

# Projet Plage Sans Vent

## Objectif

Application web React qui permet aux utilisateurs de visualiser les **plages françaises** avec leurs **conditions de vent METAR (temps réel)**, leurs **prévisions météo** et les **horaires de marées**.

Le projet s'inscrit dans l'écosystème **Thespot2be / PlageSansVent** et repose sur un backend PHP legacy exposant des données XML et JSON.

---

## Stack technique

- **Frontend** : React 18.3.1 + TypeScript
  - **Build tool** : Vite 5.4.2
  - **Carte** : Leaflet
  - **UI** : Aucun framework, styles inline (objet style={...})
  - **Backend** : PHP (legacy) – XML et JSON
  - **Encodage** : UTF-8 avec conversion depuis ISO-8859-1 si nécessaire
  - **Déploiement** : GitHub Actions → OVH /[www.PlageSansVentNewTravail](http://www.PlageSansVentNewTravail)
  - **Cache local** : localStorage (TTL par type de donnée)
- 

## Structure du projet

E:\PlageSansVentNew

```
├── .github\                # Actions GitHub pour déploiement OVH
├── frontend\              # Sources React/TypeScript (.tsx)
│   ├── main.tsx           # Entrée application
│   ├── pages\App.tsx      # Composant principal
│   ├── api\client.ts      # Gestion des appels backend + cache
│   ├── utils\haversine.ts
│   └── ...
├── react\
│   ├── index.html
│   └── assets\
└── Ressources\php\
    ├── ListeZones.php
    └── ListeSpots.php
```

|   |                                    |
|---|------------------------------------|
| — | DonneForecast.php                  |
| — | InfoSpotsXX.php                    |
| — | informations_maree3.php            |
| — | nearest_metar.php                  |
| — | logs\                              |
| — | DOCUMENTATION_APPLICATION_REACT.md |
| — | robots.txt                         |
| — | .gitignore                         |
| — | .cursor.json                       |
| — | PROJECT_CONTEXT.md                 |

## APIs principales

| Fichier                 | Méthode | Type | Description                                            |
|-------------------------|---------|------|--------------------------------------------------------|
| ListeZones.php          | POST    | XML  | Liste des zones (départements)                         |
| DonneForecast.php       | POST    | XML  | Donne les prévisions météo à partir de coordonnées GPS |
| ListeSpots.php          | POST    | XML  | Liste des plages d'une zone                            |
| InfoSpots{zoneNum}.php  | POST    | XML  | Détails complets d'un spot                             |
| informations_maree3.php | POST    | XML  | Horaires et coefficients de marée                      |
| nearest_metar.php       | GET     | JSON | Données METAR (vent, direction, heure)                 |

## Cache localStorage

| Donnée           | Clé                       | Durée  |
|------------------|---------------------------|--------|
| Zones            | zones:list                | 4 h    |
| Spots par zone   | spots:list:{zone}         | 1 h    |
| Détail spot      | spot:detail:{zone}:{spot} | 24 h   |
| Marées           | tide:{lat},{lon}:{date}   | 24 h   |
| METAR            | metar:{lat},{lon}         | 10 min |
| Rate limit METAR | metar:error               | 15 min |
| Prévision Météo  | Forecast:{lat},{lon}      | 2 h    |

## États et logique React

- **useState principaux** : zones, selected, spots, metar, userPosition, detail, view, embedded, loading, error
- **useEffect principaux** :

- Chargement initial des zones
  - Géolocalisation au démarrage
  - Chargement des spots pour la zone sélectionnée
  - Calcul et récupération du METAR le plus proche
  - Lecture des paramètres URL (embedded, mode)
- 

## Calculs géographiques

- Fonction `haversineKm(lat1, lon1, lat2, lon2)` pour distance en km
  - Seuil de 50 km pour déterminer la source de vent (user vs spot)
- 

## Fonctionnalités principales

1. **Liste par zone** : affichage des spots + METAR le plus proche
  2. **Carte Leaflet** : marqueurs cliquables, bouton “Rafraîchir”
  3. **Géolocalisation** : détection auto + fallback intelligent
  4. **Détail d’un spot** : description, horaires de marées, coefficients
  5. **Mode embedded** : affichage simplifié en iframe
- 

## Contraintes

- Structure locale et serveur **identiques** (E:\PlageSansVentNewTravail ↔ /www/PlageSansVentNewTravail)
  - Encodage et parsing XML/JSON à préserver
  - Respect du typage strict TypeScript
  - Commenter la logique métier critique
- 

