

API Documentation

Welcome to the API documentation for Yexah. This document provides information about the available endpoints and how to use them.

Register [/register]

Register User [POST]

Register a new user.

- ****Request:****
 - Method: POST
 - URL: `/register`
 - Headers:
 - `Content-Type: application/json`
 - Body:
 - ````json`
 - `{`
 - `"fullname": "John Doe",`
 - `"email": "john@example.com",`
 - `"password": "password123",`
 - `"isAdmin": false`
 - `}`
 - `````
- ****Response:****
 - Status: 201 Created
 - Body:
 - ````json`
 - `{`
 - `"success": true,`
 - `"message": "User registered successfully."`
 - `}`
 - `````

Login [/login]

Login User [POST]

Login an existing user.

- ****Request:****
 - Method: POST

```

- URL: `/login`
- Headers:
  - `Content-Type: application/json`
- Body:
  ```json
 {
 "email": "john@example.com",
 "password": "password123"
 }
  ```

- **Response:**
  - Status: 200 OK
  - Body:
    ```json
 {
 "success": true,
 "message": "Login successful.",
 "user": { /* User object */ },
 "token": "your-auth-token"
 }
    ```

```

Forgot Password [/send-reset-email]

Request Password Reset [POST]

Send a password reset email to the user.

```

- **Request:**
  - Method: POST
  - URL: `/send-reset-email`
  - Headers:
    - `Content-Type: application/json`
  - Body:
    ```json
 {
 "email": "john@example.com"
 }
    ```

- **Response:**
  - Status: 200 OK
  - Body:

```

```
```json
{
 "success": true,
 "message": "Password reset token sent successfully."
}
```
```

Reset Password [/reset-password]

Reset User Password [POST]

Reset the user's password using the provided token.

```
- **Request:**
- Method: POST
- URL: `/reset-password`
- Headers:
  - `Content-Type: application/json`
- Body:
  ```json
 {
 "email": "john@example.com",
 "token": "reset-token",
 "newPassword": "new-password123"
 }
  ```

- **Response:**
- Status: 200 OK
- Body:
  ```json
 {
 "success": true,
 "message": "Password reset successful."
 }
  ```
```

Insert Provider Plan Reference [/insert_providerplanref]

Insert Provider Plan Reference [POST]

Insert a new provider plan reference into the database.

- ****Request:****
 - Method: POST
 - URL: `/insert_providerplanref``
 - Headers:
 - ``Content-Type: application/json``
 - Body:
 - ````json`
 - ```
{
 "user_id": "user123",
 "how_often": "monthly",
 "markupPercentage": 10,
 "repairRequest": true,
 "basePrice": 100,
 "markupMoney": 15,
 "markup": "percentage",
 "confirmation_for_customer_support": true,
 "party2Name": "Provider Corp",
 "party2Address": "123 Provider St, City",
 "city": "City",
 "postalCode": "12345",
 "country": "Country",
 "website": "https://provider.com",
 "party2PAN": "ABCDE1234F",
 "party2GST": "GST12345",
 "authSignatoryName": "John Doe",
 "designation": "CEO",
 "authSignatoryEmailAddress": "john@example.com",
 "authSignatoryPhoneNumber": "+123456789",
 "authSignatoryAadharNumber": "123456789012",
 "selectedGadgets": "gadget1",
 "selectedNumberOfGadgets": 2
}
```
    - `````
- **\*\*Response:\*\***
  - Status: 200 OK
  - Body: Success message
    - ````json`
    - ```
{
  "message": "Provider plan reference inserted successfully"
}
```
 - `````
- ****Error Responses:****

- Status: 500 Internal Server Error
- Body: Error message

```
```json
{
 "error": "Internal Server Error"
}
```
```

Send PDF by Email [/sendPDFByEmail]

Send PDF by Email [POST]

Send the generated PDF by email.

- ****Request:****
 - Method: POST
 - URL: `/sendPDFByEmail`
 - Headers:
 - `Content-Type: application/json`
 - Body:

```
```json
{
 "username": "john@example.com"
}
```
```
- ****Response:****
 - Status: 200 OK
 - Body:

```
```json
{
 "message": "Email sent successfully"
}
```
```

Get Provider Plan Ref [/get_providerplanref/:user_id]

Get Provider Plan Ref [GET]

Get data of a particular user from the providerref table.

- ****Request:****

- Method: GET
- URL: `/get_providerplanref/:user_id`
- ****Response:****
 - Status: 200 OK
 - Body:

