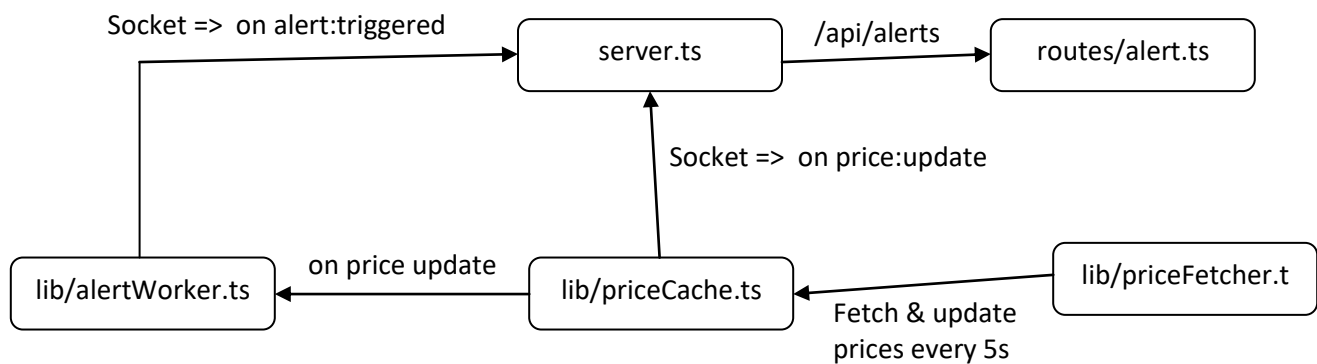


Files and their responsibilities:

- `server.ts` ⇒ Main server file responsible for initialize the socket connection, database connection (mongoDb server) and managing the environment.
- `lib/alertWorker.ts` ⇒ Responsible for listening to price changes and based on it sending alerts.
- `lib/priceCache.ts` ⇒ Responsible for managing Redis Price details Cache.
- `lib/priceFetcher.t` ⇒ Responsible for fetching price data on regular interval (5s).
- `routes/alert.ts` ⇒ Responsible for creating and sending alert data (manages REST APIs)
- `models/Alert.ts` ⇒ Contains models details of mongoDb object.

Simple Data Flow Diagram:



Challenges I Faced

The main challenge I faced when creating this app is assigning a userId to each client without implementing auth feature.

Solution: I used socket.io 's ID as a userId to store the necessary details like alerts etc.