## Déploiement

- Dépôt GitHub : [philibertb56/PlageSansVentNewTravail](https://github.com/philibertb56/PlageSansVentNewTravail)
  - Actions CI/CD : synchronisation FTP automatique vers OVH
  - Le build final est généré dans `react/assets/` puis transféré vers `/www/PlageSansVentNewTravail/react/assets/`
-



Auteur

**Bernard Philibert**

Écosystème Thespot2be / PlageSansVent

2025



## Interface et UX

L'application suit une logique **mobile-first**, avec une interface simple et épurée. Les principaux composants (dans leur ordre d'apparition de haut en bas) sont :

### AFFICHAGE PRINCIPALE

- **En-tête** : affiche le titre « Plage Sans Vent » et un icône de parasol. Les couleurs respectent le thème PWA (bleu #0066cc, fond blanc).



Plage Sans Vent

- **Sélecteur de zone** : un `<select>` natif permettant de choisir un département (Zone). Les options sont chargées via `ListeZones.php` et mises en cache (4 h). Au démarrage, la géolocalisation tente de choisir la zone la plus proche automatiquement.

87 - Haute - Vienne

- **Carte du vent actuel (METAR)** : un panneau présentant la station METAR la plus proche, la direction et la vitesse du vent (avec conversion en km/h), et l'heure d'observation. Si aucune station n'est disponible ou si les données sont trop anciennes (> 60 min), un message d'avertissement orange est affiché. Le METAR est mis en cache pendant 10 minutes et un mécanisme de rate limiting (15 min) empêche les appels répétés en cas d'erreur.

### Vent actuel

Limoges/Bellegarde Arpt, NA, FR (LFBL) - 19.6 km

**Sud (190°) à 13 Nds** (24 km/h)

Observé: 08:30

- **Sélecteur d'affichage** : Deux boutons « Liste » et « Carte » permet d'afficher les plages les plus proches de l'utilisateur en mode liste ou sur une carte Leaflet.

Liste

Carte

- **Slider de temporalité** : un slider de temporalité permet de « naviguer » entre l'instant présent « maintenant », jusqu'à la tombée de la nuit... Quand on est sur « Maintenant », on affiche les données du Metar, sinon on affiche les données du Forecast météo pour le centre de la zone des plages sélectionnées et affichées. Ce Slider reste toujours visible, même quand on scroll la page horizontalement.

 Maintenant

**S (190°) à 13 Nds** (24 km/h)



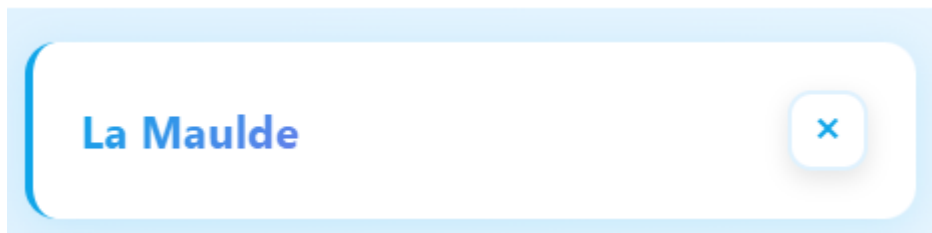
Maintenant

- **Liste des plages** : Si on est en mode « Liste », c'est une liste sous forme de `<ul>` contenant les spots de la zone courante. On affiche un **résumé des résultats** indiquant combien de plages ont été trouvées et dans quel rayon. Chaque entrée affiche le nom de la plage (bouton cliquable), la distance à l'utilisateur en kilomètres, la version des données et des pastilles rouges/vertes indiquant si la plage est ventée ou protégée. Si on est en mode « Carte », les plages sont représentées sur la carte avec des points de couleurs (rouge, orange, vert) en fonction de leur statut (plage protégée du vent, partiellement protégée ou ventée). Si l'on click sur un point, une bulle d'information s'affiche avec le nom de la plage et un lien pour accéder au détail de la plage.

|                                              |                |
|----------------------------------------------|----------------|
| 29 plages trouvées (rayon élargi à 357.6 km) |                |
| Etang de Mondon (268.1 km)                   | Plage ventée   |
| La Maulde (326.0 km)                         | Plage protégée |
| Plage de Saint-Julien-Le-Petit (326.7 km)    | Plage protégée |
| Ruisseau du Breuil (326.8 km)                | Plage ventée   |

AFFICHAGE DU PANNEAU POUR EXPLICITER LE CONTENU D'UNE PLAGE (dans l'ordre d'apparition des éléments du haut vers le bas) :

- **Nom de la plage** : On affiche le nom de la plage.

