

Score Calculator

- Pranit Patil

Instructions for Deployment

Put files provided in folder ScoreCalcProperties in a folder and put the folder path in Applications Server's System Properties as-

-Dscorecalculator.resource.location="G:\tmp\ScoreCalcProperties"

scorecalculator.properties file will be loaded from this location as Spring Property Resource.

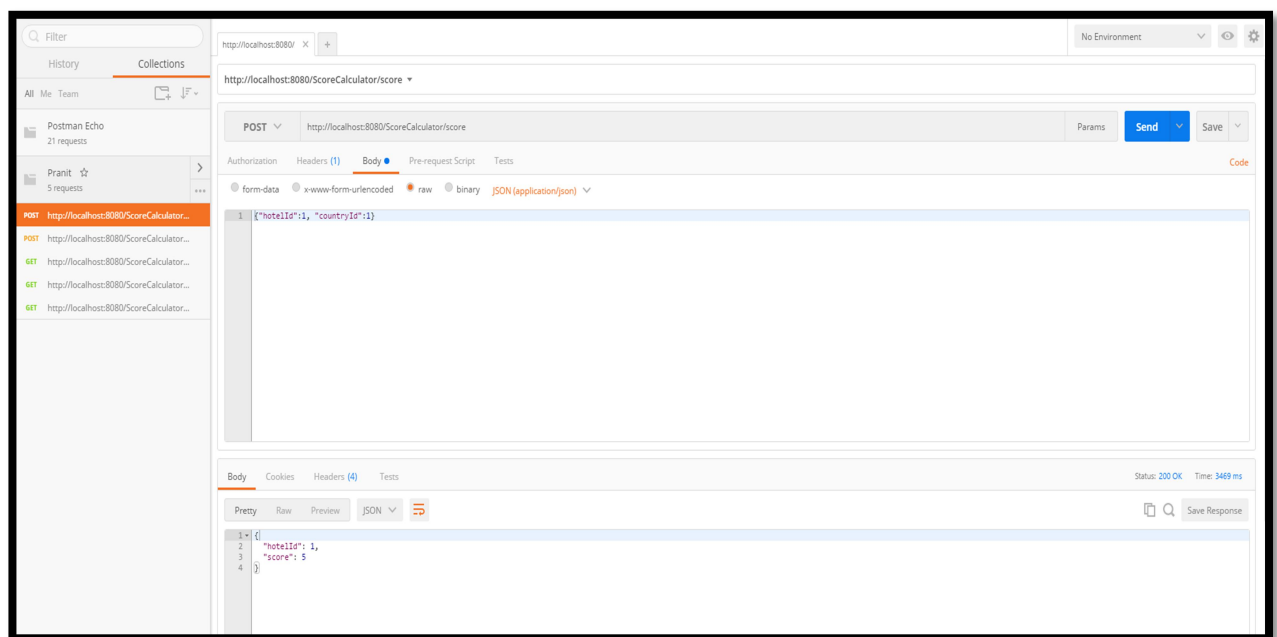
REST Services

1. Get Score

This Service is to get the score for Specific Hotel and Country

URL - <http://localhost:8080/ScoreCalculator/score>

Request – {"hotelId":1, "countryId":1}



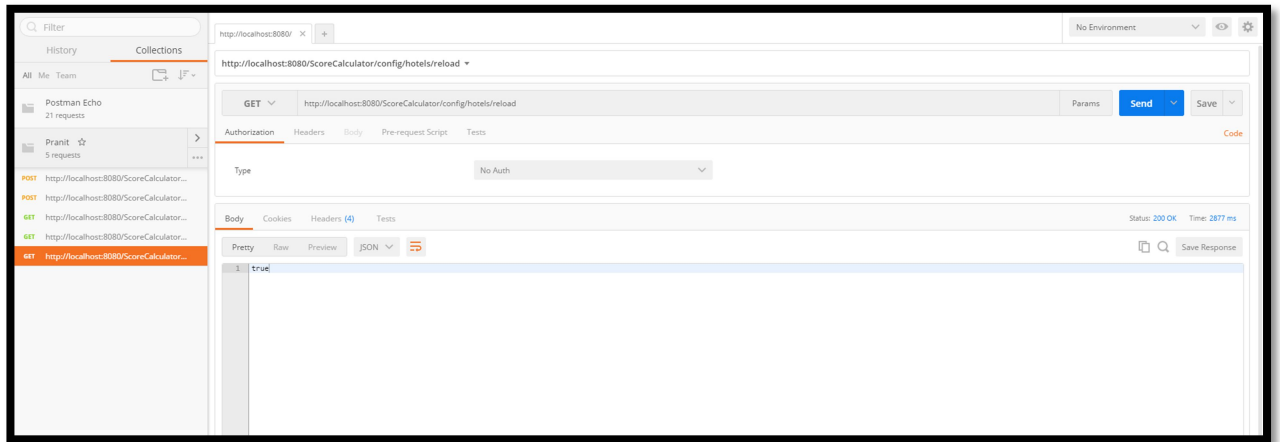
2. Reload Hotels

This Service reloads the hotels from a JSON file containing hotel information.

To add or remove the Hotels – Change the hotes.json file present at location specified in scorecalculator.properties as hotel.file.path

After changing the file following REST can be used to refresh the Hotel Information

