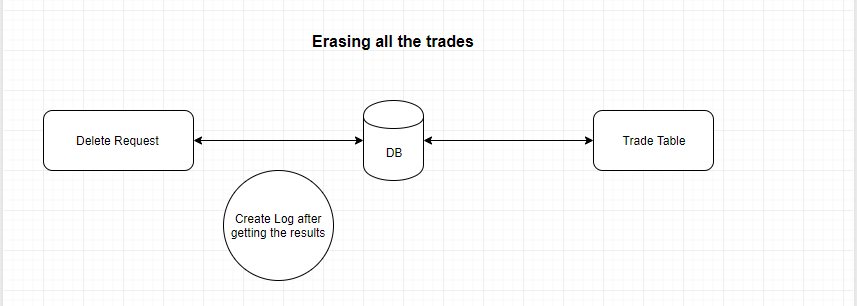
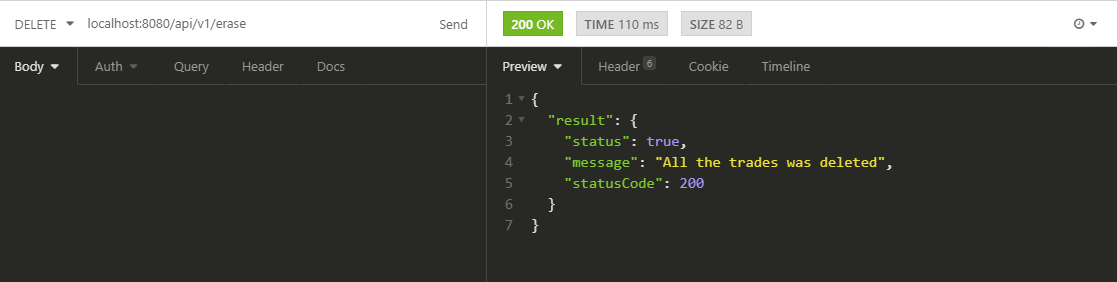
**Erasing all the trades**

**API End point:** localhost:8080/api/v1/erase

**Method**: DELETE

**Description:** Delete all trades using delete request

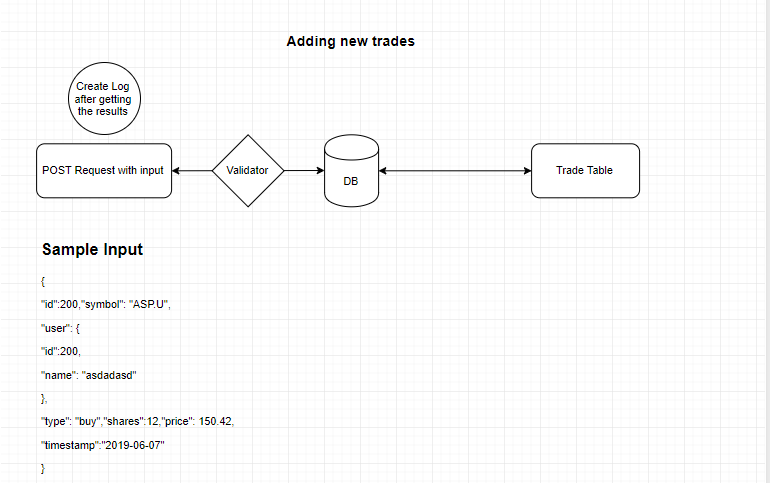


**Adding new trades**

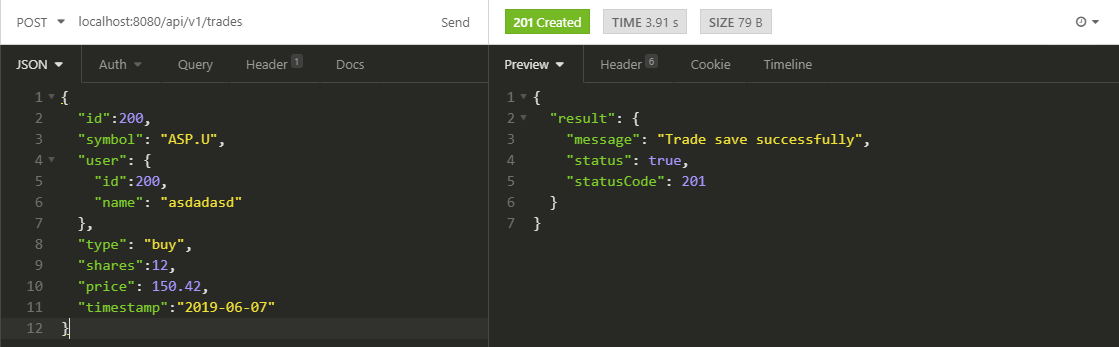
**API End point:** localhost:8080/api/v1/trades

