

# FORMATION : USAGE DES API PUBLIK

EMMANUEL CAZENAVE

METZ

09/11/2023



- 1 Introduction
- 2 Source de données
- 3 Appel Webservice (workflows)
- 4 Appel Webservice (formulaires)
- 5 API
  - API démarches
  - API connexion

Ne pas oublier de lancer l'enregistrement !

# INTRODUCTION

- Pas un logiciel mais plusieurs briques
- Écrites en Python/Django
  - ▶ Mais ce n'est pas important !
- Reliées entre elles via
  - ▶ API webservice
  - ▶ Provisionning
  - ▶ SSO

# Formation : usage des API Publiques

## └ Introduction

## └ Vue d'ensemble

- Pas un logiciel mais plusieurs briques
- Écrites en Python/Django
  - ▶ Mais ce n'est pas important !
- Reliées entre elles via
  - ▶ API webservices
  - ▶ Provisionning
  - ▶ SSO

1. Passer par les API permet justement d'utiliser son langage favori
2. Et ouais, le consommateur numéro 1 des API Publiques c'est Publiques.

- w.c.s. : démarches
- authentic : connexion
- combo : portail
- passerelle : connexion
- chrono : prise de rendez-vous

## └ Introduction

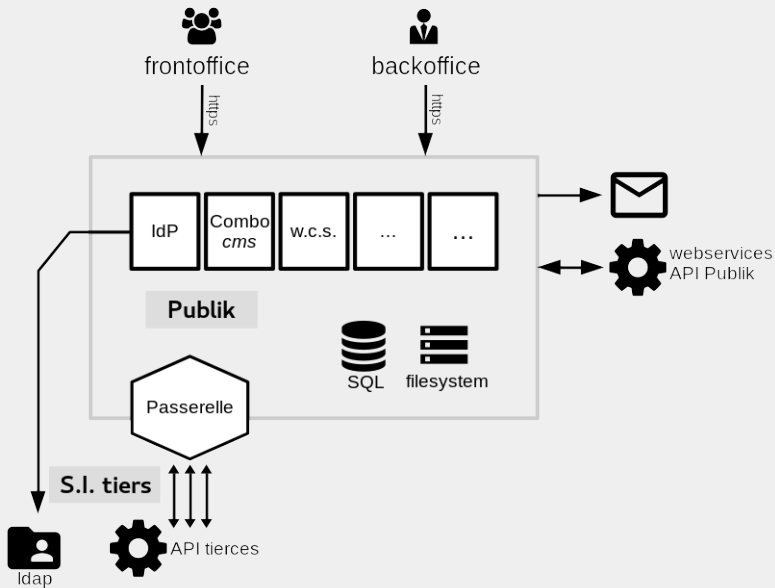
## └ Les briques et leurs fonctions

- w.c.s. : démarches
- authentic : connexion
- combo : portail
- passerelle : connexion
- chrono : prise de rendez-vous

1. On va se concentrer sur wcs.
2. Et parler un peu de l'API authentic.
3. Éventuellement un laïus sur à quel point c'est nul les connecteurs + pas le but de cette formation.
4. L'API chrono est foisonnante mais on va pas en parler sauf cas d'usage spécifique.



# SCHÉMA GÉNÉRAL



<https://doc-publik.entrouvert.com/dev/connecteurs/>



# PRÉSENTATION

- Objectif : récupérer des informations depuis un SI externe
- Dictionnaire JSON
- Structure très simple

```
{  
  "err": 0,  
  "data": [  
    {"id": "1", "text": "rue du Château"},  
    {"id": "2", "text": "avenue du Maine"},  
  ]  
}
```

# Formation : usage des API Publiques

└ Source de données

└ Présentation

- Objectif : récupérer des informations depuis un SI externe
- Dictionnaire JSON
- Structure très simple

```
{  
  "err": 0,  
  "data": [  
    {"id": "1", "text": "rue du Château"},  
    {"id": "2", "text": "avenue du Maine"},  
  ]  
}
```

## 1. Rendre une API compatible Publiques, c'est facile !

- Alimentation d'un champ liste<sup>1</sup>
- Cas d'usage avancé : autocomplétion<sup>2</sup>
  - Requis quand il y a beaucoup de données

---

<sup>1</sup>[https://doc-publik.entrouvert.com/admin-fonctionnel/fabrique-formulaires/form-champs/champ\\_liste/](https://doc-publik.entrouvert.com/admin-fonctionnel/fabrique-formulaires/form-champs/champ_liste/)