URL - <http://localhost:8080/ScoreCalculator/config/hotels/reload>



3. Reload Countries

This Service reloads the countries from a JSON file containing country information.

To add or remove the Countries – Change the countries.json file present at location specified in scorecalculator.properties as country.file.path

After changing the file following REST can be used to refresh the Hotel Information

URL - <http://localhost:8080/ScoreCalculator/config/countries/reload>

NOTE : Both Hotel and Country Reloads are scheduled as well using Spring Scheduler.

Interval of the same can be configured from scorecalculator.properties file using following

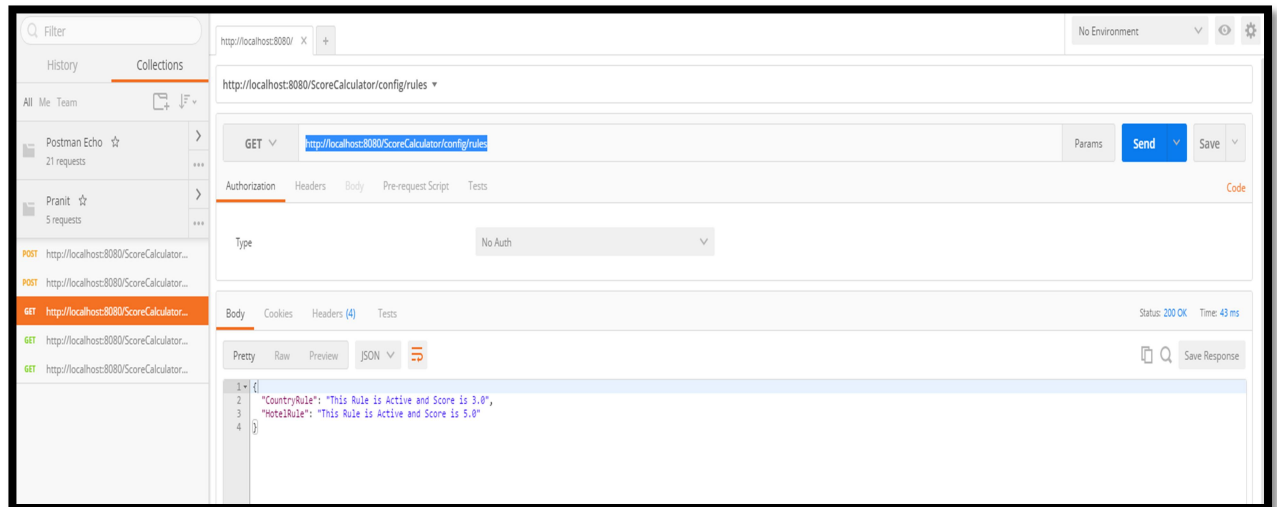
hotels.load.fixedDelay.milliseconds=18000000

countries.load.fixedDelay.milliseconds=18000000

4. View All Rules

This Service gets the List of all rules present in the System.

