Bark and Purr

Proyecto Final Desafío Latam

Team

Alejandro Botero

Janis Cáceres

Pamela Yáñez

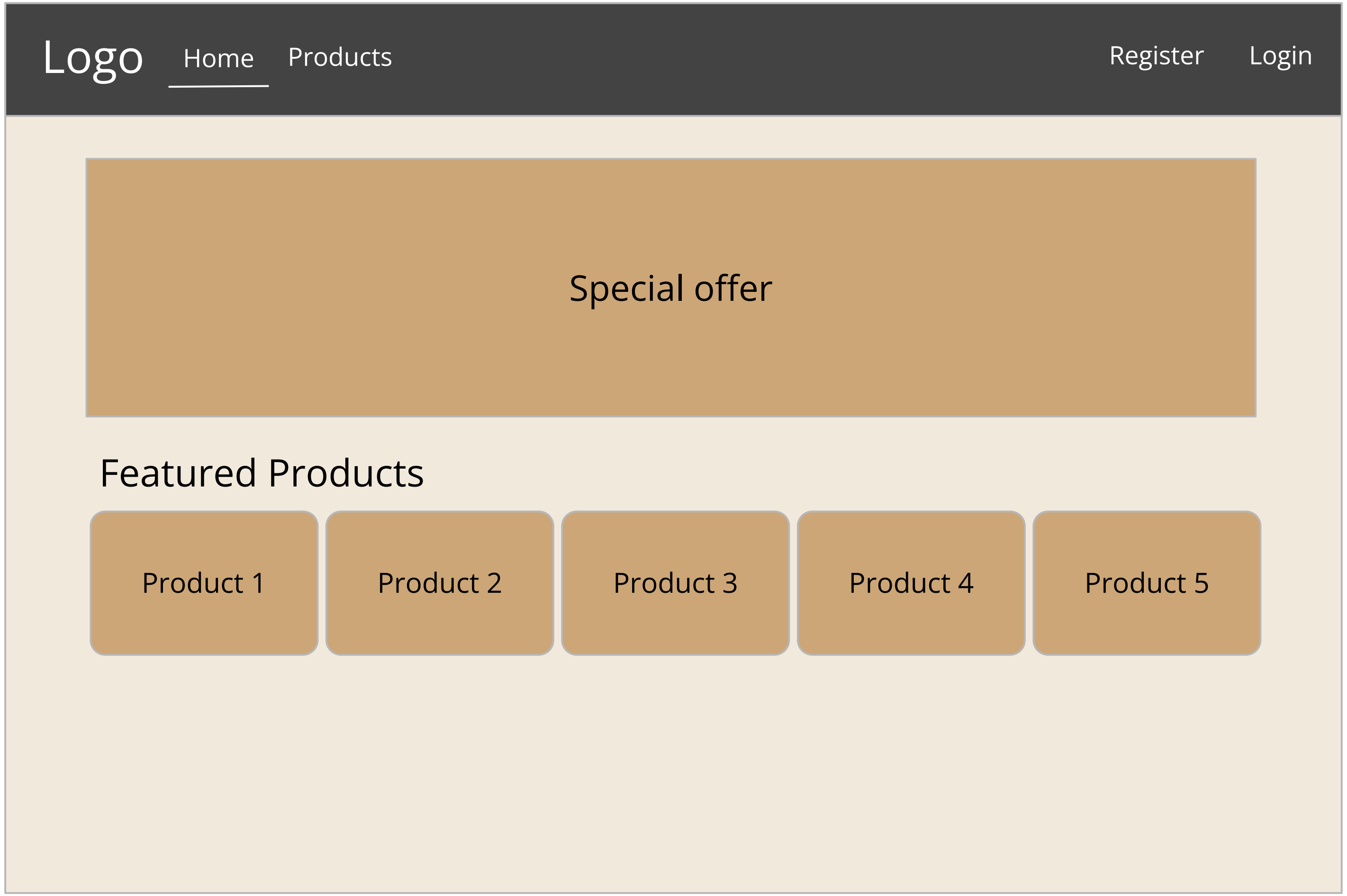
Tamara Zapata

Hito 1

01	Definición de la navegación entre las vistas	• Proveedor • Comprador
02	Definición de la navegación entre las vistas	• Las privadas • Las públicas
03	Listado de dependencias a utilizar en el proyecto	• Frontend • Backend
04	Diseño las tablas de la base de datos y sus relaciones	• Prototipo • Draw-io
05	Diseñar el contrato de datos de la API REST	• Definición de contrato • Imagen (polacode)

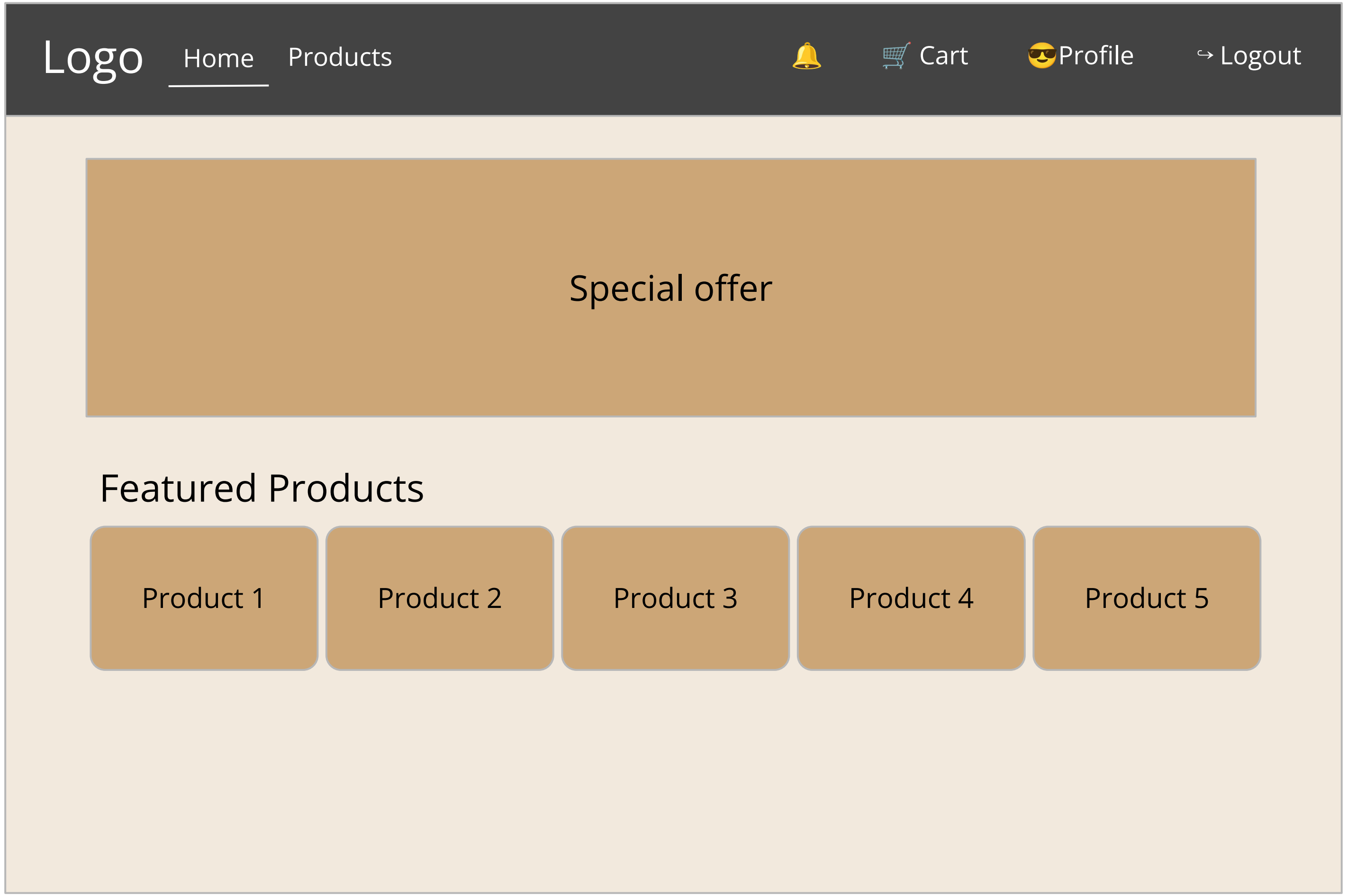
Home

Logged Out



Home

Logged In



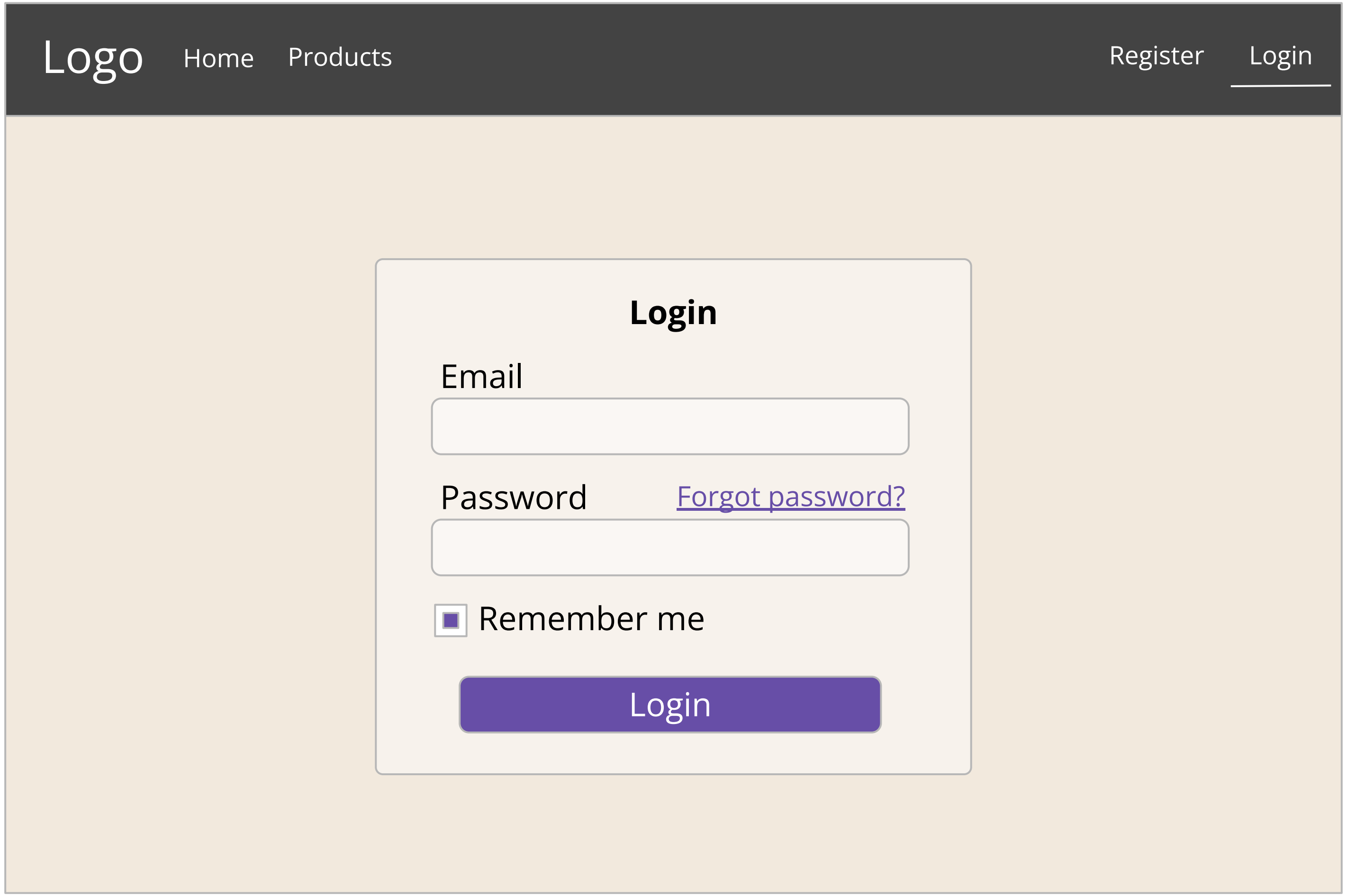
Logo Home Products 🔔 🛒 Cart 😎Profile ↪ Logout

Special offer

Featured Products

Product 1 Product 2 Product 3 Product 4 Product 5

Login



Logo Home Products Register Login

Login

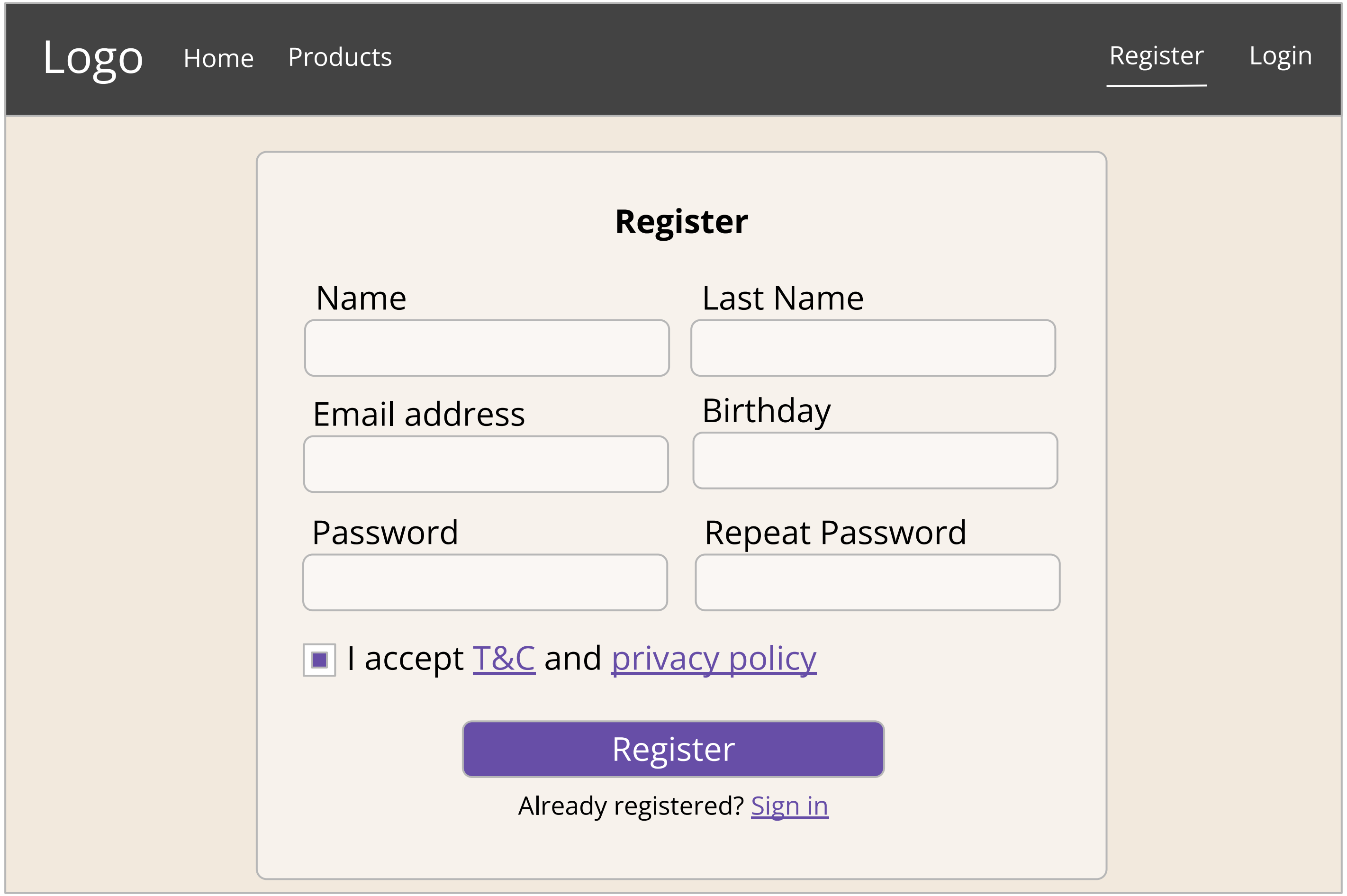
Email

Password Forgot password?

