

# Driver APP API Documentation

**Base URL** = <https://ahaadosa.eposplus.uk/driver/public/>

## 1. Signup API - [/signup](#) - POST

```
Payload = {  
  "f_name": "required,string,min:3",  
  "m_name": "string,min:3",  
  "s_name": "required,string,min:3",  
  "phone_number": "required,string",  
  "email": "required,email,unique",  
  "password": "required,string,min:8",  
  "dob": "required,date",  
  "nationality": "required,string",  
  "is_non_british": "required,boolean",  
  "home_no": "required_without:home_name,string",  
  "home_name": "required_without:home_no,string",  
  "street_name": "required,string",  
  "city": "required,string",  
  "county": "string",  
  "post_code": "required,string",  
  "home_phone": "required,string"  
}
```

```
Response = {  
  "code": "201",  
  "success": true,  
  "message": "User Created Successfully",  
  "data": {  
    "user_id": "1"  
  }  
}
```

## 2. Login API - [/login](#) - POST

```
Payload = {  
  "email": "email",  
  "password": "string, min:8"  
}
```

```
Response(200) = {
```

```
'code' => 200,  
'success' => true,  
'message' => 'Login Successful',  
'accessToken' => $accessToken,  
'refreshToken' => $refreshToken,  
'userID': "1"  
}  
Response(400) = {  
'code' => 400,  
'success' => false,  
'message' => 'Invalid Credentials'  
}
```

### 3. Send OTP API - [/send-otp](#) - POST

```
Payload = {  
  "phone_number"  
}  
Response(200) = {  
'code' => 200,  
'success' => true,  
'message' => 'OTP Sent Successfully'  
}  
Response(429) = {  
'code' => 429,  
'success' => false,  
'message' => 'Too Many Requests Sent'  
}
```

### 4. Verify OTP API - [/verify-otp](#) - POST

```
Payload = {  
  "phone_number",  
  "otp": "min:4,max:4"  
}  
Response(200) = {  
'code' => 200,  
'success' => true,  
'message' => 'OTP Verified'  
}  
Response(400) = {  
'code' => 400,  
'success' => false,  
'message' => 'Invalid Mobile Number', 'Invalid OTP', 'OTP Expired'
```

```
}
```

## 5. Generating New Access Token for Valid Refresh Token API - [/auth](#) - POST

```
Payload = {  
  "headers": {  
    "grant_type": "refresh_token"  
  },  
  "body": {  
    "refresh_token": "token"  
  }  
}  
  
Response(200) = {  
  'code' => 200,  
  'success' => true,  
  'message' => 'Login Successful',  
  'accessToken' => "newToken"  
}  
  
Response(400) = {  
  'code' => 400,  
  'success' => false,  
  'message' => 'Missing Parameters'  
}  
  
Response(401) = {  
  'code' => 401,  
  'success' => false,  
  'message' => 'Token Expired'  
}
```

## 6. Duty Status Update API - [/duty-status](#) - PUT

```
Payload = {  
  "duty_status": "on|off"  
}  
  
Response(200) = {  
  'code' => 200,  
  'success' => true,  
  'message' => 'Duty Status Updated Successful',  
}
```

## 7. Fetch Assigned Order API - [/fetch-assigned-order](#) - GET

```
Response(200) = {  
  'code' => 200,  
  'success' => true,  
  'message' => 'Order Fetched Successful',  
  'data' => $ordersData  
}
```

```
Response(200) = {  
  'code' => 200,  
  'success' => true,  
  'message' => 'No Active Orders Found',  
}
```

## **8. Update Order Status API - [/update-order-status](#) - PUT**

```
Payload = {  
  "order_id": "id",  
  "order_status": "accepted|collected"  
}
```

```
Response(200) = {  
  'code' => 200,  
  'success' => true,  
  'message' => 'Order Status Updated Successfully',  
}
```

## **9. Send deliver OTP API - [/send-delivery-otp](#) - POST**

```
Payload = {  
  "order_id": "id",  
}
```

```
Response(200) = {  
  'code' => 200,  
  'success' => true,  
  'message' => 'OTP has been successfully sent to the customer.',  
  'otp'=> $otp  
}
```

