Infosys

Responsible AI Office

Infosys Responsible AI Toolkit

API usage instructions

Contents

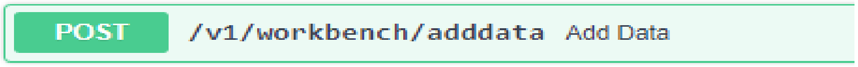
Introduction

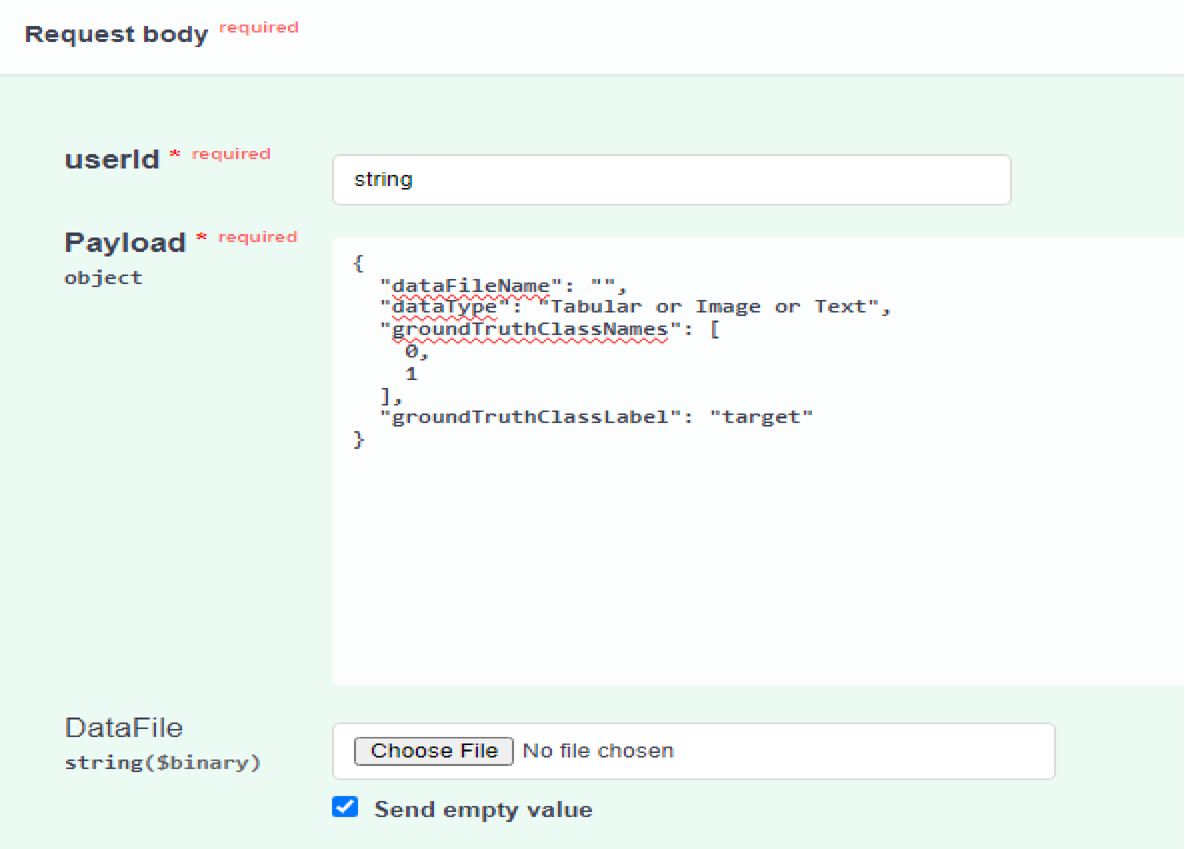
The introduction of a security module in RAI is a crucial step to safeguard the integrity and reliability of the platform. By incorporating robust security measures, RAI aims to protect user data, prevent unauthorized access, and ensure the platform's resilience against cyber threats. This module likely encompasses various security protocols, encryption techniques, authentication mechanisms, and risk management strategies to create a secure environment for users and their transactions.

