Update the Contents of FlowFile by using UpdateRecord processor

#### Article

**Short Description:**

This Tutorial describes how to add fields,removing not required fields and change values of fields in flowfile.

**Article**

**Introduction**

Using UpdateRecord processor we are can update the contents of flowfile.

**Overview of article:**Below sections describes the changes that are going to happen to the input flowfile content vs output flowfile contents.

**Input:**  
input Json record as follows and having 6 fields/elements in it.

1. [ {
2. "id" : 1,
3. "name" : "foo",
4. "age" : 20,
5. "state" : "FLORIDA",
6. "ts\_milli" : 1525526792098,
7. "ts" : "2018-05-03 10:10:10.123"
8. }]

**Expected Output:**  
I'm writing output in Avro Format but for viewing purpose i have converted the output flowfile to json.

1. [ {
2. "id" : "1",
3. "rename\_id" : "1", //newly added field based on id
4. "state" : "florida", //changed the value of field
5. "rename\_state" : "FLORIDA", //newly added field based on state
6. "unique\_id" : "1-416425265990923-FLORIDA-aab46988-6a27-4008-9a4d-65655abe9c3c",//id-filename-state-UUID
7. "ts\_milli" : 1525526792098,
8. "date" : "2018-05-05", //newly added field formatted as date based on ts\_milli
9. "ts" : "2018-05-03 10:10:10.123",
10. "ts\_tz" : "2018-05-03T10:10:10Z",//newly added field formatted as date based on ts
11. "current\_ts" : "2018-05-05 14:23:15", //current timestamp value
12. "updated\_by" : "NiFi", //newly added field based on user attribute value
13. "gmt\_time" : "2018-05-05 18:23:18.611" //newly added field gmt time.
14. }]

We are missing **name,age** fields because those fields are not required in output contents.

**UpdateRecord Processor Params:**

**Record Reader** Specifies the Controller Service to use for reading incoming data

**Record Writer**Specifies the Controller Service to use for writing out the records.

**Replacement Value Strategy**Specifies how to interpret the configured replacement values

**1.Literal Value**

The **value entered** for a Property (after Expression Language has been evaluated) is **the desired value**to update the Record Fields with. Expression Language may reference variables 'field.name', 'field.type', and 'field.value' to access information about the field and the value of the field being evaluated.