URL - <http://localhost:8080/ScoreCalculator/config/rules>



This gives names of all rules and Active/ Inactive Information which can be used to modify rules using modify rules service.

5. Modify Rules

This Service is used to modify rules.

Rules can Activated / Deactivate also Score of the rules can be modified using this.

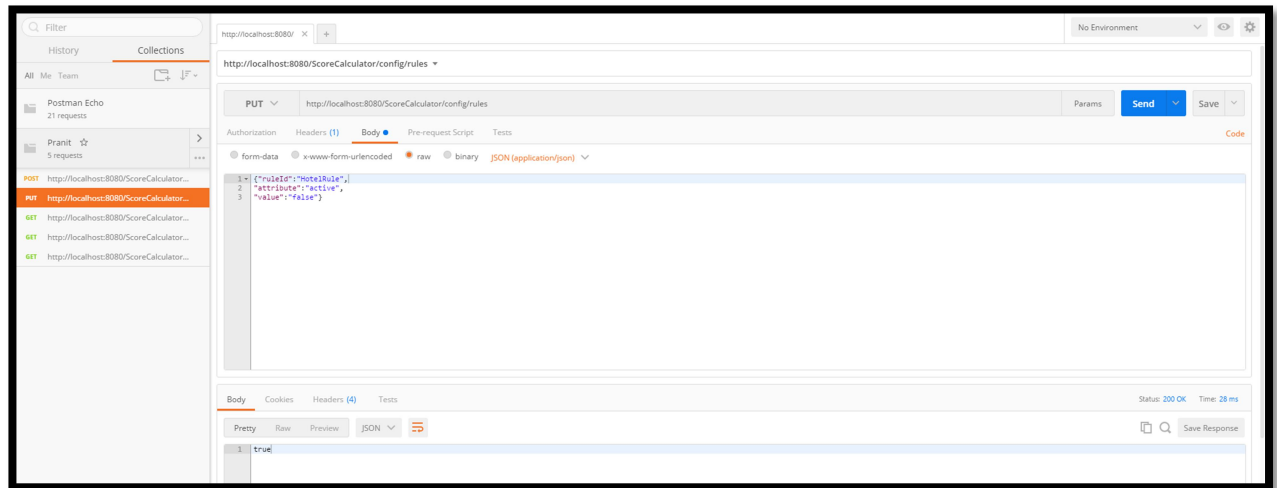
URL – <http://localhost:8080/ScoreCalculator/config/rules>

Request –

```
{
  "ruleId": "HotelRule",
  "attribute": "active",
  "value": "false"
}
```

OR

```
{
  "ruleId": "HotelRule",
  "attribute": "score",
  "value": "15"
}
```



Adding a new Rule

Any rule extends BaseRule and implements a method execute. Also each and every rule must register itself with Rule Engine using SuperClass's register method.

All Rules are present in RuleType Enum with their rule ids.

RuleEngine is responsible for executing business logic of rule along with storing and modifying the rules.

To add a new Rule –

- Create an entry in RuleType Enum.
- Create a new rule class extending BaseRule (See HotelRule for reference)
- Implement execute method and write the business logic for the rule.
- Register the rule using init method with RuleEngine.
- Any specific execution changed for the Rule can be accommodated in RuleEngine.

Improvements (Can be done)

- Hotel Store and Country Store can be loaded from DataBase and cached in corresponding Dao.
- Cache expiry and reload REST will trigger the change in the hotels.
- Rule Engine can be made more advance using configuration done from xml file. All the business logic will reside in xml and can be interpreted in Java.