Security module is a software component safeguarding digital systems by implementing robust protection mechanisms. Key features include authentication, authorization, encryption, intrusion detection, and access control to prevent unauthorized access, data breaches, and system failures.

Upload Dataset

Endpoint – /v1/workbench/adddata
Using this API, we can add dataset to the configured DB on which model has been trained as well as we want to get the assessment report.


POST /v1/workbench/adddata Add Data



Request body required

userId * required

string

Payload * required

object

```
{
  "dataFileName": "",
  "dataType": "Tabular or Image or Text",
  "groundTruthClassNames": [
    0,
    1
  ],
  "groundTruthClassLabel": "target"
}
```

DataFile

string($binary)

☑ Send empty value

userId : The "userId" parameter, which corresponds to the "usecase_name."

Payload:

dataFileName: The "dataFileName" field should contain the name of your data file.

dataType: The "dataType" field should specify the type of data, such as "tabular" ,”image” ,"text," based on the dataset.

groundTruthClassNames: "groundTruthClassNames" typically refers to the classes or categories present in a dataset, particularly in supervised learning tasks where the data is labeled. For example, in a dataset of images of fruits categorized into apples, bananas, and oranges, "apple," "banana," and "orange" would be the ground truth class names. This information helps in training machine learning models to correctly classify or predict the data. If the dataset contains class labels, they would be listed in the "groundTruthClassNames" field. If there are no class labels or categories in the dataset, this field would be left empty.

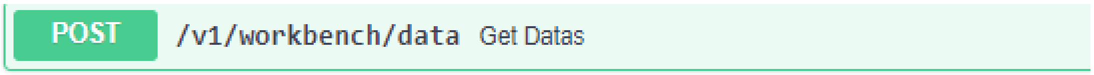
groundTruthClassLabel: Mention the target column name that your model is going to predict.

DataFile: Please select and upload the dataset file by browsing your device.

After entering all the necessary data, proceed by clicking on "execute."

If the information is successfully saved in the database, you will receive a response stating "Data added successfully.

To retrieve the details of the uploaded data, please navigate to the following API. You will receive information such as dataId and name etc.