<sup>2</sup><https://doc-publik.entrouvert.com/admin-fonctionnel/les-tutos/utiliser-l-autocompletion/>

# Formation : usage des API Publik

## └ Source de données

## └ Utilisation dans une démarche

- Alimentation d'un champ liste\*
- Cas d'usage avancé : autocomplétion\*
  - ▶ Requis quand il y a beaucoup de données

\*<https://publik.ansreseau.fr/adm/ Fonctionnel/Fabrique/Parametres/Param-champs/champ-liste/>

\*<https://publik.ansreseau.fr/adm/ Fonctionnel/Param-champs/Utiliser-l-autocomplétion/>

1. Exemple à base d'etherpad, genre <https://pad.inpt.fr/p/test-ws>
2. Exemple à partir d'un connecteur passerelle existant

# **APPEL WEBSERVICE (WORKFLOWS)**



# PRÉSENTATION

- Objectif : envoyer des données
  - Toujours en JSON
- Et en récupérer (format plus souple que pour les sources de données)

```
{  
  "err": 0,  
  "data": {  
    "xxx": "yyy",  
    "details": {  
      "aaa": "bbb"  
    }  
  }  
}
```

# Formation : usage des API Publiques

## └ Appel Webservice (workflows)

### └ Présentation

#### PRÉSENTATION

- Objectif : envoyer des données
  - Toujours en JSON
- Et en récupérer (format plus souple que pour les sources de données)

```
{  
  "err": 0,  
  "data": {  
    "xxx": "yyy",  
    "details": {  
      "aaa": "bbb"  
    }  
  }  
}
```

## 1. Exemple à base de <https://requestbin.com>

## ET LES ERREURS ?

```
{  
  "err": 1,  
  "err_class": "TypeError",  
  "err_desc": "name must be a string",  
  "data": null  
}
```

- Action « Webservice »<sup>1</sup>
- Envoyer les données de la demande

---

<sup>1</sup>[https://doc-publik.entrouvert.com/admin-fonctionnel/fabrique-de-workflows/les-actions-de-workflow/elements\\_appeler-webservices/](https://doc-publik.entrouvert.com/admin-fonctionnel/fabrique-de-workflows/les-actions-de-workflow/elements_appeler-webservices/)



- Déclaration dans les paramètres
- Condition de sortie de page, champ commentaire, préremplissage...

**API**

- Création d'un client d'API<sup>1</sup>
- Utilisation via HTTP Basic

---

<sup>1</sup><https://doc-publik.entrouvert.com/dev/wcs/api-webservices/authentication/>



## **API DÉMARCHES**

- Déclencheur d'une action à Saut automatique <sup>1</sup>
  - Ou d'une action globale
- Appel authentifié en POST vers l'adresse du formulaire à traiter <sup>2</sup>

---

<sup>1</sup>[https://doc-publik.entrouvert.com/admin-fonctionnel/fabrique-de-workflows/les-actions-de-workflow/actions\\_transition-automatique/](https://doc-publik.entrouvert.com/admin-fonctionnel/fabrique-de-workflows/les-actions-de-workflow/actions_transition-automatique/)

<sup>2</sup><https://doc-publik.entrouvert.com/dev/wcs/api-webservices/traitement-d-un-formulaire/>

- w.c.s. expose une large API
  - ▶ Récupérer la liste de tous les formulaires
  - ▶ Récupérer les données d'un formulaire
  - ▶ Compléter un formulaire
- Documentation : <https://doc-publik.entrouvert.com/dev/wcs/api-webservices/>

# Formation : usage des API Publik

└ API

└ API démarches

└ Web services exposés

- w.c.s. expose une large API
  - Récupérer la liste de tous les formulaires
  - Récupérer les données d'un formulaire
  - Compléter un formulaire
- Documentation : <https://doc-publik.entrouvert.com/dev/wcs/api-webservices/>

1. Souligner que c'est une approche alternative, push vs pull, tout ça.

## **API CONNEXION**

- authentic expose une large API
  - ▶ Lister et rechercher des utilisateurs
  - ▶ Créer ou supprimer un utilisateur
  - ▶ Agir sur les rôles
- Documentation : <https://doc-publik.entrouvert.com/dev/api-authentic/>

FIN !