

FORMATION : USAGE DES API PUBLIK

VALENTIN DENIAUD

ACADÉMIE DE VERSAILLES

29/06/2021



- 1 Introduction
- 2 Source de données
- 3 Appel Webservice
- 4 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

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

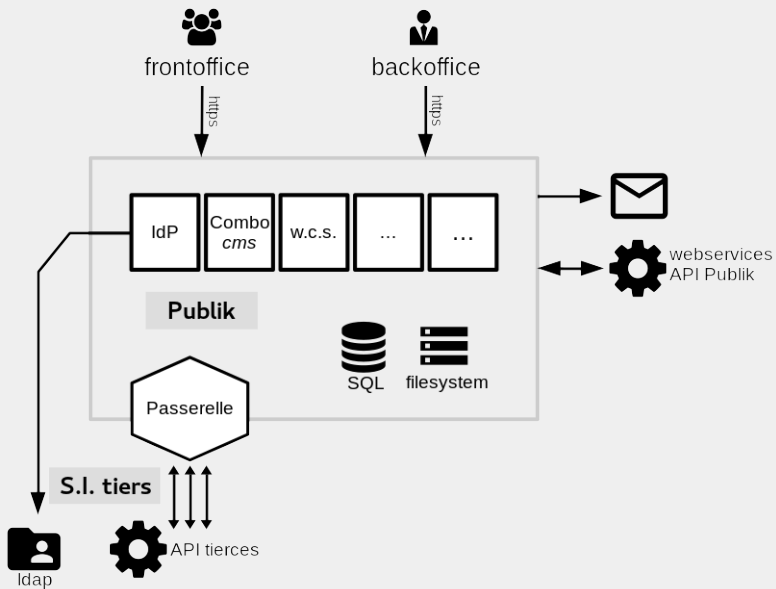
1. Passer par les API permet justement d'utiliser son langage favori
2. Et ouais, le consommateur numéro 1 des API Publik c'est Publik.

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

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

1. On va se concentrer sur w.c.s.
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/>

SOURCE DE DONNÉES

PRÉSENTATION

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

```
{  
  "err": 0,  
  "data": {  
    "text": "Donnée 1",  
    "id": 1,  
    "xxx": "clé supplémentaire"  
  }  
}
```

Formation : usage des API Publik

└ Source de données

└ Présentation

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

```
{  
  "err": 0,  
  "data": {  
    "text": "Donnée 1",  
    "id": 1,  
    "xxx": "clé supplémentaire"  
  }  
}
```

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

ET LES ERREURS ?

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

- Alimentation d'un champ liste¹
- Cas d'usage avancé : autocomplétion²
 - ▶ Requis quand il y a beaucoup de données

¹https://doc-publik.entrouvert.com/admin-fonctionnel/fabrique-formulaires/form-champs/champ_liste/

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

└ 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://pad.inpt.fr/entraineur/.../tabler=Recherche/Rechercher/Recherche/Recherche/ champ_liste/

*https://pad.inpt.fr/entraineur/.../tabler=Recherche/Rechercher/Recherche/Recherche/ champ_liste/

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

- Cellule JSON¹
- Utilisation du langage de template Django²
- Des filtres supplémentaires sont disponibles³

¹https://doc-publik.entrouvert.com/admin-fonctionnel/fabrique-formulaires/form-champs/champ_liste/

²<https://docs.djangoproject.com/fr/2.2/ref/templates/language/>

³<https://doc-publik.entrouvert.com/admin-fonctionnel/parametrage-avance/utiliser-les-filtres/>

Formation : usage des API PubliK

└ Source de données

└ Utilisation dans le portail

- Cellule JSON¹
- Utilisation du langage de template Django²
- Des filtres supplémentaires sont disponibles³

¹<https://docs.djangoproject.com/fr/2.2/ref/templates/library/#django.template.defaultfilters.TemplateStringFilter>

²<https://docs.djangoproject.com/fr/2.2/ref/templates/language/>

³<https://docs.djangoproject.com/fr/2.2/ref/templates/language/#django.template.defaultfilters.TemplateStringFilter>

1. Exemple à base d'etherpad, bis
2. Ne pas trop s'apesantir parce que relou pour nous d'intégrer ça.

APPEL WEBSERVICE

- Objectif : envoyer des données
 - ▶ Toujours en JSON
- Et en récupérer (même format que les sources de données)

- Objectif : envoyer des données
 - ▶ Toujours en JSON
- Et en récupérer (même format que les sources de données)

1. Exemple à base de `https://requestbin.com`

- Dans les workflows
 - ▶ Action « Webservice »¹
 - ▶ Envoyer les données de la demande
- Dans les formulaires
 - ▶ Déclaration dans les paramètres
 - ▶ Condition de sortie de page, champ commentaire...

¹https://doc-publik.entrouvert.com/admin-fonctionnel/fabrication-de-workflows/les-actions-de-workflow/elements_appeler-webservices/

API

- Authentification HTTP Basic, à privilégier
- Authentification par signature ¹

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

API

API DÉMARCHES

- Déclencheur d'une action « Saut automatique »¹
 - ▶ Ou d'une action globale
- Appel authentifié en POST vers l'adresse du formulaire à traiter²

¹https://doc-publik.entrouvert.com/admin-fonctionnel/fabrique-de-workflows/les-actions-de-workflow/actions_transition-automatique/

²<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-public.entrouvert.com/dev/wcs/api-webservices/`

└ 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

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 !