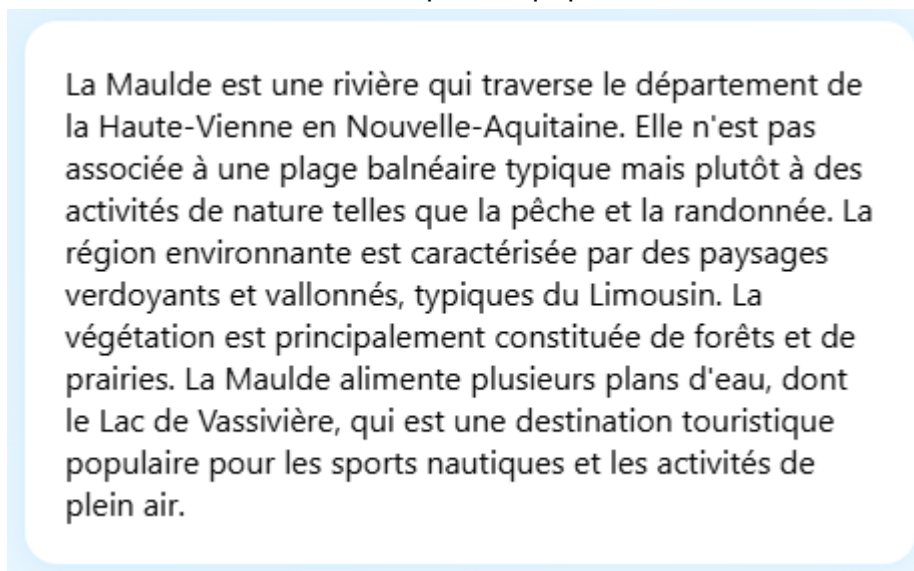


- **Statut de la plage et slider** : . Il affiche également un slider pour « naviguer » dans la journée (de maintenant jusqu'à la tombée de la nuit). Quand on est à « maintenant », on affiche les données Metar, sinon on affiche les données de forecast. On affiche l'état de la plage (protégée du vent ou pas) pour le moment de la journée qui est

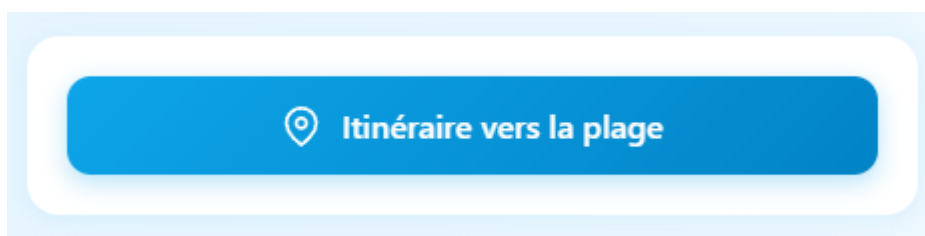
sélectionnée par le slider. Le slider et le statut de la plage (protégée ou pas du vent) reste toujours visible, même quand on scroll verticalement la page.



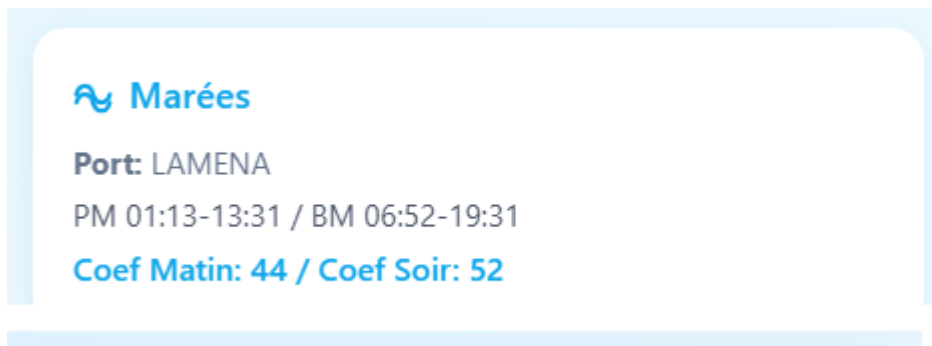
- **Informations sur la plage** : On affiche les informations textuelles sur la plage trouvées dans le fichier « InfoSpotsXX.php »