☑ Remember me

Login

Register



Register

Name

Last Name

Email address

Birthday

Password

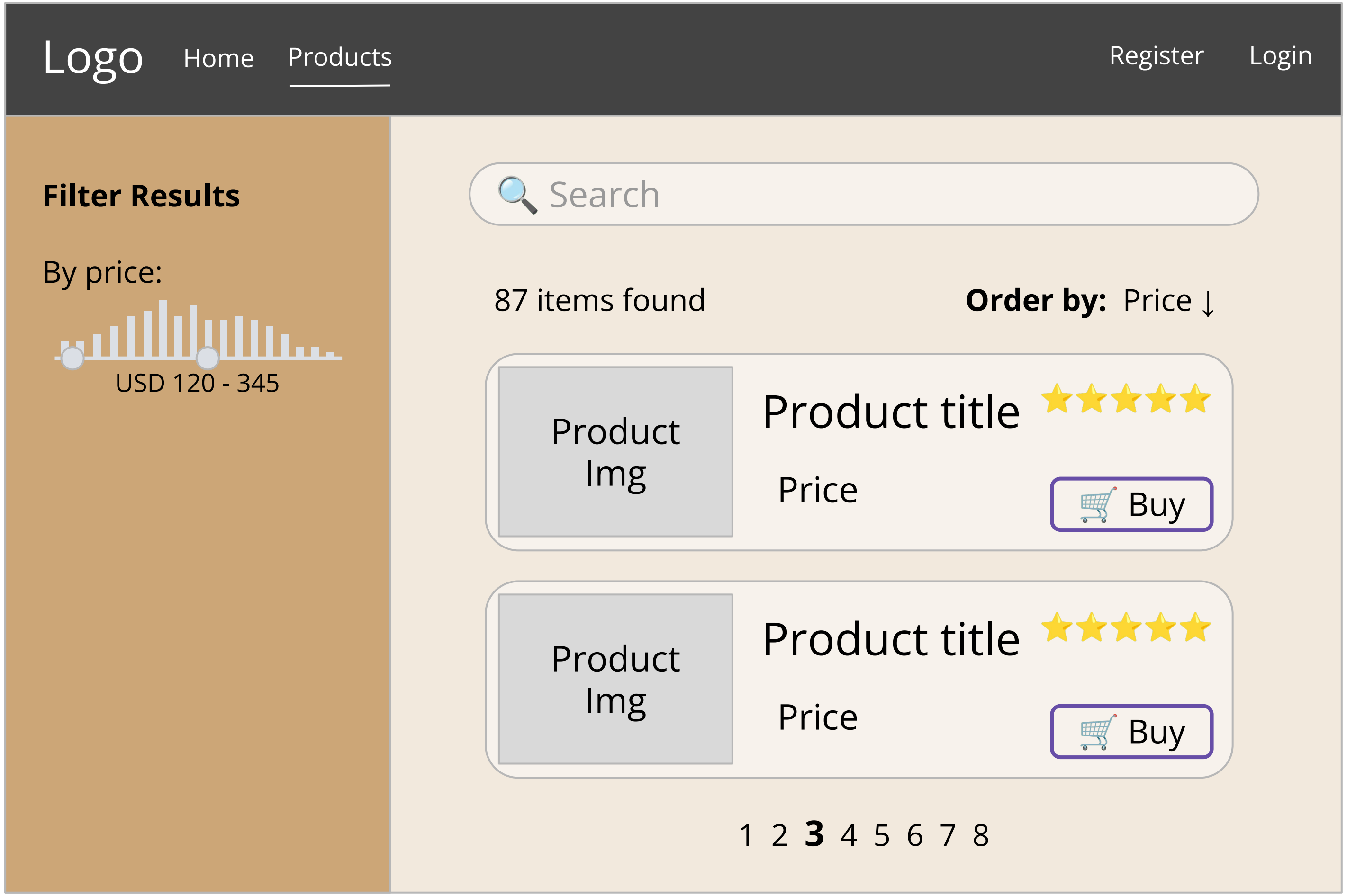
Repeat Password

■ I accept T&C and privacy policy

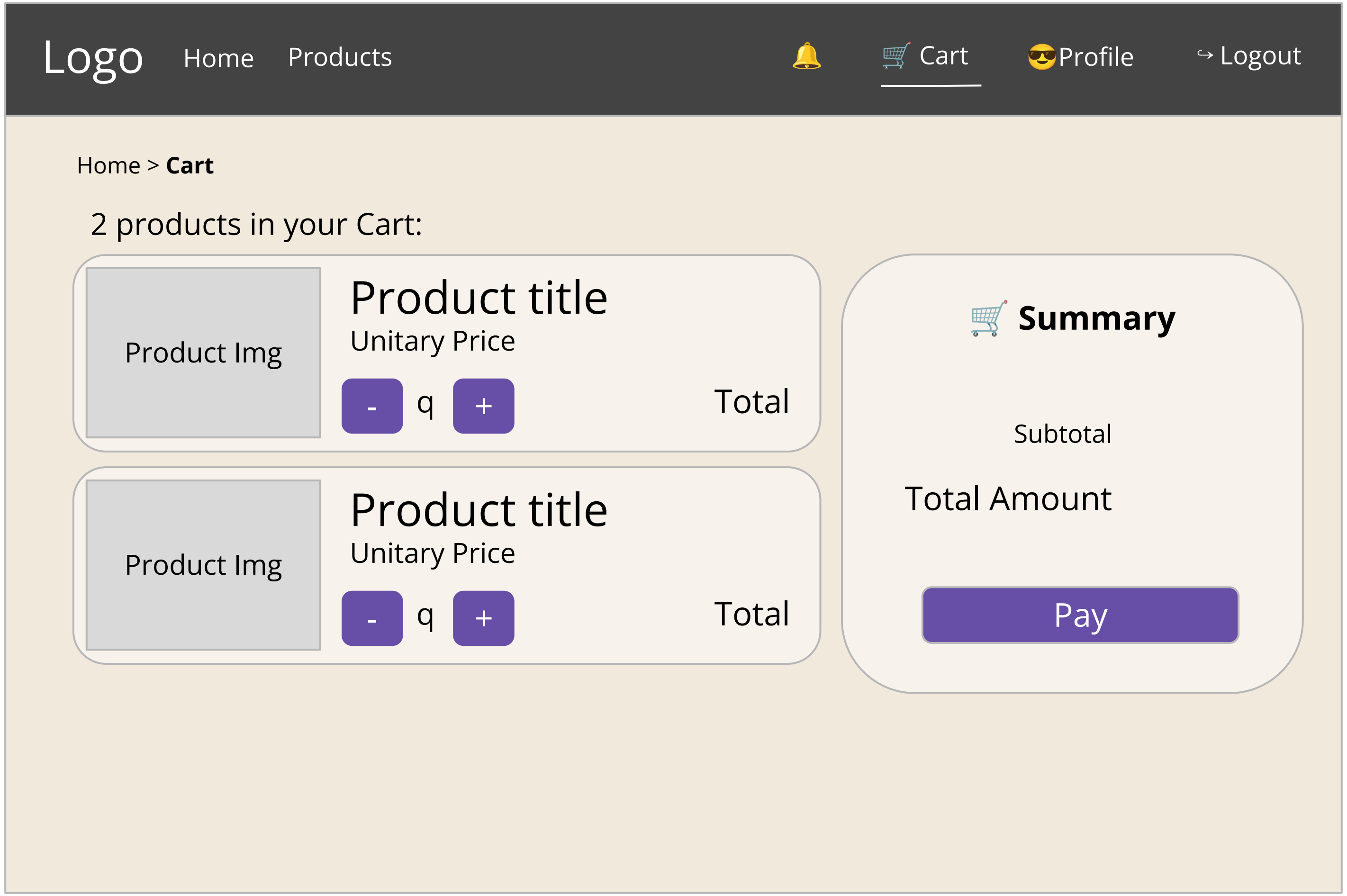
Already registered? Sign in

Diseño de Interfaz

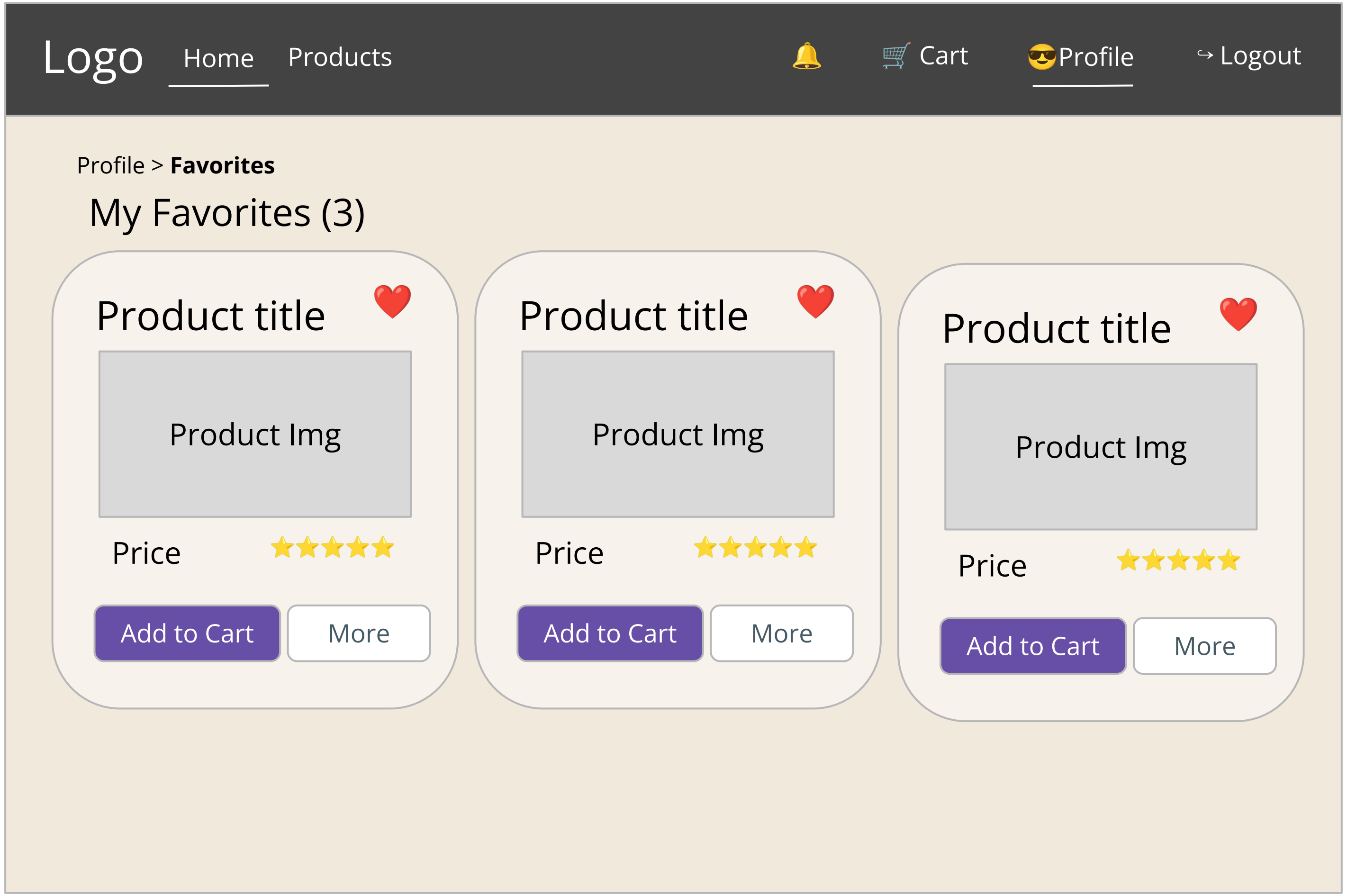
Products


Logo
Home
Products
Register
Login
Filter Results
By price:
USD 120 - 345
Search
87 items found
Order by: Price ↓
Product Img
Product title
Price
Buy
Product Img
Product title
Price
Buy
1 2 3 4 5 6 7 8

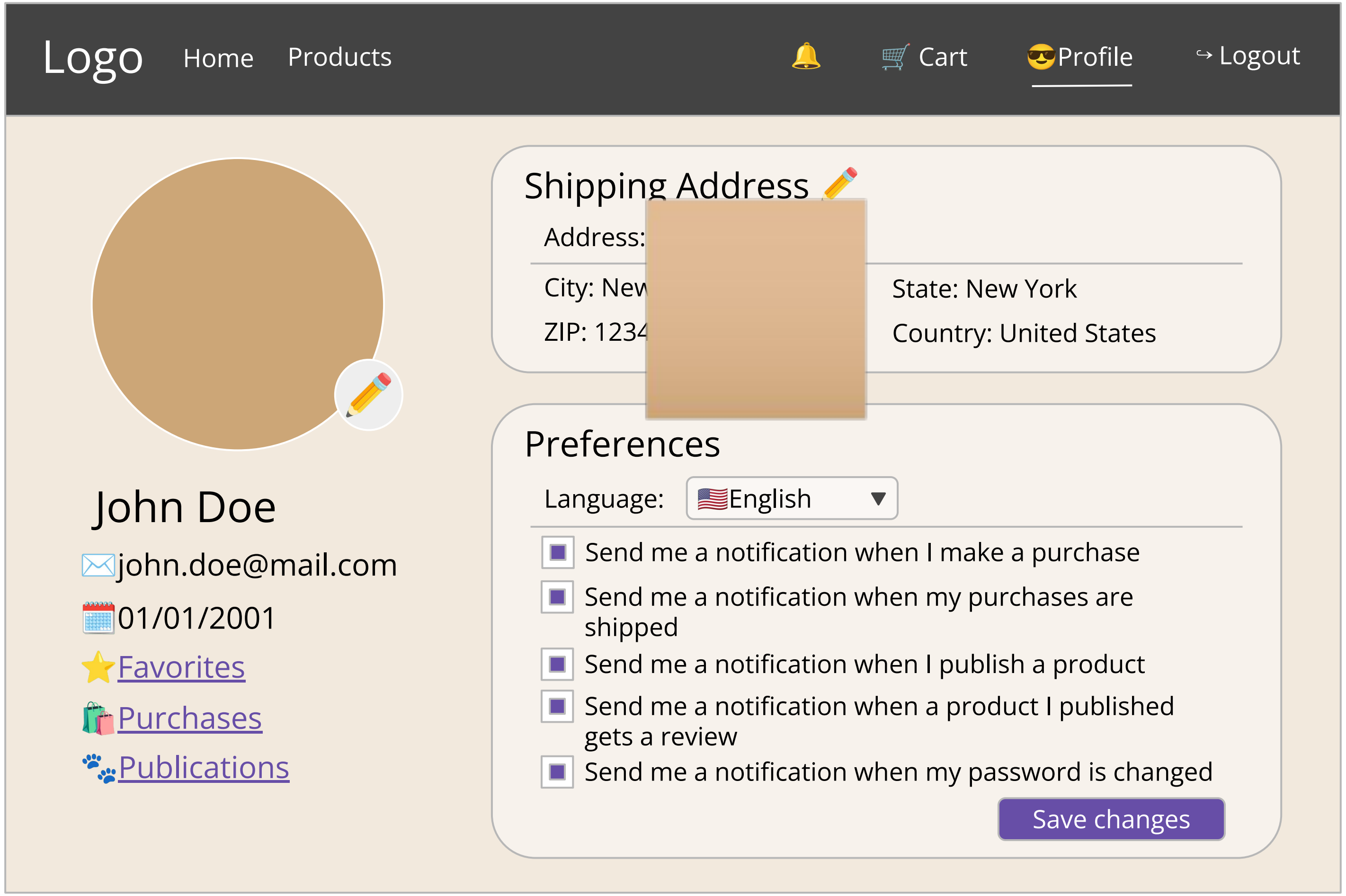
Cart


Logo
Home
Products
🔔
🛒 Cart
😎Profile
↪ Logout
Home > **Cart**
2 products in your Cart:
Product Img
Product title
Unitary Price
-
q
+
Total
Product Img
Product title
Unitary Price
-
q
+
Total
🛒 Summary
Subtotal
Total Amount
Pay

