**Types Of ItemReader:**

1. **FlatFile ItemReader:**

* Reading flat files in the Spring Batch framework is facilitated by the class called FlatFileItemReader, which provides basic functionality for reading and parsing flat files. The two most important required dependenices of FlatFileItemReader are Resouce and LineMapper.

| **Property** | **Type** | **Description** |
| --- | --- | --- |
| Comments | String[] | Specifies line prefixes that indicate comment rows. |
| Encoding | String | Specifies what text encoding to use. The default value is UTF-8. |
| **lineMapper** | **LineMapper** | **Converts a String to an Object representing the item.** |
| **linesToSkip** | **Int** | **Number of lines to ignore at the top of the file.** |
| recordSeparatorPolicy | RecordSeparatorPolicy | Used to determine where the line endings are and do things like continue over a line ending if inside a quoted string. |
| **Resource** | **Resource** | **The resource from which to read.** |
| skippedLinesCallback | LineCallbackHandler | Interface that passes the raw line content of the lines in the file to be skipped. If linesToSkip is set to 2, then this interface is called twice. |
| Strict | Boolean | In strict mode, the reader throws an exception on ExecutionContext if the input resource does not exist. Otherwise, it logs the problem and continues. |

FlatFileItemReader

Source Location of CSV File

Line Mapper

Bean Mapper

Line Tokenizer

* The two most important required dependenices of FlatFileItemReader are Resouce and LineMapper.

**LineMapper:** Interface for mapping lines (strings) to domain objects typically used to map lines read from a file to domain objects on a per line basis.

public interface LineMapper<T> {  
 T mapLine(String var1, int var2) throws Exception;  
}

**Line Tokenizer:** Interface that is used by framework to split string obtained typically from a file into tokens.

**Bean Mapper:** to map tokens into fields of entity or pojo object.

@Bean  
public FlatFileItemReader<Customer> reader() {  
 FlatFileItemReader<Customer> itemReader = new FlatFileItemReader<>();  
 itemReader.setResource(new ClassPathResource("customers.csv"));  
 itemReader.setName("csvReader");  
 itemReader.setLinesToSkip(1);  
 itemReader.setLineMapper(lineMapper());  
 return itemReader;  
}  
  
private LineMapper<Customer> lineMapper() {  
 DefaultLineMapper<Customer> lineMapper = new DefaultLineMapper<>();  
  
 DelimitedLineTokenizer lineTokenizer = new DelimitedLineTokenizer();  
 lineTokenizer.setDelimiter(",");  
 lineTokenizer.setStrict(false);  
 lineTokenizer.setNames("id", "firstName", "lastName", "email", "gender", "contactNo", "country", "dob");  
  
 BeanWrapperFieldSetMapper<Customer> fieldSetMapper = new BeanWrapperFieldSetMapper<>();  
 fieldSetMapper.setTargetType(Customer.class);  
  
 lineMapper.setLineTokenizer(lineTokenizer);  
 lineMapper.setFieldSetMapper(fieldSetMapper);  
 return lineMapper;  
  
}

1. JsonItemReader:
2. StaxEventItemReader:
3. JdbcCursorItemReader:
4. ItemReaderAdapter:

1)