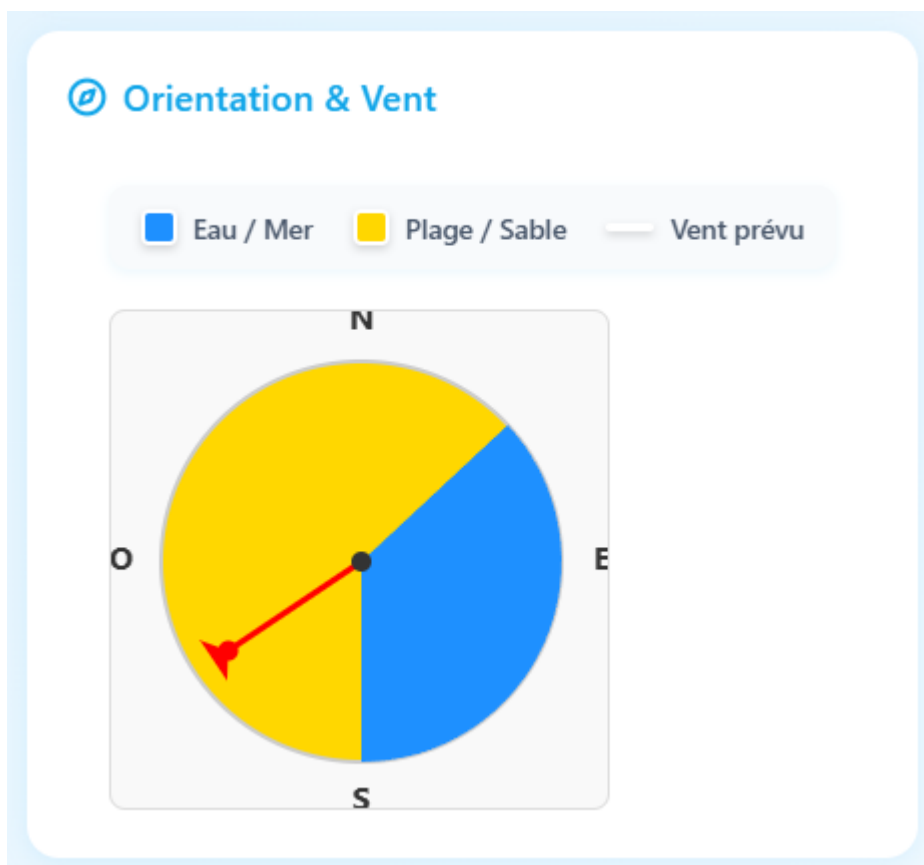
- **Itinéraire** : On affiche un bouton qui permet d'ouvrir une page Google Maps sur les coordonnées GPS de la plage



- **Marée** : On affiche les horaires de marées avec coefficients du port le plus près de la plage. Les marées sont obtenues via informations\_maree3.php (cache 24 h)



- **Orientation du vent sur la plage** : On affiche une représentation de la plage en utilisant les angles gauches et droits de la plage (fournies dans les fichiers InfoSpotsXX.php) et en affichant un cercle qui va représenter la plage (secteur en bleu, c'est la mer, secteur en jaune c'est la plage) et un rayon rouge qui va indiquer l'orientation du vent pour la période sélectionnée de la journée (dans le slider).



L'expérience utilisateur repose sur la **simplicité**, la **rapidité** (chargements progressifs, cache agressif), la **fiabilité** (gestion d'erreurs gracieuse) et l'**accessibilité** (contraste, labels explicites, boutons cliquables). Le design utilise des couleurs cohérentes, une typographie system-ui, des marges et paddings harmonisés et une mise en page responsive.

---

## Design et graphisme

La charte graphique définit une **palette de couleurs** : bleu primaire #0066cc pour les actions, bleu clair #0a58ca pour les liens, bleu très clair #f0f8ff pour les fonds de cartes, orange pâle #fff8dc pour les avertissements, rouge crimson pour les erreurs, et divers gris pour le texte secondaire et les bordures. Le fond général est blanc.