Favorites


Logo
Home
Products
🔔
🛒 Cart
😎Profile
↪ Logout
Profile > **Favorites**
My Favorites (3)
Product title
❤️
Product Img
Price
⭐⭐⭐⭐⭐
Add to Cart
More
Product title
❤️
Product Img
Price
⭐⭐⭐⭐⭐
Add to Cart
More
Product title
❤️
Product Img
Price
⭐⭐⭐⭐⭐
Add to Cart
More

Profile



John Doe

✉️john.doe@mail.com

📅01/01/2001

⭐Favorites

🛍️Purchases

🐾Publications

Shipping Address ✏️

Address: [redacted]

City: New

ZIP: 1234

State: New York

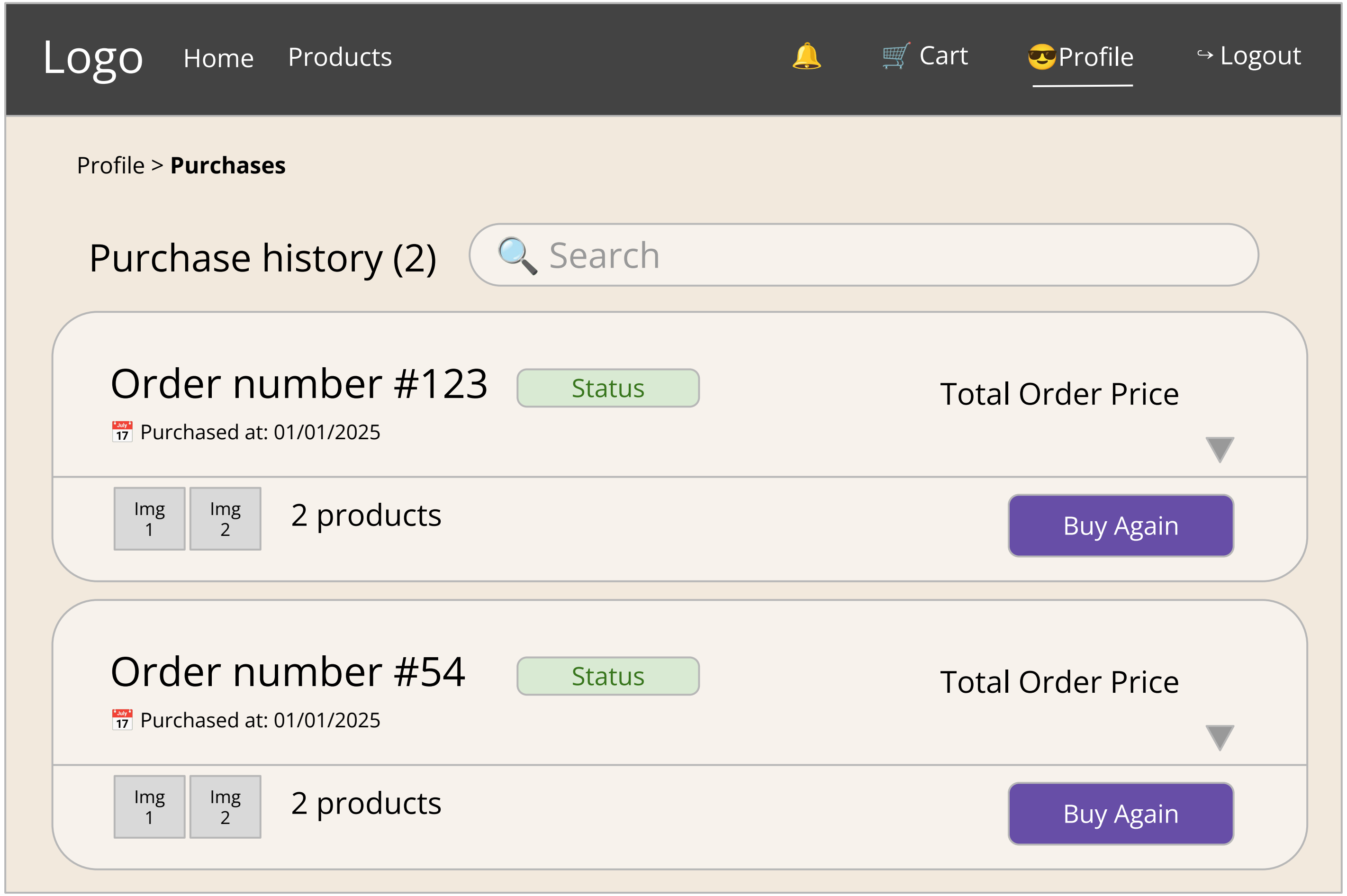
Country: United States

Preferences

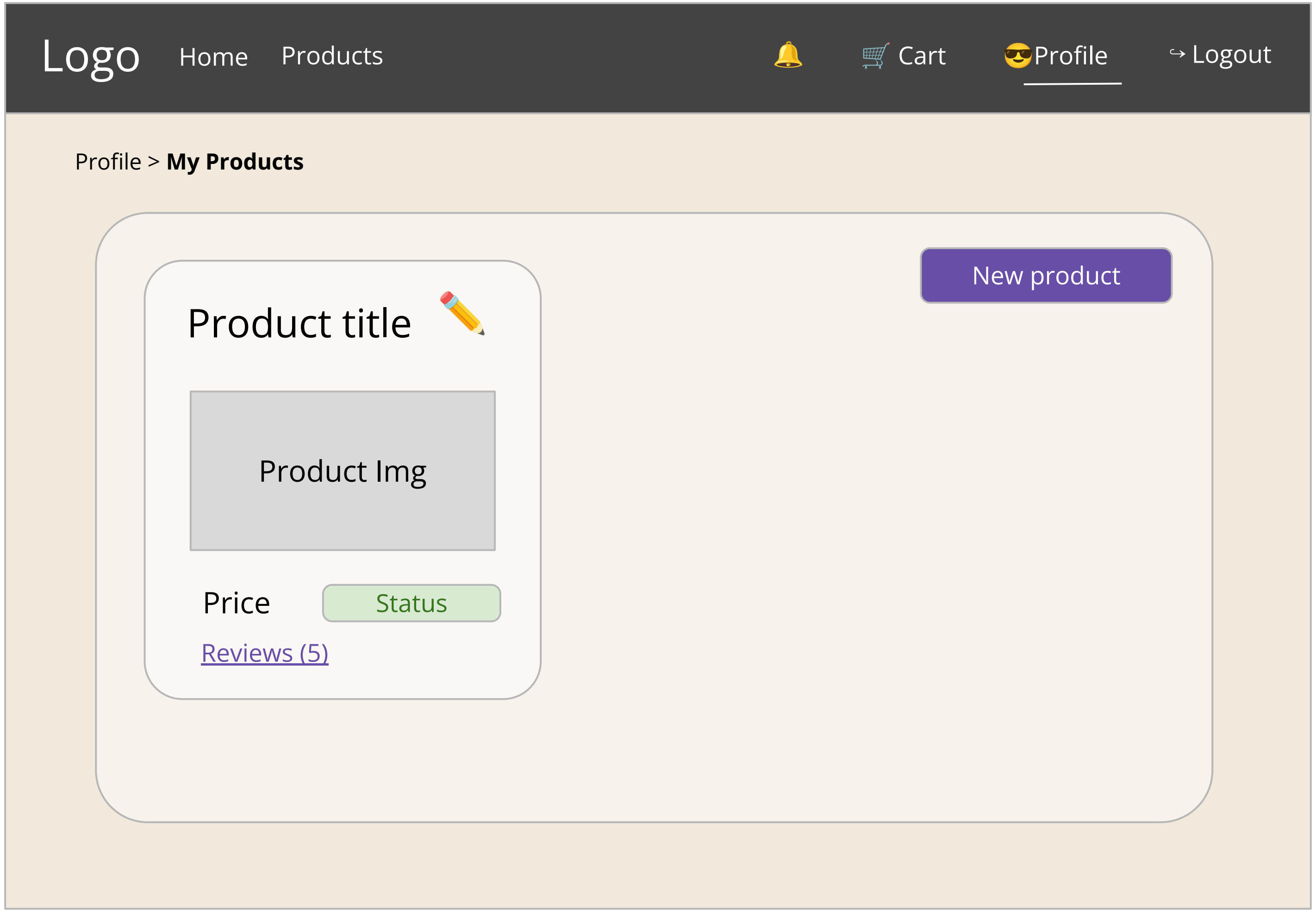
Language: 🇺🇸English ▼

- ■ Send me a notification when I make a purchase
- ■ Send me a notification when my purchases are shipped
- ■ Send me a notification when I publish a product
- ■ Send me a notification when a product I published gets a review
- ■ Send me a notification when my password is changed

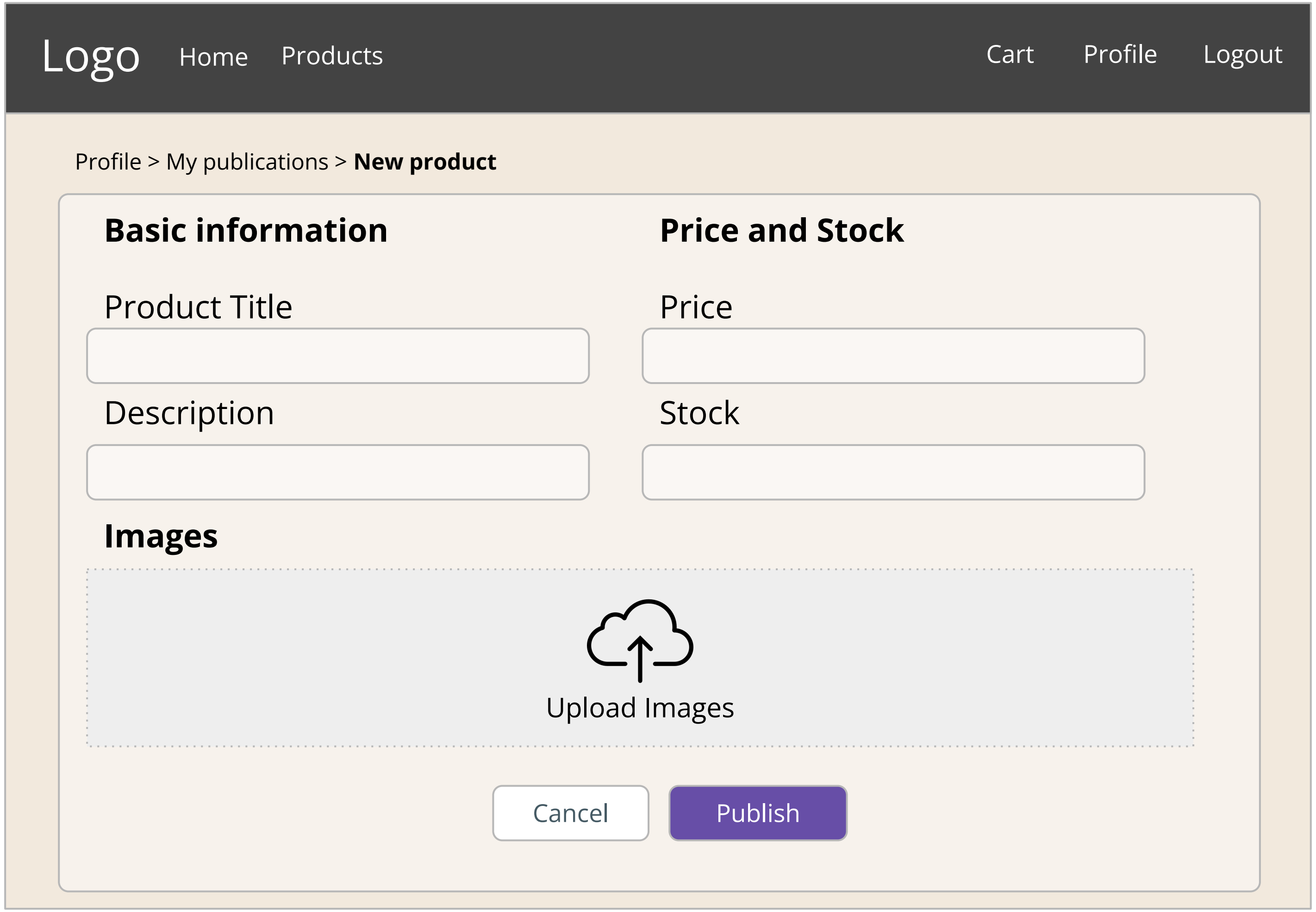
Purchases


Logo
Home
Products
🔔
🛒 Cart
😎Profile
↪ Logout
Profile > **Purchases**
Purchase history (2)
🔍 Search
Order number #123
Status
Total Order Price
📅 Purchased at: 01/01/2025
▼
Img 1
Img 2
2 products
Buy Again
Order number #54
Status
Total Order Price
📅 Purchased at: 01/01/2025
▼
Img 1
Img 2
2 products
Buy Again

My products


Logo
Home
Products
🔔
🛒 Cart
😎Profile
↪ Logout
Profile > **My Products**
New product
Product title ✏️
Product Img
Price
Status
Reviews (5)