## **10. Complete Order API - [/complete-order](#) - PUT**

```
Payload = {  
  "order_id": "id",  
  "otp": "min:4|max:4"  
}  
  
Response(200) = {  
  'code' => 200,  
  'success' => true,  
  'message' => 'Order Delivered Successfully',  
}
```

## 11. Order History API - [/order-history](#) - GET

```
Response(200) = {  
  'code' => 200,  
  'success' => true,  
  'message' => 'Order Fetched Successful',  
}
```

```
Response(200) = {  
  'code' => 200,  
  'success' => true,  
  'message' => 'No Orders Found',  
}
```

## 12. Admin Approval API - [/admin-approve](#) - POST

```
Payload = {  
  "user_id": "id",  
}  
  
Response(200) = {  
  'code' => 200,  
  'success' => true,  
  'message' => 'User approved by admin',  
}
```

## 12. Admin Rejection API - [/admin-reject](#) - POST

```
Payload = {  
  "user_id": "id",  
}
```

```

}

Response(200) = {
  'code' => 200,
  'success' => true,
  'message' => 'User Rejected by admin',
}

```

### 13. Bank Acc Info Update API - </bank/update> - PUT

```

Payload = {
  "name": "required,string",
  "acc_number": "required,string",
  "sort_code": "required,string",
  "is_account_in_your_name": "required,boolean",
  "name_on_the_account": "required,string"
}

Response(200) = {
  'code' => 200,
  'success' => true,
  'message' => 'Bank Account Info Updated Successfully',
}

```

### 13. Profile Info Update API - </profile-info-update> - PUT

**Content-Type:** multipart/form-data

\*The document file (jpeg/png/jpg/svg/pdf, max 2MB).

```

Payload = {
  "requires_work_permit": "required,boolean",

  "docs": "required,array",
  "docs.*.doc_type": "required,in:visa,passport,ni,license,sign",
  "docs.*.doc_number": "string",
  "docs.*.doc_expiry": "date",
  "docs.*.doc_url": "file|mimes:jpeg,png,jpg,svg,pdf|max:2048",

  "uses_car": "required,boolean",
  "uses_motorcycle": "required,boolean",
  "uses_bicycle": "required,boolean",

  "has_motoring_convictions": "boolean",
}

```

```

    "is_uk_licence": "required,boolean",
    "licence_country_of_issue": "string",

    "has_medical_condition": "required,boolean",
    "can_be_used_as_reference": "required,boolean",
    "is_agreed_privacy_policy": "required,boolean"
  }

```

```

Response(200) = {
  'code' => 200,
  'success' => true,
  'message' => 'Profile Info Updated Successfully',
}

```

#### 14. Profile Info API - [./profile](#) - GET

```

Response(200) = {
  'code' => 200,
  'success' => true,
  'message' => 'Profile Info retrieved Successfully',
  'data' => {
    ...userData,
    "documents": [ ]
  }
}

```

#### 15. Check Duty Status API - [./duty-status](#) - GET

```

Response(200) = {
  'code' => 200,
  'success' => true,
  'message' => 'Duty Status retrieved Successfully',
  'data' => {
    'Duty_status' => 'online' | 'offline'
  }
}

```

## 15. Check Account Status API -[/check-acc-status](#) - GET

```
Response(200) = {  
  'code' => 200,  
  'success' => true,  
  'message' => 'Account Status retrieved Successfully',  
  'data' => {  
    ...data,  
    "is_bank_info_completed": "boolean",  
    "is_other_info_completed": "boolean",  
  }  
}
```

**\*Note:** /fetch-assign-order API HTTP method changed from POST to GET.

## 16. Order Notifications Sending API -[/send-notification](#) - GET

**Query Parameter :** order\_id;

**Example:** \${API\_URL}?order\_id=7

```
Response(200) = {  
  'code' => 200,  
  'success' => true,  
  'message' => "Notification sent to 1 partners, 1 failed",  
  'data' => [  
    'partners_notified' => "number",  
    'failed_notifications' => "number",  
    'order_id' => "number",  
    'order_details' => "JSON",  
    'notification_results' => {  
      'partner_id' => "number",  
      "partner_name" => "string",  
      'status' => 'sent' || 'failed',  
    },  
  ],  
}
```



```
        'error' => " string"
    }
};
```

