JavaReader

The JavaReader is an abstract class that allows you to create a custom reader in Java.

Create a custom reader

To create a custom reader:

- Extend the JavaReader class
- Override the iterator() function

Custom reader example

The implementation of the KeysOnlyReader class shows how to create a custom reader with JavaReader:

```java
import java.util.Iterator;
import gears.GearsBuilder;

/**
 * A reader that only reads key names from the key space
 * *
 * /
 public class KeysOnlyReader extends JavaReader<String> {

 /**
 * *
 * /
 private static final long serialVersionUID = 1L;

 private String scanSize;
 private String pattern;

 /**
 * Create a new KeysOnlyReader reader
 * @param scanSize - the size to use with the scan command
 * @param pattern - the pattern of the keys to read
 * /
 public KeysOnlyReader(int scanSize, String pattern) {
 this.scanSize = Integer.toString(scanSize);
 this.pattern = pattern;
 }

 /**
 * Create a new KeysOnlyReader reader with default pattern (*) and default
 * scan size (10000)
 * /
 public KeysOnlyReader() {
 this(10000, "*");
 }
```
@Override
public Iterator<String> iterator() {
    String cursor = "0";
    int currIndex = 0;
    Object[] keys = null;
    boolean isDone = false;
    String nextKey = null;

    private String innerNext() {
        while (!isDone) {
            if (keys == null) {
                Object[] res = (Object[]) GearsBuilder.execute("scan",
                        cursor == null ? "0" : cursor,
                        "MATCH", pattern, "COUNT", scanSize);
                keys = (Object[]) res[1];
                cursor = (String) res[0];
                currIndex = 0;
            }
            if (currIndex < keys.length) {
                return (String) keys[currIndex++];
            }
            keys = null;
            if (cursor.charAt(0) == '0') {
                isDone = true;
            }
            return null;
        }
    }

    @Override
    public boolean hasNext() {
        if (nextKey == null) {
            nextKey = innerNext();
        }
        return !isDone;
    }

    @Override
    public String next() {
        String temp = nextKey != null ? nextKey : innerNext();
        nextKey = innerNext();
        return temp;
    }
};