

Crear Detalles Cliente API

Sudipta Kumar • April 29, 2024



Anonymous

```
@Post('/signup/customer_details')
@UsePipes(new ValidationPipe())
@HttpCode(HttpStatus.OK) // Set the status code to 200 (OK)
async customer_Details_Create(
  @Body() customer_info: CustomerDto,
): Promise<any> {
  try {
    const saved_customer =
      await this.customerService.Create_Customer(customer_info);
    if (saved_customer > 0) {
      return saved_customer;
    } else {
      throw new InternalServerErrorException(
        'customer data could not be saved',
      );
    }
  } catch (e) {
    throw new InternalServerErrorException({
      status: HttpStatus.INTERNAL_SERVER_ERROR,
      message: e.message,
    });
  }
}
```

give me proper documentations for this api endpoint



ChatGPT

Certainly! Here's a structured documentation for your API endpoint:

API Endpoint: Create Customer Details

Description

This API endpoint allows users to create customer details by providing necessary information.

URL

bash

Copy code

```
POST /signup/customer_details
```

Request

Headers

- **Content-Type:** application/json

Body Parameters

Parameter	Type	Description
customer_info	Object	An object containing customer details.

`customer_info` Object

Field	Type	Description
name	String	The name of the customer.
email	String	The email address of the customer.
address	String	The address of the customer.
phoneNumber	String	The phone number of the customer.

Example Request

```
json Copy code

POST /signup/customer_details
{
  "customer_info": {
    "name": "John Doe",
    "email": "johndoe@example.com",
    "address": "123 Main Street",
    "phoneNumber": "123-456-7890"
  }
}
```

Response

Success Response

- **Status Code:** 200 (OK)
- **Content-Type:** application/json

Body Parameters

Parameter	Type	Description
saved_customer	Object	The saved customer details.

`saved_customer` Object

Field	Type	Description
id	Number	The unique identifier of the customer.
name	String	The name of the customer.
email	String	The email address of the customer.
address	String	The address of the customer.
phoneNumber	String	The phone number of the customer.

Example Success Response

```
json Copy code
```

```
HTTP/1.1 200 OK
{
  "id": 1,
  "name": "John Doe",
  "email": "johndoe@example.com",
  "address": "123 Main Street",
  "phoneNumber": "123-456-7890"
}
```

Error Responses

- **Status Code:** 500 (Internal Server Error)
- **Content-Type:** application/json

Body Parameters

Parameter	Type	Description
status	Number	The HTTP status code.
message	String	The error message.

Example Error Response

```
json Copy code
HTTP/1.1 500 Internal Server Error
{
  "status": 500,
  "message": "customer data could not be saved"
}
```

Errors

- If the request body is invalid, the API will return a 400 Bad Request error.
- If there's an internal server error during the process, the API will return a 500 Internal Server Error.

Feel free to adjust the documentation as needed, and include any additional details specific to your application or requirements!



Anonymous

```
@Get('/profile')
@UseGuards(AuthGuard)
@HttpCode(HttpStatus.OK) // Set the status code to 200 (OK)
async View_own_Profile(@Request() req): Promise<any> {
  try {
    return await this.customerService.Find_Customer_By_Email(req.user.email);
  } catch (e) {
    throw new InternalServerErrorException(e.message);
  }
}
```

give me proper documentations for this api endpoint



ChatGPT

Here's the documentation for your ``/profile`` endpoint:

API Endpoint: View Own Profile

Description

This API endpoint allows authenticated users to view their own profile information.

URL

bash

Copy code

```
GET /profile
```

Request

Headers

- Authorization: Bearer [JWT Token]

Example Request

http

Copy code

```
GET /profile
Authorization: Bearer [JWT Token]
```

Response

Success Response

- **Status Code:** 200 (OK)
- **Content-Type:** application/json

Body Parameters

Parameter	Type	Description
id	Number	The unique identifier of the customer.

Parameter	Type	Description
name	String	The name of the customer.
email	String	The email address of the customer.
address	String	The address of the customer.
phoneNumber	String	The phone number of the customer.

Example Success Response

```

json Copy code

HTTP/1.1 200 OK
{
  "id": 1,
  "name": "John Doe",
  "email": "johndoe@example.com",
  "address": "123 Main Street",
  "phoneNumber": "123-456-7890"
}
```

Error Responses

- **Status Code:** 401 (Unauthorized)
 - If the request does not include a valid JWT token.
- **Status Code:** 500 (Internal Server Error)
 - If there's an internal server error during the process.

Example Error Response

```

json Copy code

HTTP/1.1 401 Unauthorized
{
  "message": "Unauthorized"
}
```

Errors

- If the request is unauthorized (i.e., missing or invalid JWT token), the API will return a 401 Unauthorized error.
- If there's an internal server error during the process, the API will return a 500 Internal Server Error.

You may need to adjust the documentation according to your application's specific authentication mechanism and error handling requirements.



Anonymous

```

// #3
@Put('/profile/update')
@UseGuards(AuthGuard)
@UsePipes(new ValidationPipe())
@HttpCode(HttpStatus.OK) // Set the status code to 200 (OK)
async Update_own_Profile(
```

```
@Request() req,
@Body() updated_data: Customer_ProfileDTO,
): Promise<any> {
  try {
    return await this.customerService.Update_Own_Profile_Details(
      req.user.email,
      updated_data,
    );
  } catch (e) {
    throw new InternalServerErrorException(e.message);
  }
}
```

give me proper documentations for this api endpoint



ChatGPT

Here's the documentation for your `/profile/update` endpoint:

API Endpoint: Update Own Profile

Description

This API endpoint allows authenticated users to update their own profile information.

URL

bash

Copy code

```
PUT /profile/update
```

Request

Headers

- Authorization: Bearer [JWT Token]
- Content-Type: application/json

Body Parameters

Parameter	Type	Description
updated_data	Object	An object containing updated profile details.