New Product



Profile > My publications > **New product**

Basic information

Product Title

Description

Price and Stock

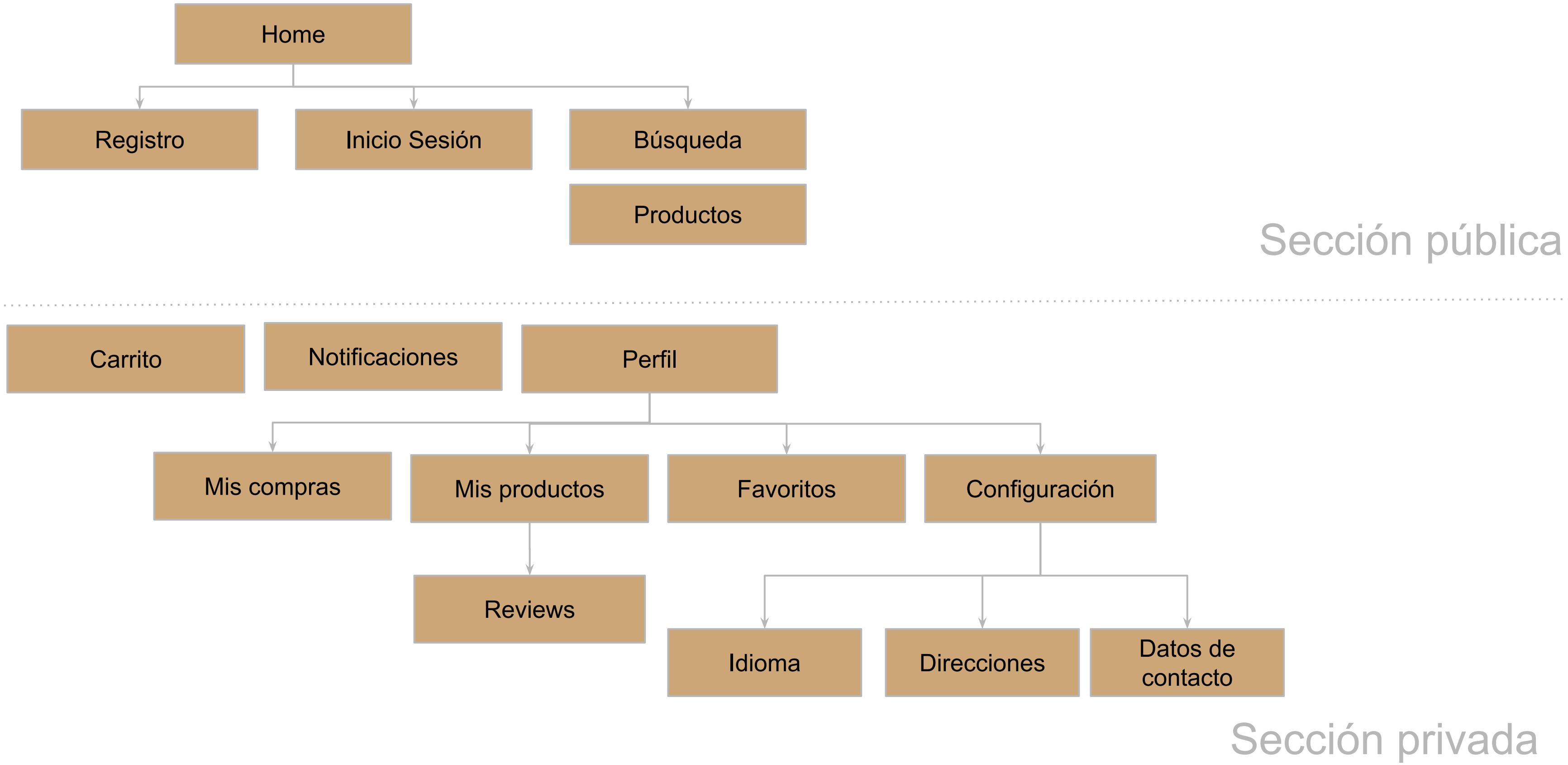
Price

Stock

Images

Definición de la navegación entre las vistas

Navegación Comprador - Proveedor



Listado de dependencias

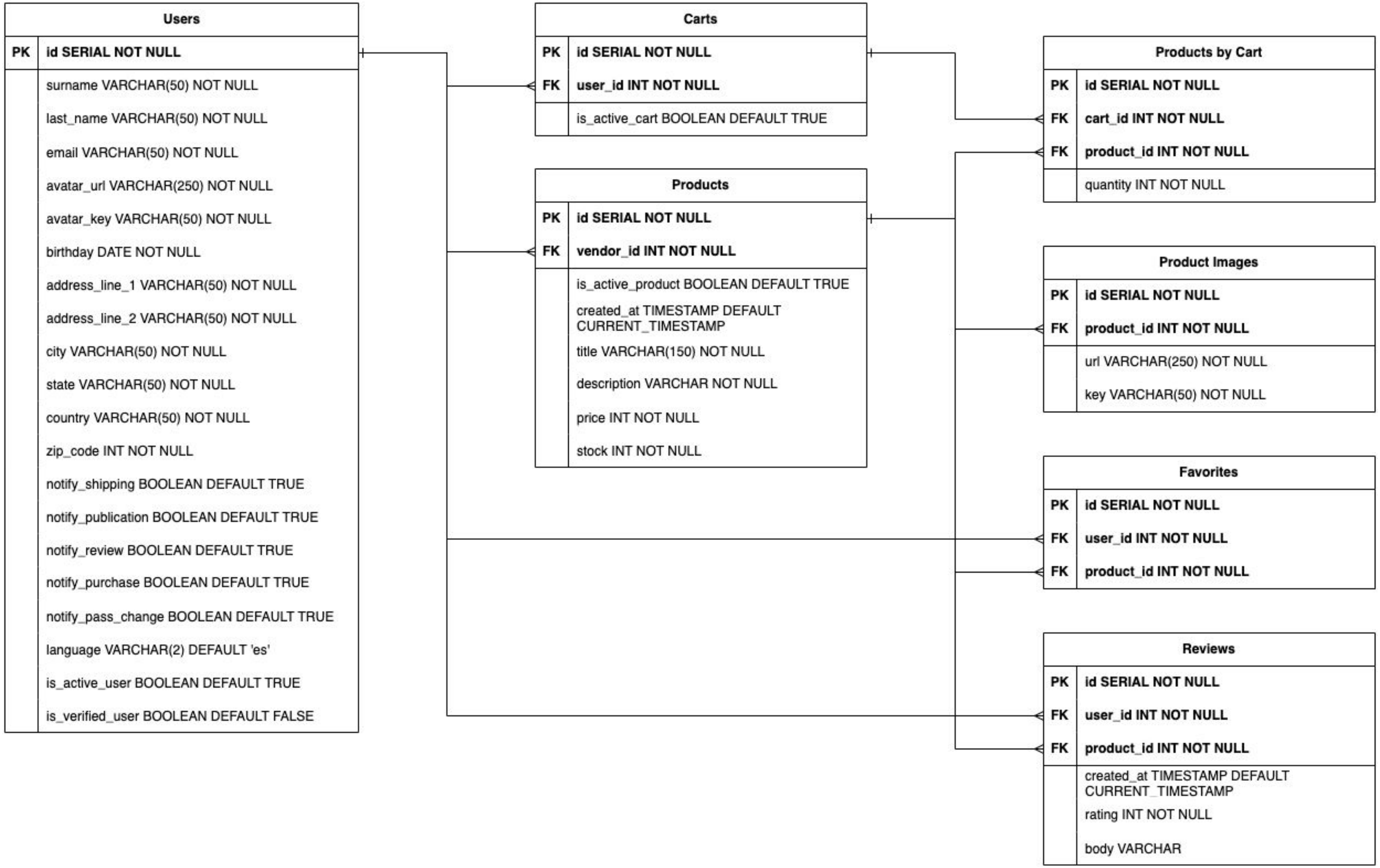
FRONTEND

★ React.js + Vite
★ react-bootstrap
★ react-router
★ axios
★ react-country-flag
★ react-loading
★ i18n (Idioma)
★ nanoid
★ bcrypt

BACKEND

★ Node.js + Express.js
★ cors
★ dotenv
★ pg

Diseño las tablas de la base de datos y sus relaciones



Users	
PK	**id SERIAL NOT NULL**
	surname VARCHAR(50) NOT NULL
	last_name VARCHAR(50) NOT NULL
	email VARCHAR(50) NOT NULL
	avatar_url VARCHAR(250) NOT NULL
	avatar_key VARCHAR(50) NOT NULL
	birthday DATE NOT NULL
	address_line_1 VARCHAR(50) NOT NULL
	address_line_2 VARCHAR(50) NOT NULL
	city VARCHAR(50) NOT NULL
	state VARCHAR(50) NOT NULL
	country VARCHAR(50) NOT NULL
	zip_code INT NOT NULL
	notify_shipping BOOLEAN DEFAULT TRUE
	notify_publication BOOLEAN DEFAULT TRUE
	notify_review BOOLEAN DEFAULT TRUE
	notify_purchase BOOLEAN DEFAULT TRUE
	notify_pass_change BOOLEAN DEFAULT TRUE
	language VARCHAR(2) DEFAULT 'es'
	is_active_user BOOLEAN DEFAULT TRUE
	is_verified_user BOOLEAN DEFAULT FALSE

Carts	
PK	**id SERIAL NOT NULL**
FK	**user_id INT NOT NULL**
	is_active_cart BOOLEAN DEFAULT TRUE

Products	
PK	**id SERIAL NOT NULL**
FK	**vendor_id INT NOT NULL**
	is_active_product BOOLEAN DEFAULT TRUE
	created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
	title VARCHAR(150) NOT NULL
	description VARCHAR NOT NULL
	price INT NOT NULL
	stock INT NOT NULL

Products by Cart	
PK	**id SERIAL NOT NULL**
FK	**cart_id INT NOT NULL**
FK	**product_id INT NOT NULL**
	quantity INT NOT NULL

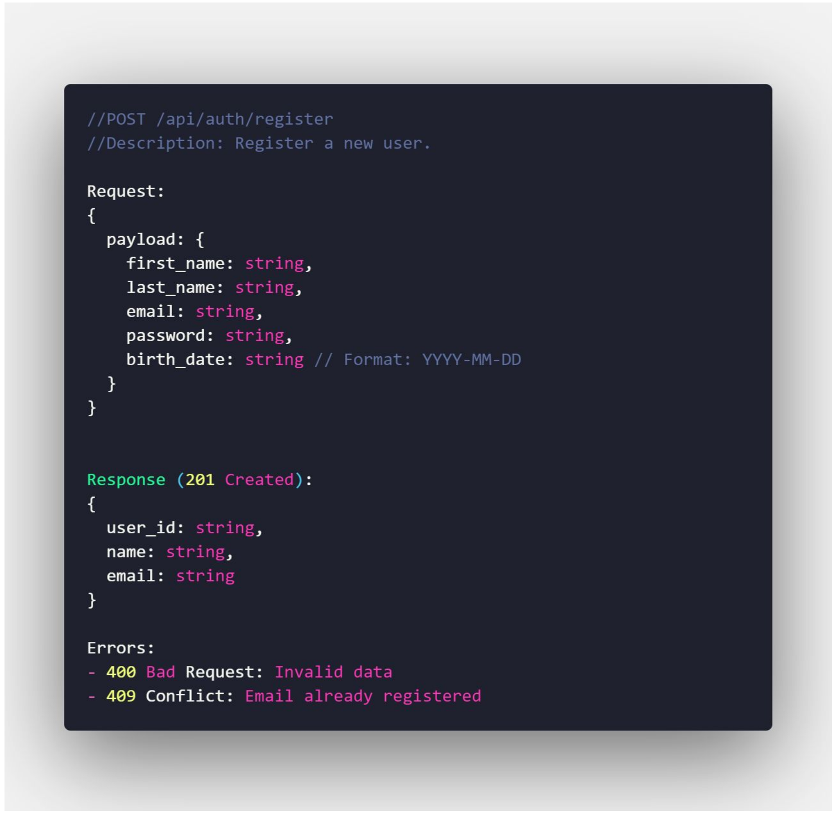
Product Images	
PK	**id SERIAL NOT NULL**
FK	**product_id INT NOT NULL**
	url VARCHAR(250) NOT NULL
	key VARCHAR(50) NOT NULL

Favorites	
PK	**id SERIAL NOT NULL**
FK	**user_id INT NOT NULL**
FK	**product_id INT NOT NULL**

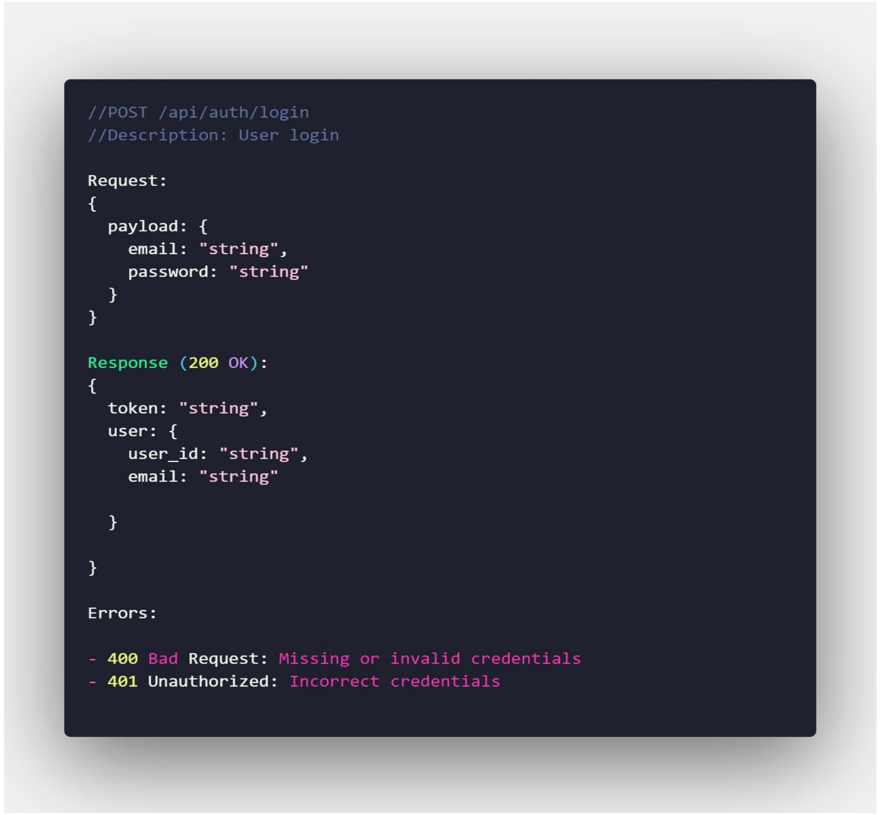
Reviews	
PK	**id SERIAL NOT NULL**
FK	**user_id INT NOT NULL**
FK	**product_id INT NOT NULL**
	created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
	rating INT NOT NULL
	body VARCHAR

Diseño del contrato de datos de la API REST

Autenticación

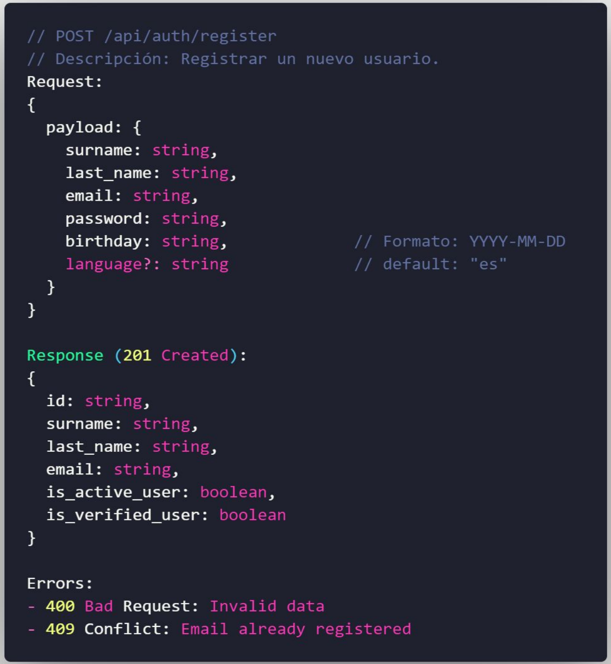


```
//POST /api/auth/register
//Description: Register a new user.

Request:
{
  payload: {
    first_name: string,
    last_name: string,
    email: string,
    password: string,
    birth_date: string // Format: YYYY-MM-DD
  }
}


Response (201 Created):
{
  user_id: string,
  name: string,
  email: string
}

Errors:
- 400 Bad Request: Invalid data
- 409 Conflict: Email already registered
```

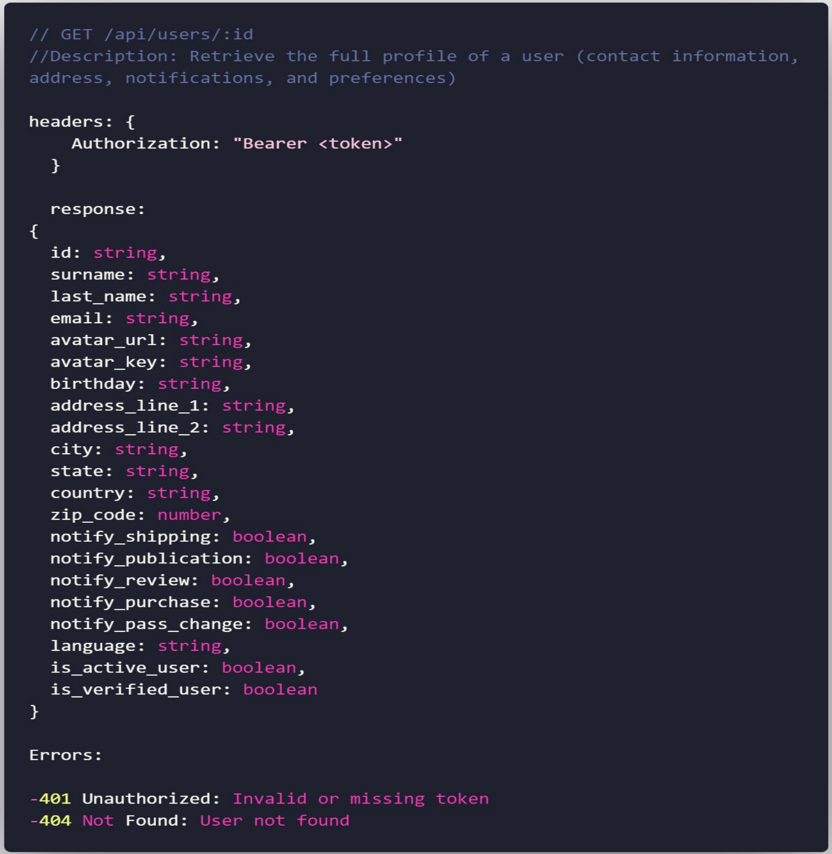


```
//POST /api/auth/login
//Description: User login

Request:
{
  payload: {
    email: "string",
    password: "string"
  }
}

Response (200 OK):
{
  token: "string",
  user: {
    user_id: "string",
    email: "string"

  }

}

Errors:

- 400 Bad Request: Missing or invalid credentials
- 401 Unauthorized: Incorrect credentials
```