## 17. Update FCM Token - [/update-fcm-token](#) - POST

```
Payload = {
  "fcm_token": "string",
}
```

```
Response(200) = {
  'code' => 200,
  'success' => true,
  'message' => "FCM token updated successfully",
};
```

**Note:** Whenever a user logs in, call this API to update the valid FCM token for pushing order notifications.

## 18. Accept Order - [/accept](#) - POST

```
Payload = {
  "order_id": "required",
}
```

```
Response(200) = {
  'code' => 200,
  'success' => true,
  'message' => "Order accepted successfully",
  'data' => {
    'order_id' => 'string'
  }
};
```

## 19. Reject Order - [/reject](#) - POST

```
Payload = {
  "order_id": "required",
}
```

```
}
```

```
Response(200) = {  
  'code' => 200,  
  'success' => true,  
  'message' => "Order rejected successfully"  
};
```

## 20. Order Details - [/details/{order\\_id}](#) - GET

```
Response(200) = {  
  'code' => 200,  
  'success' => true,  
  'message' => "Order details fetched successfully",  
  'data' => '$orderData'  
};
```

### Note:

Some APIs HTTP methods are changed from PUT to POST. Changed APIs are,

1. order/status
2. order/complete
3. duty/status
4. profile/update
5. profile/update/bank

## 21. Bank Details - [/bank/info](#) - GET

```
Response(200) = {  
  'code' => 200,  
  'success' => true,  
  'message' => "Order details fetched successfully",  
  'data' => {  
    "partner_id": "number",  
    "name": "string",  
    "acc_number": "number",  
    "sort_code": "string",  
    "is_account_in_your_name": "boolean",  
    "name_on_the_account": "string"  
  }  
};
```

## 21. Other Profile Details - [/other/info](#) - GET

```
Response(200) = {
  'code' => 200,
  'success' => true,
  'message' => "Order details fetched successfully",
  'data' => {
    "partner_id": "number",
    "requires_work_permit": "boolean",
    "uses_car": "boolean",
    "uses_motorcycle": "boolean",
    "uses_bicycle": "boolean",
    "has_motoring_convictions": "boolean",
    "is_uk_licence": "boolean",
    "licence_country_of_issue": "string",
    "has_medical_condition": "boolean",
    "can_be_used_as_reference": "boolean",
    "is_agreed_privacy_policy": "boolean",
  }
};
```

### Note:

- \*\* Update Bank Info API path changed from **/profile/update/bank** to **/profile/bank/update**
- \*\* Account Status API now includes profile completion flag, updated the api response format
- \*\* GetProfileInfo API now include documents data, updated the api response format

## 22. Update Personal Info API - [/personal/update](#) - POST

```
Payload = {
  'title' => 'nullable|string',
```

```

'f_name' => 'nullable|min:3',
's_name' => 'nullable|min:3',
'm_name' => 'nullable|min:1',
'phone_number' => 'nullable|string',
'email' => 'nullable|email',
'dob' => 'nullable|date',
'nationality' => 'nullable|string',
'is_non_british' => 'nullable|boolean',
'home_no' => 'nullable|string|required_without:home_name',
'home_name' => 'nullable|string|required_without:home_no',
'street_name' => 'nullable|string',
'city' => 'nullable|string',
'county' => 'nullable|string',
'post_code' => 'nullable|string',
'home_phone' => 'nullable|string',
}

```

```

Response = {
  "code": "201",
  "success": true,
  "message": "User Created Successfully",
  "data": {
    "user_id": "1"
  }
}

```

**For FCM Token Generation kindly follow the below instructions:**

**FIREBASE VAPID KEY =**

BDO2zsg8Ejyk630jCoqZZKfIWfFbM5OGmBow2PU9-wXIVLc9S55SCp6BEnrfEGeLqGi-j1pNKX  
s8nOK4cLFEN7k (if needed use it)

## × Add Firebase to your Flutter app

### 1 Prepare your workspace

The easiest way to get you started is to use the FlutterFire CLI.