**Method**: POST

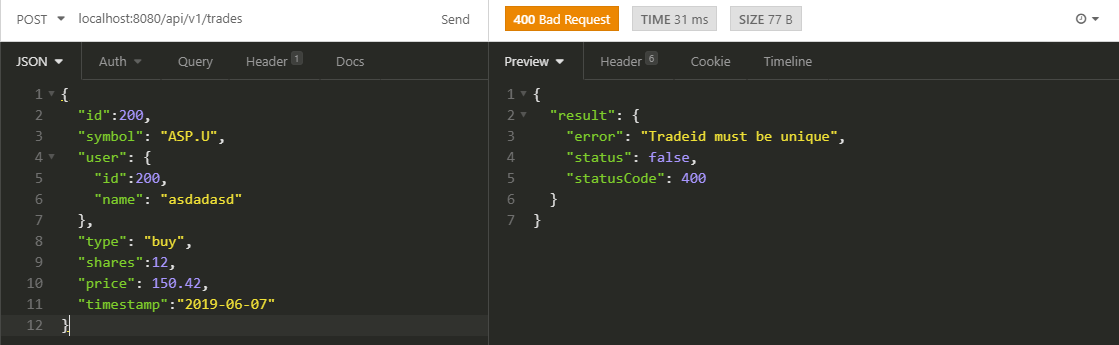
**Description:** Adding new trades: The service should be able to add a new trade by the POST request The event JSON is sent in the request body. If a trade with the same id already exists then the HTTP response code should be 400, otherwise, the response code should be 201.



**Success Output**



Bad Request

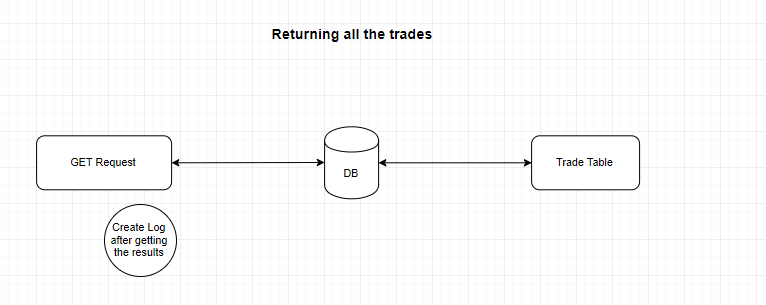


**Returning all the trades**

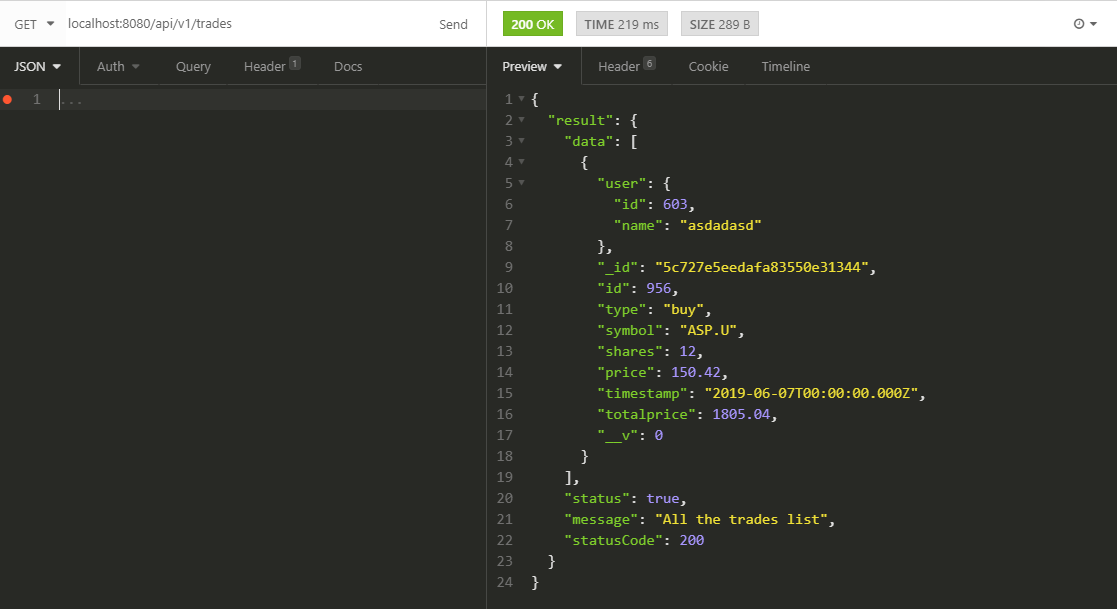
**API End point:** localhost:8080/api/v1/trades

**Method**: GET

**Description:** The service should be able to return the JSON array of all the trades through a GET request at /trades. The HTTP response code should be 200. The JSON array should be sorted in ascending order by trade ID.