La typographie s'appuie sur la police système, avec des tailles adaptées (<h1>, <h3>, corps de texte) et des opacités réduites pour les informations secondaires. Les composants utilisent des bordures fines gris clair, des coins arrondis (borderRadius : 6px) et des marges/paddings réguliers. Les icônes (parasol, vent, horloge) sont utilisées via des images SVG ou PNG dans le manifeste PWA.

Le design est **responsive** : l'affichage est optimisé pour le mobile, avec des listes verticales et une carte adaptative. Le manifest JSON et le service worker assurent l'installation en PWA, le thème couleur (#0066cc), le fond (#ffffff) et l'orientation portrait.

---

## Modèles de données

Les principales structures de données manipulées par l'application sont :

- **Zone**

```
type Zone = {
  NomZone: string    // "Loire Atlantique"
  Num: string        // "44"
  GPSZone: string    // "lat,lon"
}
```

- **Spot**

```
type Spot = {
  NomSpot: string    // Nom de la plage
  Version: string    // Version des données (pour invalider le cache)
}
```

- **SpotDetail**

```
type SpotDetail = {
  NomSpot: string    // Nom de la plage
  GPSSpot: string    // Coordonnées "lat,lon"
```

```

    TexteSpot: string // Description courte
    Description: string // Description longue (HTML possible)
}

```

- **MetarInfo**

```

type MetarInfo = {
    icao: string           // Code ICAO (ex: "LFRZ")
    name: string          // Nom de la station
    distance_km: number   // Distance en km
    metar_raw: string     // METAR brut
    windOrientation?: string // Direction formatée ("Sud (190°)")
    windSpeed?: string    // Vitesse ("14 Nds")
    obsTime?: string      // Timestamp ISO d'observation
}

```

- **TideInfo**

```

type TideInfo = {
    isAvailable: boolean
    text: string           // "PM 06:30-18:45 / BM 12:15-00:30"
    coefText: string      // "Coef Matin: 95 / Coef Soir: 102"
    timestamp: number     // Timestamp (ms)
    values: {
        heurepmm: string
        heurepms: string
        heurebmm: string
        heurebms: string
        coefmm: string
        coefms: string
    }
    portName?: string     // Nom du port de référence
}

```

---

## Règles métier et algorithmes

Les spécifications décrivent plusieurs règles métier :

- **RBM1 : Sélection automatique de la zone** – Au démarrage, l'application calcule la distance entre la position de l'utilisateur et chaque zone via la formule de Haversine et sélectionne automatiquement la plus proche (sauf si l'utilisateur choisit manuellement).
- **RBM2 : Filtrage par rayon** – Pour lister les plages, on commence par un rayon de 20 km autour de la position utilisateur. Si moins de 50 plages sont trouvées dans la zone courante, le rayon est progressivement augmenté jusqu'à atteindre au moins 50 plages. Les plages ne sortent pas de leur zone administrative.

- **RBM3 : Choix du METAR** – Si la plage la plus proche de l'utilisateur est à moins de 50 km, la recherche de METAR utilise la position de l'utilisateur. Sinon, elle utilise les coordonnées de la plage. La recherche s'effectue dans un rayon croissant (0,3° à 4,0°) et ne prend en compte que des observations de moins de 60 minutes. Un cache et un rate limiting limitent les appels.
- **RBM4 : Calcul des marées** – Lorsqu'une plage est sélectionnée, l'API `informations_maree3.php` est appelée avec les coordonnées du spot et la date du jour pour obtenir les horaires de Pleine Mer (PM) et de Basse Mer (BM) ainsi que leurs coefficients. Le port le plus proche est automatiquement choisi.

Ces règles assurent la cohérence et la pertinence des informations présentées à l'utilisateur.

BUT :

Je veux réaliser une application qui, à partir d'une zone sélectionnée (département) affiche les plages les plus proches de l'utilisateur. L'application va également afficher la météo temps réel (metar) en utilisant l'aéroport le plus proche des plages. On va pouvoir voir le vent temps réel (METAR) mais également les prévisions météo pour le secteur des plages sélectionné. On va pouvoir regarder ainsi le vent en temps réel mais également le vent pour les heures à venir jusqu'à la nuit. Et pour l'heure sélectionnée, on va afficher le status des plages (protégé du vent ou pas). Si on click sur une plage, on a alors les informations sur la plage qui s'affichent, la marée du port le plus proche de la plage et un système de sélection du moment de la journée qui nous intéresse (Maintenant -on affiche les données metar-, ou alors une heure précise jusqu'à la tombée de la nuit) afin de voir l'état de la plage en fonction du moment de la journée.

