

RAPPORT DE PROJET

PROJET PERSONNEL
« PyAurion »

Projet et rapport réalisé par

Milo Montuori



REMERCIEMENTS

Tout d'abord, je tiens à remercier tout particulièrement et à témoigner toute ma reconnaissance aux personnes qui m'ont aidé dès le début ou très récemment à concrétiser ce projet.



SOMMAIRE

I. INTRODUCTION.....	2
II. BESOINS ET OBJECTIFS DU PROJET	3
1. CONTEXTE	3
2. MOTIVATION	3
3. OBJECTIFS	4
III. GESTION DE PROJET	5
1. L'ÉQUIPE.....	5
IV. DÉVELOPPEMENT TECHNIQUE.....	6
1. LA STRATÉGIE	6
a. Architecture matérielle	6
b. Architecture logicielle	7
2. LES SOLUTIONS LOGICIELLES.....	8
a. Les technologies	8
HTML	8
CSS	8
JAVASCRIPT	8
JSON	8
PYTHON	9
FLASK	9
NGROK	9
REACT NATIVE.....	9
b. Leurs applications	10
Application mobile.....	10
Site web.....	11
Fonctionnement technique	15
Les cookies.....	15
Détails des requêtes.....	16
V. BILAN DU PROJET	17
1. Conclusion générale.....	17
VI. PERSPECTIVES	18
VII. BIBLIOGRAPHIE.....	19



I. INTRODUCTION

Dans le cadre de mes études à JUNIA ISEN, nous, les étudiants, avons l'obligation d'utiliser « Aurion », une plateforme de gestion de scolarité. Malheureusement, l'accès à certaines fonctions via un smartphone n'est pas toujours adapté. Dans cette optique, j'ai donc voulu concevoir une application permettant la simplification de l'accès à Aurion sur mobile.

Ayant une passion pour le langage de programmation Python, j'ai ainsi décidé de créer mon application avec ce langage. Cela impose de nombreuses contraintes, mais j'y étais résolu. Dès octobre 2021, PyAurion est né.



II. BESOINS ET OBJECTIFS DU PROJET

1. CONTEXTE

Un étudiant fait toujours la même chose en entrant dans son établissement : il regarde son emploi du temps pour connaître sa salle et son professeur. Les étudiants de Junia ISEN ne font pas exception, et moi non plus.

Si on pense à cela, il est important, pour le confort de tous, d'avoir un accès simple, rapide et fonctionnel à cet emploi du temps. Malheureusement, l'outil Aurion qui nous est fourni par défaut ne remplit pas complètement ces critères.

De nombreux étudiants avant moi se sont confrontés à ce problème. Il existe actuellement de nombreux projets, privés ou publics, qui essayent de résoudre cette problématique.

2. MOTIVATION

Partant de ce postulat, j'ai donc eu l'idée et l'envie de faire mon propre projet me permettant d'accéder facilement à mes informations scolaires.

J'ai donc créé PyAurion, complètement conçu en Python, un langage que j'affecte particulièrement.



3. OBJECTIFS

Le principal objectif depuis la création du projet PyAurion est de créer une application mobile simple, rapide et efficace permettant l'accès aux données les plus demandées par les étudiants.

Ainsi, j'ai projeté dans un premier temps de mettre en place un site internet qui sera le cerveau du système. Ajouté à ce cœur de projet, j'ai découvert le monde des applications mobiles pour créer une Web App : une application qui charge un site internet. Cette dernière sera l'aboutissement du projet, à même de démontrer tout l'intérêt de PyAurion.



III. GESTION DE PROJET

1. L'ÉQUIPE

Ce projet étant entièrement réfléchi et développé par moi, je suis l'unique membre de mon équipe. Cependant, j'ai échangé avec de nombreux étudiants, que je remercie, qui font partie entière de ce projet.

Je suis Milo Montuori, prépa CIR (deuxième année en septembre 2022) à JUNIA ISEN à Lille.