Success Output

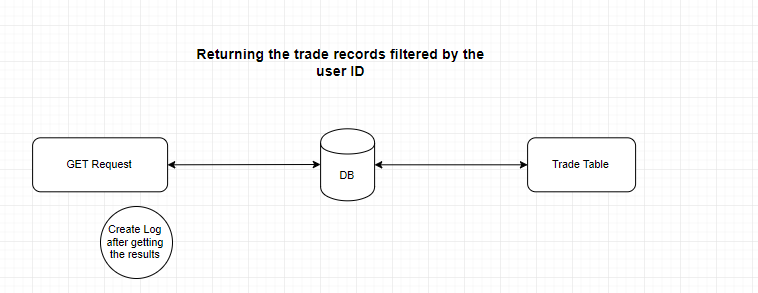


**Returning the trade records ﬁltered by the user ID**

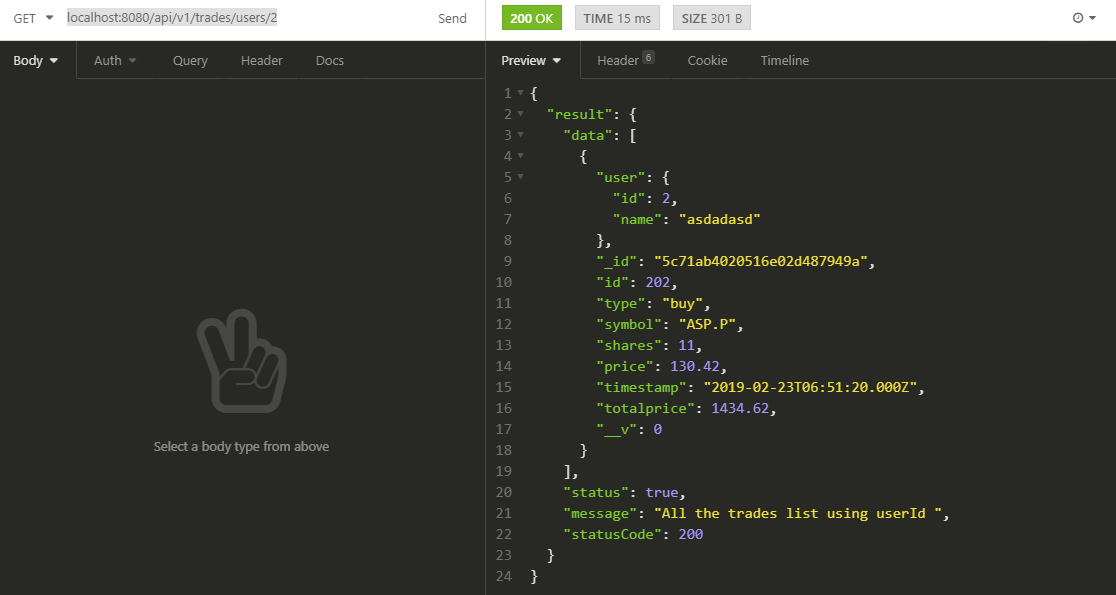
**API End point:** localhost:8080/api/v1/trades/users/2

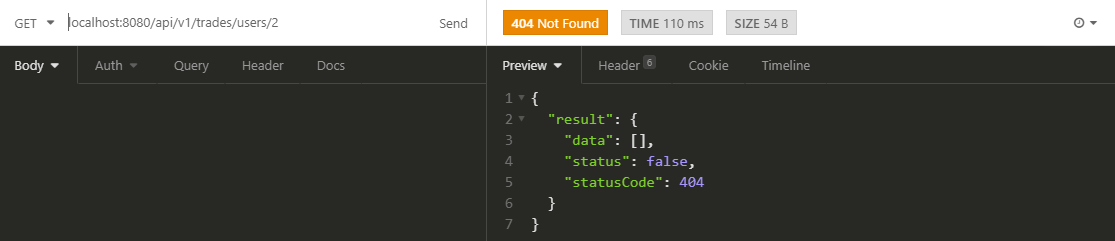
**Method**: GET

**Description:** The service should be able to return the JSON array of all the trades which are performed by the user ID through a GET request at /trades/users/{userID}. If the requested user does not exist then the HTTP response code should be 404, otherwise, the response code should be 200. The JSON array should be sorted in ascending order by trade ID.