## PREMIER DEMARRAGE

Au premier démarrage, la variable `SelectedZone` n'existe pas dans le local storage. Il n'y a donc pas de zone sélectionnée. Tu vas regarder les coordonnées de l'utilisateur et trouver la zone la plus proche de celui-ci pour définir la `SelectedZone`! Pour se faire, tu as toutes les zones disponibles dans le fichier `ListeZones.php` et les coordonnées géographiques de chaque zone ! Ainsi tu vas pouvoir calculer la distance entre l'utilisateur et chacune des zones pour choisir la zone la plus proche de l'utilisateur : Zone que l'on va appeler `XX` par la suite dans ce document d'explications fonctionnelles.

## AFFICHAGE DE LA LISTE DES PLAGES

Si tu souhaites afficher la zone `XX`. Tu vas regarder dans le local storage si la liste des plages pour cette zone est déjà enregistrée (la clef sera `"spots:list:<XX>"`). Si ce n'est pas le cas, tu vas récupérer la liste des plages de la zone `XX` au moyen du script

"ListeSpots<XX>.php" et tu pourras alors sauver les informations de la zone XX (liste des plages avec les coordonnées gps et numéro de version, etc) dans la clef "spots:list:<XX>". Si c'est le cas, tu vas retrouver la liste des plages de la zone XX dans la clef "spots:list:<XX>". Tu récupères alors dans le champ "NomSpot" de la clef "spots:list:<XX>", le nom de chaque plage de la zone XX. Tu vas alors regarder dans le local storage si tu as déjà des informations pour chacune des plages dont tu as le nom (la clef pour chacune des plages va être "zone<XX>:<NomSpot>"). Si tu trouves les clefs pour chaque plage, tu vas pouvoir récupérer les infos de ces plages dans le local storage. Tu dois juste vérifier que le numéro de version de la plage est le même entre le champ "Version" que tu trouves dans la clef "spots:list:<XX>", et le numéro de version de la plage qui est stocké dans la clef "zone<XX>:<NomSpot>". Si le numéro de version n'est pas le même ou si la clef "zone<XX>:<NomSpot>" n'existe pas, alors tu vas aller télécharger sur le serveur les informations de la plage en appelant le script "InfoSpots<XX>.php" et en passant en paramètre le nom de la plage désirée. Dès que tu as récupéré les informations de la plage <NomSpot>, tu vas pouvoir créer dans le local storage une sauvegarde de ces informations dans la clef "zone<XX>:<NomSpot>". N'oublies pas de sauver également le numéro de version de la plage (numéro de version que tu as récupéré dans la clef "spots:list:<XX>".

quand tu affiches les plages d'une zone, tu n'as pas besoin d'aller chercher le fichier InfoSpotsXX.php pour chaque plage ! ce fichier contient toutes les informations sur les plages de la zone. tu le télécharge une seule fois et tu stock les informations dans le local storage. d'ailleurs, avant de le télécharger, tu regardes si les informations ne sont pas déjà dans le local storage. Si c'est le cas, tu prends les informations du Local Storage. Tu ne vas alors re télécharger le fichier InfoSpotsXX.php uniquement si il te manque une plage dans le local storage (plage listée dans le fichier ListeSpotsXX.php) ou si le numéro de version d'une plage à changé dans ce même fichier. N'oublies pas de remettre à jour le local storage à partir des données que tu récupères suite au téléchargement du fichier InfoSpotsXX.php quand tu affiches les plages d'une zone, tu n'as pas besoin d'aller chercher le fichier InfoSpotsXX.php pour chaque plage ! ce fichier contient toutes les informations sur les plages de la zone. tu le télécharge une seule fois et tu stock les informations dans le local storage. d'ailleurs, avant de le télécharger, tu regardes si les informations ne sont pas déjà dans le local storage. Si c'est le cas, tu prends les informations du Local Storage. Tu ne vas alors re télécharger le fichier InfoSpotsXX.php uniquement si il te manque une plage dans le local storage (plage listée dans le fichier ListeSpotsXX.php) ou si le numéro de version d'une plage à changé dans ce même fichier. N'oublies pas de remettre à jour le local storage à partir des données que tu récupères suite au téléchargement du fichier InfoSpotsXX.php

