To convert the given CSV data to JSON using Spring Boot, you can create a simple Spring Boot application and use a CSV parsing library, such as OpenCSV, to read the CSV data and then convert it to JSON. Here's a step-by-step guide on how to do this:

1. Create a Spring Boot project using Spring Initializer or your favorite IDE. Make sure you include the "Spring Web" and "Lombok" dependencies.

2. Create a Java class to represent the data structure. You can use the following class to match the CSV data:

```
import lombok.Data;

@Data
public class StockData {
    private String date;
    private double open;
    private double high;
    private double low;
    private double close;
    private double adjClose;
    private long volume;
}
```

3. Create a controller class that handles the CSV to JSON conversion. Here's an example controller:

```
import com.opencsv.CSVReader;
import com.opencsv.CSVReaderBuilder;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

@RestController
public class StockDataController {

    @GetMapping("/csv-to-json")
    public List<StockData> convertCsvToJson() throws IOException {
        String csvFilePath = "your_csv_file_path.csv"; // Replace with the actual path to your CSV file
        List<StockData> stockDataList = new ArrayList<>();

        try (CSVReader reader = new CSVReaderBuilder(new FileReader(csvFilePath)).withSkipLines(1).build()) {
            String[] line;
            while ((line = reader.readNext()) != null) {
                StockData stockData = new StockData();
                stockData.setDate(line[0]);
                stockData.setOpen(Double.parseDouble(line[1]));
                stockData.setHigh(Double.parseDouble(line[2]));
                stockData.setLow(Double.parseDouble(line[3]));
                stockData.setClose(Double.parseDouble(line[4]));
                stockData.setAdjClose(Double.parseDouble(line[5]));
                stockData.setVolume(Long.parseLong(line[6]));
                stockDataList.add(stockData);
            }
        }

        return stockDataList;
    }
}
```

4. Add the OpenCSV library to your project by including the following dependency in your `pom.xml` file:

```
<dependency>
    <groupId>com.opencsv</groupId>
    <artifactId>opencsv</artifactId>
    <version>5.5</version> <!-- Use the latest version available -->
</dependency>
```

5. Run your Spring Boot application, and you can access the converted JSON data by making a GET request to `/csv-to-json`. Make sure to replace `"your_csv_file_path.csv"` with the actual path to your CSV file.

When you access the `/csv-to-json` endpoint, the CSV data will be read and converted to a JSON response in the format of a list of StockData objects.