

The background is a soft, watercolor-style illustration. It features various green leaves and foliage. In the top left, there are large, dark green leaves with prominent veins. In the top right, there are smaller, lighter green leaves. In the bottom left, there are more dark green leaves. In the bottom right, there are large, dark green leaves with white, irregular shapes cut out of them, resembling Monstera leaves. The overall color palette is a range of greens, from light and airy to deep and saturated, creating a natural, organic feel.

ADH6

Par liteapp & π (airopi)

KESKECÉ ?

- Gérer les adhésions et les cotisations
- Gérer les appareils des membres (attribution d'adresses IP)
- Gérer les chambres et leurs attributions
- Gérer le réseau (switches, ports, VLANs)
- Assurer le suivi de la trésorerie

The screenshot displays the 'dh6' user interface. On the left is a sidebar menu with categories: GENERAL (Mon Compte), ADMINISTRATION (Gestion Adhérents, Adhérents, Appareils, Chambres), Gestion réseau (Commutateurs, Ports, Commutateur Local), and Authentification (NainA, Clef d'API). The main content area features the 'dh6' logo, navigation links for 'Mes appareils' and 'Mon compte', and a status bar indicating 'Votre abonnement est actif jusqu'au' followed by a progress bar. Below this are two buttons: 'Renouveler mon abonnement' and 'Changer le mot de passe'. A 'Listes de diffusion' section contains three checked items: 'MiNET' (technical problems and association information), 'Hosting' (technical problems and complementary information for hosting users), and 'Mini-routeurs' (technical information and updates for mini-router users). At the bottom, there are input fields for 'mail', 'Votre adresse IPv4 publique est', and 'Votre sous-réseau IPv4 privé est'.



ADH = ADHÉRENTS!

SOURCE: JE CROIS + 30ANS.MINET.NET

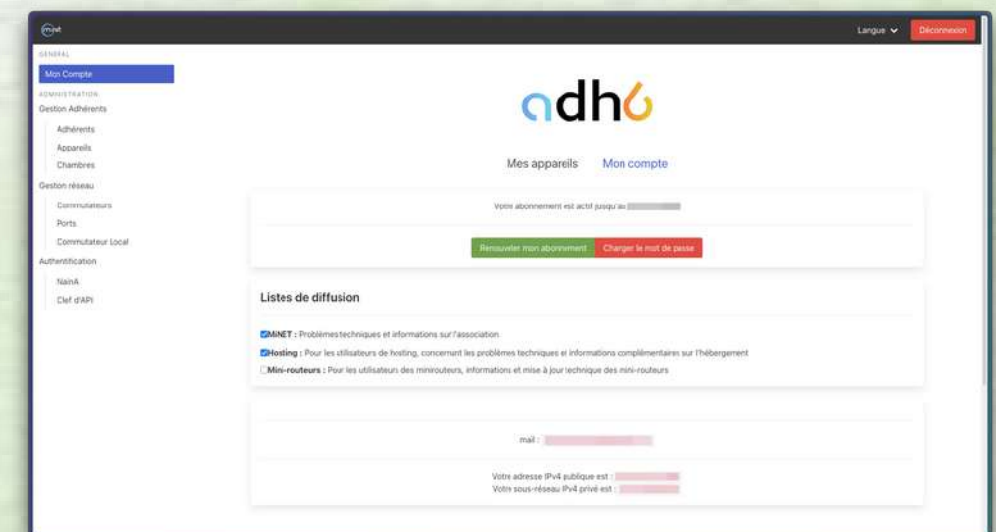
UN PEU D'HISTOIRE

"Stats adhérents via la plateforme ADH"

Appart U6: 6400
Appart Gardien U2: 2125
Appart Gardien U3: 3063

Statistiques

NOM	VALEUR
nb_users	1673
nb_preinscrits	0
ip_U1	126
ip_U2	150
ip_U3	60
ip_U4	94
ip_U5	172
ip_U6	244
ip_assos	120
clef_bf	458



adh1
(*askip*)