`updated_data` Object

Field	Type	Description
name	String	(Optional) The updated name of the customer.
address	String	(Optional) The updated address of the customer.
phoneNumber	String	(Optional) The updated phone number of the customer.

Example Request

http

Copy code

```
PUT /profile/update
Authorization: Bearer [JWT Token]
Content-Type: application/json

{
  "name": "Updated Name",
  "address": "Updated Address",
  "phoneNumber": "123-456-7890"
}
```

Response

Success Response


- **Status Code:** 200 (OK)
- **Content-Type:** application/json

Body Parameters

Parameter	Type	Description
id	Number	The unique identifier of the customer.
name	String	The updated name of the customer.
address	String	The updated address of the customer.
phoneNumber	String	The updated phone number of the customer.

Example Success Response

json

 Copy code


```
HTTP/1.1 200 OK
{
  "id": 1,
  "name": "Updated Name",
  "address": "Updated Address",
  "phoneNumber": "123-456-7890"
}
```

Error Responses

- **Status Code:** 400 (Bad Request)
 - If the request body is missing or malformed.
- **Status Code:** 401 (Unauthorized)
 - If the request does not include a valid JWT token.
- **Status Code:** 500 (Internal Server Error)
 - If there's an internal server error during the process.

Example Error Response

json

 Copy code

```
HTTP/1.1 400 Bad Request
{
```

```
"message": "Bad Request"
}
```

Errors

- If the request body is missing or malformed, the API will return a 400 Bad Request error.
- If the request is unauthorized (i.e., missing or invalid JWT token), the API will return a 401 Unauthorized error.
- If there's an internal server error during the process, the API will return a 500 Internal Server Error.

Feel free to adjust the documentation according to your application's specific requirements and error handling logic!



Anonymous

```
// # : Upload & Update Customer Image
@Put('/profile/upload')
@UseGuards(AuthGuard)
@HttpCode(HttpStatus.OK) // Set the status code to 200 (OK)
@UseInterceptors(
  FileInterceptor('myfile', {
    fileFilter: (req, file, cb) => {
      if (file.originalname.match(/^.*(.jpg|.webp|.png|.jpeg)$/))
        cb(null, true);
      else {
        cb(new MulterError('LIMIT_UNEXPECTED_FILE', 'image'), false);
      }
    },
    limits: { fileSize: 5000000 }, // 5 MB
    storage: diskStorage({
      destination: './assets/profile_images',
      filename: function (req, file, cb) {
        cb(null, Date.now() + file.originalname);
      },
    }),
  }),
)
async UploadProfileImage(
  @Request() req,
  @UploadedFile() myfileobj: Express.Multer.File,
): Promise<any> {
  console.log(myfileobj); // We can find the file name here
  if (myfileobj == null) {
    throw new BadRequestException({
      status: HttpStatus.BAD_REQUEST,
      message: 'Please Upload Image',
    });
  }
  const seller = await this.customerService.Update_Profile_Picture(
    req.user.email,
    myfileobj.filename,
  );
  if (seller != null) {
    return seller;
  }
}
```



```
} else {
  throw new NotFoundException({
    status: HttpStatus.NOT_FOUND,
    message: 'No Seller Found to Upload Seller Image',
  });
}
```

give me proper documentations for this api endpoint



ChatGPT

Sure, here's the documentation for your `/profile/upload` endpoint:

API Endpoint: Upload Profile Image

Description

This API endpoint allows authenticated users to upload and update their profile image.

URL

bash

Copy code

```
PUT /profile/upload
```

Request

Headers

- Authorization: Bearer [JWT Token]
- Content-Type: multipart/form-data

Body Parameters

Parameter	Type	Description
myfile	File (Image)	The image file to be uploaded.

Constraints

- Allowed image formats: JPG, JPEG, PNG, WEBP
- Maximum file size: 5 MB

Example Request

http

Copy code

```
PUT /profile/upload
Authorization: Bearer [JWT Token]
Content-Type: multipart/form-data

myfile: [image file]
```

Response

Success Response

- **Status Code:** 200 (OK)
- **Content-Type:** application/json

Body Parameters

Parameter	Type	Description
id	Number	The unique identifier of the customer.
profileImage	String	The filename of the uploaded profile image.

Example Success Response

```
json Copy code

HTTP/1.1 200 OK
{
  "id": 1,
  "profileImage": "example_image.jpg"
}
```

Error Responses

- **Status Code:** 400 (Bad Request)
 - If the request body is missing or malformed.
- **Status Code:** 401 (Unauthorized)
 - If the request does not include a valid JWT token.
- **Status Code:** 404 (Not Found)
 - If no seller is found to upload the profile image.
- **Status Code:** 500 (Internal Server Error)
 - If there's an internal server error during the process.

Example Error Response

```
json Copy code

HTTP/1.1 400 Bad Request
{
  "message": "Please Upload Image"
}
```

Errors

- If the request body is missing or malformed, the API will return a 400 Bad Request error.
- If the request is unauthorized (i.e., missing or invalid JWT token), the API will return a 401 Unauthorized error.
- If no seller is found to upload the profile image, the API will return a 404 Not Found error.
- If there's an internal server error during the process, the API will return a 500 Internal Server Error.

Feel free to adjust the documentation according to your application's specific requirements and error handling logic!



Anonymous

```
@Get('/profile/view_profile_image')
@UseGuards(AuthGuard)
@HttpCode(HttpStatus.OK) // Set the status code to 200 (OK)
async getSellerImages(@Request() req, @Res() res): Promise<any> {
  try {
    return this.customerService.Get_Profile_Picture(req.user.email, res);
  } catch (e) {
    throw new InternalServerErrorException(e.message);
  }
}
```

give me proper documentations for this api endpoint and success and error response



ChatGPT

Here's the documentation for your ``/profile/view_profile_image`` endpoint:

API Endpoint: View Profile Image

Description

This API endpoint allows authenticated users to view their profile image.

