Term definitions

Login – synonymous with email

ROLE\_USER – customer user

ROLE\_ADMIN – admin user

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Route** | **Authorization** | **Request** | **Response** | **Description** |
| /v1/auth/signup/customer | N/A | Body:  {  login: String,  password: String,  name: String,  address: String,  city: String,  postalCode: Integer,  country: String  } | **Status**: 201  Body:  "User signed up"  Header:  Session Cookie  **Status**: 400 (Login already in use)  Body:  "Login already in use"  **Status**: 400 (Invalid login/email)  Body:  "Invalid email"  **Status**: 400 (password length < 8)  Body:  "Password less than 8 characters" | Creates a new customer and sends back a new session cookie back. |
| /v1/auth/signin | N/A | Body:  {  login: String,  password: String  } | **Status**: 200  Header:  Session Cookie  **Status**: 400 (Missing login details)  Body:  "Missing login details"  **Status**: 400 (Malformed auth header)  Body:  "Bad Auth Header"  **Status**: 400 (Malformed auth header)  Body:  "Bad Auth Header"  **Status**: 400 (Email/password wrong)  Body:  "Invalid login" | Signs user in sending a new session cookie back. |
| /v1/auth/user/update | Anyone | Header:  Session Cookie  Body:  {  login: String,  password: String  } | **Status**: 200  Body:  "Done"  Header:  Invalidating Session Cookie  **Status**: 400 (Login already in use)  Body:  "Login already in use"  **Status**: 400 (Invalid login/email)  Body:  "Invalid email"  **Status**: 400 (password length < 8)  Body:  "Password less than 8 characters" | Allows a user to update their login credentials. If the user updates their password all previous sessions are invalidated, and user must sign in again. |
| /v1/auth/delete | Anyone | Header:  Session Cookie | **Status**: 200  Body:  "Done" | Allow user to delete their account. |
| /v1/auth/signout | Anyone | Header:  Session Cookie | **Status**: 200  Body:  "Done" | Allows user to end current session cookie. |
| /v1/auth/signout/all | Anyone | Header:  Session Cookie | **Status**: 200  Body:  "Done" | Allows user to end all sessions. |
| /v1/customer/details | ROLE\_USER | Header:  Session Cookie | **Status**: 200  Body:  {  name: String,  address: String,  city: String,  postalCode: Integer,  country: String  } | Allows customer to view their account details. |
| /v1/customer/update | ROLE\_USER | Header:  Session Cookie  Body:  {  name: String,  address: String,  city: String,  postalCode: Integer,  country: String  } | **Status**: 200  Body:  "Done" | Allows customer to update their account details |
| /v1/product/get/all | Anyone | N/A | **Status**: 200  Body:  [  {  productId: Integer,  name: String,  description: String,  units: Integer,  price: Long  }  ] | Packages all product info as a JSON list. |
| /v1/product/get/{productId} | Anyone | Path Variable:  productid: Integer | **Status**: 200  {  productId: Integer,  name: String,  description: String,  units: Integer,  price: Long  }  **Status**: 400 (product with Id unknown)  Body:  " Unknown ProductId" | Packages selected product info as a JSON object. |
| /v1/product/add | ROLE\_ADMIN | Header:  Session Cookie  Body:  {  name: String,  description: String,  units: Integer,  price: Long  } | **Status**: 200  Body:  {  productid: Integer  } | Allows admins to create new products. |
| /v1/product/update | ROLE\_ADMIN | Header:  Session Cookie  Body:  {  id: Integer,  name: String,  description: String,  units: Integer,  price: Long  } | **Status**: 200  Body:  "Done" | Allows admins to update a product’s info. |
| /v1/product/delete | ROLE\_ADMIN | Body:  Integer (productId) | **Status**: 200  Body:  "Done"  **Status**: 400 (product with Id unknown)  Body:  " Unknown ProductId" | Allows admins to delete products. |