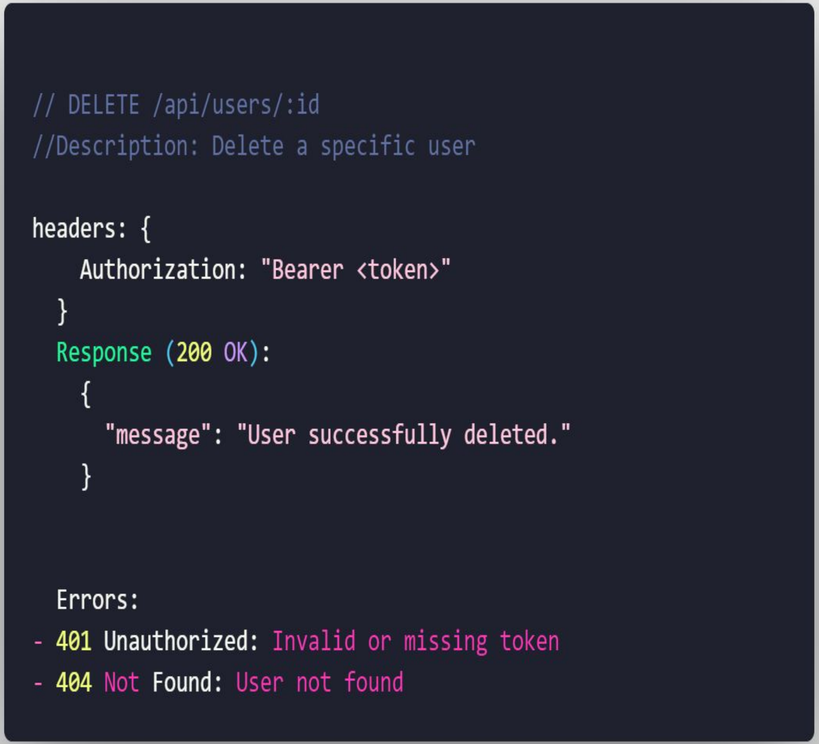
USERS



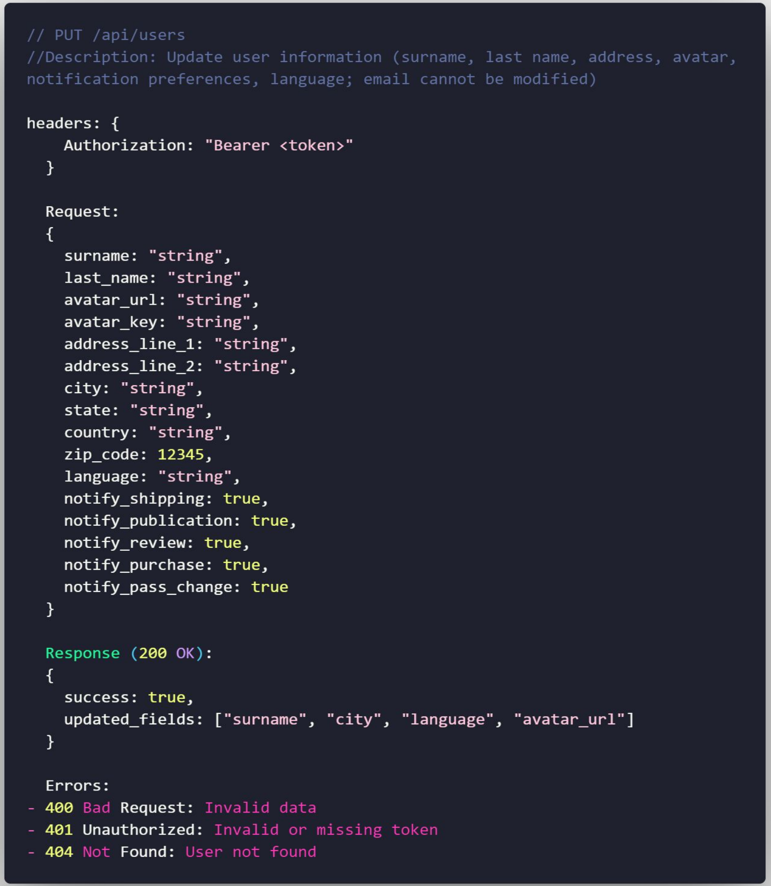
```
// POST /api/auth/register
// Descripción: Registrar un nuevo usuario.
Request:
{
  payload: {
    surname: string,
    last_name: string,
    email: string,
    password: string,
    birthday: string,              // Formato: YYYY-MM-DD
    language?: string              // default: "es"
  }
}

Response (201 Created):
{
  id: string,
  surname: string,
  last_name: string,
  email: string,
  is_active_user: boolean,
  is_verified_user: boolean
}

Errors:
- 400 Bad Request: Invalid data
- 409 Conflict: Email already registered
```



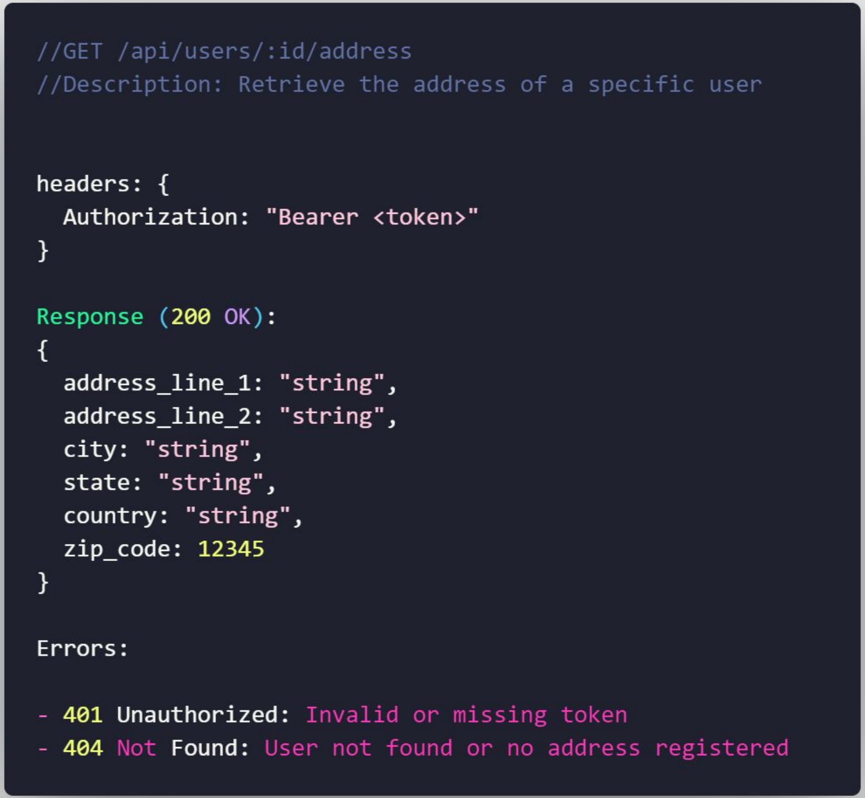
```
// GET /api/users/:id
//Description: Retrieve the full profile of a user (contact information,
address, notifications, and preferences)

headers: {
    Authorization: "Bearer <token>"
  }

  response:
{
  id: string,
  surname: string,
  last_name: string,
  email: string,
  avatar_url: string,
  avatar_key: string,
  birthday: string,
  address_line_1: string,
  address_line_2: string,
  city: string,
  state: string,
  country: string,
  zip_code: number,
  notify_shipping: boolean,
  notify_publication: boolean,
  notify_review: boolean,
  notify_purchase: boolean,
  notify_pass_change: boolean,
  language: string,
  is_active_user: boolean,
  is_verified_user: boolean
}

Errors:

-401 Unauthorized: Invalid or missing token
-404 Not Found: User not found
```



```
// DELETE /api/users/:id
//Description: Delete a specific user

headers: {
    Authorization: "Bearer <token>"
  }
  Response (200 OK):
    {
      "message": "User successfully deleted."
    }


  Errors:
- 401 Unauthorized: Invalid or missing token
- 404 Not Found: User not found
```

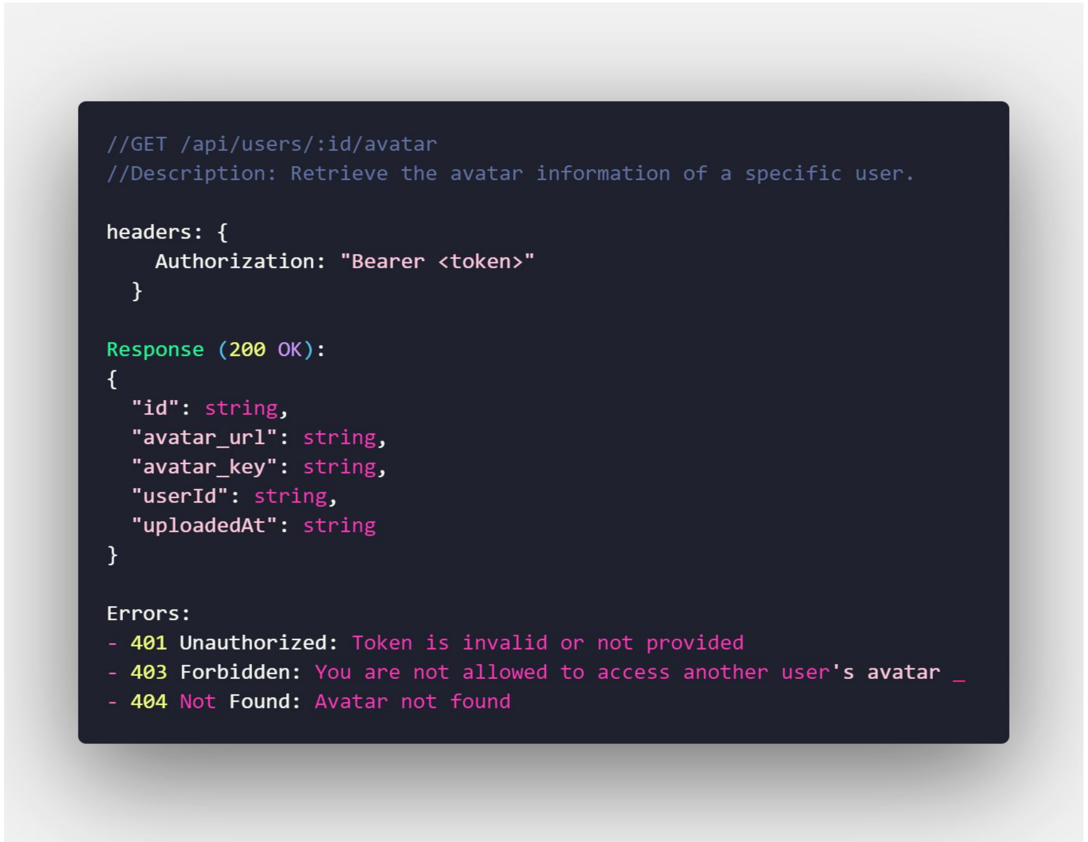


```
// PUT /api/users
//Description: Update user information (surname, last name, address, avatar,
notification preferences, language; email cannot be modified)

headers: {
    Authorization: "Bearer <token>"
  }

  Request:
  {
    surname: "string",
    last_name: "string",
    avatar_url: "string",
    avatar_key: "string",
    address_line_1: "string",
    address_line_2: "string",
    city: "string",
    state: "string",
    country: "string",
    zip_code: 12345,
    language: "string",
    notify_shipping: true,
    notify_publication: true,
    notify_review: true,
    notify_purchase: true,
    notify_pass_change: true
  }

  Response (200 OK):
  {
    success: true,
    updated_fields: ["surname", "city", "language", "avatar_url"]
  }

  Errors:
- 400 Bad Request: Invalid data
- 401 Unauthorized: Invalid or missing token
- 404 Not Found: User not found
```



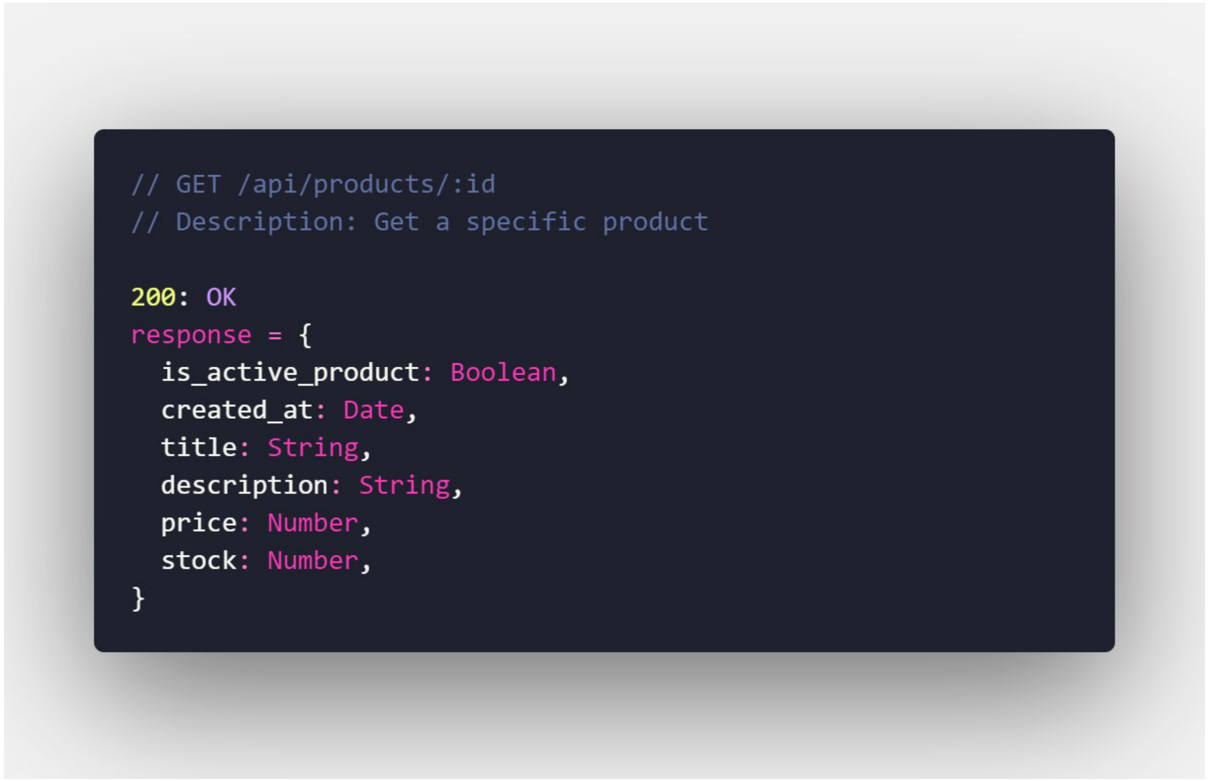
```
//GET /api/users/:id/address
//Description: Retrieve the address of a specific user


headers: {
  Authorization: "Bearer <token>"
}

Response (200 OK):
{
  address_line_1: "string",
  address_line_2: "string",
  city: "string",
  state: "string",
  country: "string",
  zip_code: 12345
}

Errors:

- 401 Unauthorized: Invalid or missing token
- 404 Not Found: User not found or no address registered
```

