**TEST CASES FOR INTERMEDIATE BACKEND DEVELOPER**

1. **Project Brief and Context**

Hyperion is a Reinsurance SAAS cloud base Web application , it connects the Insurance company(CEDANTE) to the Reinsurance Company.

The Data flow starts by the insurance company who submits slips (a file with all insurance policy cases) to share the risk with the Reinsurance Company.

This Action will start calculation and control of each case in the slips (can be more than 100K business lines)

Workflow:

* Insurance Company upload the slips
* Insurance Company review the information of the slips
* Insurance Company once ready Approve the slips in application to send it to the reinsurance company
* The reinsurance company receive the data with all calculation and control made by hyperion System
* The reinsurance Make Validation and then confirmation of the slips after that will go to the billing process ,we will not mention the details yet at this stage.

The inventory comes at the end of the working year for an accounting purpose , in this accounting we have the REC risk module that is part of the inventory

REC risk : **The provision for current risks** is defined as the amount to be provisioned in addition unearned premiums to cover the risks to be assumed and intended to cover all

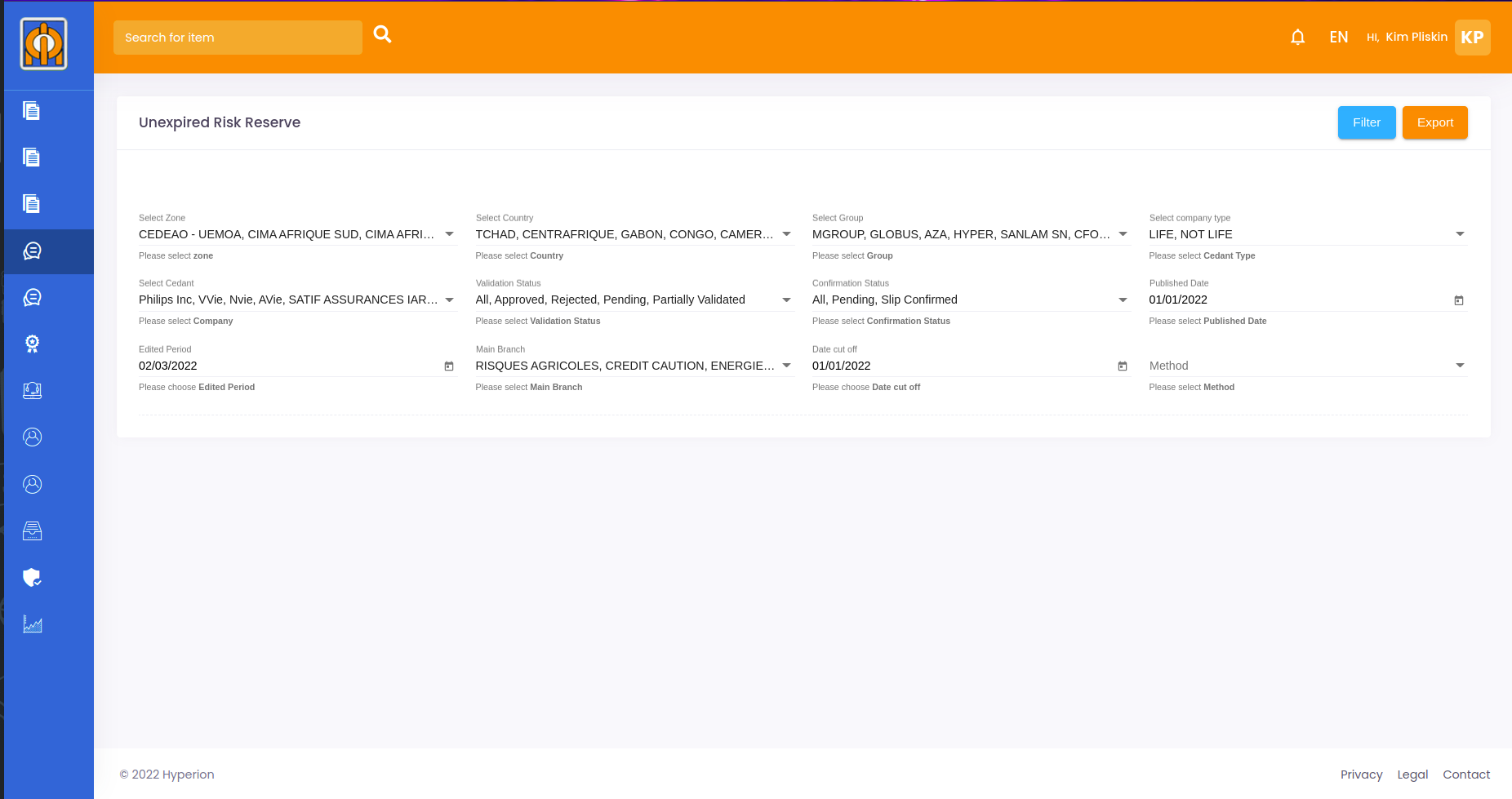
claims and all costs (including administration costs) related to the contracts

current insurance exceeding the amount of unearned premiums and related premiums due

to said contracts.

1. **TEST SCOPE**

Here your purpose will be to build the Api of the REC risk module :



This module will show a data table that aggregate the collection based on the filter (collection below in json format ) the last column of this data table will be calculated:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| References | Country | Cedants(insurance) | Validation status | Confirmation status | Publication date | Branche | Calculated REC |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

Reference : Coming from collection slips\_premium the raw is reference

Country : Coming from collection countries the row to use is name

Cedants : Coming from collection cedants the row to use is name

Validation status : Coming from collection slips\_premium the row is validation\_status

Confirmation status : Coming from collection slips\_premium the row is confirmation\_status

Publication date : Coming from collection slips\_premium the raw is publish\_date

Branche : Depends on user selected main branch coming from Branches collection row to use is name

Calculated REC :

* **In this test we gonna supposed all cedants use 36% methods(in production many others methods exist)**

Here you gonna aggregate data based on the filter selection with the cedant\_id , the slips\_id branche\_id to finish on the case\_not\_life\_premium collection, row to use is “**premium\_ht”**

So Calculated REC = Sum of the **premium\_ht** \* 36%

**This test Scenario will only include data for country “COTE D’IVOIRE” and “NOT LIFE” Company type**

The data covered over 3 companies :

ObjectID = 5de7d0ed329d8746112bef92 / name = Hyper non vie assurance

ObjectID = 5ebbfec8329d8763057a4014 / name = Kied ASSUR

ObjectID = 5ec3e07e329d8748f31aa8c2 / name = Paix assur

Collection Json files : <https://drive.google.com/file/d/1dMVTd8wQ2lOCt1RuK1d4e0EaKDyJlzRS/view?usp=sharing>

**Expected Result: REST API(Java Spring Boot GET) with filters parameters(Refer to the screenshot) and data table described above as a json response.**

**The developer must provide a Github link with clear documentation and Postman Collection.**