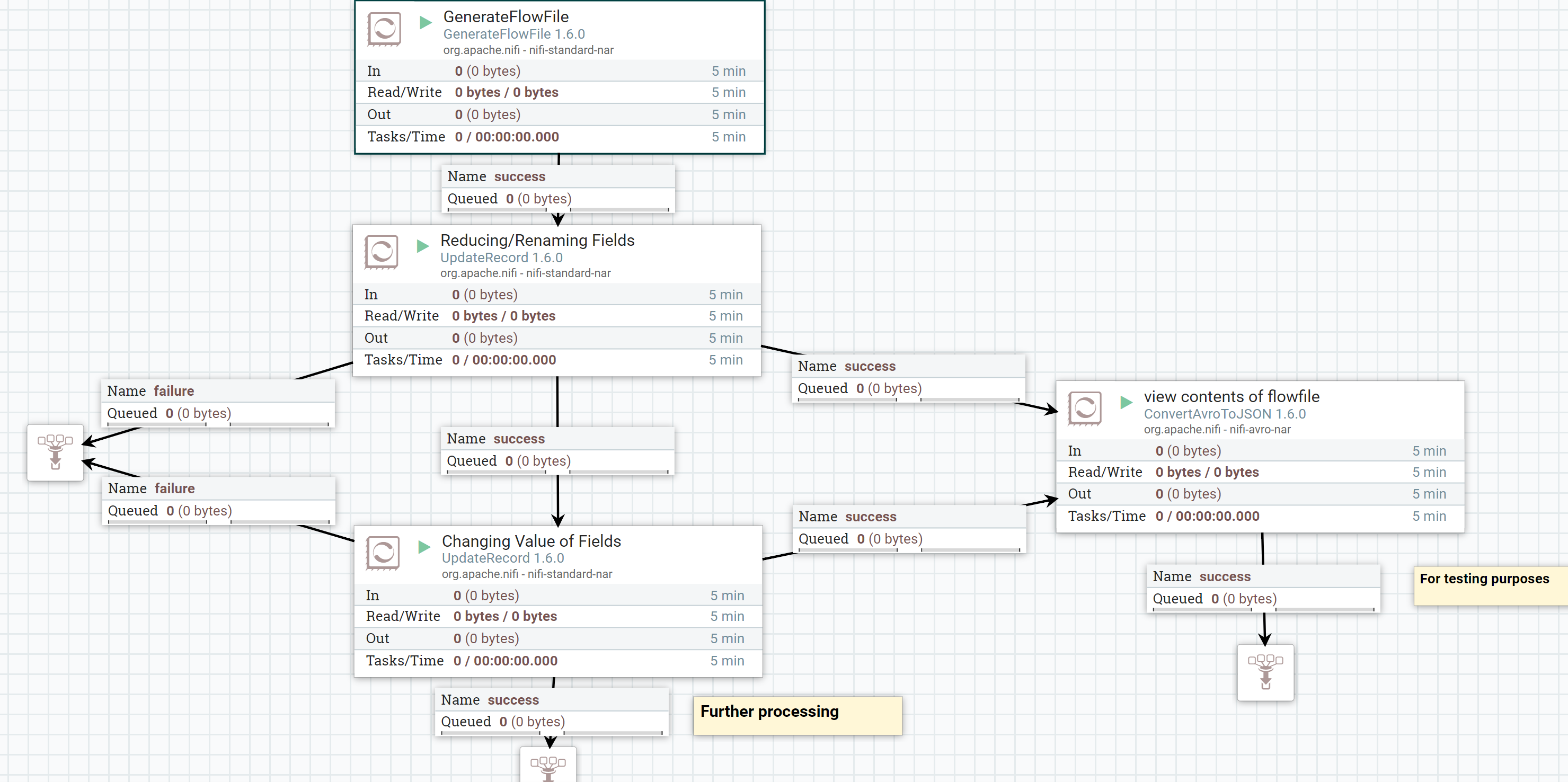
**2.Record Path Value**

The value entered for a Property (after Expression Language has been evaluated) is **not the literal value** to **use** but rather is a **Record Path** that should be evaluated against the Record, and the result of the RecordPath will be used to update the Record. if this option is selected, and the Record Path results in **multiple values for a given Record**, the input **FlowFile** will be **routed** to the '**failure**' Relationship.

**\*\*Note\*\***

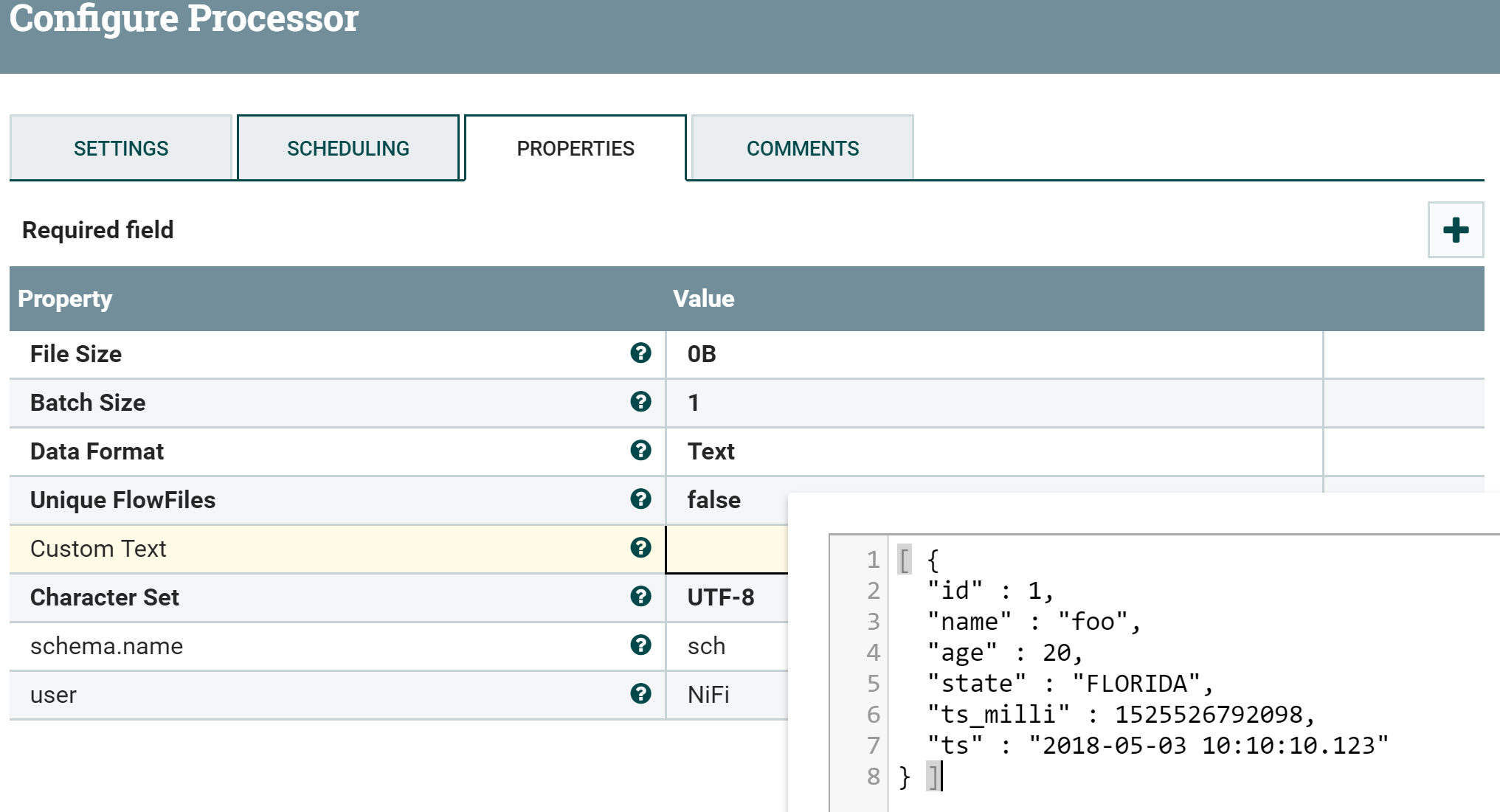
This Processor requires that at least one user-defined Property be added.