URL

bash

Copy code

```
GET /profile/view_profile_image
```

Request

Headers

- Authorization: Bearer [JWT Token]

Example Request

http

Copy code

```
GET /profile/view_profile_image
Authorization: Bearer [JWT Token]
```

Response

Success Response

- **Status Code:** 200 (OK)
- **Content-Type:** image/*

The image data will be directly sent in the response body.

Error Responses

- **Status Code:** 401 (Unauthorized)
 - If the request does not include a valid JWT token.
- **Status Code:** 404 (Not Found)
 - If no profile image is found for the user.
- **Status Code:** 500 (Internal Server Error)
 - If there's an internal server error during the process.

Example Error Response

```

json Copy code

HTTP/1.1 404 Not Found
{
  "message": "No Profile Image Found"
}
```

Errors

- If the request is unauthorized (i.e., missing or invalid JWT token), the API will return a 401 Unauthorized error.
- If no profile image is found for the user, the API will return a 404 Not Found error.
- If there's an internal server error during the process, the API will return a 500 Internal Server Error.

Feel free to adjust the documentation according to your application's specific requirements and error handling logic!



Anonymous

```

@Post('/send_money')
@UseGuards(AuthGuard)
@UsePipes(new ValidationPipe())
@HttpCode(HttpStatus.OK) // Set the status code to 200 (OK)
async Create_Billing(
  @Request() req,
  @Body() bill: Payment_ReceiverDTO,
): Promise<any> {
  try {
    const user_validity_decision = await this.customerService.user_validity(
      req.user.email,
      bill.password,
    );

    if (user_validity_decision) {
      bill.payment_type = 'Send Money';
      const decision = await this.customerService.Subtract_Credits_Amount(
        req.user.email,
        bill,
      );

      if (decision > 0) {
        return {
          success: true,
          message: 'Money has been transferred successfully',
        };
      }
    }
  }
}
```

```
} else {
    throw new InternalServerErrorException(
        'Payment Could not be completed',
    );
}
} else {
    throw new BadRequestException('Password did not matched!');
}
} catch (e) {
    throw new InternalServerErrorException(
        'Customer Controller Create Billing Error = ' + e.message,
    );
}
}
```

give me proper documentations for this api endpoint



ChatGPT

Here's the documentation for your `/send_money` endpoint:

API Endpoint: Send Money

Description

This API endpoint allows authenticated users to send money to another user.

URL

bash

Copy code

```
POST /send_money
```

Request

Headers

- Authorization: Bearer [JWT Token]
- Content-Type: application/json

Body Parameters

Parameter	Type	Description
receiver_id	String	The email address of the receiver.
amount	Number	The amount of money to be sent.
password	String	The password for validation (to confirm the transaction).

Example Request

http

Copy code

```
POST /send_money
Authorization: Bearer [JWT Token]
```

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

```
{
  "receiver_id": "receiver@example.com",
  "amount": 100,
  "password": "user_password"
}
```

Response

Success Response

- **Status Code:** 200 (OK)
- **Content-Type:** application/json

Field	Type	Description
success	Boolean	Indicates if the transaction was successful.
message	String	A message indicating the result of the transaction.

Example Success Response

```
json
```

[Copy code](#)

```
HTTP/1.1 200 OK
{
  "success": true,
  "message": "Money has been transferred successfully"
}
```

Error Responses

- **Status Code:** 400 (Bad Request)
 - If the request body is missing or malformed, or if the password does not match.
- **Status Code:** 401 (Unauthorized)
 - If the request does not include a valid JWT token.
- **Status Code:** 500 (Internal Server Error)
 - If there's an internal server error during the process.

Example Error Response

```
json
```

[Copy code](#)

```
HTTP/1.1 400 Bad Request
{
  "message": "Password did not match!"
}
```

Errors

- If the request body is missing or malformed, or if the password does not match, the API will return a 400 Bad Request error.
- If the request is unauthorized (i.e., missing or invalid JWT token), the API will return a 401 Unauthorized error.
- If there's an internal server error during the process, the API will return a 500 Internal Server Error.

Feel free to adjust the documentation according to your application's specific requirements and error handling logic!



Anonymous

```
@Post('/cash_out')
@UseGuards(AuthGuard)
@UsePipes(new ValidationPipe())
@HttpCode(HttpStatus.OK) // Set the status code to 200 (OK)
async Cash_Out(
  @Request() req,
  @Body() bill: Payment_ReceiverDTO,
): Promise<any> {
  try {
    const user_validity_decision = await this.customerService.user_validity(
      req.user.email,
      bill.password,
    );

    if (user_validity_decision) {
      bill.payment_type = 'Cash Out';
      const decision = await this.customerService.Subtract_Credits_Amount(
        req.user.email,
        bill,
      );

      if (decision > 0) {
        return {
          success: true,
          message: 'Money has been transferred successfully',
        };
      } else {
        throw new InternalServerErrorException(
          'Payment Could not be completed',
        );
      }
    } else {
      throw new BadRequestException('Password did not matched!');
    }
  } catch (e) {
    throw new InternalServerErrorException(
      'Customer Controller Create Billing Error = ' + e.message,
    );
  }
}
```

give me proper documentations for this api endpoint



ChatGPT

Here's the documentation for your `/cash_out` endpoint:

API Endpoint: Cash Out

Description

This API endpoint allows authenticated users to cash out their credits.

URL

```
bash
```

[Copy code](#)

```
POST /cash_out
```

Request

Headers

- Authorization: Bearer [JWT Token]
- Content-Type: application/json

Body Parameters

Parameter	Type	Description
amount	Number	The amount of money to be cashed out.
password	String	The password for validation (to confirm the transaction).