N'oublies pas que je souhaite que tu affiches uniquement les plages de la zone sélectionnée qui sont dans un rayon des 20km de l'utilisateur. Et si tu as moins de 50 plages dans ces 20km, alors tu élargies le rayon jusqu'à avoir 50 plages DE LA ZONE

SELECTIONNE (!). Attention, pour trouver les plages que tu vas afficher, tu as toutes les informations des plages de la zone XX dans "spots:list:<XX>" (les coordonnées GPS des plages notamment). Ainsi, tu pourras télécharger uniquement les informations des plages à afficher et ne pas télécharger inutilement les informations des autres plages (les plages non sélectionnées)

je souhaite que tu mettes en place la barre de progression en haut de l'écran dès que tu changes de zone. Cette barre doit disparaître quand toutes les plages sont affichées et leurs statut affichés. il est à noter que cette barre doit également s'afficher quand il n'y a aucune zone de sélectionnée et que tu cherche quelle est la zone la plus proche de l'utilisateur afin de déterminer la zone sélectionnée..

Tu vas créer 2 boutons en dessous de la sélection de la zone, un bouton "Liste" et un bouton "Carte". Le bouton "Liste" affiche la liste des plages de la zone (en respectant les consignes de sélections des plages en fonction de coordonnées GPS de l'utilisateur) et le click sur "Carte" va faire exactement la même chose mais en affichant les plages sur une carte OpenStreetMap.

## AFFICHAGE DES INFORMATIONS DE VENT

## AFFICHAGE DE L'ETAT DES PLAGES

Tu vas pouvoir afficher l'état des plages en fonctions des indications de vent (la plage est à l'abrit ou pas globalement)

Il est à noter que sous les boutons "liste" et "carte", on va afficher un slider qui va permettre de sélectionner le moment de la journée qui nous intéresse (Maintenant -on affiche les données metar-, ou alors une heure précise jusqu'à la tombée de la nuit et là on va utiliser les données du forecast pour la zone des plages affichées) afin de voir l'état de ces plages en fonction du moment de la journée sélectionné. Il est à noter que le slider de sélection du moment de la journée doit toujours rester visible, même quand on scroll verticalement la page.

Note : quand on utilise DonneForecast pour récupérer le forecast d'un point GPS, il faut que l'on récupère le forecast de la journée entière. On stock ces prévisions dans le local storage avec un time stamp de 2 heures. Si on a besoin de prévision météo pour un point GPS, on regarde en premier dans le local storage si elles sont présentes et si elles sont suffisamment fraîches avant de retourner solliciter le serveur !

Attention, calcul UNIQUEMENT l'état des plages (pas de vent, plage protégé, etc) [calculateWindProtection] pour les plages de la zone XX que tu affiches !! Tu ne dois pas le faire pour les autres plages (plages non sélectionnées de la zone XX et plages des autres zones non sélectionnées)...

Quand tu as récupéré toutes les plages à afficher de la zone sélectionnée (SelectedZone), tu vas pouvoir trouver quel est l'aéroport le plus proche de ces plages à afficher en utilisant l'information de géolocalisation de ces plages ! Quand tu as déterminé l'aéroport le plus proche de ces plages à afficher de la zone sélectionnée, alors tu vas pouvoir récupérer le METAR associé afin d'afficher le vent en temps réel !

Concernant le statut de la plage (est-elle protégée ou pas du vent), voici les regles de fonctionnement. Pour une plage <NomSpot> figurant dans la zone XX, on peut récupérer des informations d'orientation de la plage dans le Local Storage (si les informations des cette plage ont été sauvegardées dans la clef "zone<XX>:<NomSpot>"). Si ce n'est pas le cas, alors il faudra appeler le script "InfoSpots<XX>.php" (et en lui passant en parametre le nom de la plage désirée) pour récupérer ces informations. Si tu as l'information par le script ce sont les champs \$LimiteGauchePlage et \$LimiteDroitePlage qui nous intéressent. Dans le Local Storage ce sont les champs LimiteGauchePlage et LimiteDroitePlage qui nous intéressent. La limite gauche et la limite droite d'une plage forment un angle (dans le sens des aiguille d'une montre) qui est face à la mer. Globalement, si le vent est dans cet angle, la plage est ventée, sinon elle ne l'est pas. (Exemple 1, si la limite gauche de la plage est à 50 degres et la limite droite de la plage est à 120 degrés, si le vent est à 90 degrés la plage est ventée. Alors que si le vent est à 220 degrés, la plage est protégée. Exemple 2, si la limite gauche de la plage est à 310 degres et la limite droite de la plage est à 30 degrés, si le vent est à 10 degrés la plage est ventée. Alors que si le vent est à 180 degres, la plage est protégée). Voici des regles un peu plus précises :