Success output



Not Found output

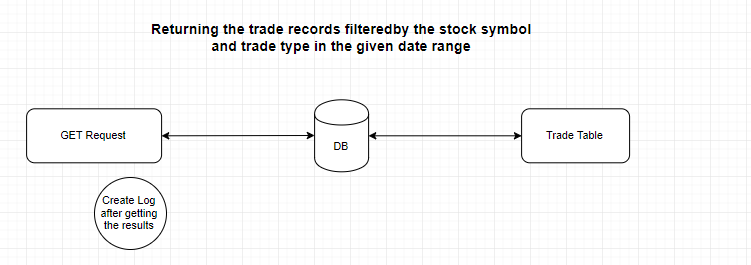
**Returning the trade records filtered by the stock symbol and trade type in the given date range**

**API End point:** localhost:8080/api/v1/stocks/ASP.U/trades?type=buy&start=2016-10-10&end=2016-10-10

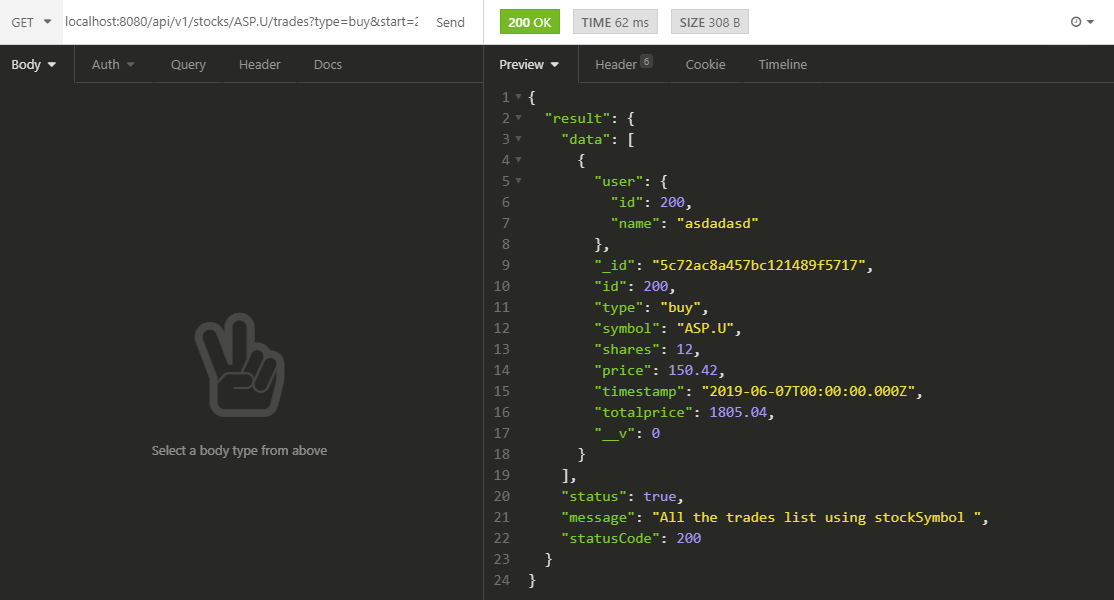
**Method**: GET

**Description:** The service should be able to return the JSON array of all the trades which are associated with the stock symbol and the given trade type, i.e., either buy or sell, in the given date range speciﬁed by start date and end date inclusive, by the GET request at /stocks/{stockSymbol}/trades?type={tradeType}&start=