Example Request

```
http
```

[Copy code](#)

```
POST /cash_out
Authorization: Bearer [JWT Token]
Content-Type: application/json

{
  "amount": 100,
  "password": "user_password"
}
```

Response

Success Response

- **Status Code:** 200 (OK)
- **Content-Type:** application/json

Field	Type	Description
success	Boolean	Indicates if the transaction was successful.
message	String	A message indicating the result of the transaction.

Example Success Response

```
json
```

[Copy code](#)

```
HTTP/1.1 200 OK
{
  "success": true,
```



```
"message": "Money has been transferred successfully"
}
```

Error Responses

- **Status Code:** 400 (Bad Request)
 - If the request body is missing or malformed, or if the password does not match.
- **Status Code:** 401 (Unauthorized)
 - If the request does not include a valid JWT token.
- **Status Code:** 500 (Internal Server Error)
 - If there's an internal server error during the process.

Example Error Response

```
json Copy code

HTTP/1.1 400 Bad Request
{
  "message": "Password did not match!"
}
```

Errors

- If the request body is missing or malformed, or if the password does not match, the API will return a 400 Bad Request error.
- If the request is unauthorized (i.e., missing or invalid JWT token), the API will return a 401 Unauthorized error.
- If there's an internal server error during the process, the API will return a 500 Internal Server Error.

Feel free to adjust the documentation according to your application's specific requirements and error handling logic!



Anonymous

```
@Post('/bill_payment')
@UseGuards(AuthGuard)
@UsePipes(new ValidationPipe())
@HttpCode(HttpStatus.OK) // Set the status code to 200 (OK)
async Bill_Payment(
  @Request() req,
  @Body() bill: Payment_ReceiverDTO,
): Promise<any> {
  try {
    const user_validity_decision = await this.customerService.user_validity(
      req.user.email,
      bill.password,
    );

    if (user_validity_decision) {
      bill.payment_type = 'Bill Payment';
      const decision = await this.customerService.Subtract_Credits_Amount(
        req.user.email,
        bill,
      );
    }
  }
}
```

```
if (decision > 0) {
    return {
        success: true,
        message: 'Money has been transferred successfully',
    };
} else {
    throw new InternalServerErrorException(
        'Payment Could not be completed',
    );
}
} else {
    throw new BadRequestException('Password did not matched!');
}
} catch (e) {
    throw new InternalServerErrorException(
        'Customer Controller Create Billing Error = ' + e.message,
    );
}
}
```

give me proper documentations for this api endpoint



ChatGPT

Here's the documentation for your `/bill_payment` endpoint:

API Endpoint: Bill Payment

Description

This API endpoint allows authenticated users to make bill payments.

URL

bash

Copy code

```
POST /bill_payment
```

Request

Headers


- Authorization: Bearer [JWT Token]
- Content-Type: application/json

Body Parameters

Parameter	Type	Description
amount	Number	The amount of money to be paid for the bill.
password	String	The password for validation (to confirm the transaction).

Example Request

http

 Copy code

```
POST /bill_payment
Authorization: Bearer [JWT Token]
Content-Type: application/json

{
  "amount": 100,
  "password": "user_password"
}
```

Response


Success Response

- **Status Code:** 200 (OK)
- **Content-Type:** application/json

Field	Type	Description
success	Boolean	Indicates if the transaction was successful.
message	String	A message indicating the result of the transaction.

Example Success Response

json

 Copy code


```
HTTP/1.1 200 OK
{
  "success": true,
  "message": "Money has been transferred successfully"
}
```

Error Responses

- **Status Code:** 400 (Bad Request)
 - If the request body is missing or malformed, or if the password does not match.
- **Status Code:** 401 (Unauthorized)
 - If the request does not include a valid JWT token.
- **Status Code:** 500 (Internal Server Error)
 - If there's an internal server error during the process.

Example Error Response

json

 Copy code

```
HTTP/1.1 400 Bad Request
{
  "message": "Password did not match!"
}
```

Errors

- If the request body is missing or malformed, or if the password does not match, the API will return a 400 Bad Request error.

- If the request is unauthorized (i.e., missing or invalid JWT token), the API will return a 401 Unauthorized error.
- If there's an internal server error during the process, the API will return a 500 Internal Server Error.

Feel free to adjust the documentation according to your application's specific requirements and error handling logic!



Anonymous

```
@Post('/cash_in')
@UseGuards(AuthGuard)
@UsePipes(new ValidationPipe())
@HttpCode(HttpStatus.OK) // Set the status code to 200 (OK)
async Cash_In(
  @Request() req,
  @Body() bill: Payment_ReceiverDTO,
): Promise<any> {
  try {
    const user_validity_decision = await this.customerService.user_validity(
      req.user.email,
      bill.password,
    );

    if (user_validity_decision) {
      bill.payment_type = 'Cash In';
      const decision = await this.customerService.Add_Credits_Amount(
        req.user.email,
        bill,
      );

      if (decision > 0) {
        return {
          success: true,
          message: 'Money has been transferred successfully',
        };
      } else {
        throw new InternalServerErrorException(
          'Payment Could not be completed',
        );
      }
    } else {
      throw new BadRequestException('Password did not matched!');
    }
  } catch (e) {
    throw new InternalServerErrorException(
      'Customer Controller Create Billing Error = ' + e.message,
    );
  }
}
```

give me proper documentations for this api endpoint



ChatGPT

Here's the documentation for your `/cash_in` endpoint:


API Endpoint: Cash In

Description

This API endpoint allows authenticated users to add credits to their account.

URL

bash

 Copy code

```
POST /cash_in
```

Request

Headers


- Authorization: Bearer [JWT Token]
- Content-Type: application/json

Body Parameters

Parameter	Type	Description
amount	Number	The amount of money to be added to the account.
password	String	The password for validation (to confirm the transaction).