Please refer to [this](https://nifi.apache.org/docs/nifi-docs/components/org.apache.nifi/nifi-standard-nar/1.6.0/org.apache.nifi.processors.standard.UpdateRecord/index.html) link for more details regarding UpdateRecord Processor.  
**Flow:**



**GenerateFlowFile Configs:**

Configure the processor as shown below



Add new properties as

**schema.name**

1. sch

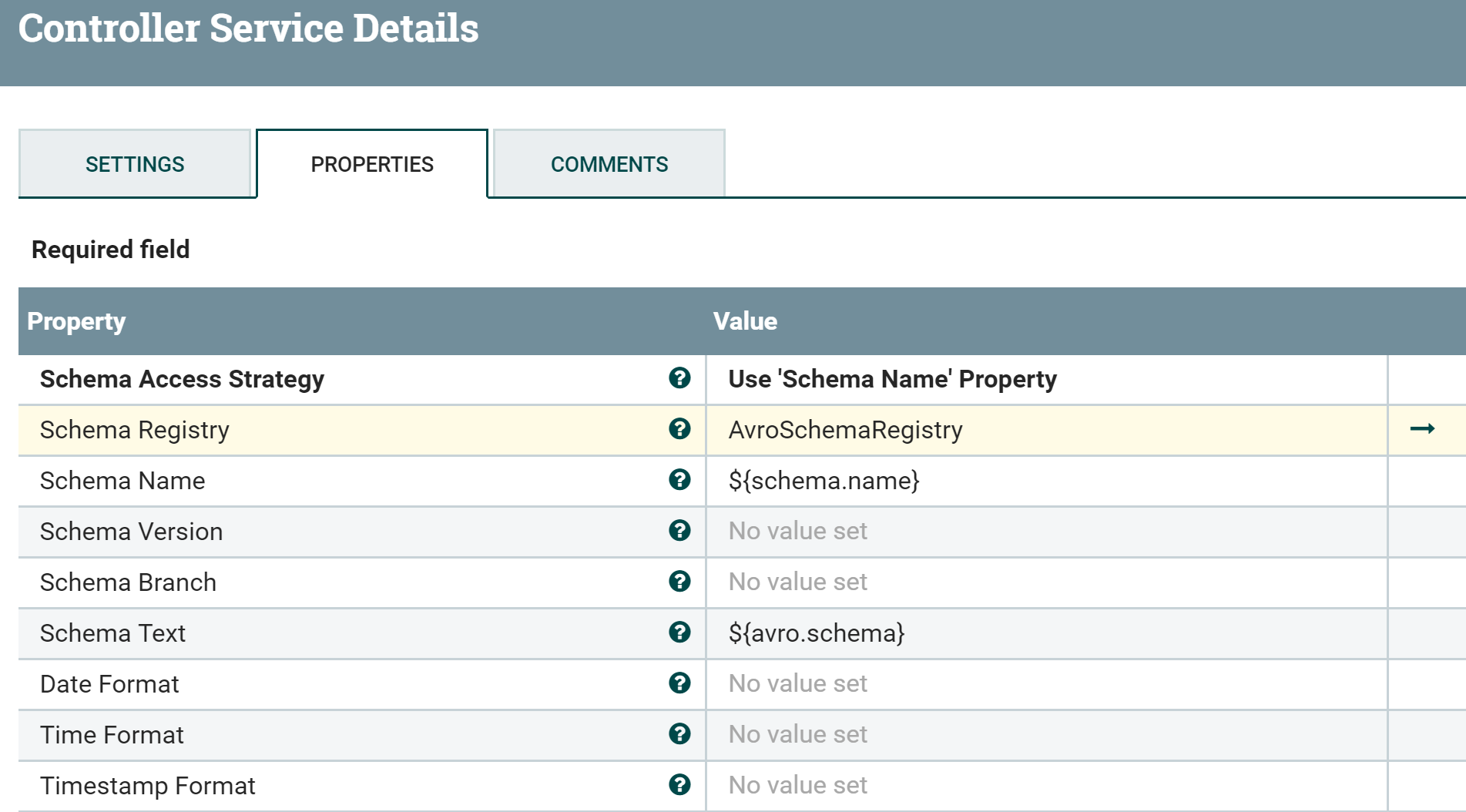
**user**

1. NiFi

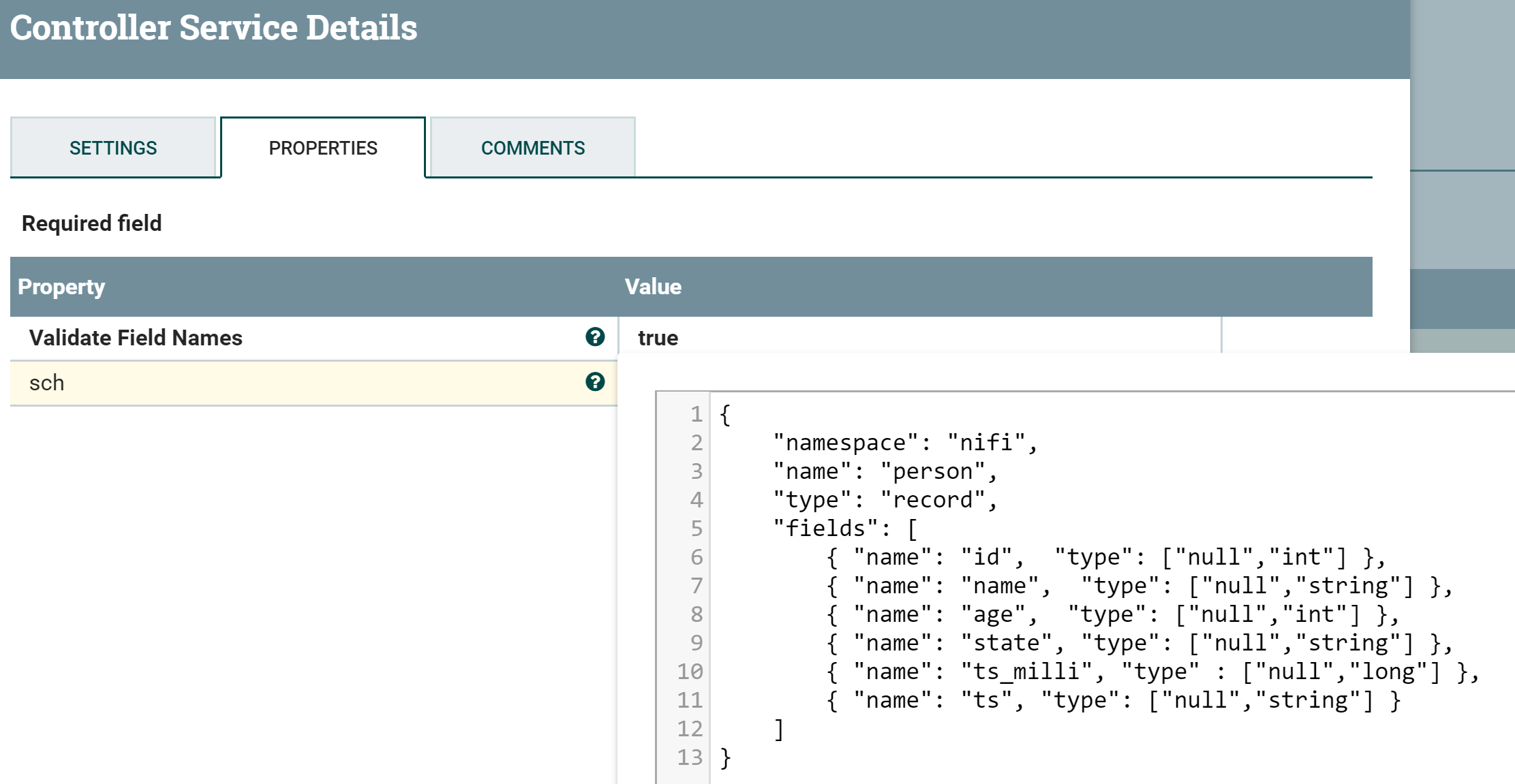
**UpdateRecord Strategy as Record Path Configs:-**  
As **Replacement Value Strategy**

1. Record Path Value