```
```json
{
 "success": true,
 "providerData": { /* Provider data */ }
}
```
```

Get Transactions [/get_transactions/:user_id]

Get Transactions [GET]

Get transactions data of a particular user.

- ****Request:****
 - Method: GET
 - URL: `/get_transactions/:user_id`
- ****Response:****
 - Status: 200 OK
 - Body:

```
```json
{
 "success": true,
 "transactionsData": [/* Array of transactions */]
}
```
```

Create Transaction [/transactions]

Create Transaction [POST]

Create a new transaction.

- ****Request:****
 - Method: POST
 - URL: `/transactions`
 - Headers:
 - `Content-Type: application/json`

```

- Body:
  ```json
 {
 /* Transaction data */
 }
  ```

- **Response:**
  - Status: 200 OK
  - Body:
    ```json
 {
 "success": true,
 "message": "Transaction created successfully."
 }
    ```

```

Upload Excel [/upload]

Upload Excel File [POST]

Upload an Excel file containing transaction data.

```

- **Request:**
  - Method: POST
  - URL: `/upload`
  - Headers:
    - `Content-Type: multipart/form-data`
  - Form Data:
    - `excelFile`: Excel file to upload

- **Response:**
  - Status: 200 OK
  - Body:
    ```json
 {
 "message": "Data inserted successfully."
 }
    ```

```

Service Tickets [/servicetickets]

Create Service Ticket [POST]

Create a new service ticket.

```
- **Request:**
- Method: POST
- URL: `/servicetickets`
- Headers:
  - `Content-Type: application/json`
- Body:
  ```json
 {
 "providerrefno": "PR123",
 "status": "Open",
 "date": "2023-08-23",
 "issue": "Device not working",
 "planstart": "2023-08-25",
 "planvalidity": "2023-12-25",
 "customer": "John Doe",
 "myinvoiceno": "INV456",
 "myinvoicedate": "2023-08-22",
 "user_id": 1
 }
  ```

- **Response:**
- Status: 200 OK
- Body:
  ```json
 {
 "success": true,
 "message": "Service ticket raised successfully."
 }
  ```
```

Get Service Tickets [GET]

Get service tickets for a specific user.

```
- **Request:**
- Method: GET
- URL: `/servicetickets/:user_id`

- **Response:**
- Status: 200 OK
```



```
- Body:
  ```json
 {
 "success": true,
 "serviceTickets": [/* Array of service ticket objects */]
 }
  ```
```

Provider Reference [/providerref]

Create Provider Reference [POST]

Create a new provider reference.

```
- **Request:**
  - Method: POST
  - URL: `/providerref`
  - Headers:
    - `Content-Type: application/json`
  - Body:
    ```json
 {
 /* Provider reference data */
 }
    ```

- **Response:**
  - Status: 200 OK
  - Body:
    ```json
 {
 "message": "Data successfully stored in the database"
 }
    ```
```

Send PDF by Email [/sendPDFByEmail]

Send PDF by Email [POST]

Send a generated PDF by email.

```
- **Request:**
  - Method: POST
  - URL: `/sendPDFByEmail`
```

- Headers:
 - `Content-Type: application/json`
- Body:


```
```json
{
 "username": "john@example.com"
}
```
```
- ****Response:****
 - Status: 200 OK
 - Body:


```
```json
{
 "message": "Email sent successfully"
}
```
```

Document Management [/documents]

Get Document by ID [GET]

Get a document by its ID.

- ****Request:****
 - Method: GET
 - URL: `/documents/:doc_id`
- ****Response:****
 - Status: 200 OK
 - Body: Document object

Save Signatures [/save-signature]

Save User Signature [POST]

Save a user's signature.

- ****Request:****
 - Method: POST
 - URL: `/save-signature`
 - Headers:
 - `Content-Type: application/json`
 - Body:

```

    ````json
 {
 "user_id": 1,
 "signature_image": "base64-encoded-image-data"
 }
    ````

- **Response:**
  - Status: 200 OK
  - Body:
    ````json
 {
 "message": "Signature saved successfully"
 }
    ````

## Save Images and Documents [/save-imagedata]

### Save User Images and Documents [POST]

Save user images and documents.

