Practical 5 API docs

API Documentation

Users

Login

Request Type

- Method: POST
- Content-Type: application/json

Request

```
{
    type: 'login',
    email: 'email',
    password: 'password'
}
```

Return

```
{ //Success
    status: 'success',
    message: 'Login successful'
  user: {
  user_id : 'user_id',
    name : 'name',
  email : 'email',
  user_type : 'user_type'
  apikey: 'apikey'
  }
}

{ //Error
    status: 'error',
    message: 'Invalid email or password'
}
```

Registration

Request Type

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

Request

```
{
    type: 'Register',
    email: 'email',
    name: 'name',
    password: 'password',
    user_type: 'customer/store owner',
    *registrationNo: '2025/123456/01'
}
```

*Only required if user is a store owner. Format of registration numbers found at https://www.inv24.co.za/za/blog/invoice_reg_nr/.

Return

```
{ //Success
    status: 'success',
    name: 'name',
    type: 'type',
    apikey: 'apikey'
}

{ //Error
    status: 'error',
    message: 'some error message'
}
```

Update user settings

Request Type

- **Method**: POST

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

Request

```
{
    type: 'SavePrefrences',
    apikey: 'apikey',
    *theme: 0,
    *min_price: 0.0,
    *max_price: 1000.0,
}
```

*at least one of the parameters is required

Return

```
{ //Success
    status: 'success',
    message: 'Updated user settings'
}

{ //Error
    status: 'error',
    message: 'Error updating the user settings'
}
```

Product

GetAllProducts

Request Type

- **Method**: POST

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

Request

```
{
    type: 'GetAllProducts'
}
```

Return

```
{
    status: 'success',
    message: 'Products fetched successfully',
    data: [
        {
            product_id: 1,
            title: 'title',
            price: 50.00,
            product_link: 'link to products google shopping page',
            description: 'description',
            launch_date: 'Date the product was added to our database',
            thumbnail: 'link to load image of product',
            category: 'category',
            brand_name: 'name of brand'
        },
        {
            ...
        },
        ...
    ]
}
```

Adding Product

Request Type

- **Method**: POST

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

Request

```
{
    type: 'AddProduct',
    title: 'Product title',
    price: 55.00,
    product_link: 'link to products google shopping page',
    description: 'description',
```

```
   *launch_date: 'Date the product was added to our database',
   thumbnail: 'link to load image of product',
   category: 'category',
   brand_name: 'brand name', //Name of brand
   store_id: 1 //id of store
   apikey: 'apikey' //apikey of the user
}
```

*Maybe should be omitted and then the backend sets the date to today

Return

```
{
   status: 'success',
   message: 'Product added successfully to the database.'
}


{
   status: 'error',
   message: 'some error message'
}
```

Delete Product

Request Type

- **Method**: POST

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

Request

```
{
   type: 'DeleteProduct',
   prod_id: 1, //id of the product
   store_id: 1, //id of the store to which a product belongs to
   apikey: 'apikey of the user'
}
```

Return

```
{
   status: 'success',
   message: 'Product deleted successfully in the database'
}


{
   status: 'error',
   message: 'some error message'
}
```

Edit Product

Request Type

- **Method**: POST

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

Request

```
{
   type: 'EditProduct',
   prod_id: 1, //id of the product you want to delete
   store_id: 1, //id of the store in which the product is in
   *title: 'new title',
   *price: 'new price',
   *product_link: 'new product_link',
   *description: 'new description',
   *launch_date: 'new launch_date',
   *thumbnail: 'new thumbnail',
   *category: 'new category',
   *brand_name: 'new brand'
}
```

*at least one of the parameters is required

Return

```
{
   status: 'success',
```

```
    message: 'Product updated successfully.'
}


{
    status: 'error',
    message: 'some error message'
}
```

Filter Products

Request Type

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

Request

```
{
    type: 'GetFilteredProducts',
    *prod_id: 1, //id of a product
    *brand: 'brand_id',
    *category: 'category',
    *min_price: 'min_price',
    *max_price: 'max_price',
    *search: 'search',
    *store_id: 1 //id of store that owns product
}
```

*at least one of the parameters is required

Return

```
{
    status: 'success',
    data: [
        {
            product_id: 1,
            title: 'title',
            price: 50.00,
            product_link: 'link to products google shopping page',
            description: 'description',
            launch_date: 'Date the product was added to our database',
            thumbnail: 'link to load image of product',
            category: 'category',
            brand_id: 1 //Should be fixed to return brand name
        },
        {
            ...
        },
        ...
    ]
}


{
    status: 'error',
    message: 'some error message'
}
```

Brands

Get Brands

Request Type

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

Request

```
{
    type: 'GetBrands'
}
```

Return

```
{
    status: 'success',
    message: 'Brands successfully retrieved',
    data: [
```

```
        {
            brand_id: 1,
            name: 'name'
        },
        {
            ...
        },
        ...
    ]
}
```

