1. User Management Service

a. Register New User

- **Endpoint:** `POST /users/register`

Request Body:
json
Copy code

```
{
  "username": "string",
  "email": "string",
  "password": "string"
}
```

- `201 Created` - User successfully registered.
- `400 Bad Request` - Invalid request data.
- `409 Conflict` - User already exists.

Response:
json
Copy code

```
{
  "userId": "string",
  "username": "string",
  "email": "string"
}
```

-

b. Authenticate User

- **Endpoint:** `POST /users/login`

Request Body:
json
Copy code

```
{
  "username": "string",
  "password": "string"
}
```

- `200 OK` - Authentication successful, token issued.
- `400 Bad Request` - Invalid username or password.
- `401 Unauthorized` - Authentication failed.

Response:
json
Copy code

```
{
  "token": "string"
}
```

-

c. Retrieve User Profile

- **Endpoint:** `GET /users/{userId}`

Response:
json
Copy code

```
{
  "userId": "string",
  "username": "string",
  "email": "string",
  "notifications": [
    {
      "notificationId": "string",
      "type": "string",
      "message": "string"
    }
  ]
}
```

- `200 OK` - User profile retrieved successfully.
- `404 Not Found` - User not found.

d. Update User Profile

- **Endpoint:** `PUT /users/{userId}`

Request Body:
json
Copy code

```json
{
  "username": "string",
  "email": "string"
}
```

- `200 OK` - User profile updated successfully.
- `400 Bad Request` - Invalid data.
- `404 Not Found` - User not found.

Response:
json
Copy code

```json
{
  "userId": "string",
  "username": "string",
  "email": "string"
}
```

-

e. Change User Password

- **Endpoint:** `PUT /users/{userId}/password`

Request Body:
json
Copy code

```json
{
  "oldPassword": "string",
  "newPassword": "string"
}
```

- `200 OK` - Password updated successfully.
- `400 Bad Request` - Invalid old password or new password.
- `404 Not Found` - User not found.

Response:

json

Copy code

```
{
  "message": "Password updated successfully"
}
```

-

f. Delete User Account

- **Endpoint:** `DELETE /users/{userId}`

Response:

json

Copy code

```
{
  "message": "User deleted successfully"
}
```

- `200 OK` - User account deleted successfully.
- `404 Not Found` - User not found.

g. Retrieve User Notifications

- **Endpoint:** `GET /users/{userId}/notifications`

Response:

json

Copy code

```
{
  "notifications": [
    {
      "notificationId": "string",
      "type": "string",
      "message": "string"
    }
  ]
}
```

- `200 OK` - Notifications retrieved successfully.

- `404 Not Found` - User not found.

2. Trade Service

a. Initiate Buy Request

- **Endpoint:** `POST /trade/buy`

Request Body:
json
Copy code

```
{
  "sourceStationId": "string",
  "destinationStationId": "string",
  "goodsId": "string",
  "quantity": "integer"
}
```

- `201 Created` - Buy request initiated successfully.
- `400 Bad Request` - Invalid request data.
- `404 Not Found` - Space station or goods not found.

Response:
json
Copy code

```
{
  "transactionId": "string",
  "status": "string",
  "message": "Buy request initiated successfully"
}
```

-

b. Initiate Sell Request

- **Endpoint:** `POST /trade/sell`

Request Body:
json

Copy code

```
{
  "sourceStationId": "string",
  "destinationStationId": "string",
  "goodsId": "string",
  "quantity": "integer"
}
```

- `201 Created` - Sell request initiated successfully.
- `400 Bad Request` - Invalid request data.
- `404 Not Found` - Space station or goods not found.

Response:
json
Copy code

```
{
  "transactionId": "string",
  "status": "string",
  "message": "Sell request initiated successfully"
}
```

c. Check Trade Status

- **Endpoint:** `GET /trade/status/{transactionId}`

Response:
json
Copy code

```
{
  "transactionId": "string",
  "status": "string",
  "details": "string"
}
```

- `200 OK` - Trade status retrieved successfully.
- `404 Not Found` - Transaction not found.

3. Inventory Service

a. Check Stock Levels

- **Endpoint:** `GET /inventory/check/{goodsId}`
- **Query Parameter:** `stationId`

Response:
json
Copy code

```
{
  "stationId": "string",
  "goodsId": "string",
  "stockLevel": "integer"
}
```

- `200 OK` - Stock levels retrieved successfully.
- `400 Bad Request` - Missing or invalid query parameters.
- `404 Not Found` - Goods or space station not found.

b. Update Stock Levels

- **Endpoint:** `POST /inventory/update`

Request Body:
json
Copy code

```
{
  "stationId": "string",
  "goodsId": "string",
  "newStockLevel": "integer"
}
```

- `200 OK` - Stock levels updated successfully.
- `400 Bad Request` - Invalid request data.
- `404 Not Found` - Space station or goods not found.