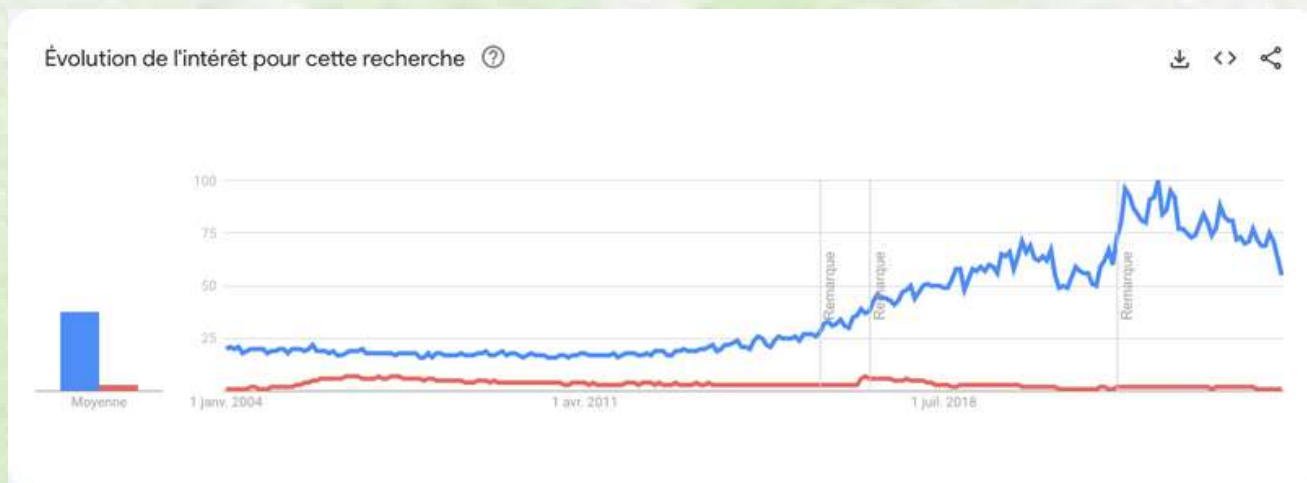
adh 2 & 3
(chill dans les
backrooms)

adh4
(pas beau)

adh5
(stylé !)

adh6
(plus stylé !)

POURKOI ADH6(66) ?



python vs ruby, de 2004 à mai 2025

From: Anthony VEREZ <netantho@minet.net> ☆
Subject: [equipe] ADH666
To: equipe@listes.minet.net ★

Salut,

Bon avant de lâcher le troll de l'année :
ADH5 est déjà très bien foutu. La qualité de code est très
impressionnante, tu as fais un boulot formidable Guillaume.

L'idée d'ADH6(66) vient plutôt du fait que l'équipe actuelle n'ose pas
toucher au code, même pour corriger les bugs. Ce qui est très gênant sur
le moyen/long terme.
J'ai essayé de corriger des bugs, apporter quelques améliorations
mineures mais je n'ai clairement pas le temps à moi seul (ou alors il
faut que je me décharge d'autres tâches dans le même cas ...) pour me
consacrer à ADH5.

6/16/12, 3:07 AM

thread Re: [equipe] réunion ADH6

Anatole Lefort

10 Jun 2016 12:16 p.m.

Salut,

Sur ton framadate, tu proposes 3 dates, dont 2 qui sont déjà passées :p

On 06/10/2016 02:11 PM, capolrik wrote:

Search this list

Les stagiaires MINET de cet été vont commencer à bosser sur adh6.
C'est un projet qui demande de la préparation et un discussion sur le
cahier des charge, et d'autres aspects techniques.

Pour trouvez une date, on remplit le framadate suivant SVP:
<https://framadate.org/oQ62qZh1st5m9yn5>

us avez des conseils, des erreurs à ne pas commettre, n'hésitez pas
donner sur ce pad:
https://imagine.minet.net/pad/p/Commentaires_sur_ADH6

d'avance à tous

Ce n'est pas une nouvelle idée ! Évoquée
en 2012 et commencée en 2016.

KAHIER DÉ CHARGES

- Application ultra-modulaire
- Backend et frontend séparés
(parce qu'on a dit MODULAIRE)
- Fiable
(on commente pas les tests quand ça marche pas)
- Simple
(“on réinvente pas la roue” même si ça reste à démontrer parfois)