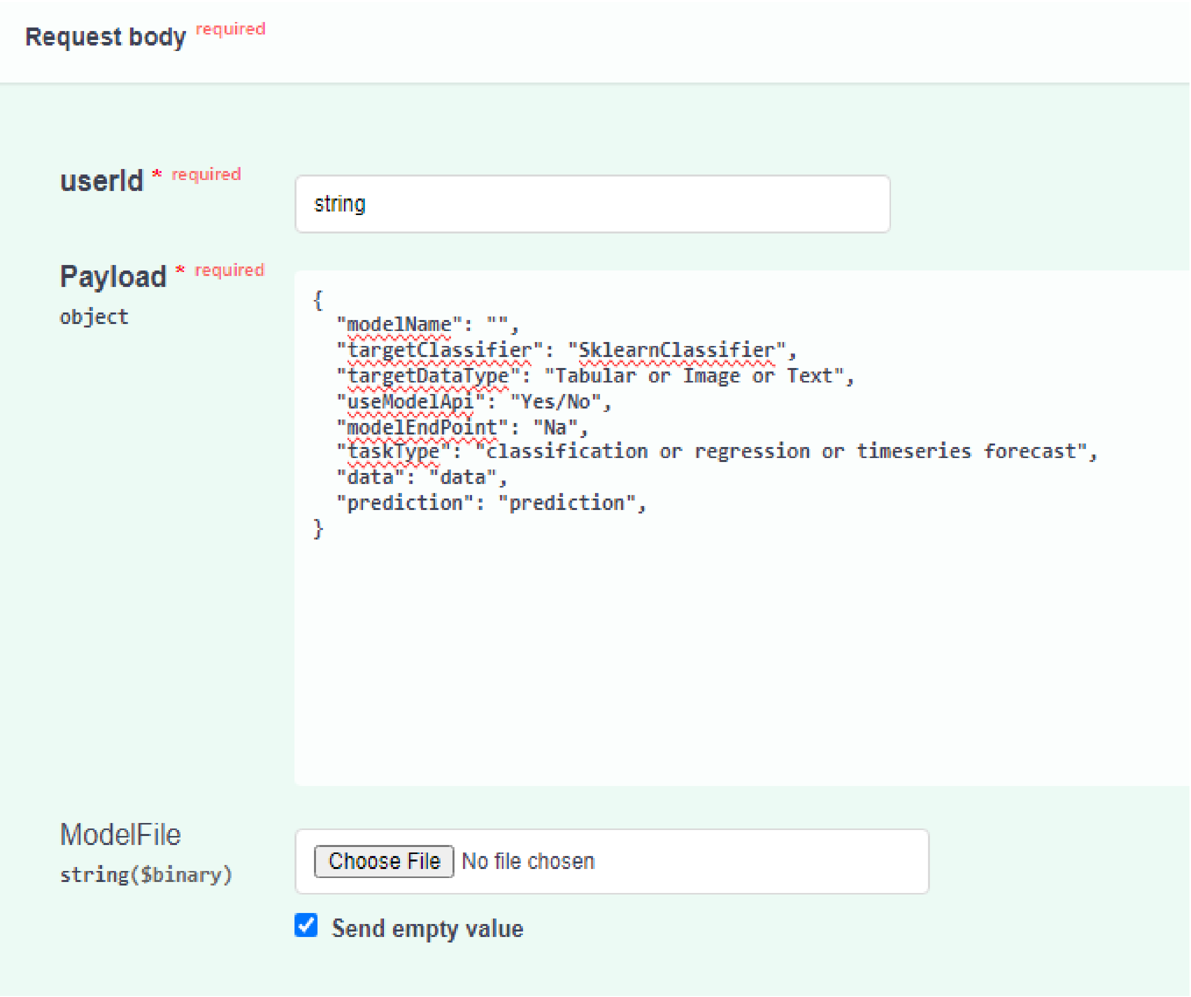


POST /v1/workbench/data Get Datas

Upload ModelFile

Endpoint – /v1/workbench/addmodel

Using this API, we can add ModelFile to the configured DB on which we want to get the assessment report. The requested payload is attached below:-



Request body required

userId * required

string

Payload * required

object

```
{
  "modelName": "",
  "targetClassifier": "SklearnClassifier",
  "targetDataType": "Tabular or Image or Text",
  "useModelApi": "Yes/No",
  "modelEndPoint": "Na",
  "taskType": "classification or regression or timeseries forecast",
  "data": "data",
  "prediction": "prediction",
}
```

ModelFile

string($binary)

Choose File No file chosen

☑ Send empty value

userId : The "userId" parameter, which corresponds to the "user_name."

Payload:

modelName : The "modelName" field should contain the name you choose for your model.

targetDataType: The "targetDataType" field should specify the type of data your model is designed to work with such as "tabular" , “image” , "text" based on the dataset it will be trained on.

taskType: The "taskType" field should specify the type of task your model is intended for. You can provide "CLASSIFICATION" if the model is for classification tasks, "REGRESSION" if it's for regression tasks, or "TIMESERIESFORECAST" if it's for time series forecasting.

targetClassifier: It is type of algorithm which is used to train the model.

useModelApi: If you are uploading the model, please provide "**No**." Otherwise, if you are providing the model via an endpoint, provide "**Yes**."

If "**useModelApi**" is set to "**Yes**":

modelEndpoint: If you are accessing the model via an endpoint, please specify your endpoint here in the "modelEndpoint" field.

data: If you are accessing the model via an endpoint, the "data" field should contain the input parameter of the endpoint, which binds input data to the endpoint.

prediction: If you are accessing the model via an endpoint, the "prediction" field should contain the output parameter of the endpoint, which delivers data from the endpoint.

