The task is to build a REST API using the FastAPI framework in Python. The API will serve trade data from a mocked database.

The requirements for the API:

1. **Listing trades:** Create an endpoint to fetch a list of trades.
2. **Single trade:** Provide an endpoint to fetch a trade by its ID.
3. **Searching trades:** Implement an endpoint that allows users to search for trades based on fields such as counterparty, instrumentId, instrumentName, and trader.
4. **Advanced filtering:** Add support for optional query parameters to filter trades based on assetClass, tradeDateTime (start and end), tradeDetails.price (minPrice and maxPrice), and tradeDetails.buySellIndicator (tradeType).
5. **Bonus points:** Implement pagination and sorting on the list of trades.

To steps complete this task:

1. Install FastAPI and Pydantic using the links provided in the resources section.
2. Create a new Python file for your API.
3. Import the necessary modules: fastapi, pydantic, and datetime.
4. Define the Pydantic models for the Trade and TradeDetails classes based on the provided schema model.
5. Create an instance of the FastAPI application.
6. Implement the endpoints for listing trades, fetching a single trade by ID, searching trades, and filtering trades. Use the appropriate HTTP methods (GET, POST, etc.) and path parameters/query parameters as specified.
7. Implement the logic for retrieving trade data from the mocked database or any other data source you choose to use. You can generate random data for the purposes of this test.
8. Add support for pagination and sorting if you want to earn bonus points. You can use query parameters to control the page size, current page, and sorting order.Remember that the familiarity with FastAPI is not a requirement, and your submission will be reviewed based on the functionality of the API you build. However, try to produce a functional API that meets the requirements specified.