

# Quick Profile API Documentation

This API allows for the rapid creation of customer profiles (both Individuals and Corporate entities) through a single unified endpoint.

## Endpoint Overview

- **URL:** `{{baseUrl}}/quick-profile/create`
- **Method:** `POST`
- **Auth Required:** Yes (Bearer Token)
- **Content-Type:** `multipart/form-data`

## Request Fields (multipart/form-data)

Since this endpoint supports file uploads for ID documents, you must send the data as `form-data`.

### 1. General Fields

Field Name	Type	Required	Description
customerType	String	Yes	Either Individual or Corporate.
customerName	String	Yes	Name of the person or company.
dob	Date String	Yes	<b>Individual:</b> Date of Birth. <b>Corporate:</b> Incorporation Date.
nationality	String	Yes	<b>Individual:</b> Citizenship. <b>Corporate:</b> Country of Incorporation.
birthCountry	String	No	<b>Individual:</b> Country of Birth.
profession	String	No	<b>Individual:</b> Job title/Profession.
email	String	No	Contact email address.
mobile	String	No	Contact mobile number.
pepStatus	String	No	YES or NO. Defaults to NO.
residentStatus	String	No	e.g., Resident, Non-Resident.

### 2. Address Fields

Field Name	Type	Required	Description
addressLine1	String	No	Street address or Area.
city	String	No	City or Emirate.
poBox	String	No	P.O. Box number.

### 3. Identity / License Details

These fields use array-like bracket notation to support deep object mapping in the backend.

Field Name	Type	Required	Description
idDetails[0][idType]	String	Yes	e.g. Passport, Trade License, National ID.
idDetails[0][idNumber]	String	Yes	The ID or License number.
idDetails[0][expiryDate]	Date String	Yes	Expiry date of the document.
idDetails[0][file]	File	Yes	<b>The actual file upload binary.</b>

## Backend Routing Logic

The backend intelligently routes the data based on `customerType` :

- If **Individual** : Saves to the standard `IndividualProfile` collection.
- If **Corporate** : Saves to the `CorporateProfile` collection. We automatically map:
  - `dob` -> `incorporationDate`
  - `nationality` -> `country`
  - `idNumber` -> `tradeLicenseNumber`
  - `idDetails[0][file]` -> `tradeLicenseFile`

## Sample Response

### Success (201 Created)

```
{
  "success": true,
  "message": "Corporate Quick Profile created successfully",
  "profile": {
    "id": "67a1da...",
    "userId": "67a102...",
    "customerType": "Corporate",
    "name": "Global Tech Ventures LLC",
    "incorporationDate": "2015-03-10T00:00:00.000Z",
    "country": "United Arab Emirates",
    "screening": {
      "status": "PENDING",
      "searchBy": "Name"
    },
    "meta": {
      "createdAt": "2024-02-04T11:05:00.000Z",
      "updatedAt": "2024-02-04T11:05:00.000Z"
    }
  }
}
```

### Error (400 Bad Request)

```
{
  "success": false,
  "message": "ValidationError: idDetails.0.idType is required"
}
```

---

## Integration Tip

When using `axios` or `fetch`, do **not** manually set the `Content-Type` header to `multipart/form-data`. Let the browser/environment set it automatically with the correct boundary.

```
const formData = new FormData();
formData.append('customerType', 'Individual');
formData.append('customerName', 'John Doe');
// ... append other fields ...
formData.append('idDetails[0][file]', fileInput.files[0]);

axios.post('/quick-profile/create', formData);
```