**Record Reader:**JsonTreeReader  
Configure the controller service as shown below



**AvroSchemaRegistry Configs:**



Add new property

**sch**

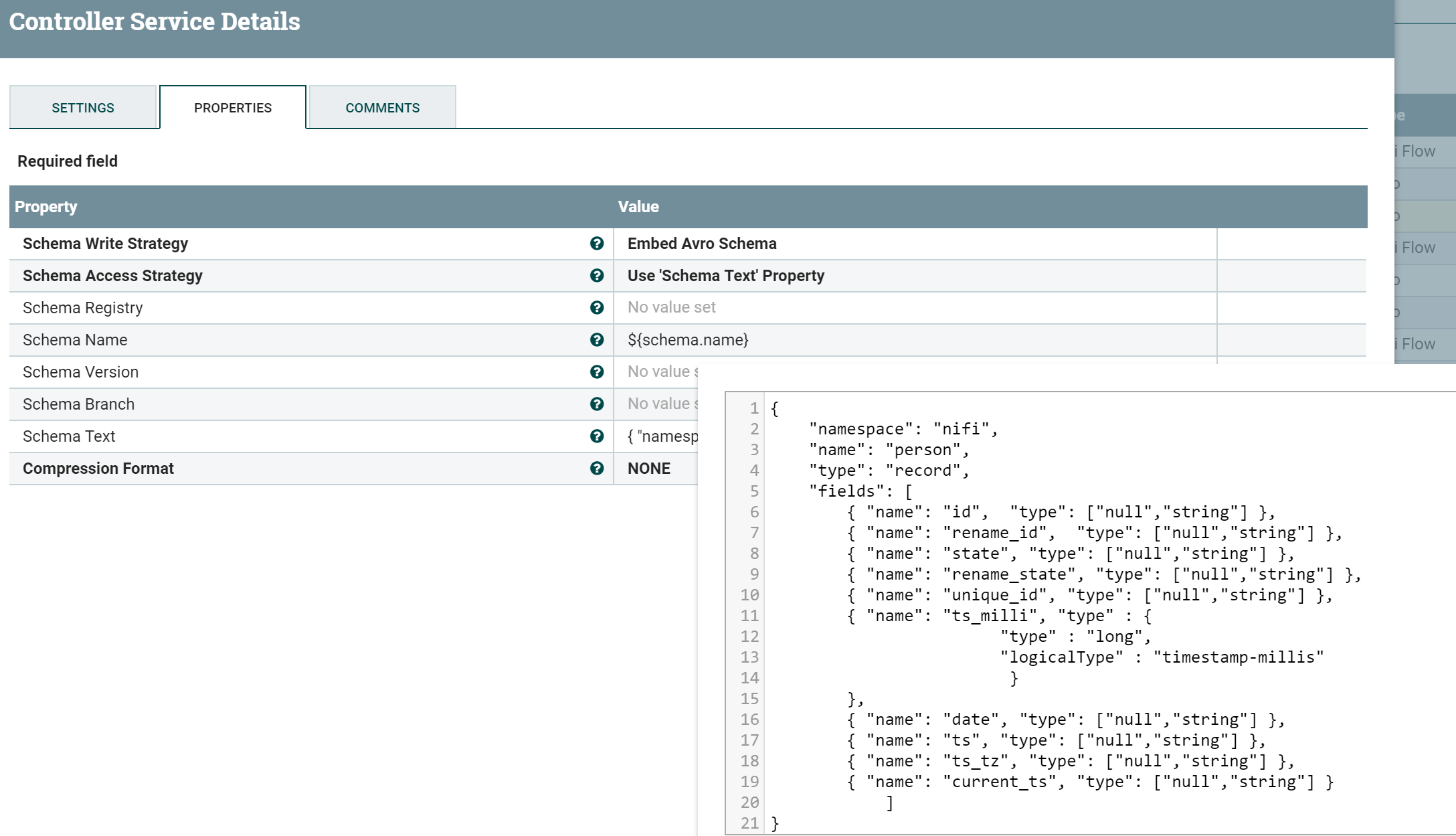
1. {
2. "namespace": "nifi",
3. "name": "person",
4. "type": "record",
5. "fields": [
6. { "name": "id", "type": ["null","int"] },
7. { "name": "name", "type": ["null","string"] },
8. { "name": "age", "type": ["null","int"] },
9. { "name": "state", "type": ["null","string"] },
10. { "name": "ts\_milli", "type" : ["null","long"] },
11. { "name": "ts", "type": ["null","string"] }
12. ]
13. }

