

GraphQL Endpoints - Input and Output

Update Inventory Quantity (Mutation)

Input:

quantity: Float!, shopId: Float!, itemId: Float!

Output:

updateInventoryQuantity

Get Item By ID (Query)

Input:

itemId: Float!

Output:

```
item {  
  item_id  
  name  
  description  
  price  
  item_image  
  quantity  
}  
success
```

Search Items (Query)

Input:

query: String!

Output:

```
item_id  
name  
description  
price  
item_image  
quantity
```

Create Item (Mutation)

Input:

```
quantity: Float!  
shopId: Float!  
categoryId: Float!  
price: Float!  
description: String!  
name: String!  
itemImage: String
```

Output:

```
success  
item {  
  quantity  
  name  
  price  
  item_image  
  item_id  
  description  
}
```

Update Item (Mutation)

Input:

```
quantity: Float!  
categoryId: Float!  
price: Float!  
description: String!  
name: String!  
itemId: Float!  
itemImage: String
```

Output:

updateItem

Delete Item (Mutation)

Input:

itemId: Float!

Output:

deleteItem

Get Items By Category (Query)

Input:

categoryId: Float!

Output:

item_id
name
description
price
item_image
quantity

Get Shop By Item ID (Query)

Input:

itemId: Float!

Output:

shop_id
name
address
open
contact_number

Get User Notifications (Query)

Input:

userId: Float!

Output:

notification_id

```
message
created_at
```

Create Notification (Mutation)

Input:

message: String!, userId: Float!

Output:

```
notification_id
message
created_at
```

Get Order By ID (Query)

Input:

orderId: Float!

Output:

```
order {
  order_id
  total_price
  order_date
  status
  user_id
}
success
```

Get User Order History (Query)

Input:

userId: Float!

Output:

```
order_id
total_price
order_date
status
```

user_id

Create Order (Mutation)

Input:

input: CreateOrderInput!

Output:

```
success
order {
  total_price
  status
  order_id
  order_date
  user_id
}
```

Get Shop By ID (Query)

Input:

shopId: Float!

Output:

```
shop {
  shop_id
  name
  address
  open
  contact_number
}
success
```

Create Shop (Mutation)

Input:

```
ownerId: Float!
address: String!
name: String!
```

contactNumber: String
open: Boolean

Output:

```
shop {  
  shop_id  
  name  
  address  
  open  
  contact_number  
}  
success
```

Get User By ID (Query)

Input:

userId: Float!

Output:

```
user {  
  user_id  
  name  
  email  
  password_hash  
  user_address  
}  
success  
token
```

Create User (Mutation)

Input:

roleId: Float!
password: String!
email: String!
name: String!
userAddress: String

Output:

```
user {
  user_id
  name
  email
  password_hash
  user_address
}
success
token
```

Search User By Email (Query)

Input:

email: String!

Output:

```
user {
  user_id
  name
  email
  password_hash
  user_address
}
success
token
```

Get User Address (Query)

Input:

userId: Float!

Output:

getUserAddress

Update User (Mutation)

Input:

roleId: Float!

email: String!
name: String!
userId: Float!
userAddress: String

Output:
updateUser

Delete User (Mutation)

Input:
userId: Float!

Output:
deleteUser

Get User Role (Query)

Input:
userId: Float!

Output:

role_id
role_name

Get User Shops (Query)

Input:
userId: Float!

Output:

address
contact_number
name
open
shop_id

Get User Shop ID (Query)

Input:

userId: Float!

Output:

getUserShopId

Assign User To Shop (Mutation)

Input:

roleId: Float!

shopId: Float!

userId: Float!

Output:

assignUserToShop

Login With Email Password (Mutation)

Input:

password: String!

email: String!

Output:

```
user {  
  user_id  
  name  
  email  
  password_hash  
  user_address  
}  
success  
token
```

Register With Email Password (Mutation)

Input:

```
userAddress: String!  
roleId: Float!  
password: String!  
email: String!  
name: String!
```

Output:

```
user {  
  user_id  
  name  
  email  
  password_hash  
  user_address  
}  
success  
token
```