Avatar



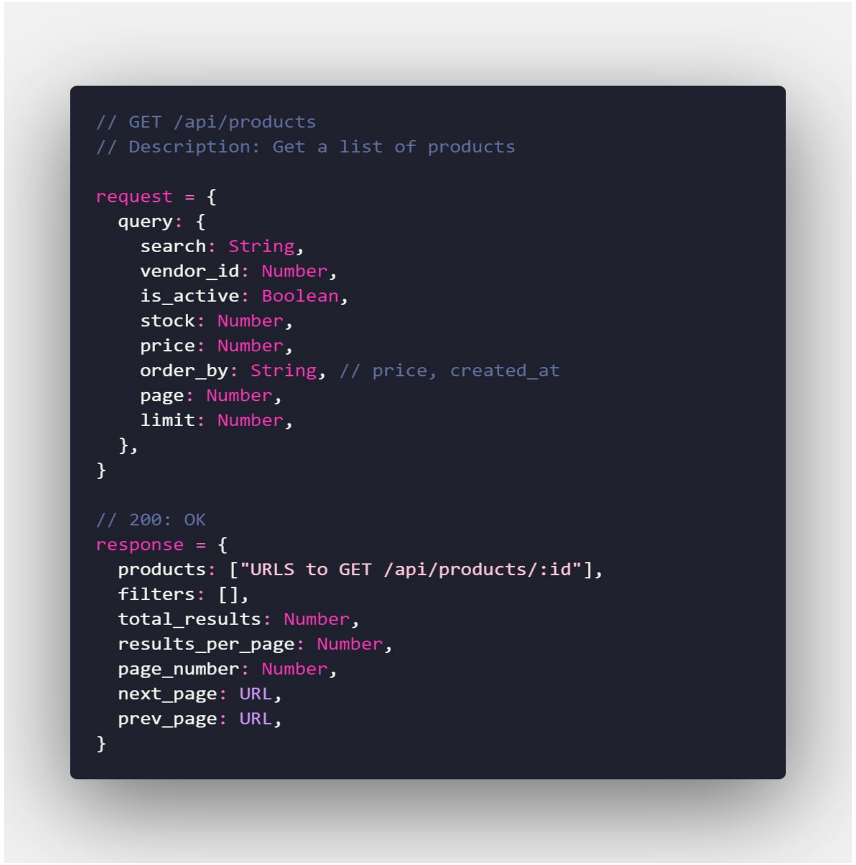
```
//GET /api/users/:id/avatar
//Description: Retrieve the avatar information of a specific user.

headers: {
    Authorization: "Bearer <token>"
  }

Response (200 OK):
{
  "id": string,
  "avatar_url": string,
  "avatar_key": string,
  "userId": string,
  "uploadedAt": string
}

Errors:
- 401 Unauthorized: Token is invalid or not provided
- 403 Forbidden: You are not allowed to access another user's avatar _
- 404 Not Found: Avatar not found
```

GET Product



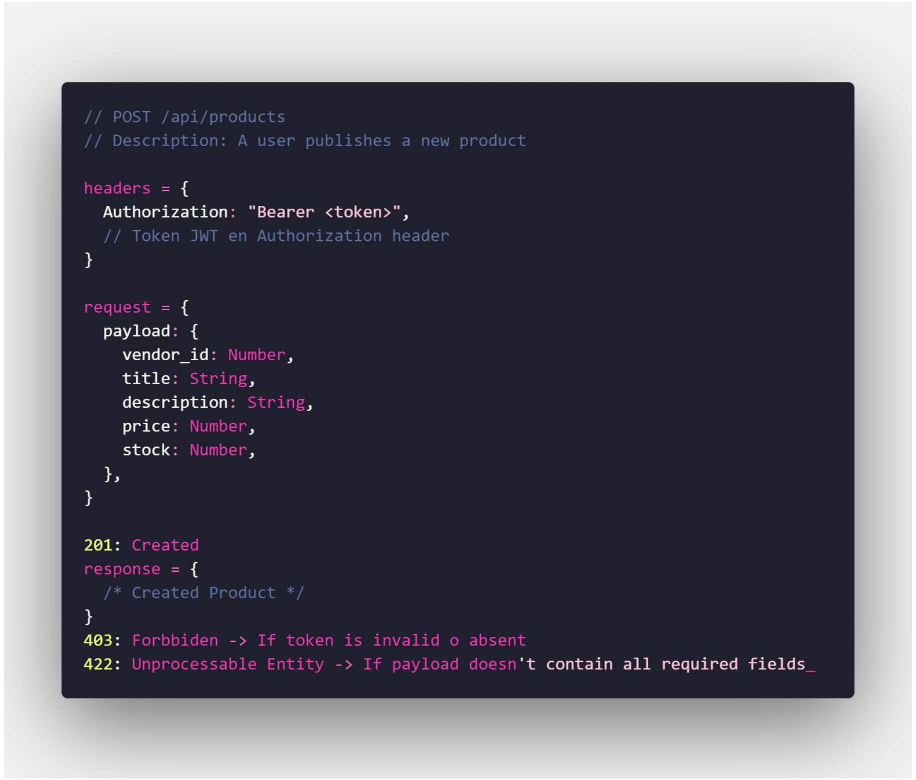
```
// GET /api/products/:id
// Description: Get a specific product

200: OK
response = {
  is_active_product: Boolean,
  created_at: Date,
  title: String,
  description: String,
  price: Number,
  stock: Number,
}
```

GET Products



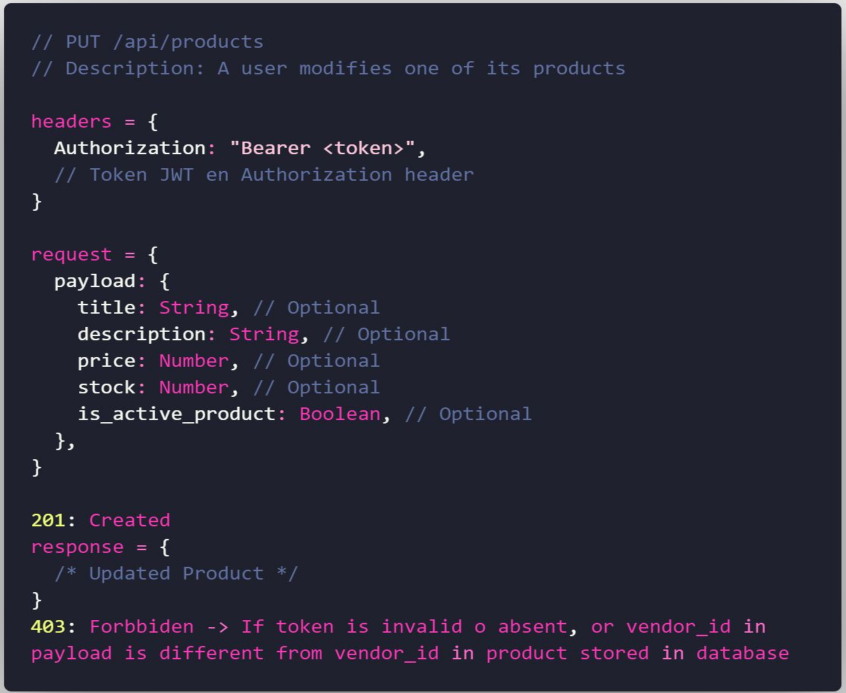
```
// GET /api/products
// Description: Get a list of products

request = {
  query: {
    search: String,
    vendor_id: Number,
    is_active: Boolean,
    stock: Number,
    price: Number,
    order_by: String, // price, created_at
    page: Number,
    limit: Number,
  },
}

// 200: OK
response = {
  products: ["URLS to GET /api/products/:id"],
  filters: [],
  total_results: Number,
  results_per_page: Number,
  page_number: Number,
  next_page: URL,
  prev_page: URL,
}
```

POST Products



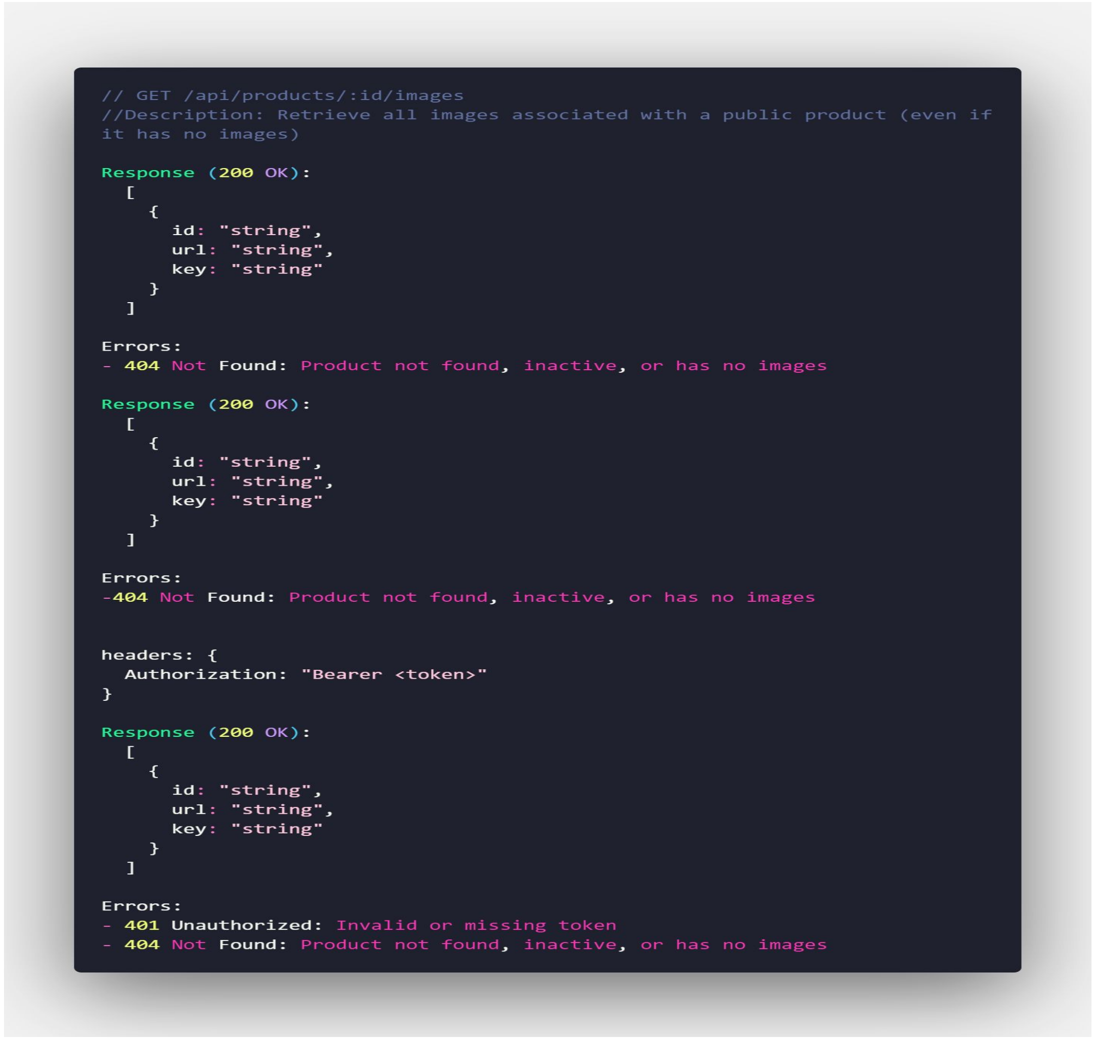
```
// POST /api/products
// Description: A user publishes a new product

headers = {
  Authorization: "Bearer <token>",
  // Token JWT en Authorization header
}

request = {
  payload: {
    vendor_id: Number,
    title: String,
    description: String,
    price: Number,
    stock: Number,
  },
}

201: Created
response = {
  /* Created Product */
}
403: Forbbiden -> If token is invalid o absent
422: Unprocessable Entity -> If payload doesn't contain all required fields_
```

PUT Products

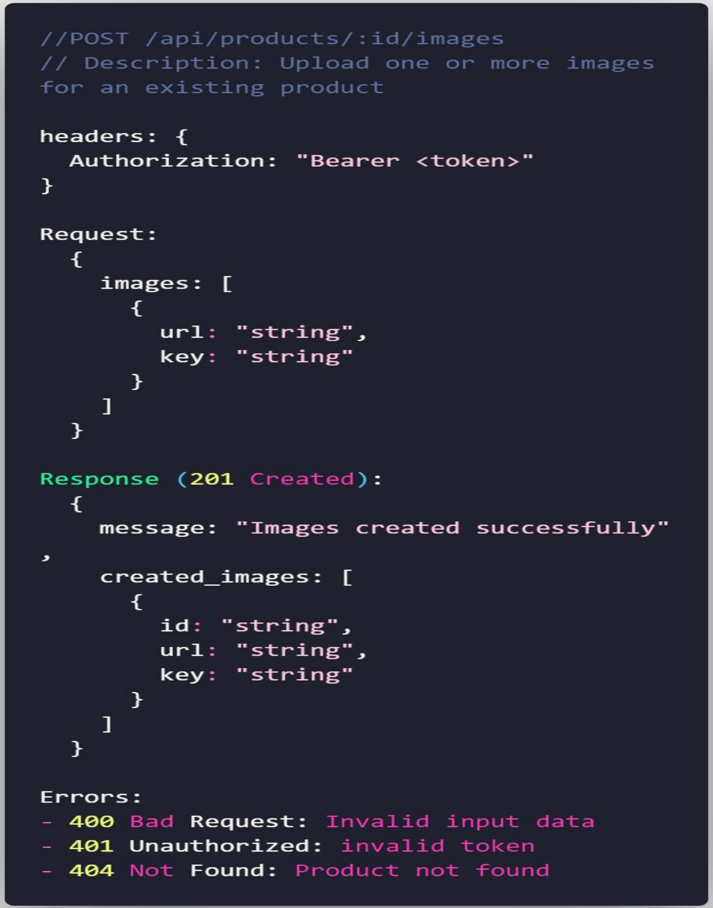


```
// PUT /api/products
// Description: A user modifies one of its products

headers = {
  Authorization: "Bearer <token>",
  // Token JWT en Authorization header
}

request = {
  payload: {
    title: String, // Optional
    description: String, // Optional
    price: Number, // Optional
    stock: Number, // Optional
    is_active_product: Boolean, // Optional
  },
}

201: Created
response = {
  /* Updated Product */
}
403: Forbbiden -> If token is invalid o absent, or vendor_id in
payload is different from vendor_id in product stored in database
```

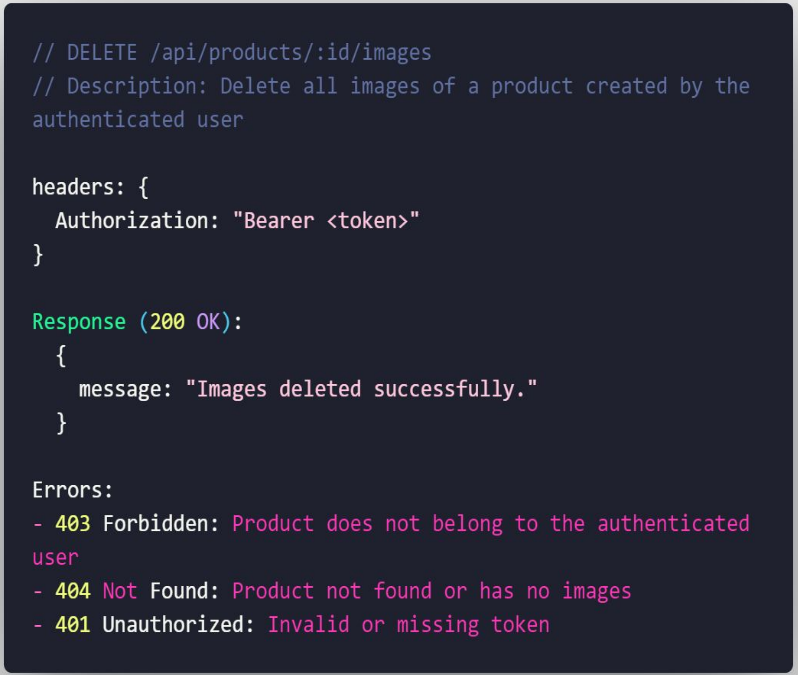
Images



```
// GET /api/products/:id/images
//Description: Retrieve all images associated with a public product (even if
it has no images)

Response (200 OK):
  [
    {
      id: "string",
      url: "string",
      key: "string"
    }
  ]

Errors:
- 404 Not Found: Product not found, inactive, or has no images

Response (200 OK):
  [
    {
      id: "string",
      url: "string",
      key: "string"
    }
  ]

Errors:
-404 Not Found: Product not found, inactive, or has no images


headers: {
  Authorization: "Bearer <token>"
}

Response (200 OK):
  [
    {
      id: "string",
      url: "string",
      key: "string"
    }
  ]

Errors:
- 401 Unauthorized: Invalid or missing token
- 404 Not Found: Product not found, inactive, or has no images
```