By using above schema we are reading the incoming json message (or) you can even specify only the required fields in this schema then processor will read only those fields and all the names are case sensitive.

**Record Writer:**

I'm using AvroRecordSetWriter i.e we are reading input in json format and writing as avro format.

Configure the Controller service as shown below.

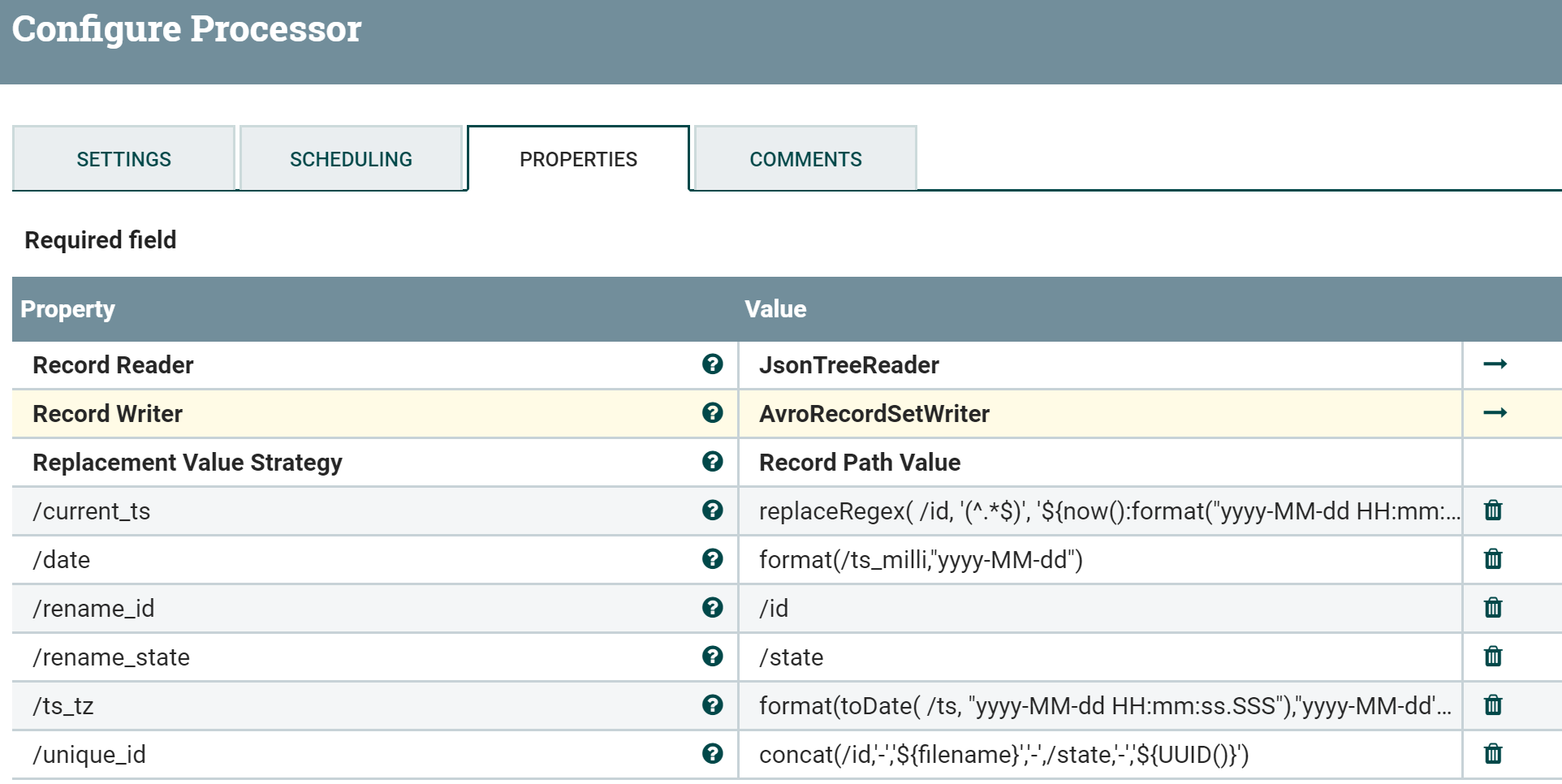


**Schema Text:**

1. {
2. "namespace": "nifi",
3. "name": "person",
4. "type": "record",
5. "fields": [
6. { "name": "id", "type": ["null","string"] },
7. { "name": "rename\_id", "type": ["null","string"] },
8. { "name": "state", "type": ["null","string"] },
9. { "name": "rename\_state", "type": ["null","string"] },
10. { "name": "unique\_id", "type": ["null","string"] },
11. { "name": "ts\_milli", "type" : {
12. "type" : "long",
13. "logicalType" : "timestamp-millis"
14. }
15. },
16. { "name": "date", "type": ["null","string"] },
17. { "name": "ts", "type": ["null","string"] },
18. { "name": "ts\_tz", "type": ["null","string"] },
19. { "name": "current\_ts", "type": ["null","string"] },
20. { "name": "gmt\_time", "type": ["null","string"] },
21. { "name": "updated\_by", "type": ["null","string"] }
22. ]
23. }