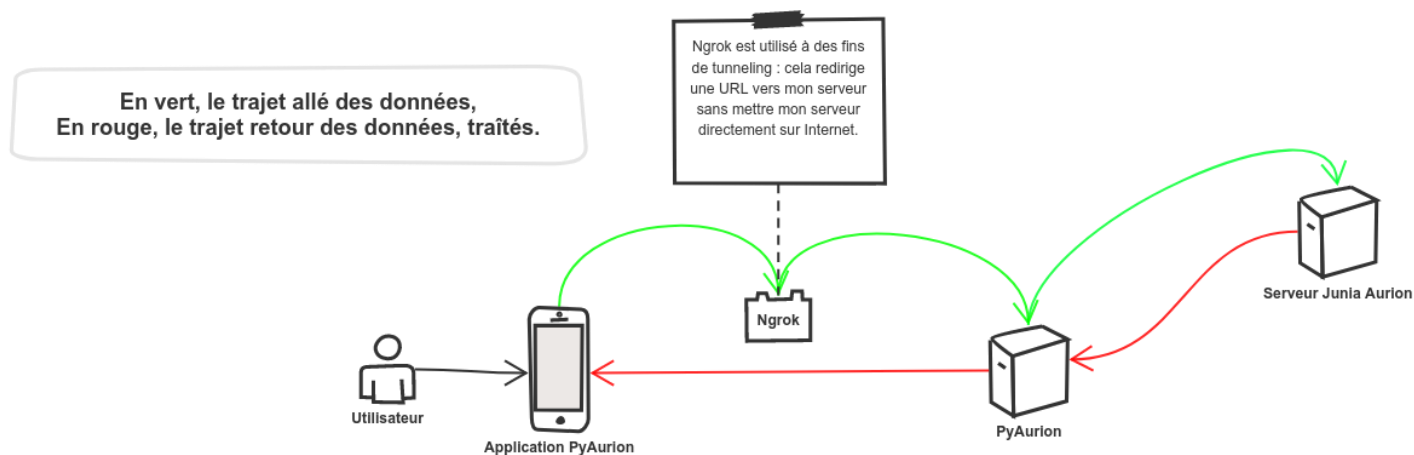


IV. DÉVELOPPEMENT TECHNIQUE

1. LA STRATÉGIE

a. Architecture matérielle

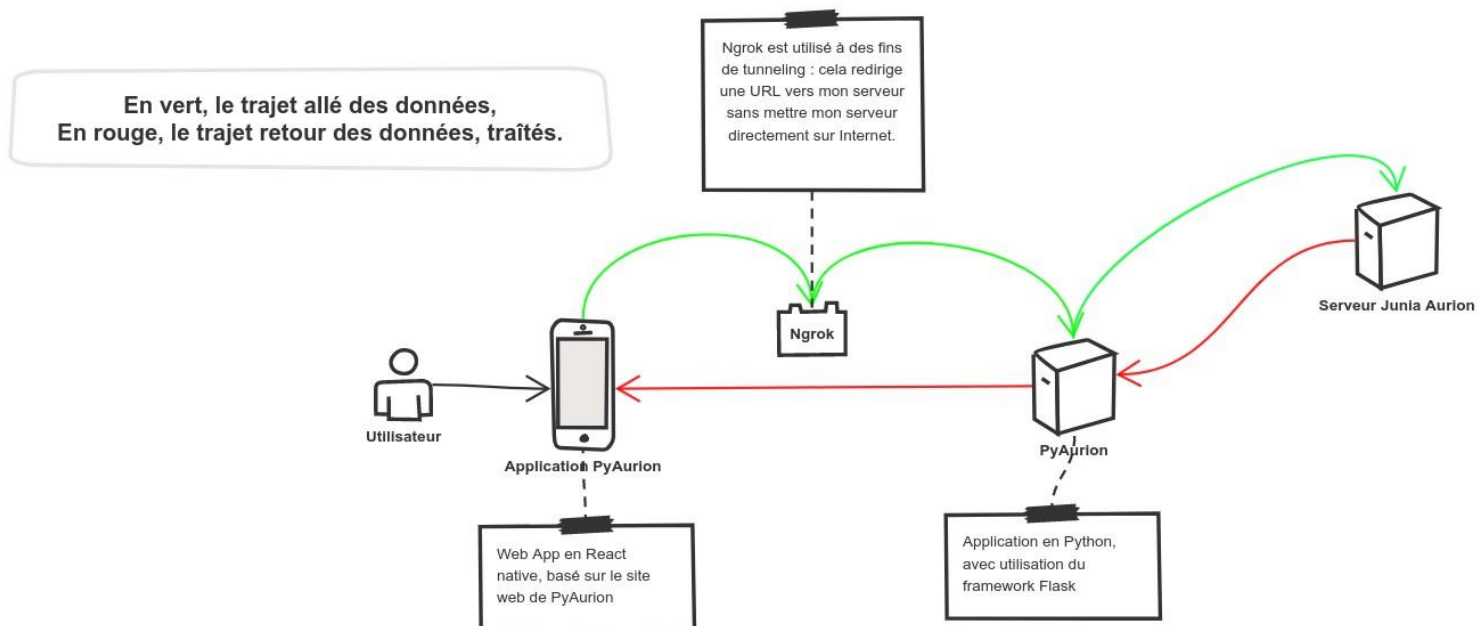
Le diagramme ci-dessous résume l'architecture matériel sous forme d'un schéma :