- **Request:**
  - Method: POST
  - URL: `/save-imagedata`
  - Headers:
    - `Content-Type: application/json`
  - Body:
    ````json
 {
 "user_id": 1,
 "gst_image": "base64-encoded-image-data",
 "adharcard_image": "base64-encoded-image-data",
 "pancard_image": "base64-encoded-image-data",
 "logo_image": "base64-encoded-image-data"
 }
    ````

- **Response:**
  - Status: 200 OK
  - Body:
    ````json
 {
 "message": "Documents saved successfully"
 }
    ````

```

```
}  
```
```

## Save Admin Signature [/save-admin-signature]

### Save Admin Signature [POST]

Save an admin's signature.

```
- **Request:**
 - Method: POST
 - URL: `/save-admin-signature`
 - Headers:
 - `Content-Type: application/json`
 - Body:
 ``json
 {
 "user_id": 1,
 "signature_image": "base64-encoded-image-data"
 }
 ``

- **Response:**
 - Status: 200 OK
 - Body:
 ``json
 {
 "message": "Signature saved successfully"
 }
 ``
```

Generate an agreement document with Yexah's signature.

```
- **Request:**
 - Method: POST
 - URL: `/generateAgreementyexahSignature1`
 - Headers:
 - `Content-Type: application/json`
 - Body:
 ``json
 {
 "id": 1,
 "party2Name": "Party 2 Name",
 "party2Address": "Party 2 Address",
 }
```

```
"party2PAN": "PAN123",
"party2GST": "GST456",
"authSignatoryName": "John Doe",
"designation": "CEO",
"authSignatoryEmailAddress": "john@example.com",
"authSignatoryPhoneNumber": "123-456-7890",
"signatureImageData": "base64-encoded-image-data"
}
```
```

- ****Response:****
 - Status: 200 OK
 - Body: Agreement document in PDF format

Generate an draft agreement document .

- ****Request:****
 - Method: POST
 - URL: `/generateAgreementyexahSignature2`
 - Headers:
 - `Content-Type: application/json`
 - Body:

```
```json
{
 "id": 1,
 "party2Name": "Party 2 Name",
 "party2Address": "Party 2 Address",
 "party2PAN": "PAN123",
 "party2GST": "GST456",
 "authSignatoryName": "John Doe",
 "designation": "CEO",
 "authSignatoryEmailAddress": "john@example.com",
 "authSignatoryPhoneNumber": "123-456-7890",

}
```
```

- ****Response:****
 - Status: 200 OK
 - Body: Agreement document in PDF format

Generate a report based on transaction details.

- ****Request:****

- Method: POST
- URL: `/generatereport``
- Headers:
 - `Content-Type: application/json``
- Body:

```
```json
{
 "user_id": 123,
 "providerefno": "PR0123",
 "planstart": "2023-08-01",
 "planvalidity": "2024-08-01",
 "invoicedate": "2023-08-15",
 "invoiced": true,
 "customername": "John Doe",
 "email": "john@example.com",
 "transaction_value": 500
}
```
```
- ****Response:****
 - Status: 200 OK
 - Body: Report document in PDF format

Get Users `[/getusers]`

Get All Users `[GET]`

Get a list of all users.

- ****Request:****
 - Method: GET
 - URL: `/getusers``
 - Headers:
 - `Content-Type: application/json``
- ****Response:****
 - Status: 200 OK
 - Body: List of user objects

```
```json
[
```

```

{
 "id": 1,
 "name": "John Doe",
 "email": "john@example.com",
 ...
},
{
 "id": 2,
 "name": "Jane Smith",
 "email": "jane@example.com",
 ...
},
...
]
```

```

Get Signatures [/getsignatures]

Get All Signatures [GET]

Get a list of all user signatures.

```

- **Request:**
  - Method: GET
  - URL: `/getsignatures`
  - Headers:
    - `Content-Type: application/json`

- **Response:**
  - Status: 200 OK
  - Body: List of signature objects
    ```json
 [
 {
 "id": 1,
 "user_id": 1,
 "signature_image": "base64signaturedata...",
 ...
 },
 {
 "id": 2,
 "user_id": 2,
 "signature_image": "base64signaturedata...",
 ...
 }
]
    ```

```

```
    },  
    ...  
  ]  
  ...  
}
```

Approve Contract [/approveContract/{userId}]

Approve Contract [POST]

Approve a user's contract.

- ****Request:****
 - Method: POST
 - URL: ``/approveContract/{userId}``
 - URL Parameters:
 - ``userId`` (string): ID of the user to approve
 - Headers:
 - ``Content-Type: application/json``
- ****Response:****
 - Status: 200 OK

Configure Contract [/configureContract/{userId}]

Configure Contract [POST]

Mark a user's contract as configured.

- ****Request:****
 - Method: POST
 - URL: ``/configureContract/{userId}``
 - URL Parameters:
 - ``userId`` (string): ID of the user to mark as configured
 - Headers:
 - ``Content-Type: application/json``
- ****Response:****
 - Status: 200 OK

Get Configured Users [/getconfiguredusers]

Get Configured Users [GET]

Get a list of all users who have configured their contract.