As you can see there are bunch of new fields are added in the above avro schema.

Now we are going to**add user-defined** properties in update record processor



**1./current\_ts**

1. replaceRegex( /id, '(^.\*$)', '${now():format("yyyy-MM-dd HH:mm:ss")}')

To get current\_ts field value we are using replaceRegex function of **RecordPathDomainSpecific**language and replacing the id value with current timestamp with the specific format

**2./date**

1. format(/ts\_milli,"yyyy-MM-dd")

Based on **ts\_milli**record path value and changing the format to get **only the date**from epoch time in milliseconds  
 **3./rename\_id**

1. /id

Assigning the id value to new **rename\_id** field we are going to have same id value for both fields  
  
4.**/rename\_state**

1. /state

Assigning the state value to new **rename\_state** field we are going to have same state value for both fields because in our output avro schema includes both fields.

**5./ts\_tz**

1. format(toDate( /ts, "yyyy-MM-dd HH:mm:ss.SSS"),"yyyy-MM-dd'T'HH:mm:ss'Z'")

Changing the format of the **ts field** value and **adding T and Z** and assigning the new value to **ts\_tz field.**  
**6./unique\_id**

1. concat(/id,'-','${filename}','-',/state,'-','${UUID()}')

In this field value we are **concatenating id,filename,state,UUID() values** with - to make the value unique and assigning the value to**unique\_id field**.

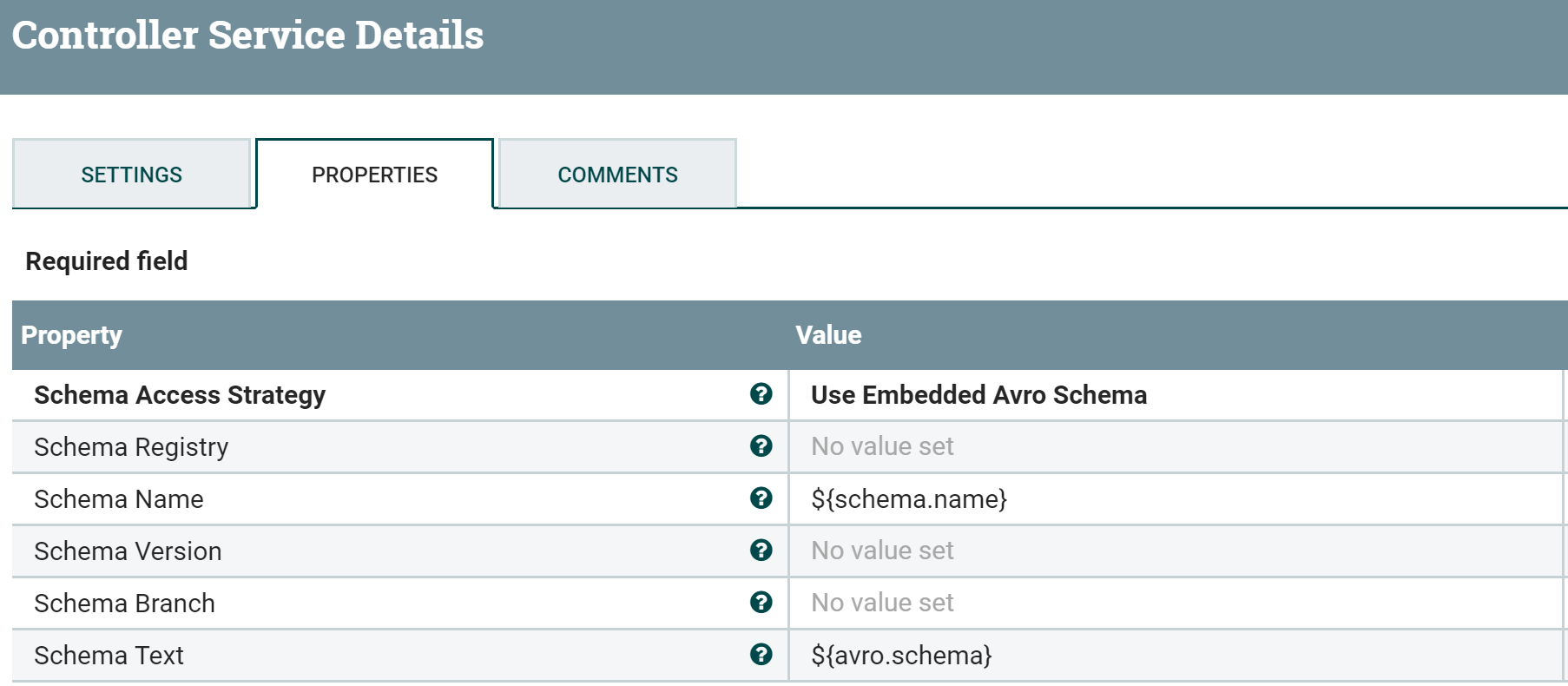
Our output flowfile will be in avro format to view the flowfile content we are using ConvertAvroToJSON processor to view each processor output.