Add Brands

Request Type

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

Can only be done by an admin account

Request

```
{
    type: 'AddBrand',
    apikey: 'apikey',
    brand_name: 'brand_name'
}
```

Return

```
{
    status: 'success',
    message: 'Brand successfully added'
}
```

```
{
    status: 'error',
    message: 'some error message'
}
```

Remove Brands

Request Type

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

Can only be done by an admin account

Request

```
{
    type: 'RemoveBrand',
    apikey: 'apikey',
    brand_name: 'brand_id'
}
```

Return

```
{
    status: 'success',
    message: 'Brand successfully removed'
}
```

```
{
    status: 'error',
    message: 'some error message'
}
```

Rating

Add Rating

Request Type

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

Request

```
{
    type: 'SubmitRating',
    apikey: 'apikey',
    prod_id: 1, //id of the product
    rating: 5, //a number out of 5
    comment: 'comment'
}
```

Return

```
{
    status: 'success',
    message: 'Rating submitted successfully.'
}

{
    status: 'error',
    message: 'some error message'
}
```

Get Ratings

Request Type

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

Request

```
{
    type: 'GetRatings',
    prod_id: 1 //id of the product
}
```

Return

```
{
    status: 'success',
    data: [
        {
            name: 'users name',
            rating: 5, //a number out of 5
            comment: 'comment'
        },
        {
            ...
        },
        ...
    ]
}

{
    status: 'error',
    message: 'some error message'
}
```

Delete Rating

Request Type

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

Request

```
{
    type: 'DeleteRating',
    rating_id: 1, //id of the rating
    apikey: 'apikey of the user'
}
```

Return

```
{
    status: 'success',
    message: 'Rating deleted successfully.'
}
```

```
{
    status: 'error',
    message: 'some error message'
}
```

Edit Rating

Request Type

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

Request

```
{
    type: 'EditRating',
    apikey: 'apikey',
    rating_id: 1, //id of the rating
    *rating: 5, //a number out of 5
    *comment: 'comment'
}
```

*at least one of the parameters is required

Return

```
{
    status: 'success',
    message: 'Rating updated successfully.'
}

{
    status: 'error',
    message: 'some error message'
}
```

Stores

Register Store Owner

Request Type

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

Request

```
{
    type: 'RegisterStoreOwner', //registers the user as store owner and makes a store
    apikey: 'apikey',
    store_name: 'name of store',
    store_url: 'url to store's page',
    type: 'type of store'
}
```

Return

```
{
    status: 'success',
    message: 'Successfully added store and assigned owner'
}

{
    status: 'error',
    message: 'some error message'
}
```

Get Stores

Request Type

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

Request

```
{
    type: 'GetStores'
}
```

Return

```
{
    status: 'success',
    message: 'All stores available, retrieved successfully.' or 'There are currently no stores available',
    data: [
        {
            store_id: 1,
            name: 'name',
            type: 'type',
            url: 'link to store's page'
        },
        {
            ...
        },
        ...
    ]
}

{
    status: 'error',
    message: 'some error message'
}
```

Get users Store

Request Type

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

Request

```
{
    type: 'GetUsersStores',
    apikey: 'apikey'
}
```

Return

```
{
    status: 'success',
    message: 'All stores available, retrieved successfully.' or 'There are currently no stores available',
    data: [
        {
            store_id: 1,
            name: 'name',
            type: 'type',
            url: 'link to store's page'
        },
        {
            ...
        },
        ...
    ]
}

{
    status: 'error',
    message: 'some error message'
}
```

Get Filtered Stores

Request Type

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

Request

```
{
    type: 'getFilteredStores',
    store_id: 1 //id of the store
}
```

Return

```
{
    status: 'success',
    data:  {
        store_id: 1,
        name: 'name',
        type: 'type',
        url: 'link to store's page'
    }
}


{
    status: 'error',
    message: 'some error message'
}
```

Follow Store

Request Type

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

Request

```
{
    type: 'Follow',
    apikey: 'apikey',
    store_id: 1 //id of store being followed
}
```

Return

```
{
    status: 'success',
    message: 'Successfully followed store'
}

{
    status: 'error',
    message: 'some error message'
}
```

Get the stores that a user follows

Request Type

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

Request

```
{
    type: 'GetFollowing', //gets the stores that the user follows
    apikey: 'apikey'
}
```

Return

```
{
    status: 'success',
    message: 'Successfully retrieved followed stores' or 'User is not following any stores',
    data: [
      {
        store_id: 1,
        name: 'name',
        type: 'type',
        url: 'link to store's page'
      },
      {
        ...
      },
      ...
    ]
}


{
    status: 'error',
```

```
      message: 'some error message'
    }
```

Unfollow a store

Request Type

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

Request

```
{
    type: 'Unfollow',
    apikey: 'apikey',
    store_id: 1 //id of the store being unfollowed
}
```

Return

```
{
    status: 'success',
    message: 'Successfully removed follow',
    data: 1 //id of the store that was unfollowed
}

{
    status: 'error',
    message: 'some error message'
}
```