

## Interfaces Client-Serveur de l'application

### **AIS APP**

Le 17 juin 2025,

**Equipe de recherche :**

**FAVÉ Xavier**

**BEHAREL Armand**

**TOURNEUX Antoine**

**Groupe 2 : Trinôme 5**

**Enseignants :** Napoléon Thibault, Kaddah Wissam

**Année :** 2024/2025

1)

### **GET /api/ship\_names.php** *Auto-complétion « Nom » (add\_ship.html)*

Permet de récupérer une liste de noms de navires pour alimenter automatiquement la liste déroulante du champ "Nom". Cela aide l'utilisateur à éviter de retaper un nom existant.

---

2.

### **POST /api/positions\_crud.php** *Ajouter une position AIS*

Envoie un point AIS (Automatic Identification System) contenant les données d'un navire (position, vitesse, cap, etc.) vers le serveur. Cela permet d'enregistrer une nouvelle position sur la carte.

---

3.

### **POST /api/boats.php** *Formulaire « Nouveau navire » (add\_ship.html / ships.js)*

Ajoute un nouveau navire en envoyant toutes les informations (fiche complète) le concernant : MMSI, nom, dimensions, état, etc.

---

4.

### **GET /api/boats.php** *Dashboard – navires (dashboard.html / map.js)*

Récupère la liste complète de tous les navires enregistrés. Sert à remplir le tableau de bord et à afficher les navires sur la carte.

---

5.

### **GET /api/positions.php?id\_bateau={MMSI}** *Dashboard – tracer la route (dashboard.html / map.js)*

Retourne les positions enregistrées pour un navire donné, identifié par son MMSI. Utilisé pour tracer la route réelle du navire.

---

6.

### **GET /api/boats.php** *Admin – liste navires (admin.html / admin.js)*

Même que la requête n°4, mais utilisée dans l'interface d'administration pour afficher la liste des navires.

---

7.

**POST /api/boats.php** *Admin – ajout navire (admin.html / admin.js)*

Permet à un administrateur d'ajouter un navire via le back-office en envoyant ses informations complètes.

---

8.

**PUT /api/boats.php** *Admin – édition navire (admin.html / admin.js)*

Modifie un navire existant. L'admin peut changer ses données (dimensions, nom, etc.) via un formulaire d'édition.

---

9.

**DELETE /api/boats.php** *Admin – suppression navire (admin.html / admin.js)*

Supprime un navire en fonction de son identifiant MMSI. Action irréversible dans l'interface admin.

---

10.

**GET /api/position\_pages.php?page=&limit=** *Positions paginées (admin / monitoring)*

Cette requête permet de récupérer des positions AIS (Automatic Identification System) paginées depuis la base de données, utilisées dans la page admin, où l'on veut afficher les positions par lots.

---

11.

**GET /api/predict\_type.php?SOG=...&COG=...&Heading=...&Length=...&Width=...&Draft=...** *Prédiction – type de navire (predict.html?mode=type / predict.js)*

**Page :** predict.html?mode=type / predict.js

Envoie les caractéristiques du navire dans l'URL pour obtenir une prédiction de son type (cargo, tanker, etc.) via un modèle IA.

---

12.

**GET /api/predict\_route.php?id={MMSI}** *GeoJSON de la trajectoire prédite (predict.html?mode=route)*

Cette requête permet de récupérer la trajectoire prédite d'un navire identifié par son MMSI, sous forme de GeoJSON, un format standard pour représenter des données géospatiales. Elle est utilisée dans la page `predict.html?mode=route`

---

13.

**DELETE /api/positions\_crud.php** *Supprimer une position AIS*

Permet de supprimer une position. Utilisée dans la page admin.

---

14.

**GET /api/predict\_horizon.php?mmsi={MMSI}&horizon={5,10,15}** *Prédiction – horizon (predict.html?mode=route / predict.js)*

Prédit la trajectoire du navire sur 5, 10 ou 15 minutes dans le futur en se basant sur un modèle d'IA.

---

15.

**GET /api/predict\_cluster.php** *Clustering (clusters.html / clusters.js)*

Récupère les points regroupés en clusters pour les afficher sur une carte. Sert à identifier des zones de regroupement ou comportements similaires.

---

16.

**POST /api/login.php** *Authentication (login.html)*

Permet à un utilisateur de se connecter en envoyant son email et mot de passe. Retourne le statut de connexion et son rôle (admin, utilisateur...).

---

17.

**POST /api/logout.php** *Déconnexion (menu)*

Déconnecte l'utilisateur et détruit la session. Retourne une confirmation (success:true).

---

18.

**GET /api/check\_auth.php** *Vérifier session (menu, pages protégées)*

Vérifie si l'utilisateur a une session active. Sert à sécuriser les pages réservées à l'admin.

---