CommandReader

The CommandReader allows you to run RedisGears functions on command when you:

1. Pass the CommandReader to the `GearsBuilder.CreateGearsBuilder()` function in your Java code.
2. Call the `register()` function.
3. Run `RG.JEXECUTE` to register your code.
4. Use `RG.TRIGGER` to run your code on command:

   `RG.TRIGGER <Trigger name> [arg1 arg2 ...]`

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>trigger</td>
<td>string</td>
<td>The command name that triggers the registered RedisGears functions to run</td>
</tr>
</tbody>
</table>

Output records

Outputs a record with the command trigger’s name and arguments.

Example

The following example shows how to create a custom command to update an item’s stock. It also adds a timestamp to track when the last restock occurred.

Add a hash to the database that represents an inventory item:

```
redis> HSET inventory:headphones:1 color "blue" stock 5 price 30.00
(integer) 3
```

Example code:
Create the reader that will pass data to the pipe
CommandReader reader = new CommandReader();
Set the name of the custom command
reader.setTrigger("Restock");
Create the data pipe builder
GearsBuilder.CreateGearsBuilder(reader).map(r -> {
  Parse the command arguments to get the key name and new stock value
  String itemKey = new String((byte[]) r[1], StandardCharsets.UTF_8);
  String newStock = new String((byte[]) r[2], StandardCharsets.UTF_8);

  // Update the item's stock and add a timestamp
  GearsBuilder.execute("HSET", itemKey, "stock", newStock,
                      "last_restocked", Long.toString(System.currentTimeMillis()));

  return "OK restocked " + itemKey;
}).register();