**Output:**

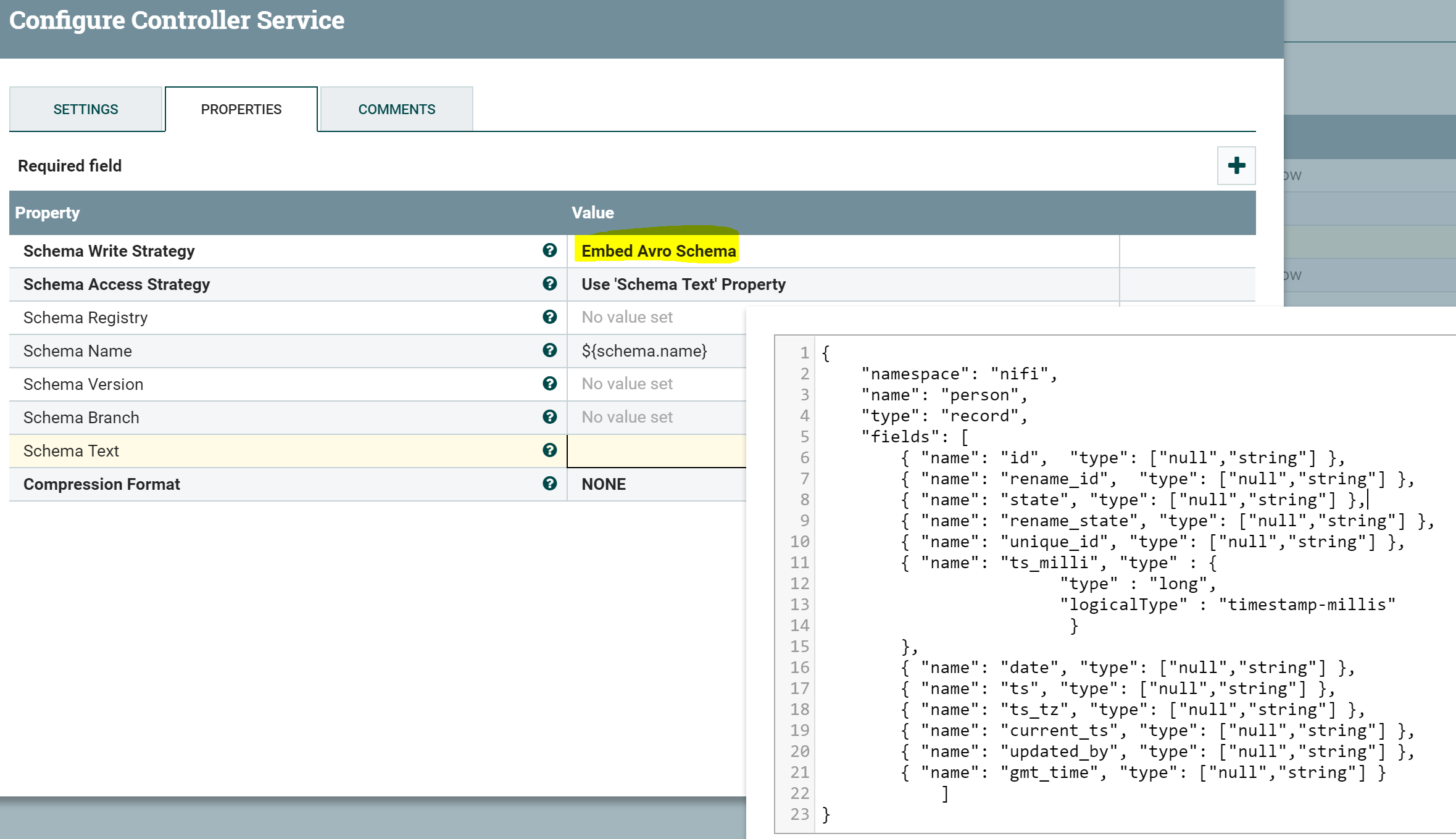
1. [ {
2. "id" : "1",
3. "rename\_id" : "1",
4. "state" : "FLORIDA",
5. "rename\_state" : "FLORIDA",
6. "unique\_id" : "1-423955681497409-FLORIDA-3d71d47d-da7a-44bb-b506-9f7a027933e2",
7. "ts\_milli" : 1525526792098,
8. "date" : "2018-05-05",
9. "ts" : "2018-05-03 10:10:10.123",
10. "ts\_tz" : "2018-05-03T10:10:10Z",
11. "current\_ts" : "2018-05-05 16:28:39",
12. "gmt\_time" : null, //this field is going to populated in next UpdateRecord processor
13. "updated\_by" : null //this field is going to populated in next UpdateRecord processor
14. }]

**Update Record Strategy as Literal value configs:-**

Record Reader as Avro reader as we are having**feeding avro file with embedded schema** so we are keeping **Schema Access Strategy as Embedded Avro Schema**