www.sketchboard.io



b. Architecture logicielle



www.sketchboard.io

Pour une question de lisibilité et de compréhension, l'intégralité du fonctionnement logiciel sera exprimée dans les paragraphes suivants.



2. LES SOLUTIONS LOGICIELLES

a. Les technologies

HTML

L'Hypertext Markup Language, abrégé HTML, est un langage conçu pour représenter les pages web. Ce langage de balisage permet de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, etc.

CSS

De l'anglais Cascading Style Sheets (les feuilles de style en cascade), le CSS constitue un langage informatique permettant d'ajouter des contraintes de mise en forme graphique dans des documents web, dont les normes sont établies par le World Wide Web Consortium (W3C).

JAVASCRIPT

JavaScript, abrégé JS, est un langage de programmation de scripts principalement utilisé dans les pages web interactives.

JSON

JavaScript Object Notation, abrégé JSON, est un format de données textuelles et génériques permettant de représenter de l'information structurée comme le permet le langage XML par exemple.



PYTHON

Python est un langage de programmation structuré et orienté objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.

FLASK

Flask est un micro-framework Python facile et simple qui permet de faire des applications web évolutives. Flask dépend de la boîte à outils WSGI et du moteur de templates Jinja. Le micro dans le « micro framework » signifie que Flask vise à garder le code de base simple mais extensible.

NGROK

Ngrok est un outil qui permet de rendre un site web local accessible depuis l'extérieur, via un tunnel sécurisé, depuis une url du type <https://azerty.ngrok.io>. Cette url est accessible par n'importe qui et depuis n'importe où.

REACT NATIVE

React Native est un framework d'applications mobiles open source créé par Facebook. Il est utilisé pour développer des applications pour Android, iOS et UWP en permettant aux développeurs d'utiliser React avec les fonctionnalités natives de ces plateformes.



b. Leurs applications

Application mobile

L'application mobile est l'un des points d'entrée vers PyAurion. Je vais donc commencer par expliquer son fonctionnement. L'application est développée en React Native. Voici un exemple d'une partie du code permettant la visualisation d'un site internet sur une Web App :

```
JS Appjs M X
JS Appjs > App
1 import * as React from "react";
2 import Constants from 'expo-constants';
3 import { WebView } from "react-native-webview";
4
5 export default class App extends React.Component {
6   render() {
7     return (
8       <WebView
9         source={{ uri: "https://mylow.fr/pa" }}
10        style={{
11          marginTop: Constants.statusBarHeight,
12          flex: 1,
13        }}
14      />
15      <WebView
16        overScrollMode="never"
17      />
18    );
19  }
20 }
```

Pour exporter un tel code en application mobile, j'ai utilisé le site internet <https://expo.dev/> qui s'occupe de compiler le React vers un format d'application mobile tel que *.ipa* pour les appareils Apple, ou *.app-bundle* pour les appareils Android.