Example Request

http

 Copy code

```
POST /cash_in
Authorization: Bearer [JWT Token]
Content-Type: application/json

{
  "amount": 100,
  "password": "user_password"
}
```

Response


Success Response

- **Status Code:** 200 (OK)
- **Content-Type:** application/json

Field	Type	Description
success	Boolean	Indicates if the transaction was successful.
message	String	A message indicating the result of the transaction.

Example Success Response

json

 Copy code

```
HTTP/1.1 200 OK
{
  "success": true,
  "message": "Money has been transferred successfully"
}
```

Error Responses

- **Status Code:** 400 (Bad Request)
 - If the request body is missing or malformed, or if the password does not match.
- **Status Code:** 401 (Unauthorized)
 - If the request does not include a valid JWT token.
- **Status Code:** 500 (Internal Server Error)
 - If there's an internal server error during the process.

Example Error Response

```
json Copy code

HTTP/1.1 400 Bad Request
{
  "message": "Password did not match!"
}
```

Errors

- If the request body is missing or malformed, or if the password does not match, the API will return a 400 Bad Request error.
- If the request is unauthorized (i.e., missing or invalid JWT token), the API will return a 401 Unauthorized error.
- If there's an internal server error during the process, the API will return a 500 Internal Server Error.

Feel free to adjust the documentation according to your application's specific requirements and error handling logic!



Anonymous

```
@Post('/add_money/wallet_to_bank')
@UseGuards(AuthGuard)
@UsePipes(new ValidationPipe())
@HttpCode(HttpStatus.OK) // Set the status code to 200 (OK)
async Wallet_to_Bank(
  @Request() req,
  @Body() bill: Payment_ReceiverDTO,
): Promise<any> {
  try {
    const user_validity_decision = await this.customerService.user_validity(
      req.user.email,
      bill.password,
    );

    if (user_validity_decision) {
      bill.payment_type = 'Wallet to Bank';
    }
  }
}
```

```
const decision = await this.customerService.Subtract_Credits_Amount(
  req.user.email,
  bill,
);

if (decision > 0) {
  return {
    success: true,
    message: 'Money has been transferred successfully',
  };
} else {
  throw new InternalServerErrorException(
    'Payment Could not be completed',
  );
}
} else {
  throw new BadRequestException('Password did not matched!');
}
} catch (e) {
  throw new InternalServerErrorException(
    'Customer Controller Create Billing Error = ' + e.message,
  );
}
}
```

give me proper documentations for this api endpoint



ChatGPT

Here's the documentation for your `/add_money/wallet_to_bank` endpoint:

API Endpoint: Wallet to Bank Transfer

Description

This API endpoint allows authenticated users to transfer money from their wallet to their bank account.

URL

bash

Copy code

```
POST /add_money/wallet_to_bank
```

Request

Headers

- Authorization: Bearer [JWT Token]
- Content-Type: application/json

Body Parameters

Parameter	Type	Description
amount	Number	The amount of money to be transferred to the bank account.
password	String	The password for validation (to confirm the transaction).

Example Request

```

http
Copy code

POST /add_money/wallet_to_bank
Authorization: Bearer [JWT Token]
Content-Type: application/json

{
  "amount": 100,
  "password": "user_password"
}
```

Response

Success Response

- **Status Code:** 200 (OK)
- **Content-Type:** application/json

Field	Type	Description
success	Boolean	Indicates if the transaction was successful.
message	String	A message indicating the result of the transaction.

Example Success Response

```

json
Copy code

HTTP/1.1 200 OK
{
  "success": true,
  "message": "Money has been transferred successfully"
}
```

Error Responses

- **Status Code:** 400 (Bad Request)
 - If the request body is missing or malformed, or if the password does not match.
- **Status Code:** 401 (Unauthorized)
 - If the request does not include a valid JWT token.
- **Status Code:** 500 (Internal Server Error)
 - If there's an internal server error during the process.

Example Error Response

```

json
Copy code

HTTP/1.1 400 Bad Request
{
```



```
"message": "Password did not match!"
}
```

Errors

- If the request body is missing or malformed, or if the password does not match, the API will return a 400 Bad Request error.
- If the request is unauthorized (i.e., missing or invalid JWT token), the API will return a 401 Unauthorized error.
- If there's an internal server error during the process, the API will return a 500 Internal Server Error.

Feel free to adjust the documentation according to your application's specific requirements and error handling logic!



Anonymous

```
@Post('/add_money/wallet_to_card')
@UseGuards(AuthGuard)
@UsePipes(new ValidationPipe())
@HttpCode(HttpStatus.OK) // Set the status code to 200 (OK)
async Wallet_to_Card(
  @Request() req,
  @Body() bill: Payment_ReceiverDTO,
): Promise<any> {
  try {
    const user_validity_decision = await this.customerService.user_validity(
      req.user.email,
      bill.password,
    );

    if (user_validity_decision) {
      bill.payment_type = 'Wallet to Card';
      const decision = await this.customerService.Subtract_Credits_Amount(
        req.user.email,
        bill,
      );

      if (decision > 0) {
        return {
          success: true,
          message: 'Money has been transferred successfully',
        };
      } else {
        throw new InternalServerErrorException(
          'Payment Could not be completed',
        );
      }
    } else {
      throw new BadRequestException('Password did not matched!');
    }
  } catch (e) {
    throw new InternalServerErrorException(
      'Customer Controller Create Billing Error = ' + e.message,
    );
  }
}
```

```
}  
}
```

give me proper documentations for this api endpoint



ChatGPT

Here's the documentation for your `/add_money/wallet_to_card` endpoint:

API Endpoint: Wallet to Card Transfer

Description

This API endpoint allows authenticated users to transfer money from their wallet to their card.

URL

bash Copy code

POST /add_money/wallet_to_card

Request

Headers

- Authorization: Bearer [JWT Token]
- Content-Type: application/json

Body Parameters

Parameter	Type	Description
amount	Number	The amount of money to be transferred to the card.
password	String	The password for validation (to confirm the transaction).