Response:
json

Copy code

```
{
  "stationId": "string",
  "goodsId": "string",
  "newStockLevel": "integer"
}
```

-

4. Cargo Management Service

a. Create Shipment Order

- **Endpoint:** `POST /shipment/create`

Request Body:
json
Copy code

```
{
  "sourceStationId": "string",
  "destinationStationId": "string",
  "goodsId": "string",
  "quantity": "integer",
  "shipmentDetails": "string"
}
```

- `201 Created` - Shipment order created successfully.
- `400 Bad Request` - Invalid request data.
- `404 Not Found` - Space station or goods not found.

Response:
json
Copy code

```
{
  "shipmentId": "string",
  "status": "string",
  "message": "Shipment order created successfully"
}
```

b. Track Shipments

- **Endpoint:** `GET /shipment/tracking/{trackingId}`

Response:
json
Copy code

```
{
  "trackingId": "string",
  "status": "string",
  "currentLocation": "string",
  "estimatedArrival": "string"
}
```

- `200 OK` - Shipment status retrieved successfully.
- `404 Not Found` - Tracking ID not found.

c. Report Shipment Issues

- **Endpoint:** `POST /shipment/issues`

Request Body:
json
Copy code

```
{
  "shipmentId": "string",
  "issueDescription": "string",
  "reportedBy": "string"
}
```

-

Response:
json
Copy code

```
{
  "message": "Issue reported successfully"
}
```

- `201 Created` - Issue reported successfully.
- `400 Bad Request` - Invalid request data.
- `404 Not Found` - Shipment not found.

5. Real-Time Notification Service

a. Notify Users

- **Endpoint:** `POST /notifications/user`

Request Body:
json
Copy code

```
{
  "userId": "string",
  "notificationType": "string",
  "message": "string"
}
```

- `201 Created` - Notification sent successfully.
- `400 Bad Request` - Invalid request data.
- `404 Not Found` - User not found.

Response:
json
Copy code

```
{
  "message": "Notification sent successfully"
}
```

6. Space Station Service

a. List All Space Stations

- **Endpoint:** `GET /locations/space-stations`

Response:
json
Copy code

```
[
  {
    "id": "string",
    "name": "string",
    "location": "string",
    "status": "string"
  }
]
```

- `200 OK` - List of space stations retrieved successfully.

b. Get Space Station Details

- **Endpoint:** `GET /locations/space-stations/{id}`

Response:
json
Copy code

```
{
  "id": "string",
  "name": "string",
  "location": "string",
  "status": "string",
  "inventory": [
    {
      "goodsId": "string",
      "stockLevel": "integer"
    }
  ]
}
```

- `200 OK` - Space station details retrieved successfully.
- `404 Not Found` - Space station not found.

c. Update Inventory at Space Station

- **Endpoint:** `POST /locations/space-stations/{id}/inventory`

Request Body:
json
Copy code

```
{
  "goodsId": "string",
  "newStockLevel": "integer"
}
```

- `200 OK` - Inventory updated successfully.
- `400 Bad Request` - Invalid request data.
- `404 Not Found` - Space station or goods not found.

Response:
json
Copy code

```
{
  "stationId": "string",
  "goodsId": "string",
  "newStockLevel": "integer"
}
```

-

7. Item Management Service

a. Create Item

- **Endpoint:** `POST /items`

Request Body:
json
Copy code

```
{
  "name": "string",
  "description": "string",
  "category": "string",
  "price": "number"
}
```

- `201 Created` - Item created successfully.
- `400 Bad Request` - Invalid request data.

Response:
json
Copy code

```
{
  "itemId": "string",
  "name": "string",
  "description": "string",
  "category": "string",
  "price": "number"
}
```

-

b. Retrieve Item Details

- **Endpoint:** `GET /items/{itemId}`

Response:
json
Copy code

```
{
  "itemId": "string",
  "name": "string",
  "description": "string",
  "category": "string",
  "price": "number"
}
```

- `200 OK` - Item details retrieved successfully.
- `404 Not Found` - Item not found.

c. Update Item

- **Endpoint:** `PUT /items/{itemId}`

Request Body:
json
Copy code

```
{
  "name": "string",
  "description": "string",
  "category": "string",
  "price": "number"
}
```

- `200 OK` - Item updated successfully.
- `400 Bad Request` - Invalid request data.
- `404 Not Found` - Item not found.

Response:
json
Copy code

```
{
  "itemId": "string",
  "name": "string",
  "description": "string",
  "category": "string",
  "price": "number"
}
```

-

d. Delete Item

- **Endpoint:** `DELETE /items/{itemId}`

Response:
json
Copy code

```
{
  "message": "Item deleted successfully"
}
```

- `200 OK` - Item deleted successfully.
- `404 Not Found` - Item not found