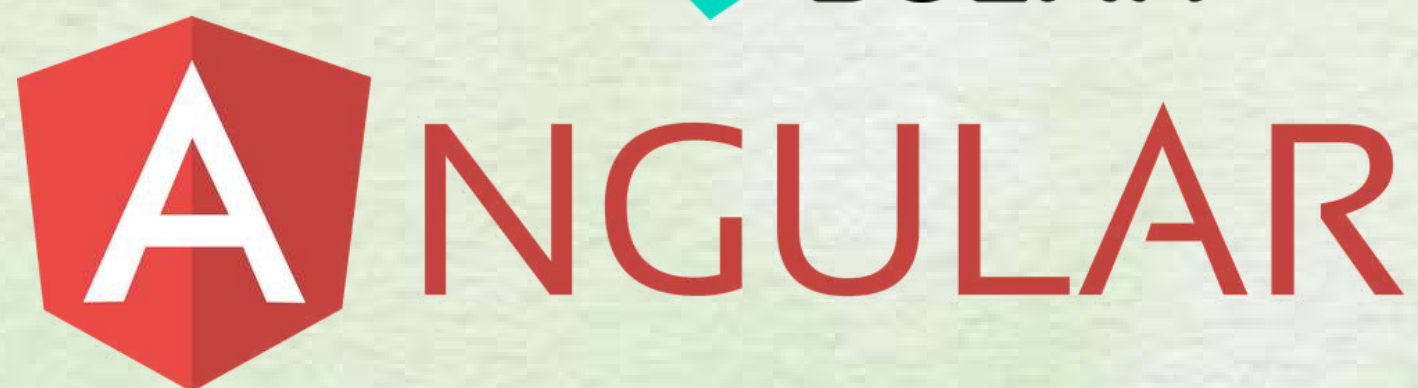
CONNEXION
API-FIRST FOR ALL



YA KOI DEDAN ?



Flask



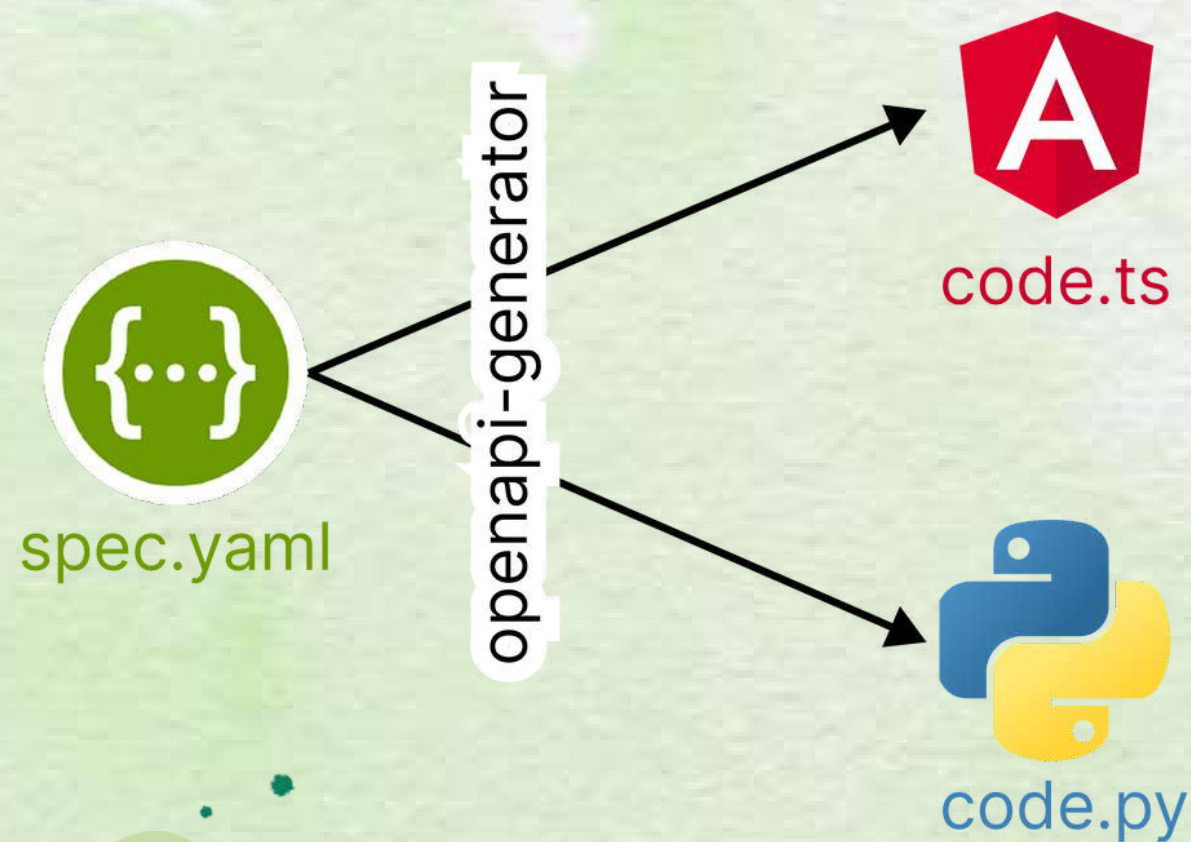
```
openapi: "3.0.3"
info:
  title: ADH6 API
  description: >
    This is the specification for **MiNET**'s ADH6 plaform. Its aim is to manage our users,
    devices and treasury.
  license:
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
  version: 2.0.0
  contact:
    name: MiNET
    email: equipe@minet.net
    url: https://minet.net
servers:
  - url: https://adh6.minet.net/api/
    description: ADH6 production server
  - url: https://adh6-dev.minet.net/api/
    description: ADH6 staging server
paths:
  /health:
    get:
      summary: Retrieve the health of the API server
      description: >
        This endpoint allows for better monitoring of the state of the API.
        **TODO**: Improve the amount of information returned
      operationId: health
      responses:
        '200':
          description: ADH6 is alive!
        '503':
          description: Database is down.
  security:
    - OAuth2:
        - admin:read
        - ApiKeyAdminAuth: []
  tags:
    - misc
  /profile:
    get:
      summary: Introspection endpoint
      description: >
        This endpoint returns information about the currently logged-in user.
    ...
```



```
openapi: "3.0.3"
info:
  title: ADH6 API
  description: >
    This is the specification for **MiNET**'s ADH6 plaform. Its aim is to
    manage our users, devices and treasury.
```

```
paths:
  /health:
    get:
      summary: Retrieve the health of the API server
      description: >
        This endpoint allows for better monitoring of the state of
        the API.
        **TODO**: Improve the amount of information returned
      operationId: health
      responses:
        '200':
          description: ADH6 is alive!
        '503':
          description: Database is down.
  security:
    - OAuth2:
        - admin:read
        - ApiKeyAdminAuth: []
  tags:
    - misc
  /profile:
    get:
      summary: Introspection endpoint
      description: >
        This endpoint returns information about the currently
        logged-in user.
    ...
```