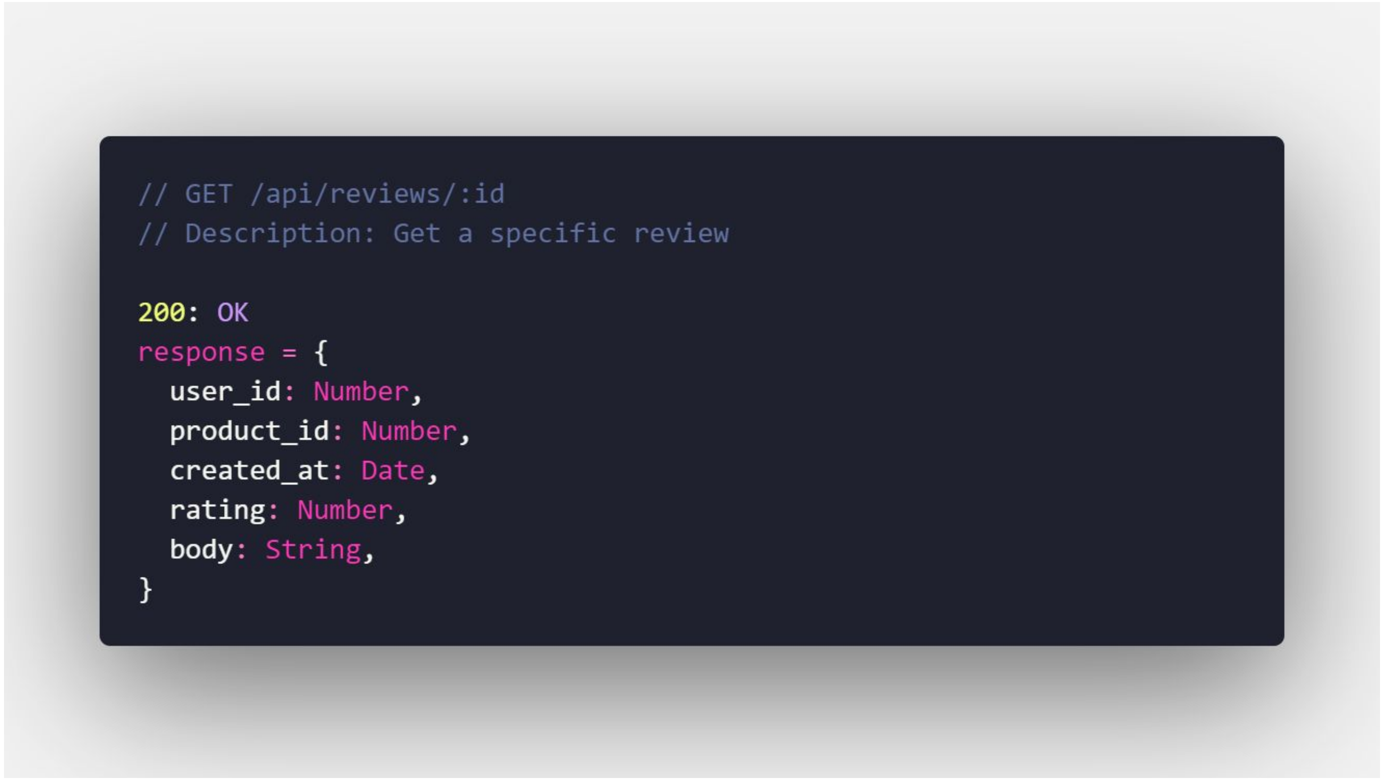


```
//POST /api/products/:id/images
// Description: Upload one or more images
for an existing product

headers: {
  Authorization: "Bearer <token>"
}

Request:
  {
    images: [
      {
        url: "string",
        key: "string"
      }
    ]
  }

Response (201 Created):
  {
    message: "Images created successfully"
,
    created_images: [
      {
        id: "string",
        url: "string",
        key: "string"
      }
    ]
  }

Errors:
- 400 Bad Request: Invalid input data
- 401 Unauthorized: invalid token
- 404 Not Found: Product not found
```

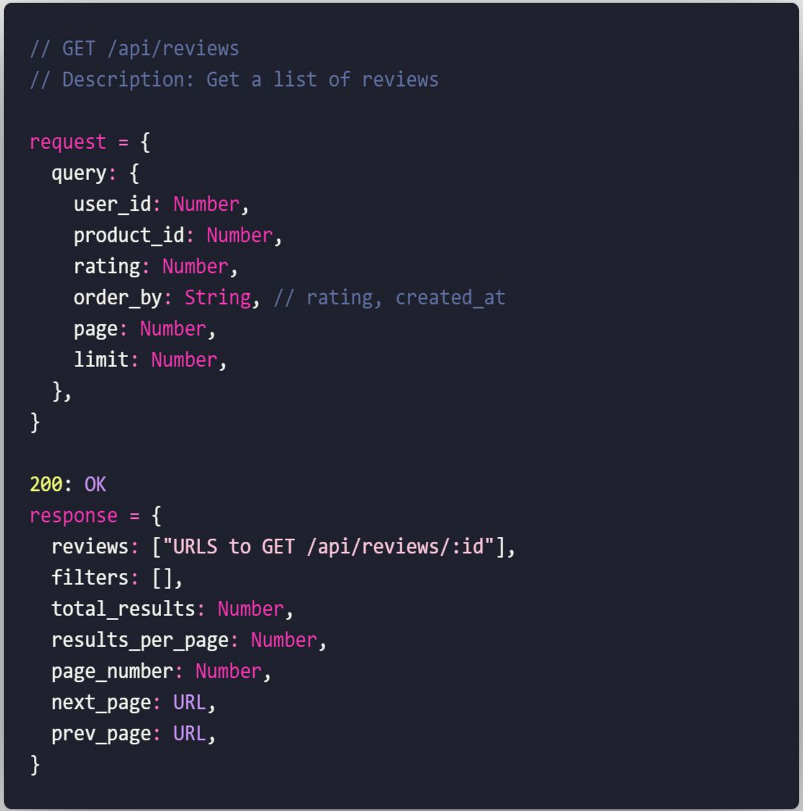


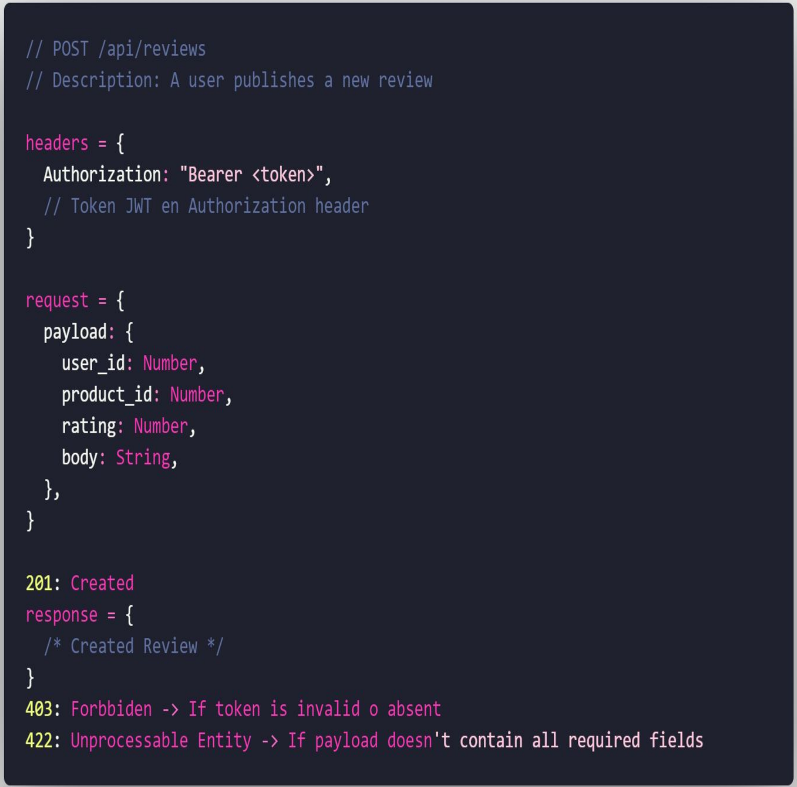
```
// DELETE /api/products/:id/images
// Description: Delete all images of a product created by the
authenticated user

headers: {
  Authorization: "Bearer <token>"
}

Response (200 OK):
  {
    message: "Images deleted successfully."
  }

Errors:
- 403 Forbidden: Product does not belong to the authenticated
user
- 404 Not Found: Product not found or has no images
- 401 Unauthorized: Invalid or missing token
```

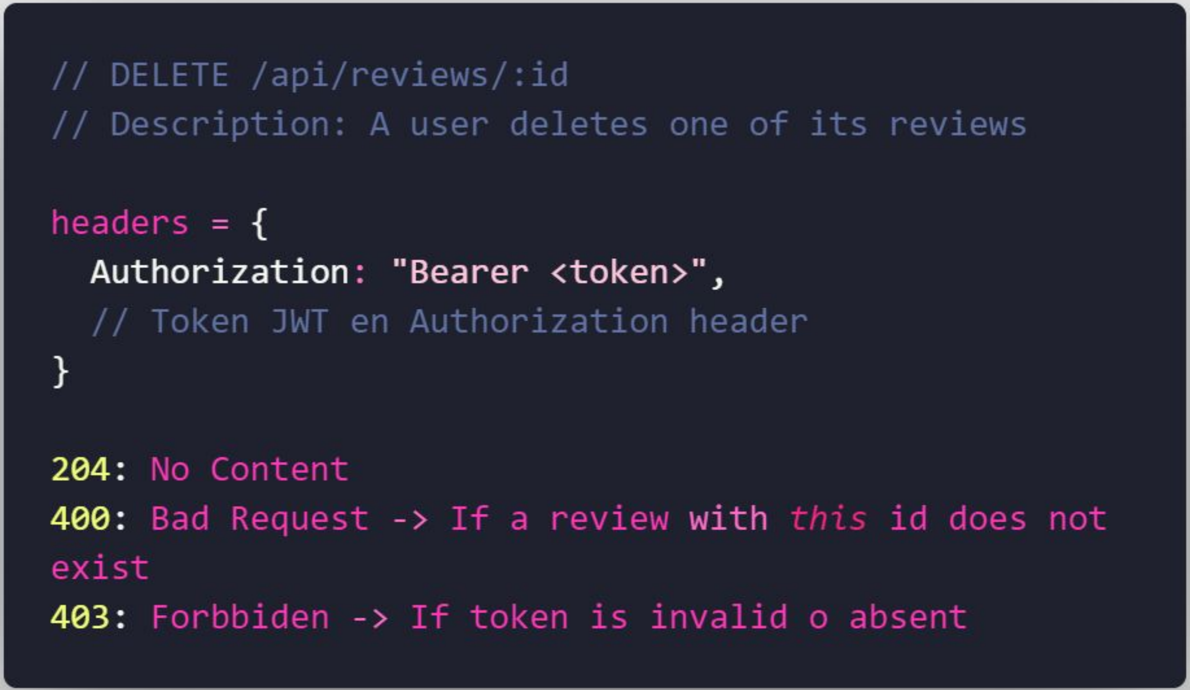
Reviews



```
// GET /api/reviews/:id
// Description: Get a specific review

200: OK
response = {
  user_id: Number,
  product_id: Number,
  created_at: Date,
  rating: Number,
  body: String,
}
```

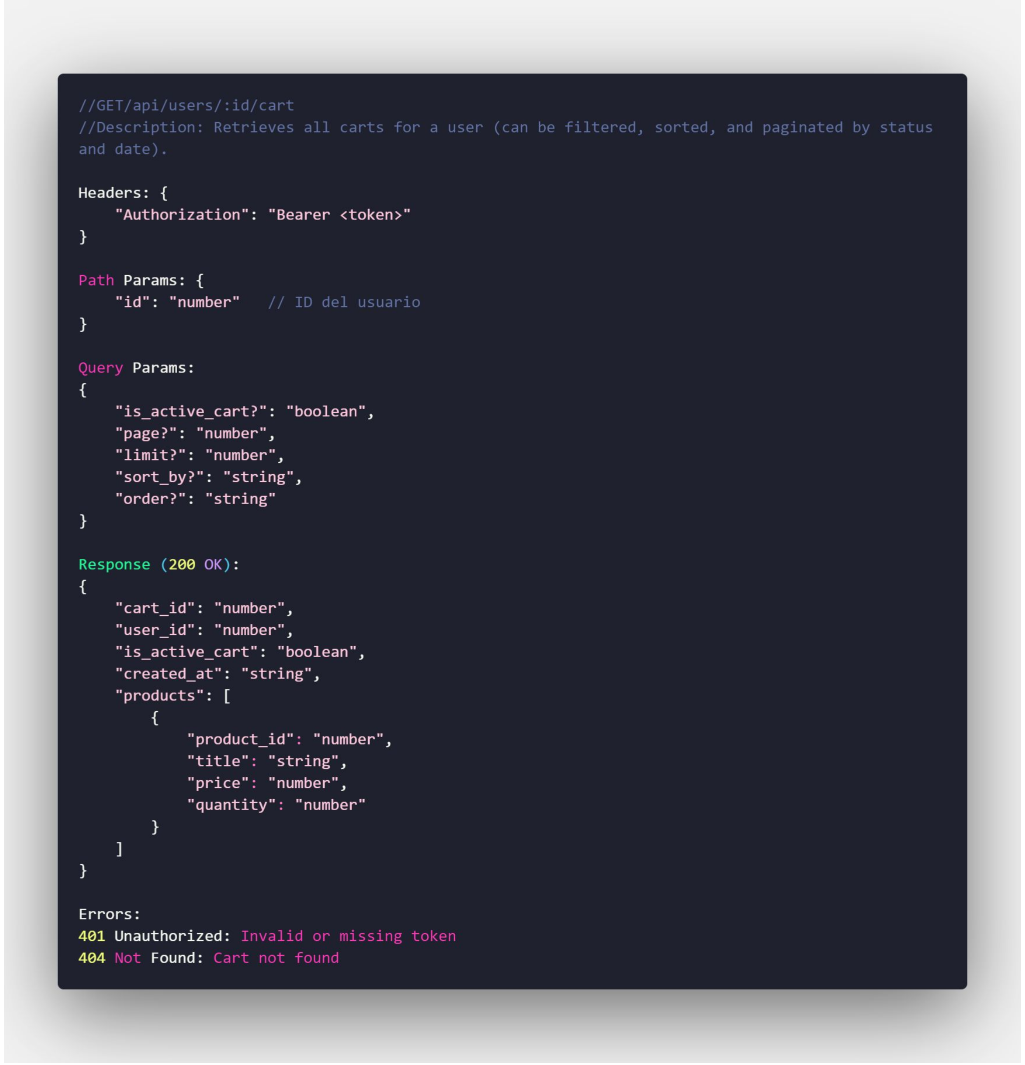

```
// GET /api/reviews
// Description: Get a list of reviews

request = {
  query: {
    user_id: Number,
    product_id: Number,
    rating: Number,
    order_by: String, // rating, created_at
    page: Number,
    limit: Number,
  },
}

200: OK
response = {
  reviews: ["URLS to GET /api/reviews/:id"],
  filters: [],
  total_results: Number,
  results_per_page: Number,
  page_number: Number,
  next_page: URL,
  prev_page: URL,
}
```


```
// POST /api/reviews
// Description: A user publishes a new review

headers = {
  Authorization: "Bearer <token>",
  // Token JWT en Authorization header
}

request = {
  payload: {
    user_id: Number,
    product_id: Number,
    rating: Number,
    body: String,
  },
}

201: Created
response = {
  /* Created Review */
}
403: Forbbiden -> If token is invalid o absent
422: Unprocessable Entity -> If payload doesn't contain all required fields
```

