

# Watchlist Feature for Stock Broking Application

This feature allows users to create multiple watchlists to track their favorite stocks. Users can create, view, update, and delete watchlists as well as add and remove stocks from them.

## Database Schema

The feature adds three new tables to the existing database:

- 1. `user_watchlist` - Links users to their watchlists (one per user)
- 2. `watchlist` - Stores the actual watchlists belonging to a user
- 3. `watchlist_item` - Stores stock symbols within each watchlist

## API Endpoints

Method	Endpoint	Description
GET	<code>/api/watchlists</code>	Get all user's watchlists
GET	<code>/api/watchlists/</code>	Get watchlist by ID with all items
POST	<code>/api/watchlists</code>	Create a new watchlist
POST	<code>/api/watchlists/ /items</code>	Add a stock symbol to a watchlist
DELETE	<code>/api/watchlists/ /items/</code>	Remove a stock from a watchlist
DELETE	<code>/api/watchlists/</code>	Delete a watchlist

## Request/Response Format

### Create Watchlist

Request:

json

Copy

```
{
  "name": "My Tech Stocks",
  "description": "Technology companies I'm tracking"
}
```

Response:

json

 Copy

```
{
  "success": true,
  "data": {
    "id": 1,
    "name": "My Tech Stocks",
    "description": "Technology companies I'm tracking",
    "is_default": false,
    "created_at": "2025-04-04T12:00:00.000Z",
    "updated_at": "2025-04-04T12:00:00.000Z",
    "items_count": 0
  }
}
```

## Add to Watchlist

### Request:

json

 Copy

```
{
  "symbol": "AAPL"
}
```

### Response:

json

 Copy

```
{
  "success": true,
  "data": {
    "id": 1,
    "symbol": "AAPL",
    "added_at": "2025-04-04T12:05:00.000Z"
  }
}
```

## Features

- **Default Watchlist:** When a user creates their first watchlist, it's automatically marked as the default
- **Duplicate Prevention:** System prevents adding the same stock to a watchlist multiple times

- **Cascade Delete:** When a watchlist is deleted, all its items are automatically deleted
- **Error Handling:** Comprehensive error handling with appropriate status codes and error messages

## Implementation Files

1. **Database Schema:** Update to DB definition with Watchlist interfaces
2. **Types:** TypeScript definitions for requests and responses
3. **Database Service:** Database operations for watchlists
4. **Controller:** Business logic and request/response handling 5