PK OPENAPI ? INITIATIVE



OpenAPITools / [openapi-generator](#)

OpenAPI Generator allows generation of API client libraries (SDK generation), server stubs, documentation and configuration automatically given an OpenAPI Spec (v2, v3)



23672 stars

3480 Contributors



open api generator

```
docker run --rm -u $(CURRENT_UID):$(CURRENT_GID) -v ${PWD}:/local openapitools/openapi-generator-cli:v7.12.0 generate -i /local/openapi/spec.yaml -g python-flask -o /local/tmpsrc --additional-properties packageName=adh6 --additional-properties=modelPackage=entity
```

```
docker run --rm -u $(CURRENT_UID):$(CURRENT_GID) -v ${PWD}:/local openapitools/openapi-generator-cli:v7.11.0 generate -i /local/openapi/spec.yaml -g typescript-angular -o "/local/frontend_angular/src/app/api" --additional-properties=queryParamObjectFormat=key
```

ET CONNEXION ?

- créer les routes
- valides les entrées / sorties
- appel les fonctions
- met à disposition un swagger UI (<https://adh6.minet.net/api/ui/>)

```
main.py

from connexion import FlaskApp

app = FlaskApp(__name__)

def post_greeting(name: str):
    return f"Hello {name}", 200

app.add_api("openapi.yaml")
```