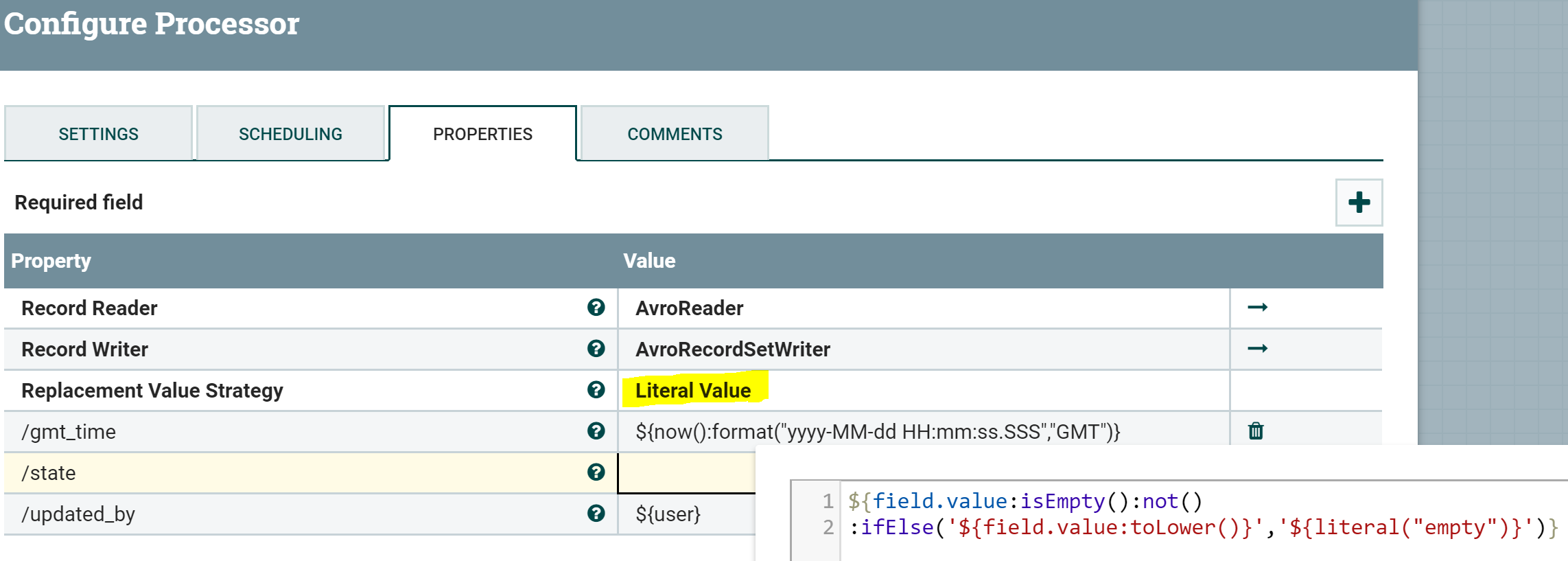
**AvroSet writer configs:**



**Schema Text:**

1. {
2. "namespace": "nifi",
3. "name": "person",
4. "type": "record",
5. "fields": [
6. { "name": "id", "type": ["null","string"] },
7. { "name": "rename\_id", "type": ["null","string"] },
8. { "name": "state", "type": ["null","string"] },
9. { "name": "rename\_state", "type": ["null","string"] },
10. { "name": "unique\_id", "type": ["null","string"] },
11. { "name": "ts\_milli", "type" : {
12. "type" : "long",
13. "logicalType" : "timestamp-millis"
14. }
15. },
16. { "name": "date", "type": ["null","string"] },
17. { "name": "ts", "type": ["null","string"] },
18. { "name": "ts\_tz", "type": ["null","string"] },
19. { "name": "current\_ts", "type": ["null","string"] },
20. { "name": "gmt\_time", "type": ["null","string"] },
21. { "name": "updated\_by", "type": ["null","string"] }
22. ]
23. }

Add **new user-defined properties** as



**1./gmt\_time**

1. ${now():format("yyyy-MM-dd HH:mm:ss.SSS","GMT")}

We are adding gmt time by using NiFi expression language  
 **2./state**

1. ${field.value:isEmpty():not():ifElse('${field.value:toLower()}','${literal("empty")}')}

in this value we are checking state field value and if it's not empty then changing the value to **lower case,**if the value **is null or empty string or blank** then keeping the value as literal **"empty"**  
  
**3./updated\_by**

1. ${user}

We have added **user attribute in GenerateFlowfile** processor and now we assigning the user attribute value to **update\_by**field value.

**Final Output:**

1. [ {
2. "id" : "1",
3. "rename\_id" : "1",
4. "state" : "florida",
5. "rename\_state" : "FLORIDA",
6. "unique\_id" : "1-425389131155321-FLORIDA-c81faa7a-4f2e-414e-bff5-d8722485921c",
7. "ts\_milli" : 1525526792098,
8. "date" : "2018-05-05",
9. "ts" : "2018-05-03 10:10:10.123",
10. "ts\_tz" : "2018-05-03T10:10:10Z",
11. "current\_ts" : "2018-05-05 16:52:32",
12. "gmt\_time" : "2018-05-05 20:52:36.088",
13. "updated\_by" : "NiFi"
14. }]

As i have used output format as Avro but we can use same **avro schema** for other formats(json,csv..).  
Now we have **prepared each record** by a**dding new fields to it and removing unnecessary fields** from the record, then Use **success relation** for further processing.  
Please refer to [this](https://nifi.apache.org/docs/nifi-docs/html/record-path-guide.html) link for more details regarding **RecordPathDomainSpecific Language.**

Please refer [this](https://nifi.apache.org/docs/nifi-docs/html/expression-language-guide.html) link for **NiFi Expression Language**.

Reference flow.xml  
[update-contents-using-updaterecord.xml](https://community.hortonworks.com/storage/attachments/72627-update-contents-using-updaterecord.xml)