```

- **Request:**
  - Method: GET
  - URL: `/getconfiguredusers`
  - Headers:
    - `Content-Type: application/json`

- **Response:**
  - Status: 200 OK
  - Body: List of configured user objects
    ```json
 [
 {
 "id": 1,
 "name": "John Doe",
 "email": "john@example.com",
 ...
 },
 {
 "id": 2,
 "name": "Jane Smith",
 "email": "jane@example.com",
 ...
 },
 ...
]
    ```

```

Get Configured Users Not Approved [/getconfiguredusersnotapproved]

Get Configured Users Not Approved [GET]

Get a list of configured users whose contract is not yet approved.

```

- **Request:**
  - Method: GET
  - URL: `/getconfiguredusersnotapproved`
  - Headers:
    - `Content-Type: application/json`

- **Response:**
  - Status: 200 OK
  - Body: List of configured user objects

```

Get Approved Configured Users [/getapprovedconfiguredusers]

Get Approved Configured Users [GET]

Get a list of configured users whose contract is approved.

- ****Request:****
 - Method: GET
 - URL: `/getapprovedconfiguredusers`
 - Headers:
 - `Content-Type: application/json`
- ****Response:****
 - Status: 200 OK
 - Body: List of configured user objects

Get Transactions [/get_transactions]

Get All Transactions [GET]

Get a list of all transactions.

- ****Request:****
 - Method: GET
 - URL: `/get_transactions`
 - Headers:
 - `Content-Type: application/json`
- ****Response:****
 - Status: 200 OK
 - Body: List of transaction objects

Send Email [/send-email]

Send Email [POST]

Send an email to a recipient.

- ****Request:****
 - Method: POST
 - URL: `/send-email`
 - Headers:
 - `Content-Type: application/json`
 - Body:

```
```json
{
 "email": "recipient@example.com",
 "text": "Hello, this is the email text."
}
```
```

```
- **Response:**
- Status: 200 OK
```

Send Email New Contract for Approval [/send-email-new-contract-is-for-approve]

Send Email New Contract for Approval [POST]

Send an email notification about a new contract that requires approval.

```
- **Request:**
- Method: POST
- URL: `/send-email-new-contract-is-for-approve`
- Headers:
  - `Content-Type: application/json`
- Body:
  ```json
 {
 "email": "recipient@example.com",
 "text": "A new contract is awaiting approval."
 }
  ```
```

```
- **Response:**
- Status: 200 OK
```

Send Email Contract Approved [/send-email-contract-approved]

Send Email Contract Approved [POST]

Send an email notification about a contract being approved.

```
- **Request:**
- Method: POST
- URL: `/send-email-contract-approved`
- Headers:
```

```

- `Content-Type: application/json`
- Body:
  ```json
 {
 "email": "recipient@example.com",
 "text": "Congratulations! Your contract has been approved."
 }
  ```

- **Response:**
  - Status: 200 OK

```

Get User by ID [/getusers/{id}]

Get User by ID [POST]

Get user details by their ID.

```

- **Request:**
  - Method: GET
  - URL: `/getusers/{id}`
  - URL Parameters:
    - `id` (string): ID of the user to retrieve
  - Headers:
    - `Content-Type: application/json`

- **Response:**
  - Status: 200 OK
  - Body: User object

```

API Documentation

Upload PDF [/upload-pdf/{userId}]

Upload and Save PDF [POST]

Upload a PDF file and save it as a BLOB in the database for a specific user.

```

- **Request:**
  - Method: POST
  - URL: `/upload-pdf/{userId}`
  - URL Parameters:
    - `userId` (string): ID of the user to associate the PDF with

```

- Headers:
 - `Content-Type: multipart/form-data`
- Form Data:
 - `pdf` (file): The PDF file to upload
- ****Response:****
 - Status: 200 OK
 - Body: Success message

```
```json
{
 "message": "PDF uploaded and saved successfully"
}
```
```
- ****Error Responses:****
 - Status: 500 Internal Server Error
 - Body: Error message

```
```json
{
 "error": "Internal Server Error"
}
```
```