CONNEXION
API-FIRST FOR ALL



spec.yaml

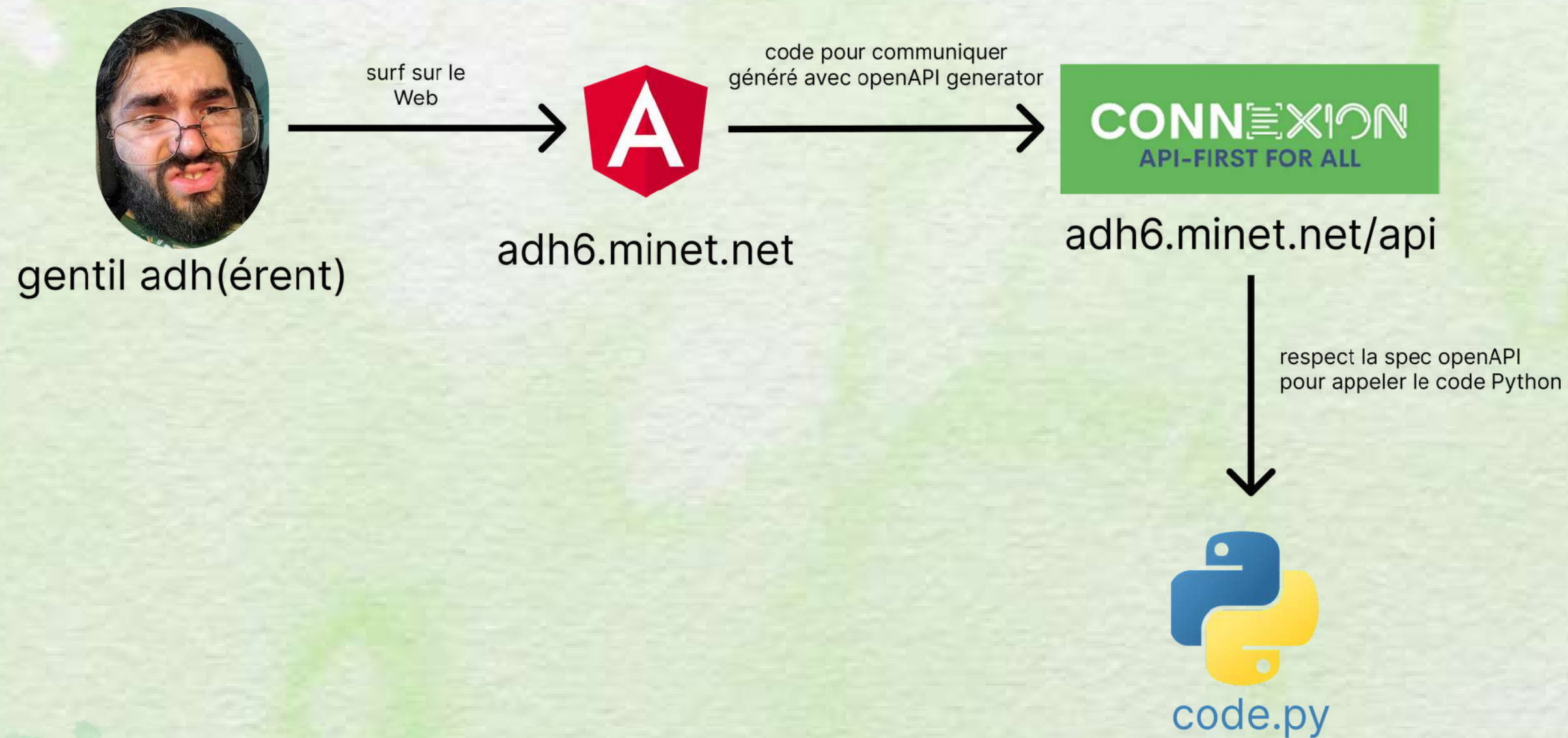


code.py

```
spec.yaml

openapi: "3.0.0"
info:
  title: Greeting application
  version: 0.0.1
paths:
  /greeting/{name}:
    post:
      operationId: run.post_greeting
      responses:
        '200':
          description: "Greeting response"
          content:
            text/plain:
              schema:
                type: string
      parameters:
        - name: name
          in: path
          required: true
          schema:
            type: string
```

CONCRÈTEMENT ?



SUR LE RÉSEAU ?

SNMP
SYMPA DE SE
CONNECTER, MAIS COMMENT
ON INTÈGRE ÇA AU NÉAU ??
RADIUS

SNMP

SNMP (Simple Network Management Protocol)

- Les données sont organisées dans des MIBs (Management Information Base) sous forme d'arbre avec des numéros sur les noeuds :
 - vmMembershipTable (vlan associé à un port) = 1.3.6.1.4.1.9.9.68.1.2.2.1
- ADH6 cache tous ces trucs horribles derrière des jolies fonctions :

```
@log_call
def get_port_vlan(self, port_id: int) -> int:
    """
    Get the VLAN assigned to a port.

    :raise PortNotFound
    """
    oid, ip, community = self.get_oid_switch_ipand_community_from_port_id(port_id)
    try:
        return get SNMP_value(community, ip, 'CISCO-VLAN-MEMBERSHIP-MIB', 'vmVlan', oid)
    except NetworkManagerReadError:
        raise
```

RADIUS

Bonne nouvelle :

C'est SQL qui fait la traduction entre la table ADH6 et la table freeradius (enfin... en grande partie)