ModelFile: Please ignore this field.

If "**useModelApi**" is set to "**No**":

In this case, you can either remove these three fields or leave them as they are:

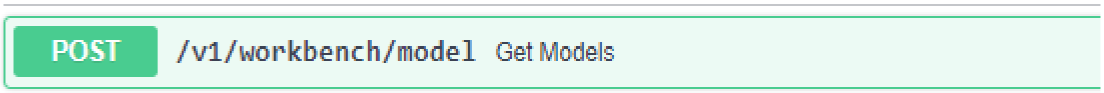
- modelEndpoint
- data
- prediction

ModelFile: Please select and upload the model file by browsing your device.

After entering all the necessary data, click on "execute."

If the model is successfully saved in the database, you will receive a response stating "Model added successfully."

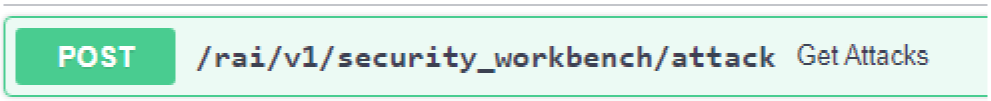
To retrieve the details of the uploaded model, please navigate to the following API. You will receive information such as modelId and name etc.



POST /v1/workbench/model Get Models

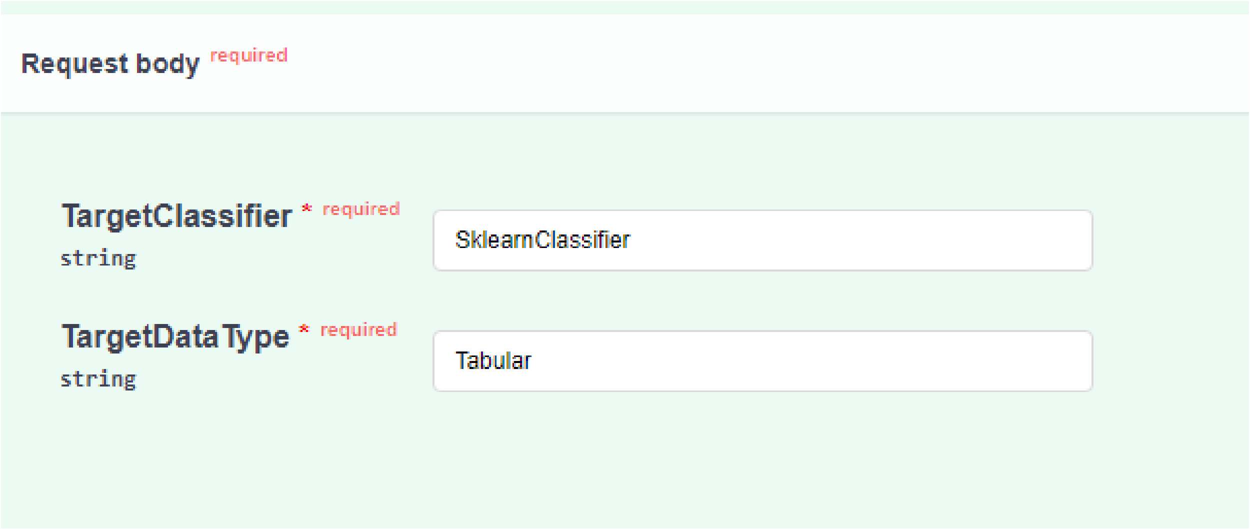
Get Applicable Attacks

Endpoint – rai/v1/security_workbench/attack
Using this API, we can get the list of applicable attacks which can be run on your provided model. To check which attacks is applicable, you have to provide classifier and data type to below API endpoint.