Before you continue, make sure to:

- Install the [Firebase CLI](#) and log in (run `firebase login`)
- Install the [Flutter SDK](#)
- Create a Flutter project (run `flutter create`)

Next

### 2 Install and run the FlutterFire CLI

From any directory, run this command:

```
$ dart pub global activate flutterfire_cli
```



Then, at the root of your Flutter project directory, run this command:

```
$ flutterfire configure --project=driver-app-38a97
```



This automatically registers your per-platform apps with Firebase and adds a `lib/firebase_options.dart` configuration file to your Flutter project.

Previous

Next

### 3 Initialize Firebase and add plugins

To initialize Firebase, call `Firebase.initializeApp` from the `firebase_core` package with the configuration from your new `firebase_options.dart` file:

```
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';

// ...

await Firebase.initializeApp(
  options: DefaultFirebaseOptions.currentPlatform,
);
```

Then, add and begin using the [Flutter plugins](#) for the Firebase products you'd like to use.

Note: If you're using Analytics or Performance Monitoring, you may need to follow a few additional setup steps.

[Previous](#)

[Continue to console](#)

## Note:

- Updated Order Details API - [/details/{order\\_id}](#) POST method to GET method
- Token expiration time has been increased to 1 day.

## 23. Forgot Password API - [/forgot-password](#) - POST

```
Payload = {
  "email": "required" | "string"
}
Response(200) = {
  'code' => 200,
  'success' => true,
  'message' => 'If the email exists in our system, you will receive a password reset code shortly.'
  'otp' => $otp    // for development attached otp code to response
}
```

## 24. Verify Email OTP API - </otp/verify/email> - POST

```
Payload = {  
  "email": "required|string",  
  "otp": "min:4,max:4"  
}  
  
Response(200) = {  
  'code' => 200,  
  'success' => true,  
  'message' => 'OTP Verified',  
  'data' => {  
    'user_id' : 'string| number'  
  }  
}
```

## 25. Reset Password API - </reset-password> - POST

```
Payload = {  
  "user_id": "required" | "string", // will provided on the verify email otp response  
  "new_password": "required|string", // min: 1-uppercase, 1-lowercase, 1-number,  
  1-special_chars  
  "confirm_password": "required|string"  
}  
  
Response(200) = {  
  'code' => 200,  
  'success' => true,  
  'message' => 'Password updated successfully'  
}
```

## 26. Delete Document API - </document/{id}> - DELETE

```
Response(200) = {  
  'code' => 200,  
  'success' => true,  
  'message' => 'Document deleted successfully.'  
}
```

## Note:

- Get Profile API now includes address and document information in the following format:

```
{
  "code": 200,
  "success": true,
  "message": "Profile Info retrieved Successfully",
  "data": {
    "id": 1,
    "title": "Mr.",
    "phone_number": "+4479111123456",
    "duty_status": "online",
    "gender": "male",
    "email": "johndoe@example.com",
    "nationality": "British",
    "f_name": "John",
    "s_name": "Michael",
    "m_name": null,
    "dob": "1990-05-15",
    "is_non_british": false,
    "acc_created_at": "30-06-2025",
    "address_info": {
      "home_no": "221B",
      "home_name": null,
      "street_name": "Baker Street",
      "city": "London",
      "county": null,
      "post_code": "NW1 6XE",
      "home_phone": "+442079110000"
    },
    "documents": [
      {
        "id": 2,
        "doc_type": "license",
        "doc_number": "1234",
        "doc_expiry": "14-11-2025",
        "doc_url": "image"
      }
    ]
  }
}
```



- Send the documents["id"] to delete a particular document
- Now, login and user profile-related APIs allow users to update their details only until the admin verifies their account. Once verified or rejected, the API will no longer allow profile updates.