CREATE ALGORITHM=UNDEFINED DEFINER='thunder'@'%' SQL SECURITY DEFINER VIEW 'login_authorize' AS select 'adh5-prod','adherents','id' AS 'id','adh5-prod','adherents','login' AS 'login','adh5-prod','vians','numero' AS 'vian_nb','adh5-prod','adherents','date_de_depart' AS 'date_de_depart','adh5-prod','adherents','mode_association' AS 'mode_association' from (('adh5-prod','adherents' join 'adh5-prod','chambres') join 'adh5-prod','vians') where (('adh5-prod','adherents','chambre_id' = 'adh5-prod','chambres','id') and ('adh5-prod','chambres','vian_id' = 'adh5-prod','vians','id'));

Authentification:

- Radius ← vue radcheck ← Tables adh5-prod et adherents ← ADH6

Vérification de l'adhérent :

- Vue login_authorize (date de départ)
- Vue login_to_macf (Adresse MAC filaire)
- Vue login_to_macf_mab (MAB)
- Vue login_to_macw (Adresse MAC)

LES LOGS

- Les logs sont stockés et récupérés depuis ElasticSearch en production.
- On récupère les logs de freeradius (Authentification, adresses MAC) et kea (DHCP)
- Ils sont envoyés sur les serveurs de logs (es-padon, es-magnus) pour être stockés
- ADH6 les récupère pour les montrer aux admins sur la page des adhérents.



JE VEUX MODIFIER DES TRUCS

Voici le plan pour modifier des trucs sur adh6 :

- **api_server** : Code source du backend Python
- **frontend_angular** : Code source du frontend Angular
- **openapi** : Spécification de l'API
- **reverse_proxy** : Configuration du reverse proxy (NGINX)
- **doc** : Documentation supplémentaire (un seul pdf lol)

The background is a soft, watercolor-style wash of light green and white. It is decorated with various green leaves and foliage. In the top left, there are large, dark green leaves with prominent veins. In the top right, there are smaller, rounded green leaves. In the bottom left, there are more rounded green leaves. In the bottom right, there are large, dark green leaves with prominent veins, similar to the ones in the top left.

MOMENT TP

ON KLONE

*Y a plus de lien entre Github et Gitlab, donc pour le moment,
vous pouvez clone ici :*



clone ADH7

```
git clone git@github.com:airopi/adh7
```

ON RUN (FAUT DOCKER)



run ADH7

```
cp adh7  
docker compose up -d --build  
docker compose logs -f
```

ON SEED

La base de donnée est construite lors de l'exécution, mais elle est vide ! On peut donc la remplir avec des données !



seed ADH7

```
# vous pouvez ctrl + C pour quitter les logs  
cd api_server  
./manage.sh seed  
./manage.sh fake ${LDAP}
```



ON TEST

<https://adh6-local.minet.net/>

Connectez vous avec votre LDAP !

(Faites pas attention aux couleurs du thème)

Si ça marche, c'est beau.

La première fois que j'ai voulu faire tourner ADH6 en local, ça m'a pris
environ 2h (et ça marchait a moitié)



ON KOMPREN

OK, ça tourne, mais comment ?

Dans /adh7, regardez le fichier docker-compose.yml : il contient plusieurs services

- le frontend
- l'api
- un proxy
- et la base de donnée !

ON KOMPREN

Dans /adh7/api_server, regardez le fichier Dockerfile.

Il s'y passe pleins de choses, mais entre autres:

- on télécharge un fichier "OUIs.txt" qui contient les noms des fabricants par adresses MAC
- on extrait les MIBs et on génère le code python associé
- on génère le code python depuis la spec OpenAPI
- on installe les paquets pythons et autres dépendances
- on lance le backend !



ON KOMPREN

Dans /adh7/frontend, regardez le fichier Dockerfile.

Il s'y passe pleins de choses, mais entre autres:

- on génère le code angular depuis la spec OpenAPI
- on optimise les fichiers SVGs
- on installe les dépendances
- on build le front !

The background is a light green watercolor wash. It is decorated with various green leaves and foliage. In the top left, there are large, dark green monstera leaves. In the top right, there are smaller, rounded green leaves. In the bottom left, there are more rounded green leaves. In the bottom right, there are large monstera leaves and some smaller rounded leaves.

ON KOMPREN

Y a encore beaucoup beaucoup de choses à regarder, à
étudier, à comprendre, etc.

C'est en adh6-sant qu'on devient adh6-seur.

The background is a soft, watercolor-style wash of light green and pale yellow. It is decorated with various green leaves and foliage. In the top left, there are large, dark green monstera leaves. In the top right, there are smaller, rounded green leaves. In the bottom left, there are more rounded green leaves. In the bottom right, there are large monstera leaves and some smaller, pointed green leaves. The overall feel is fresh, natural, and serene.

THANK
YOU