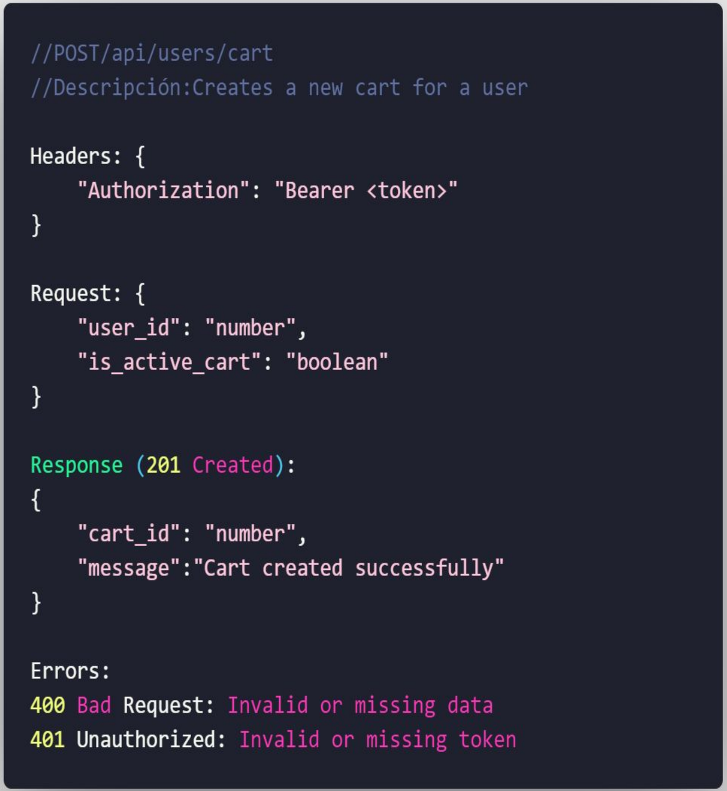


```
// DELETE /api/reviews/:id
// Description: A user deletes one of its reviews

headers = {
  Authorization: "Bearer <token>",
  // Token JWT en Authorization header
}

204: No Content
400: Bad Request -> If a review with this id does not
exist
403: Forbbiden -> If token is invalid o absent
```

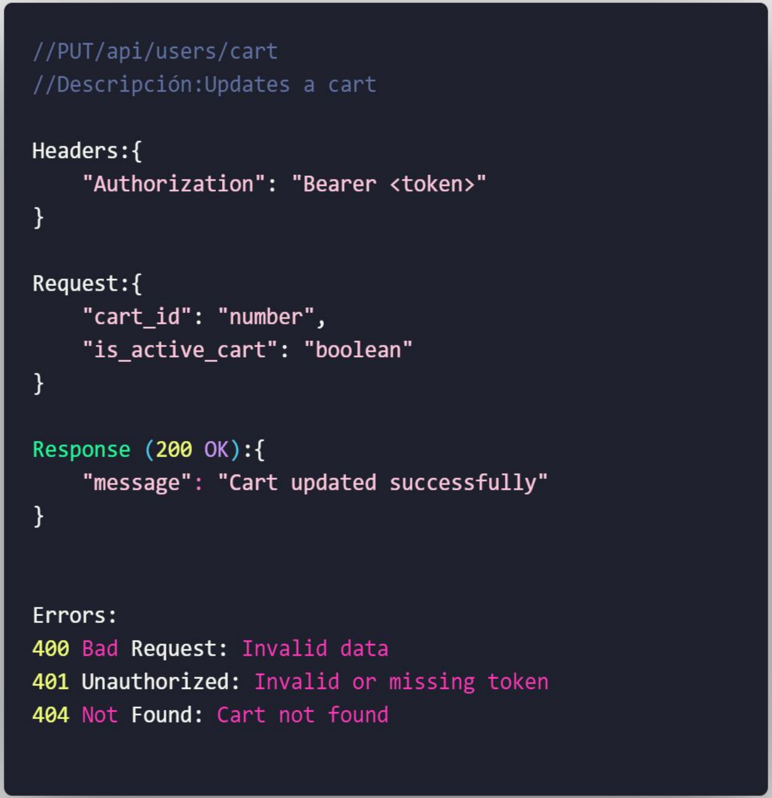
Cart



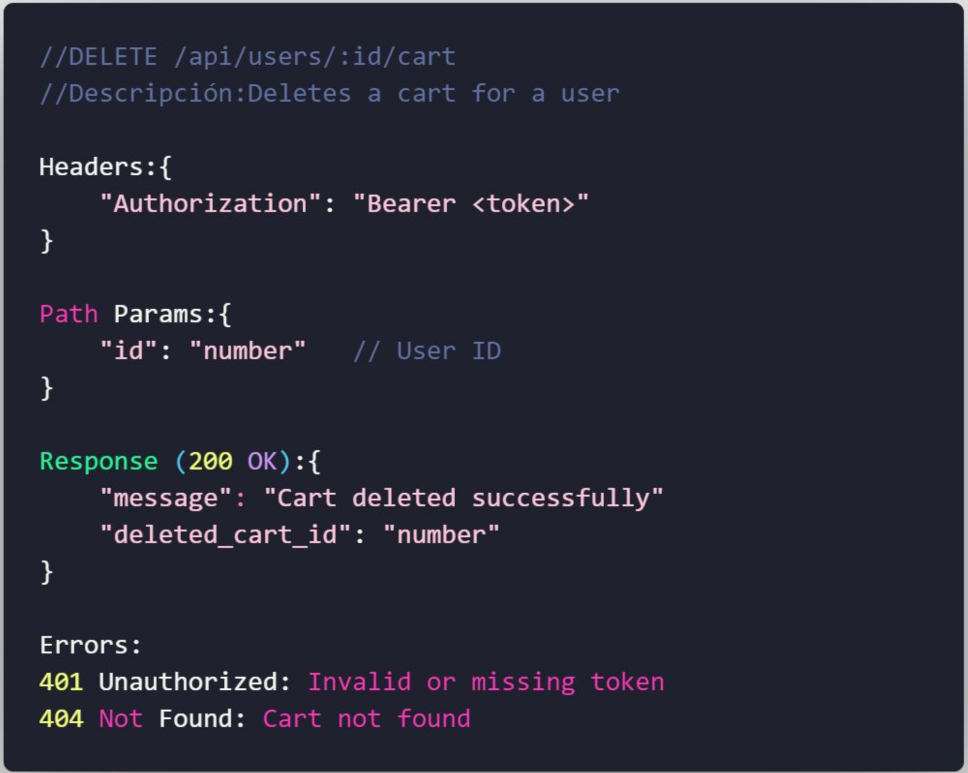
```
//GET/api/users/:id/cart
//Description: Retrieves all carts for a user (can be filtered, sorted, and paginated by status
and date).

Headers: {
    "Authorization": "Bearer <token>"
}

Path Params: {
    "id": "number"   // ID del usuario
}

Query Params:
{
    "is_active_cart?": "boolean",
    "page?": "number",
    "limit?": "number",
    "sort_by?": "string",
    "order?": "string"
}

Response (200 OK):
{
    "cart_id": "number",
    "user_id": "number",
    "is_active_cart": "boolean",
    "created_at": "string",
    "products": [
        {
            "product_id": "number",
            "title": "string",
            "price": "number",
            "quantity": "number"
        }
    ]
}

Errors:
401 Unauthorized: Invalid or missing token
404 Not Found: Cart not found
```



```
//POST/api/users/cart
//Descripción:Creates a new cart for a user

Headers: {
    "Authorization": "Bearer <token>"
}

Request: {
    "user_id": "number",
    "is_active_cart": "boolean"
}

Response (201 Created):
{
    "cart_id": "number",
    "message":"Cart created successfully"
}

Errors:
400 Bad Request: Invalid or missing data
401 Unauthorized: Invalid or missing token
```

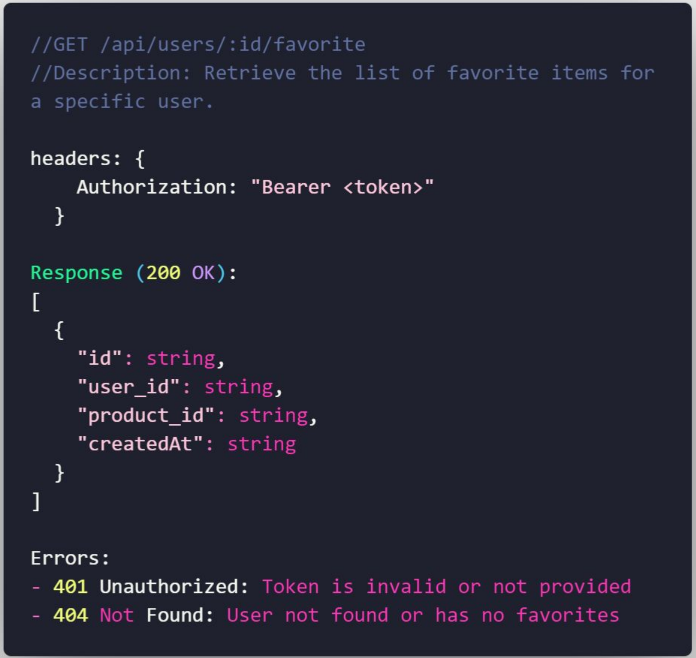


```
//PUT/api/users/cart
//Descripción:Updates a cart

Headers:{
    "Authorization": "Bearer <token>"
}

Request:{
    "cart_id": "number",
    "is_active_cart": "boolean"
}

Response (200 OK):{
    "message": "Cart updated successfully"
}


Errors:
400 Bad Request: Invalid data
401 Unauthorized: Invalid or missing token
404 Not Found: Cart not found
```

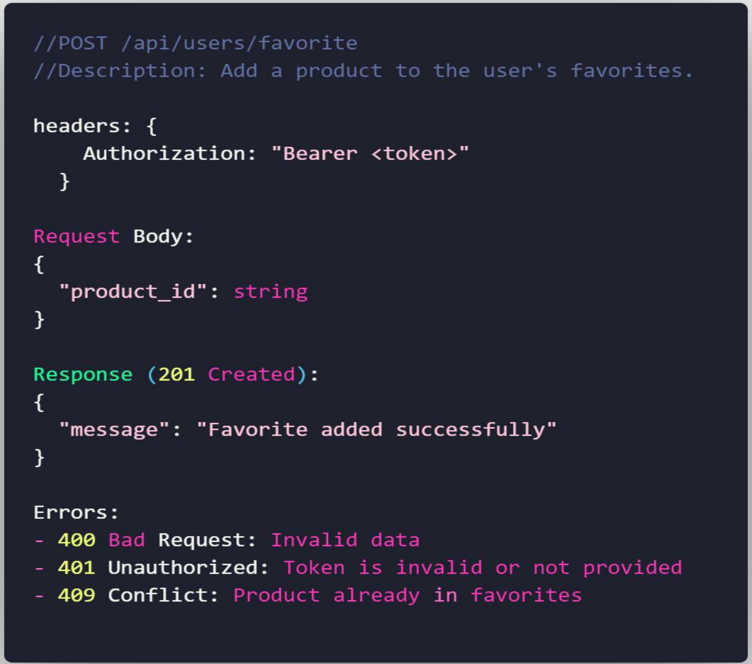


```
//DELETE /api/users/:id/cart
//Descripción:Deletes a cart for a user

Headers:{
    "Authorization": "Bearer <token>"
}

Path Params:{
    "id": "number"   // User ID
}

Response (200 OK):{
    "message": "Cart deleted successfully"
    "deleted_cart_id": "number"
}

Errors:
401 Unauthorized: Invalid or missing token
404 Not Found: Cart not found
```

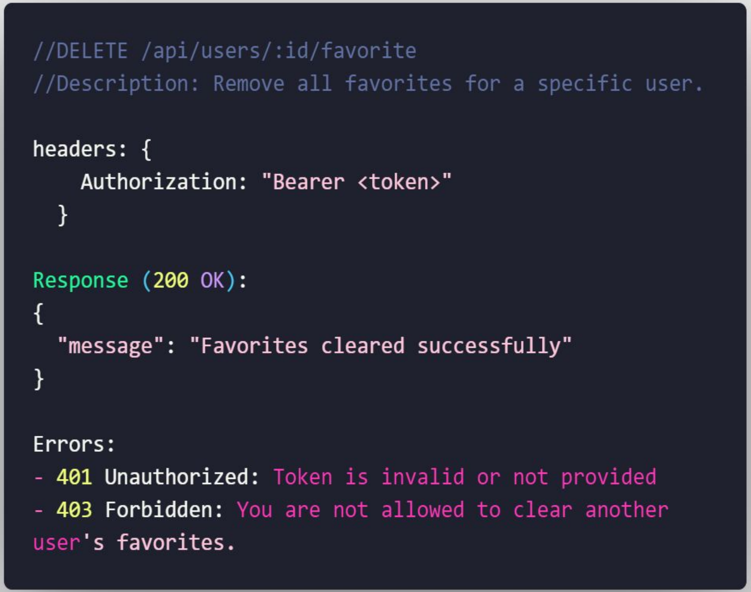
Favorites



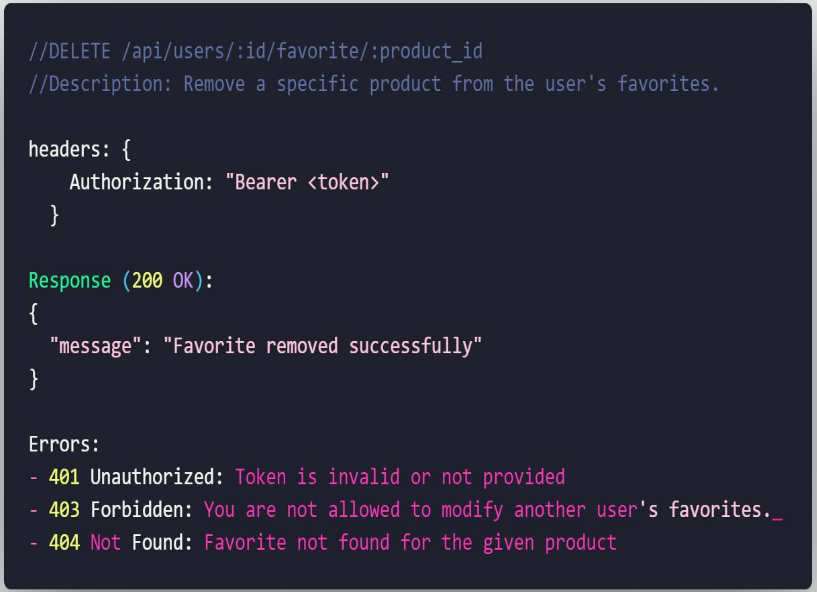
```
//GET /api/users/:id/favorite
//Description: Retrieve the list of favorite items for
a specific user.

headers: {
    Authorization: "Bearer <token>"
  }

Response (200 OK):
[
  {
    "id": string,
    "user_id": string,
    "product_id": string,
    "createdAt": string
  }
]

Errors:
- 401 Unauthorized: Token is invalid or not provided
- 404 Not Found: User not found or has no favorites
```



```
//POST /api/users/favorite
//Description: Add a product to the user's favorites.

headers: {
    Authorization: "Bearer <token>"
  }

Request Body:
{
  "product_id": string
}

Response (201 Created):
{
  "message": "Favorite added successfully"
}

Errors:
- 400 Bad Request: Invalid data
- 401 Unauthorized: Token is invalid or not provided
- 409 Conflict: Product already in favorites
```



```
//DELETE /api/users/:id/favorite
//Description: Remove all favorites for a specific user.

headers: {
    Authorization: "Bearer <token>"
  }

Response (200 OK):
{
  "message": "Favorites cleared successfully"
}

Errors:
- 401 Unauthorized: Token is invalid or not provided
- 403 Forbidden: You are not allowed to clear another
user's favorites.
```



```
//DELETE /api/users/:id/favorite/:product_id
//Description: Remove a specific product from the user's favorites.

headers: {
   Authorization: "Bearer <token>"
  }

Response (200 OK):
{
  "message": "Favorite removed successfully"
}

Errors:
- 401 Unauthorized: Token is invalid or not provided
- 403 Forbidden: You are not allowed to modify another user's favorites._
- 404 Not Found: Favorite not found for the given product
```