:
    type: "object"
    properties:
      response:
        type: "string"
        description: "Le nom de la requête appelée."
        xml:
          name: "response"
```


CAP ADRESSE – 2 RUE DE MALLEVILLE, 95880 ENGHIEN LES BAINS - Tél. 01.84.17.99.99
SARL capital social 100 000 euros – RCS Pontoise n°500 283 601 – Code NAF 5829C
N°TVA INTRA COMMUNAUTAIRE FR 79500283601

Dans l'exemple au-dessus, la fonction ne possède aucun paramètre d'entré et de sortie sauf ceux obligatoire pour une requête JSON « request » en entré et « response » en sortie.

1.1 VARIABLES EN ENTREE

Les paramètres d'entré s'ajoutent après la clé « parameters », la déclaration se fait comme ci-dessous :

```
- name: "maVariable"
  in: "query"
  required: true
  type: "string"
  decription: "description sur maVariable"
```

La valeur de la clé « in » est toujours égale à query, si le nouvel attribut en entré est obligatoire on peut insérer la clé required et l'initialisé à true.
Sinon il n'est pas nécessaire de l'ajouter, passé directement à la clé type.

```
- name: "maVariable"
  in: "query"
  required: true
  type: "integer"
  format: "int32"
  decription: "description sur maVariable"
```

L'attribut type peut recevoir les valeurs suivantes :

- Integer
- String

Dans le cas de l'entier, on peut spécifier la taille via la clé format mais optionnel.

On peut ajouter l'élément « enum » si le paramètre d'entré peut recevoir qu'un choix limité de valeurs, mais pas obligatoire.

1.2 VARIABLES EN SORTIE

Les paramètres d'entré s'ajoutent après la clé « properties », la déclaration se fait comme ci-dessous :

```
maVariable
  type: "string"
  decription: "description sur maVariable"
  xml:
    name: "maVariable"
```


CAP ADRESSE – 2 RUE DE MALLEVILLE, 95880 ENGHIEN LES BAINS - Tél. 01.84.17.99.99
SARL capital social 100 000 euros – RCS Pontoise n°500 283 601 – Code NAF 5829C
N°TVA INTRA COMMUNAUTAIRE FR 79500283601

L'attribut type peut recevoir les valeurs suivantes :

- Integer
- String
- Array
- Object

```
 maVariable
    type: "string"
    decription: "description sur maVariable"
    xml:
       name: "maVariable"
 maVariableEntier:
    type: "integer"
    format: "int32"
    description: "description sur maVariableEntier."
    xml:
      name: "iRet"
  monTableau:
    type: "array"
    description: "decription de monTableau."
    xml:
      name: "monTableau"
      wrapped: true
    items:
      $ref: "#/definitions/monTableau"
```

Dans le cas où le type a pour valeur « array » ou « object », il est nécessaire d'ajouter la clé « items » afin de préciser la structure de l'objet ou du tableau ainsi que l'élément « wrapped » à true.

La valeur de la clé $ref pointe vers la structure de « monTableau », voir code ci-dessous. Cette partie du code est ajouté à la suite de la clé « definitions ».
L'élément « monTableau » doit avoir la même indentation que « NomDeLaFonctionResponse » issue de l'exemple type.

```
monTableau:
  type: "object"
  properties:
    maVariable:
      type: "string"
  xml:
    name: « maVariable"
```

2 TEST JAVASCRIPT

Sur la page de l'éditeur en ligne, onglet « Generate Client » choisir javascript.

3 TEST PHP

Sur la page de l'éditeur en ligne, onglet « Generate Client » choisir php.

CAP ADRESSE – 2 RUE DE MALLEVILLE, 95880 ENGHIEN LES BAINS - Tél. 01.84.17.99.99
SARL capital social 100 000 euros – RCS Pontoise n°500 283 601 – Code NAF 5829C
N°TVA INTRA COMMUNAUTAIRE FR 79500283601

CAP ADRESSE
2 RUE DE MALLEVILLE
95880 ENGHIEN LES BAINS

Standard : 01 84 17 99 99
www.capadresse.com
info@capadresse.com