Example Request

http Copy code

POST /add_money/wallet_to_card
Authorization: Bearer [JWT Token]
Content-Type: application/json

{
 "amount": 100,
 "password": "user_password"
}

Response

Success Response

- **Status Code:** 200 (OK)
- **Content-Type:** application/json

Field	Type	Description
success	Boolean	Indicates if the transaction was successful.
message	String	A message indicating the result of the transaction.

Example Success Response

```

json Copy code

HTTP/1.1 200 OK
{
  "success": true,
  "message": "Money has been transferred successfully"
}
```

Error Responses

- **Status Code: 400** (Bad Request)
 - If the request body is missing or malformed, or if the password does not match.
- **Status Code: 401** (Unauthorized)
 - If the request does not include a valid JWT token.
- **Status Code: 500** (Internal Server Error)
 - If there's an internal server error during the process.

Example Error Response

```

json Copy code

HTTP/1.1 400 Bad Request
{
  "message": "Password did not match!"
}
```

Errors

- If the request body is missing or malformed, or if the password does not match, the API will return a 400 Bad Request error.
- If the request is unauthorized (i.e., missing or invalid JWT token), the API will return a 401 Unauthorized error.
- If there's an internal server error during the process, the API will return a 500 Internal Server Error.

Feel free to adjust the documentation according to your application's specific requirements and error handling logic!



Anonymous

```

@Post('/add_money/bank_to_wallet')
@UseGuards(AuthGuard)
@UsePipes(new ValidationPipe())
@HttpCode(HttpStatus.OK) // Set the status code to 200 (OK)
async Bank_to_Wallet(
  @Request() req,
  @Body() bill: Payment_ReceiverDTO,
): Promise<any> {
  try {
```

```
const user_validity_decision = await this.customerService.user_validity(
  req.user.email,
  bill.password,
);

if (user_validity_decision) {
  bill.payment_type = 'Bank to Wallet';
  const decision = await this.customerService.Add_Credits_Amount(
    req.user.email,
    bill,
  );

  if (decision > 0) {
    return {
      success: true,
      message: 'Money has been transferred successfully',
    };
  } else {
    throw new InternalServerErrorException(
      'Payment Could not be completed',
    );
  }
} else {
  throw new BadRequestException('Password did not matched!');
}
} catch (e) {
  throw new InternalServerErrorException(
    'Customer Controller Create Billing Error = ' + e.message,
  );
}
}
```

give me proper documentations for this api endpoint



ChatGPT

Here's the documentation for your `/add_money/bank_to_wallet` endpoint:

API Endpoint: Bank to Wallet Transfer

Description

This API endpoint allows authenticated users to transfer money from their bank account to their wallet.

URL

bash

Copy code

```
POST /add_money/bank_to_wallet
```

Request

Headers

- Authorization: Bearer [JWT Token]
- Content-Type: application/json

Body Parameters

Parameter	Type	Description
amount	Number	The amount of money to be transferred to the wallet.
password	String	The password for validation (to confirm the transaction).

Example Request

```
http Copy code

POST /add_money/bank_to_wallet
Authorization: Bearer [JWT Token]
Content-Type: application/json

{
  "amount": 100,
  "password": "user_password"
}
```

Response

Success Response

- **Status Code:** 200 (OK)
- **Content-Type:** application/json

Field	Type	Description
success	Boolean	Indicates if the transaction was successful.
message	String	A message indicating the result of the transaction.

Example Success Response

```
json Copy code

HTTP/1.1 200 OK
{
  "success": true,
  "message": "Money has been transferred successfully"
}
```

Error Responses

- **Status Code:** 400 (Bad Request)
 - If the request body is missing or malformed, or if the password does not match.
- **Status Code:** 401 (Unauthorized)
 - If the request does not include a valid JWT token.
- **Status Code:** 500 (Internal Server Error)
 - If there's an internal server error during the process.

Example Error Response

```
json Copy code
```

```
HTTP/1.1 400 Bad Request
{
  "message": "Password did not match!"
}
```

Errors

- If the request body is missing or malformed, or if the password does not match, the API will return a 400 Bad Request error.
- If the request is unauthorized (i.e., missing or invalid JWT token), the API will return a 401 Unauthorized error.
- If there's an internal server error during the process, the API will return a 500 Internal Server Error.

Feel free to adjust the documentation according to your application's specific requirements and error handling logic!



Anonymous

```
@Get('payment/list')
@UseGuards(AuthGuard)
@UsePipes(new ValidationPipe())
@HttpCode(HttpStatus.OK) // Set the status code to 200 (OK)
async Get_All_Billing(@Request() req): Promise<any> {
  try {
    const payment_list = this.customerService.Get_All_Billing_Payment(
      req.user.email,
    );

    if (payment_list != null) {
      return payment_list;
    } else {
      throw new NotFoundException('Data not found');
    }
  } catch (e) {
    throw new InternalServerErrorException(e.message);
  }
}
```

give me proper documentations for this api endpoint



ChatGPT

Here's the documentation for your `/payment/list` endpoint:


API Endpoint: Get Payment List

Description

This API endpoint allows authenticated users to retrieve a list of their billing payments.

URL

bash

 Copy code

```
GET /payment/list
```


Request

Headers

- Authorization: Bearer [JWT Token]

Example Request

http

 Copy code

```
GET /payment/list
Authorization: Bearer [JWT Token]
```

Response


Success Response

- **Status Code:** 200 (OK)
- **Content-Type:** application/json

The response will contain a list of billing payments.