/rai/v1/security_workbench/attack Get Attacks

The requested payload is attached below

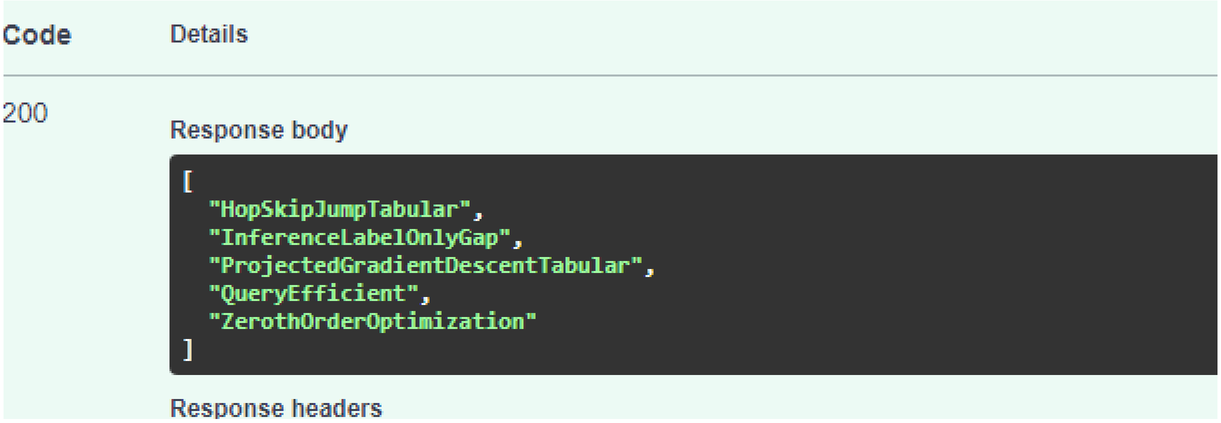

Request body required

TargetClassifier * required
string
SklearnClassifier

TargetDataType * required
string
Tabular

TargetClassifier : It is type of algorithm which is used to train the model like SklearnClassifier, KerasClassifier etc.
TargetDataType : It refer to the nature or format of the input features that are used to train a model like tabular,image or text.

Response :