- Si le vent est inférieur ou égale à 3 noeuds :

Quelque soit l'orientation du vent par rapport à la plage, on affiche "Peu de vent" en vert comme statut de la plage

- Si le vent est entre 4 noeuds et 7 noeuds :

Si le vent vient dans le secteur entre la Limite Gauche de la plage et sa limite droite (dans le sens des aiguilles d'une montre), on affiche "Un peu de vent" en jaune comme statut de la plage

Si le vent n'est pas dans ce secteur, on affiche "Plage protégée" en vert comme statut de la plage

- Si le vent est entre 8 noeuds et 15 noeuds :

Si le vent vient dans le secteur entre la Limite Gauche de la plage et sa limite droite (dans le sens des aiguilles d'une montre), on affiche "Plage ventée" en rouge comme statut de la plage

Si le vent n'est pas dans ce secteur, on affiche "Plage protégée" en vert comme statut de la plage

- Si le vent est supérieur ou égal à 16 noeuds :

Si le vent vient dans le secteur entre la Limite Gauche (X) de la plage et sa limite droite (Y) (dans le sens des aiguilles d'une montre), on affiche "Plage ventée" en rouge comme statut de la plage

Si le vent vient dans l'un de ces 2 secteurs :

secteur 1 : entre la Limite Gauche de la plage moins 10 degrés et la Limite Gauche de la plage (entre  $X-10$  degrés (modulo 360) et X, dans le sens des aiguilles d'une montre) .  
Exemple pour une plage dont la limite gauche est à 50 degrés, ça fait un secteur entre 40 et 50 degrés et pour une plage dont la limite gauche est à 5 degrés, ça fait un secteur: entre 355 et 5 degrés .

Secteur 2 : si le vent vient entre la limite droite de la plage et la limite droite de la plage plus 10 degrés (entre Y et  $Y+10$  degrés (modulo 360) dans le sens des aiguilles d'une montre)  
Exemple pour une plage dont la limite droite est à 50 degrés, ça fait un secteur entre 50 et 60 degrés et pour une plage dont la limite droite est à 355 degrés, ça fait un secteur: entre 355 et 5 degrés

Alors, si le vent vient dans l'un de ces 2 secteurs, on affiche "Un peu de vent" en jaune comme statut de la plage

Si le vent n'est pas dans ces secteurs (donc si il est dans le secteur entre la limite droite de la plage + 10 degrés et la Limite Gauche de la plage - 10 degrés, dans le sens des aiguilles d'une montre), on affiche "Plage protégée" en vert comme statut de la plage

(Exemple 1, pour un vent supérieur ou égal à 16 noeuds, si la limite gauche de la plage est à 50 degrés et la limite droite de la plage est à 120 degrés, si le vent est à 125 degrés, la plage est alors affichée comme ayant "Un peu de vent". Exemple 2, si la limite gauche de la plage est à 310 degrés et la limite droite de la plage est à 30 degrés, si le vent est à 295 degrés, la plage est affichée comme étant "Plage protégée". Exemple 3, si la limite gauche de la plage

est à 20 degrés et la limite droite de la plage est à 60 degrés, si le vent est à 50 degrés, la plage est affichée comme étant "Plage ventée". )

#### AFFICHAGE DE L'INFORMATION D'UNE PLAGE :

Si on click sur une plage (sur la liste ou sur depuis la carte) , on a alors les informations sur la plage qui s'affichent dans un panneau latéral. A partir des informations stockées dans le Local Storage (et qui viennent du fichier InfoSpotsXX.php) on va afficher les infos sur la plage, la marée du port le plus proche de la plage et un système de sélection du moment de la journée qui nous intéresse (Maintenant -on affiche les données metar-, ou alors une heure précise jusqu'à la tombée de la nuit) afin de voir l'état de la plage en fonction du moment de la journée sélectionné. En utilisant les angles de la plage, on va afficher un cercle qui va représenter la plage (en bleu c'est la mer, en jaune c'est la plage) et on va positionner dessus l'orientation du vent en fonction de l'heure sélectionnée. Il est à noter que le slider de sélection du moment de la journée et le status de la plage doivent toujours rester visibles, même quand on scroll verticalement la page.