Pour publier une application, il est nécessaire d'avoir un compte développeur sur les deux plateformes : Apple et Google. Concernant le compte Apple, je n'en ai pas. Je n'ai donc pas pu partager mon application sur l'*AppStore*. À propos de Google, je me suis créé un compte *Google Developer* qui me permet de préparer et publier une application, après vérification de Google. Voici l'interface me permettant de voir les versions d'une application :

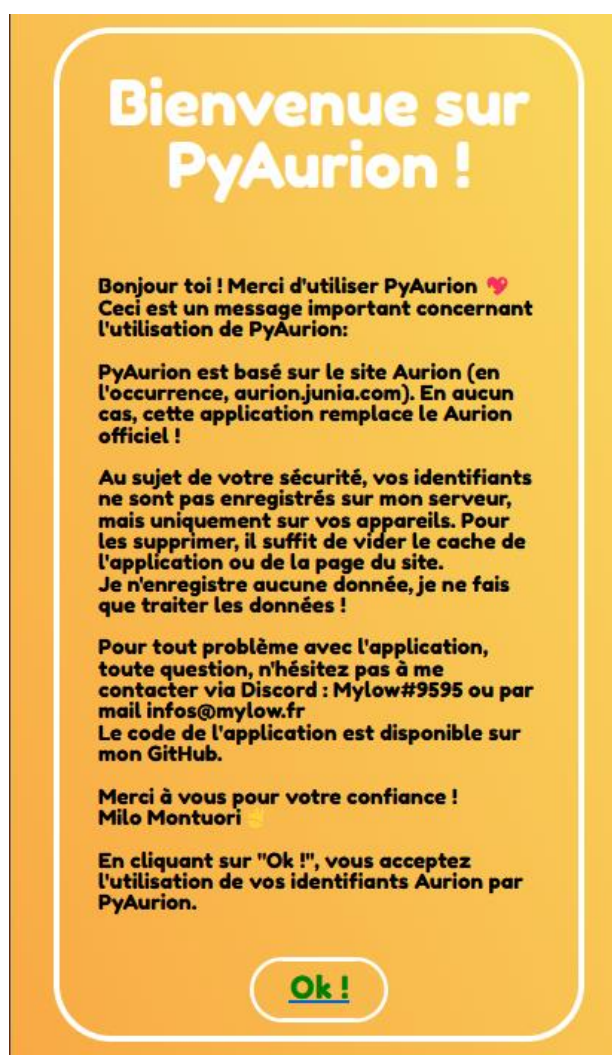
Version	Dernière version	Canal	État de la release	Dernière mise à jour	Pays/Régions
Icon's Update + fix topbar	4	Production	Disponible sur Google Play Déploiement complet	28 août 2022 16:12	177 sur 177
2 (1.0.1)	2	Tests internes	Accessible aux testeurs internes Déploiement complet	6 août 2022 21:32	-



Site web

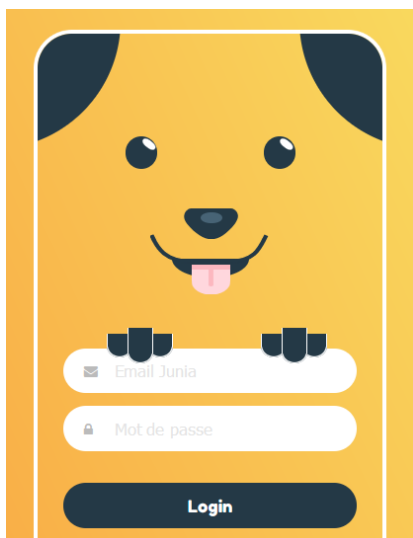
Le site web de PyAurion est le cœur de l'application. Dans ce paragraphe, je vais expliquer le fonctionnement du site internet, et par extension de l'application. Dans un second temps, j'expliquerai le *backend*. Les éléments suivants correspondent à la dernière version de PyAurion sortie en septembre 2022. Il est possible que des mises à jour aient eu lieu entre-temps.

Dans la dernière version de PyAurion, un *pop-up* s'affiche dès l'ouverture de l'application. Cette dernière prévient que PyAurion ne remplace pas Aurion et prévient également de l'utilisation des données personnelles des utilisateurs par l'application. L'utilisateur doit nécessairement accepter le fait d'utiliser ses identifiants JUNIA Aurion pour accéder à PyAurion.



