Portage de l’application mobile L’Ambulance sous flutter

# Plateforme

* Windows 10 (version 1803)
* Android Studio 3.4
* flutter\_windows\_v1.5.4-hotfix.2-stable
* swagger/postman (api)

# Installation

<https://flutter.dev/docs/get-started/install>

<https://developer.android.com/studio>

<https://swagger.io/>

<https://www.getpostman.com/>

# Best practices links

Architect your Flutter project using BLOC pattern :

<https://medium.com/flutterpub/architecting-your-flutter-project-bd04e144a8f1>

# Interface/UI

L’interface proposera un menu de 4 items en bas de menu toujours présents permettant de sélectionner les 4 fonctionnalités suivantes :

## Accueil

La page d’accueil présentera :

* Le logo de la société
* Un descriptif sommaire de l’application.
* En bas de page un encart publicitaire

## Simulateur

Le simulateur proposera un choix de formulaire pour les ambulances, les vsl, les taxis.

Pour chacun des formulaires :

* Chaque changement de sélection d’item du formulaire enverra une requête de service afin de calculer le coût de l’intervention en temps réel
* Service de type SOAP pour les ambulances, vsl et paramètres et récemment REST/JSON pour les taxis (voir le chapitre « Web services ».

## Paramètres

Le calcul du coût d’intervention pour chaque formulaire (ambulances, vsl, taxis) dépend des paramètres (globaux).

Chaque modification de ce formulaire doit être impacté immédiatement dans les formulaires cités ci-dessus car leurs calculs de coûts en dépende.

## Enregistrement

# Web services

## Service Ambulance et VSL

Le web service propose une api SOAP.

La documentation se trouve à l’url : ???

## Service Taxi

### Documentation

Le web service propose une api REST/JSON en méthode d’envoi de type POST.

La documentation est disponible à l’url suivante :

<http://liburgence.aidesnsoft.fr/swagger/ui/index>

### Génération des objets JSON / Gestion de la sérialisation

L’API propose plusieurs services avec de très nombreux paramètres.

L’API est en V1.0 et peut être amenée à évoluer.

Il faut donc pouvoir générer des objets modèles (POJO) en dart qui est le langage de programmation.

Pour cela le projet utilise la library « json\_serializable »

Installation/utilisation ci-dessous :

<https://flutter.dev/docs/development/data-and-backend/json>