Example Success Response

json

 Copy code


```
HTTP/1.1 200 OK
[
  {
    "id": 1,
    "payment_type": "Bill Payment",
    "amount": 100,
    "date": "2024-04-28T12:00:00Z"
  },
  {
    "id": 2,
    "payment_type": "Wallet to Bank",
    "amount": 50,
    "date": "2024-04-27T12:00:00Z"
  }
]
```

Error Responses

- **Status Code:** 401 (Unauthorized)
 - If the request does not include a valid JWT token.
- **Status Code:** 404 (Not Found)
 - If no payment data is found for the user.
- **Status Code:** 500 (Internal Server Error)
 - If there's an internal server error during the process.

Example Error Response

json

 Copy code

```
HTTP/1.1 404 Not Found
{
  "message": "Data not found"
}
```

Errors

- If the request is unauthorized (i.e., missing or invalid JWT token), the API will return a 401 Unauthorized error.
- If no payment data is found for the user, the API will return a 404 Not Found error.
- If there's an internal server error during the process, the API will return a 500 Internal Server Error.

Feel free to adjust the documentation according to your application's specific requirements and error handling logic!



Anonymous

```
@Post('/forget_password')
@UsePipes(new ValidationPipe())
@HttpCode(HttpStatus.OK) // Set the status code to 200 (OK)
async Forget_Password(
  @Body() forgetPassword_DTO: ForgetPasswordDTO,
): Promise<any> {
  try {
    return await this.customerService.ForgetPassword(
      forgetPassword_DTO.email,
    );
  } catch (e) {
    throw new InternalServerErrorException(e.message);
  }
}
```

give me proper documentations for this api endpoint



ChatGPT

Here's the documentation for your `/forget_password` endpoint:


API Endpoint: Forget Password

Description

This API endpoint allows users to initiate the process of resetting their password by providing their email address.

URL

bash

 Copy code

```
POST /forget_password
```


Request

Headers

- **Content-Type:** application/json

Body Parameters

Parameter	Type	Description
email	String	The email address for which the password reset is requested.

Example Request

```
http Copy code  
  
POST /forget_password  
Content-Type: application/json  
  
{  
  "email": "user@example.com"  
}
```

Response

Success Response

- **Status Code:** 200 (OK)
- **Content-Type:** application/json

Upon successful initiation of the password reset process, a success message will be returned.

Example Success Response

```
json Copy code  
  
HTTP/1.1 200 OK  
{  
  "message": "Password reset process initiated. Please check your email for further instructions."  
}
```

Error Responses

- **Status Code:** 400 (Bad Request)
 - If the request body is missing or malformed.
- **Status Code:** 500 (Internal Server Error)
 - If there's an internal server error during the process.

Example Error Response

```
json Copy code  
  
HTTP/1.1 500 Internal Server Error  
{  
  "message": "An error occurred while processing your request."  
}
```

Errors

- If the request body is missing or malformed, the API will return a 400 Bad Request error.
- If there's an internal server error during the process, the API will return a 500 Internal Server Error.

Feel free to adjust the documentation according to your application's specific requirements and error handling logic!



Anonymous

```
@Post('/otp')
@UsePipes(new ValidationPipe())
@HttpCode(HttpStatus.OK) // Set the status code to 200 (OK)
async OTP_Verification(
  @Request() req,
  @Body() OTP_Object: OTP_ReceiverDTO,
): Promise<any> {
  try {
    console.log('User provided otp = ' + OTP_Object.otp);
    const decision = await this.customerService.otp_verification(
      req,
      OTP_Object.otp,
    );
    if (decision) {
      return {
        success: true,
        message: 'OTP verification successful',
      };
    } else {
      return new BadRequestException('OTP did not matched!');
    }
  } catch (e) {
    // throw new InternalServerErrorException(e.message);
  }
}
```

give me proper documentations for this api endpoint



ChatGPT

Here's the documentation for your `/otp` endpoint:

API Endpoint: OTP Verification

Description

This API endpoint allows users to verify an OTP (One-Time Password) for authentication purposes.

URL

```
bash
```

Copy code

POST /otp

Request

Headers


- Content-Type: application/json

Body Parameters

Parameter	Type	Description
otp	String	The OTP provided by the user for verification.

Example Request

http

 Copy code

```
POST /otp
Content-Type: application/json

{
  "otp": "123456"
}
```

Response


Success Response

- **Status Code:** 200 (OK)
- **Content-Type:** application/json

Upon successful verification of the OTP, a success message will be returned.

Example Success Response

json

 Copy code


```
HTTP/1.1 200 OK
{
  "success": true,
  "message": "OTP verification successful"
}
```

Error Responses

- **Status Code:** 400 (Bad Request)
 - If the OTP provided by the user does not match the expected OTP.
- **Status Code:** 500 (Internal Server Error)
 - If there's an internal server error during the process.

Example Error Response

json

 Copy code

```
HTTP/1.1 400 Bad Request
{
```

```
"message": "OTP did not match!"
}
```

Errors

- If the OTP provided by the user does not match the expected OTP, the API will return a 400 Bad Request error.
- If there's an internal server error during the process

, the API will return a 500 Internal Server Error.

Feel free to adjust the documentation according to your application's specific requirements and error handling logic!



Anonymous

```
@Post('/signup')
// @UsePipes(new ValidationPipe())
@HttpCode(HttpStatus.OK) // Set the status code to 200 (OK)
async Signup(@Body() signup_info: LoginDTO): Promise<any> {
  try {
    signup_info.password = await bcrypt.hash(signup_info.password, 12);

    const user_id = await this.authService.signUp(signup_info);
    if (user_id < 0) {
      throw new BadRequestException({
        status: HttpStatus.BAD_REQUEST,
        message: 'Email Already Exists',
      });
    } else {
      return user_id;
    }
  } catch (e) {
    throw new BadRequestException({
      status: HttpStatus.BAD_REQUEST,
      message: e.message,
    });
  }
}
```

give me proper documentations for this api endpoint



ChatGPT

Here's the documentation for your `/signup` endpoint:


API Endpoint: User Signup

Description

This API endpoint allows users to sign up by providing their email and password.