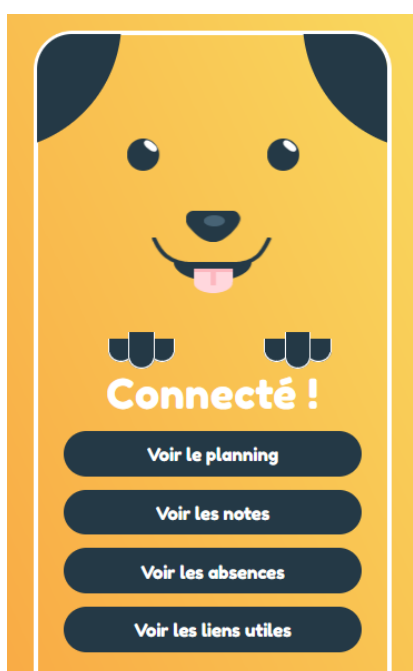


PyAurion étant une application basée sur le site Aurion, il est donc requis pour l'application d'avoir les identifiants (email JUNIA et mot de passe) de l'utilisateur. Ce dernier doit donc entrer ses informations qui seront stockées sur son appareil (je reviendrai sur ce point plus tard dans le rapport) :



Une fois connecté, l'utilisateur a accès à quatre options étant les trois fonctions principales de PyAurion et une option permettant l'accès à certains liens utiles :

- Voir le planning
- Voir les notes
- Voir les absences
- Voir les liens utiles





Le planning permet de consulter son emploi du temps. L’affichage se fait jour par jour, affichant les cours verticalement. Il est possible de naviguer entre les jours avec les flèches disposés en haut du planning. Un bouton « Aujourd’hui » permet le retour à la date du jour. Dans la partie basse de l’application se trouve deux boutons : « Reload » permet de mettre à jour les informations du planning (je reviendrai sur ce point plus tard dans le rapport) et le bouton « Home » faisant revenir l’utilisateur au menu vu précédemment, le menu principal de l’application. Les données visibles sont les horaires du cours, le bâtiment ainsi que la salle de cours, le contenu du cours puis l’enseignant.



PyAurion permet aussi la visualisation des notes de l'utilisateur. Les notes sont affichées de la plus récente à la plus ancienne. Il est prévu de pouvoir afficher seulement les notes de l'année en cours. Les données visibles sont la date de l'évaluation, le code de l'évaluation, le sujet de l'évaluation ainsi que la note reçue par l'utilisateur.





PyAurion permet également la visualisation des absences de l'utilisateur. Les absences sont affichées de la plus récente à la plus ancienne. Il est prévu de pouvoir afficher seulement les absences de l'année en cours. Les données visibles sont la date de l'absence, le type d'absence, le nombre d'heures concernés par l'absence, les informations sur le cours manqués (horaires, type de cours, professeur). Une somme des absences de l'utilisateur est affichée également.

Nombre d'heures total: 16:40

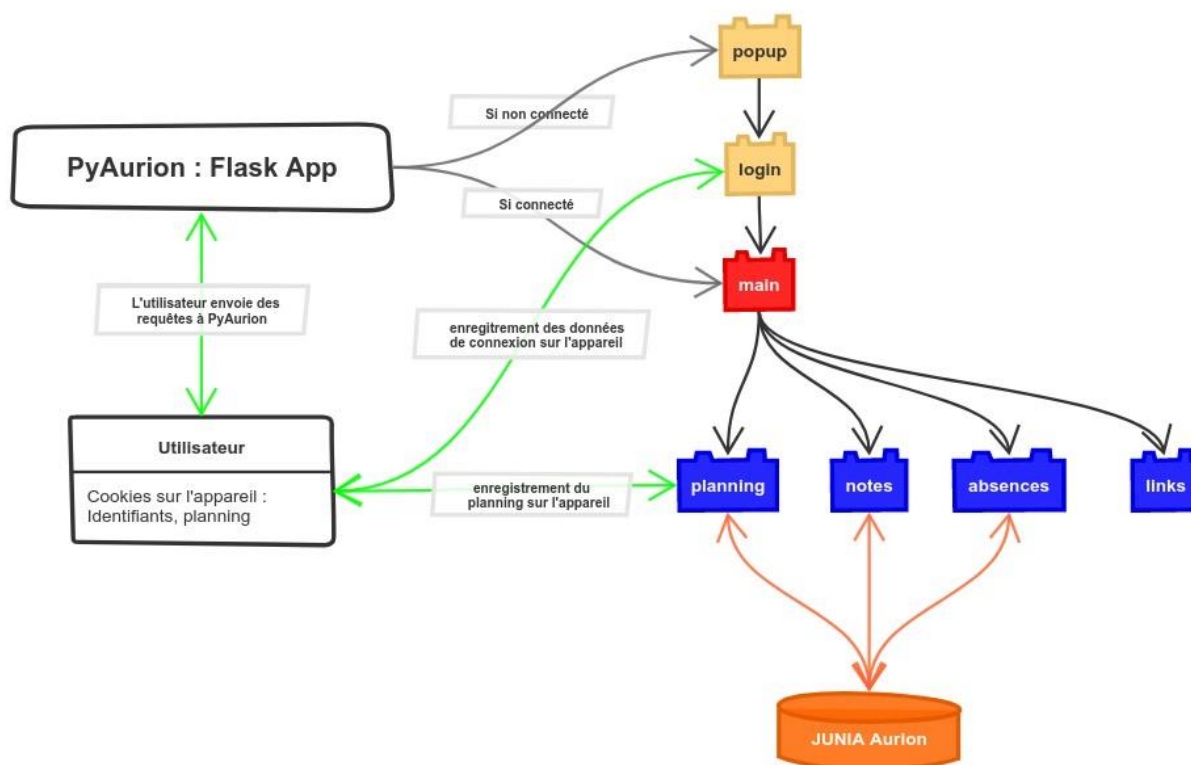
17/12/21
Absence excusée sans justificatif
1:00
11:25 - 12:25
Electronique Analogique
[redacted]
Pédagogique