| /v1/order/create/simple | ROLE\_USER | Header:  Session Cookie  Body:  {  productId: Integer,  quantity: Integer,  } | **Status**: 200  Body:  {  productid: Integer  }  **Status**: 400 (product with Id unknown)  Body:  " Unknown ProductId" | Creates a new order with the customer’s location details as the order’s destination. |
| /v1/order/create | ROLE\_USER | Header:  Session Cookie  Body:  {  productId: Integer,  quantity: Integer,  address: String,  city: String,  postalCode: Integer,  country: String  } | **Status**: 200  Body:  {  productid: Integer  }  **Status**: 400 (product with Id unknown)  Body:  " Unknown ProductId" | Creates a new order. |
| /v1/order/admin/create | ROLE\_ADMIN | Header:  Session Cookie  Body:  {  login: String,  productId: Integer,  quantity: Integer,  address: String,  city: String,  postalCode: Integer,  country: String  } | **Status**: 200  Body:  {  productid: Integer  }  **Status**: 400 (product with Id unknown)  Body:  " Unknown ProductId"  **Status**: 400 (associated login is not a customer/ login does not exist)  Body:  “Bad request” | Allows admins to create an order for selected customer. |
| /v1/order/admin/update/status | ROLE\_ADMIN | Header:  Session Cookie  Body:  {  orderId: UUID,  orderStatus: OrderStatuses.statuses (String)  } | **Status**: 200  Body:  "Done" | Updates the status of an order. |
| /v1/order/update | ROLE\_USER | Header:  Session Cookie  Body:  {  orderId: UUID,  productId: Integer,  quantity: Integer,  address: String,  city: String,  postalCode: Integer,  country: String  } | **Status**: 200  Body:  "Done"  **Status**: 400 (product with Id unknown)  Body:  "Bad OrderId"  **Status**: 400 (order has already begun or been cancelled)  Body:  "Order underway or cancelled" | Allows a customer to change their order, but only if the order has not already begun or been cancelled. |
| /v1/order/admin/update | ROLE\_ADMIN | Header:  Session Cookie  Body:  {  orderId: UUID,  customerLogin: String,  productId: Integer,  quantity: Integer,  address: String,  city: String,  postalCode: Integer,  country: String  } | **Status**: 200  Body:  "Done"  **Status**: 400 (product with Id unknown)  Body:  "Bad OrderId"  **Status**: 400 (associated login is not a customer/ login does not exist)  Body:  “Bad customer login” | Allows admins to update order’s details but not its status as this is done by /v1/order/admin/update/status. |
| /v1/order/get/active | ROLE\_USER | Header:  Session Cookie | **Status**: 200  Body:  [  {  orderId: UUID,  customerLogin: String,  productId: Integer,  quantity: Integer,  address: String,  city: String,  postalCode: Integer,  country: String  }  ] | Packages the customer’s active orders as a JSON list. |
| /v1/order/get/all | ROLE\_USER | Header:  Session Cookie | **Status**: 200  Body:  [  {  orderId: UUID,  customerLogin: String,  productId: Integer,  quantity: Integer,  address: String,  city: String,  postalCode: Integer,  country: String  }  ] | Packages the all customer’s orders as a JSON list. |
| /v1/order/get/{orderId} | ROLE\_USER | Header:  Session Cookie  Path Variable:  productid: String | **Status**: 200  Body:  {  orderId: UUID,  customerLogin: String,  productId: Integer,  quantity: Integer,  address: String,  city: String,  postalCode: Integer,  country: String  } | Packages the customer’s select order as a JSON object. |
| /v1/hello | ROLE\_USER, ROLE\_ADMIN | Header:  Session Cookie | **Status**: 200  Body:  "Hello, World!" | Testing route. |
| /v1/hello/user | ROLE\_USER | Header:  Session Cookie | **Status**: 200  Body:  "Hello, admin!" | Testing route. |
| /v1/hello/admin | ROLE\_ADMIN | Header:  Session Cookie | **Status**: 200  Body:  "Hello, admin!" | Testing route. |

-