URL

bash

 Copy code

POST /signup

Request

Headers


- Content-Type: application/json

Body Parameters

Parameter	Type	Description
email	String	The email address of the user.
password	String	The password chosen by the user.

Example Request

http

 Copy code

```
POST /signup
Content-Type: application/json

{
  "email": "user@example.com",
  "password": "password123"
}
```

Response


Success Response

- **Status Code:** 200 (OK)
- **Content-Type:** application/json

Upon successful signup, the response will contain the newly created user ID.

Example Success Response

json

 Copy code


```
HTTP/1.1 200 OK
{
  "userId": "abc123"
}
```

Error Responses

- **Status Code:** 400 (Bad Request)
 - If the provided email already exists in the database, or if there's an error during the signup process.

Example Error Response

json

 Copy code

```
HTTP/1.1 400 Bad Request
{
  "status": 400,
  "message": "Email Already Exists"
}
```

Errors

- If the provided email already exists in the database, the API will return a 400 Bad Request error with a message indicating the email already exists.
- If there's an error during the signup process, the API will return a 400 Bad Request error with an appropriate error message.

Feel free to adjust the documentation according to your application's specific requirements and error handling logic!



Anonymous

```
@Post('/login')
@HttpCode(HttpStatus.OK) // Set the status code to 200 (OK)
@UsePipes(new ValidationPipe())
async Login(@Body() login_info: LoginDTO): Promise<any> {
  return await this.authService.signIn(login_info);
}
```

give me proper documentations for this api endpoint



ChatGPT

Here's the documentation for your `/login` endpoint:

API Endpoint: User Login

Description

This API endpoint allows users to log in by providing their email and password.

URL

```
bash
```

[Copy code](#)

```
POST /login
```

Request

Headers

- Content-Type: application/json

Body Parameters

Parameter	Type	Description
email	String	The email address of the user.
password	String	The password provided by the user.

Example Request

```
http Copy code

POST /login
Content-Type: application/json

{
  "email": "user@example.com",
  "password": "password123"
}
```

Response

Success Response

- **Status Code:** 200 (OK)
- **Content-Type:** application/json

Upon successful login, the response will contain the user's authentication token.

Example Success Response

```
json Copy code

HTTP/1.1 200 OK
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

Error Responses

- **Status Code:** 400 (Bad Request)
 - If the provided email or password is incorrect, or if there's an error during the login process.

Example Error Response

```
json Copy code

HTTP/1.1 400 Bad Request
{
  "message": "Invalid email or password"
}
```

Errors

- If the provided email or password is incorrect, the API will return a 400 Bad Request error with a message indicating invalid credentials.
- If there's an error during the login process, the API will return a 400 Bad Request error with an appropriate error message.

Feel free to adjust the documentation according to your application's specific requirements and error handling logic!



Anonymous

```
@Get('/logout')
@UseGuards(AuthGuard)
@UsePipes(new ValidationPipe())
async Logout(@Request() req): Promise<any> {
  try {
    const token = await this.authService.extractTokenFromHeader(req);
    if (token != null && token != "") {
      return await this.authService.logout(req.user.email, token);
    } else {
      throw new BadRequestException(
        'Please provide the token inside header, along with the request',
      );
    }
  } catch (e) {
    throw new InternalServerErrorException(e.message);
  }
}
```

give me proper documentations for this api endpoint



ChatGPT

Here's the documentation for your `/logout` endpoint:

API Endpoint: User Logout

Description

This API endpoint allows authenticated users to log out by invalidating their authentication token.

URL

bash

Copy code

GET /logout

Request

Headers

- Authorization: Bearer [JWT Token]

Example Request

http

Copy code

```
GET /logout
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```


Response

Success Response

- **Status Code:** 200 (OK)
- **Content-Type:** application/json

Upon successful logout, the response will indicate that the user has been logged out.

Example Success Response

```
json Copy code

HTTP/1.1 200 OK
{
  "message": "User logged out successfully"
}
```

Error Responses

- **Status Code:** 400 (Bad Request)
 - If the authentication token is missing or invalid.

Example Error Response

```
json Copy code

HTTP/1.1 400 Bad Request
{
  "message": "Please provide a valid authentication token"
}
```

Errors

- If the authentication token is missing or invalid, the API will return a 400 Bad Request error with an appropriate error message.

Feel free to adjust the documentation according to your application's specific requirements and error handling logic!



Anonymous

```
@Post('/change_password')
@UsePipes(new ValidationPipe())
@HttpCode(HttpStatus.OK) // Set the status code to 200 (OK)
async Change_password(
  @Request() req,
  new_Password_Object_DTO: New_PasswordDTO,
): Promise<any> {
  try {
    const result = await this.authService.UpdatePassword(
      req,
      new_Password_Object_DTO.password,
    );
    console.log('Result = ' + result);
```

```
} catch (e) {
  throw new InternalServerErrorException(
    'Change Password Auth Controller error = ' + e.message,
  );
} finally {
  // Destroy the JWT
  return await this.authService.destroy_temporary_JWT(req);
}
}
```

give me proper documentations for this api endpoint



ChatGPT

Here's the documentation for your `/change_password` endpoint:

API Endpoint: Change Password

Description

This API endpoint allows authenticated users to change their password.

URL

bash

Copy code

POST /change_password

Request

Headers

- Authorization: Bearer [JWT Token]
- Content-Type: application/json

Body Parameters

Parameter	Type	Description
password	String	The new password chosen by the user.

Example Request

http

Copy code

```
POST /change_password
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

{
  "password": "newPassword123"
}
```

Response

Success Response

- **Status Code:** 200 (OK)
- **Content-Type:** application/json

Sign up to chat

[Report content](#) | [Terms of use](#) | [Privacy policy](#) |