17/12/21
Absence excusée sans justificatif
1:00
10:20 - 11:20
Mathématiques 2 : Analyse -
Mathématiques 2 : Analyse
[redacted]



Fonctionnement technique

Pour visualiser le fonctionnement technique derrière PyAurion, voici un schéma simplifié de l'application et les différents transferts de données :



www.sketchboard.io

Les cookies

PyAurion enregistre seulement les données nécessaires, pour éviter à l'utilisateur de se reconnecter à chaque ouverture de l'application. Les identifiants en font partie, ainsi que les données du planning : PyAurion enregistre le prochain mois à partir de la connexion de l'utilisateur dans le stockage, permettant un chargement rapide des informations de l'emploi du temps. Il est nécessaire d'actualiser ses données de temps en temps pour ne pas avoir de mauvaises surprises (d'où la présence du bouton « Reload » vu précédemment).



V. BILAN DE PROJET

1. Conclusion générale

PyAurion est avant tout un outil basé sur le site Aurion. Mais, du fait de son utilité publique, l'application devient un véritable projet personnel long-terme.

D'autre part, une Web App a cette particularité de rassembler divers domaines de l'informatique : Développement Web, Python, React (JavaScript), du réseau, et bien d'autres.

L'entièreté de mon code est publiée sur mon GitHub. Cela m'a permis de recevoir des messages de toute la France au sujet de mon projet, un élève de l'ISEN de Nantes m'a contacté pour développer une application similaire par exemple.

PyAurion est donc plus qu'une simple application gérée par un étudiant, mais bien un projet concret, utile à tous, y compris à moi. J'ai pu développer mes compétences comme je ne l'aurai jamais imaginé.



VI. PERSPECTIVES

Ce projet est avant tout un projet personnel, m'ayant permis d'apprendre de nombreuses choses tout au long du développement de PyAurion.

Il conviendra, cependant, d'élargir le nombre d'options possible au sein de l'application pour en faire un véritable outil pour les étudiants JUNIA, non uniquement ISEN.

Je pense par exemple à :

- L'affichage des notes et absences en fonction de l'année actuelle
- Un bloc-notes permettant l'enregistrement des devoirs
- Un système de notification pour toute nouvelle notes
- Une application mobile sur l'AppStore pour les utilisateurs Apple

La seule limitation reste l'imagination.



VII. BIBLIOGRAPHIE

Site Aurion sur le serveur JUNIA

<https://aurion.junia.com/>

Site internet PyAurion

<https://mylow.fr/pa>

Documentation Python

<https://www.python.org/>

Reqbin

<https://reqbin.com/curl>

Expo (pour React)

<https://expo.dev/>

Documentation Mozilla

<https://developer.mozilla.org/fr/>

Ngrok

<https://ngrok.com/>

Site Google Play

<https://play.google.com/>

Site Apple Developer

<https://developer.apple.com>