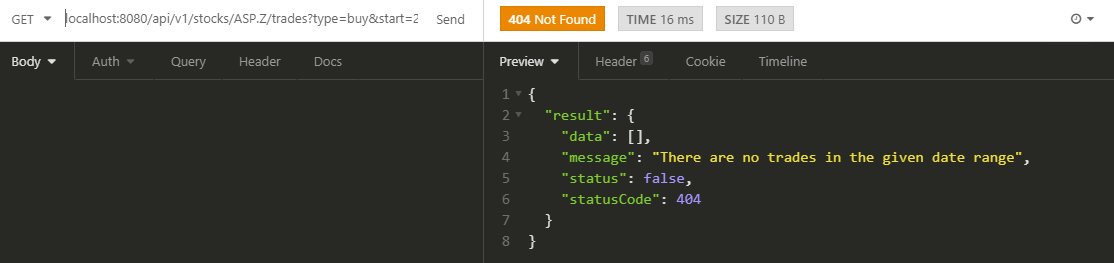
{startDate}&end={endDate}. If the requested stock symbol does not exist then the HTTP response code should be 404, otherwise, the response code should be 200irrespective of whether or not there are trades associated with the stock symbol in the given date range. The JSON array should be sorted in ascending order by trade ID.



Success Output



Not found output

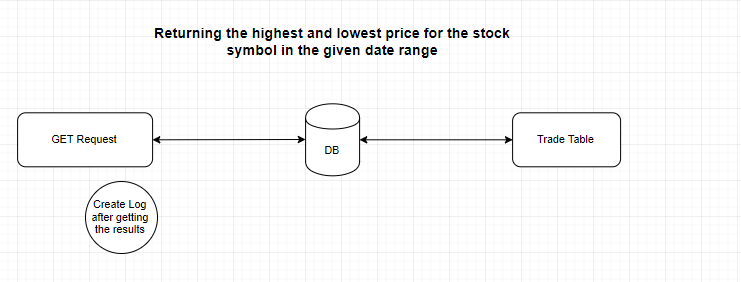


**Returning the highest and lowest price for the stock symbol in the given date range**

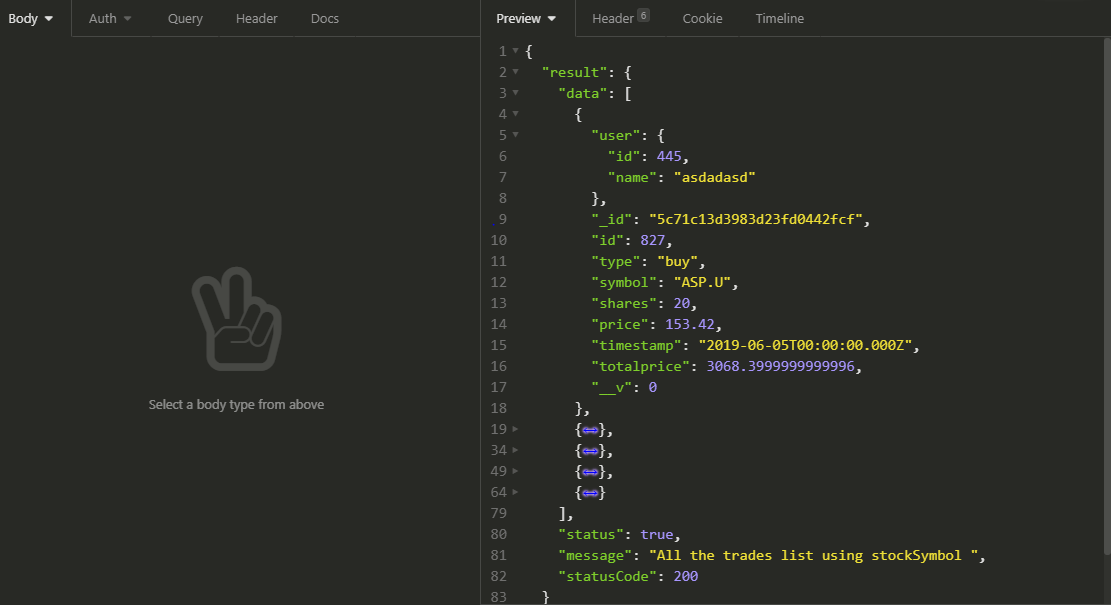
**API End point:** localhost:8080/api/v1/stocks/ASP.U/price?start=2019-06-01&end=2019-06-20

**Method**: GET

**Description:** The service should be able to return the JSON object describing the information of highest and lowest price in the given date range speciﬁed by start date and end date inclusive, through a GET request at /stocks/{stockSymbol}/price?start={startDate}&end= {endDate}. If the requested stock symbol does not exist then the HTTP response code should be 404, otherwise, the response code should be 200. The response JSON should consist of the following three ﬁelds: symbol: the symbol for the requested stock highest: the highest price for the requested stock symbol in the given date range lowest: the lowest price for the requested stock symbol in the given date range



Success output



Not Found Output

