**Portfolio Management API Documentation**

**Base URL:**

http://example.com/api/portfolio

**Endpoints:**

**1. Get Entire Portfolio**

- Endpoint: /

- Method: GET

- Description: Retrieves the entire portfolio with trades.

- Request: None

- Response:

{

"success": true,

"data": {

// Portfolio data here

}

}

**2. Get Holdings**

- Endpoint: /holdings

- Method: GET

- Description: Retrieves holdings in an aggregate view.

- Request: None

- Response:

{

"success": true,

"data": {

// Holdings data here

}

}

**3. Get Cumulative Returns**

- Endpoint: /returns

- Method: GET

- Description: Retrieves cumulative returns.

- Request: None

- Response:

{

"success": true,

"data": {

"cumulativeReturn": 10.5 // Cumulative return percentage

}

}

**4. Add a New Trade**

- Endpoint: /addTrade

- Method: POST

- Description: Adds a new trade to the portfolio.

- Request:

{

"stock": "stock\_id",

"date": "trade\_date",

"price": 50.25,

"type": "BUY" // or "SELL"

}

- Response:

{

"success": true,

"data": {

// Newly added trade data here

}

}

**5. Update an Existing Trade**

- Endpoint: /updateTrade

- Method: POST

- Description: Updates an existing trade in the portfolio.

- Request:

{

"id": "trade\_id",

"date": "new\_trade\_date",

"price": 55.75,

"type": "SELL" // or "BUY"

}

- Response:

{

"success": true,

"data": {

// Updated trade data here

}

}

6. Remove a Trade

- Endpoint: /removeTrade

- Method: POST

- Description: Removes a trade from the portfolio.

- Request:

{

"id": "trade\_id"

}

- Response:

{

"success": true,

"message": "Trade removed successfully"

}

**Response Status Codes:**

- 200 OK - Successful request

- 500 Internal Server Error - Server error, failed request

**Error Responses:**

In case of errors, the response will include a "success" flag set to "false" and a "message" indicating the error encountered.

**Notes:**

- All routes are prefixed with /api/portfolio.

- Replace placeholders like stock\_id and trade\_id with actual values in the requests.

- Prices are in currency units (e.g., USD).

- Dates should be in a valid date format.

- Types for trades can be either "BUY" or "SELL".

- Ensure proper error handling on the client side to handle potential errors.