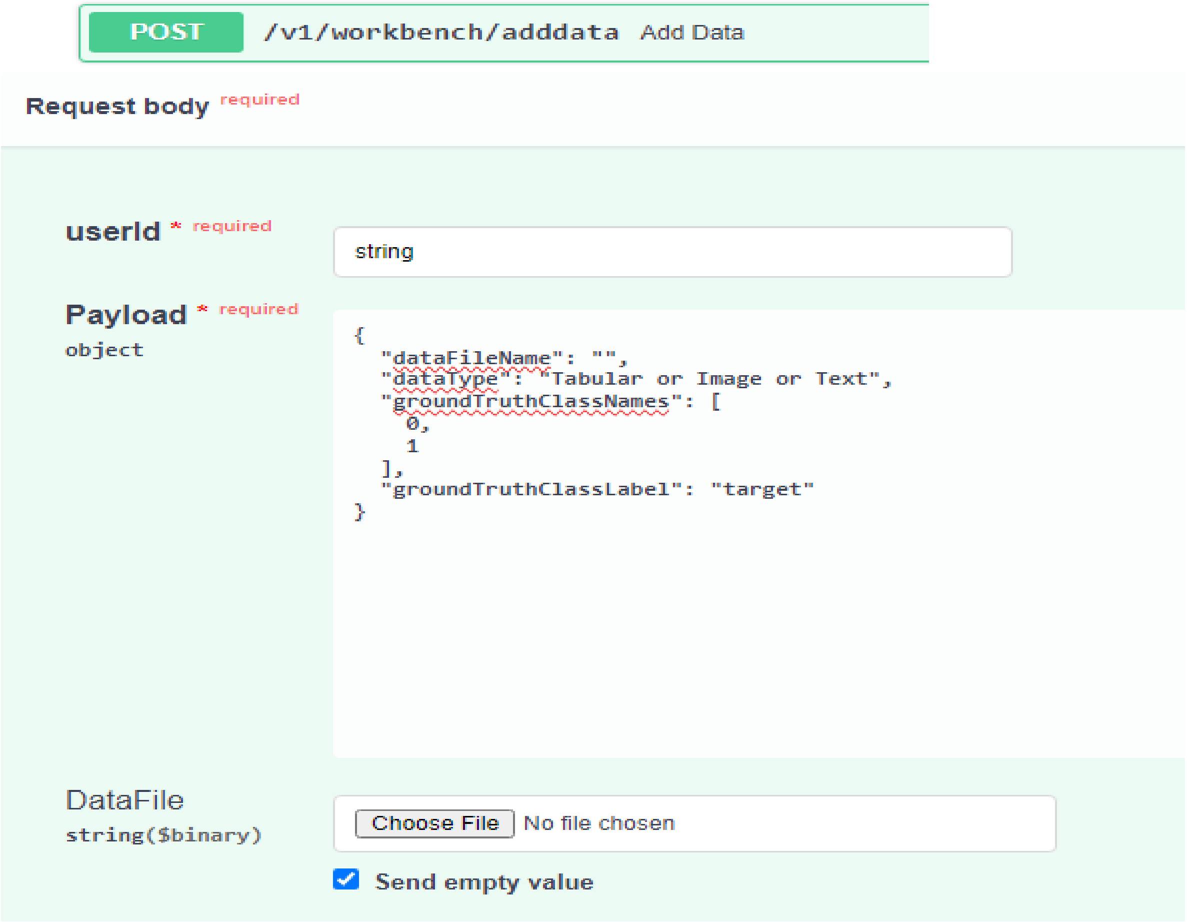


200

Response body

```
[
  "HopSkipJumpTabular",
  "InferenceLabelOnlyGap",
  "ProjectedGradientDescentTabular",
  "QueryEfficient",
  "ZerothOrderOptimization"
]
```

This is the list of applicable attacks on SklearnClassifier and tabular data.



/v1/workbench/adddata

Request body required

userId * required: string

Payload * required
object

```
{
  "dataFileName": "",
  "dataType": "Tabular or Image or Text",
  "groundTruthClassNames": [
    0,
    1
  ],
  "groundTruthClassLabel": "target"
}
```

DataFile
string($binary)

No file chosen

☑ Send empty value

userId : The "userId" parameter, which corresponds to the "usecase_name."

Payload:

dataFileName: The "dataFileName" field should contain the name of your data file.

dataType: The "dataType" field should specify the type of data, such as "tabular" ,”image” ,"text," based on the dataset.

groundTruthClassNames: "groundTruthClassNames" typically refers to the classes or categories present in a dataset, particularly in supervised learning tasks where the data is labeled. For example, in a dataset of images of fruits categorized into apples, bananas, and oranges, "apple," "banana," and "orange" would be the ground truth class names. This information helps in training machine learning models to correctly classify or predict the data. If the dataset contains class labels, they would be listed in the "groundTruthClassNames" field. If there are no class labels or categories in the dataset, this field would be left empty.

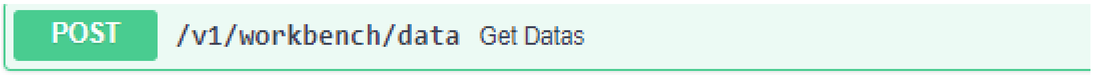
groundTruthClassLabel: Mention the target column name that your model is going to predict.

DataFile: Please select and upload the dataset file by browsing your device.

After entering all the necessary data, proceed by clicking on "execute."

If the information is successfully saved in the database, you will receive a response stating "Data added successfully.

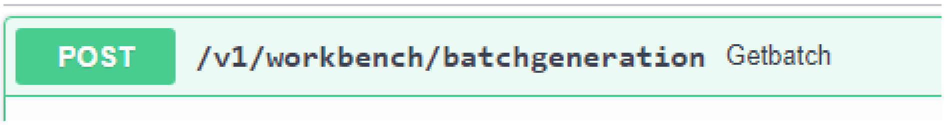
To retrieve the details of the uploaded data, please navigate to the following API. You will receive information such as dataId and name etc.



POST /v1/workbench/data Get Datas

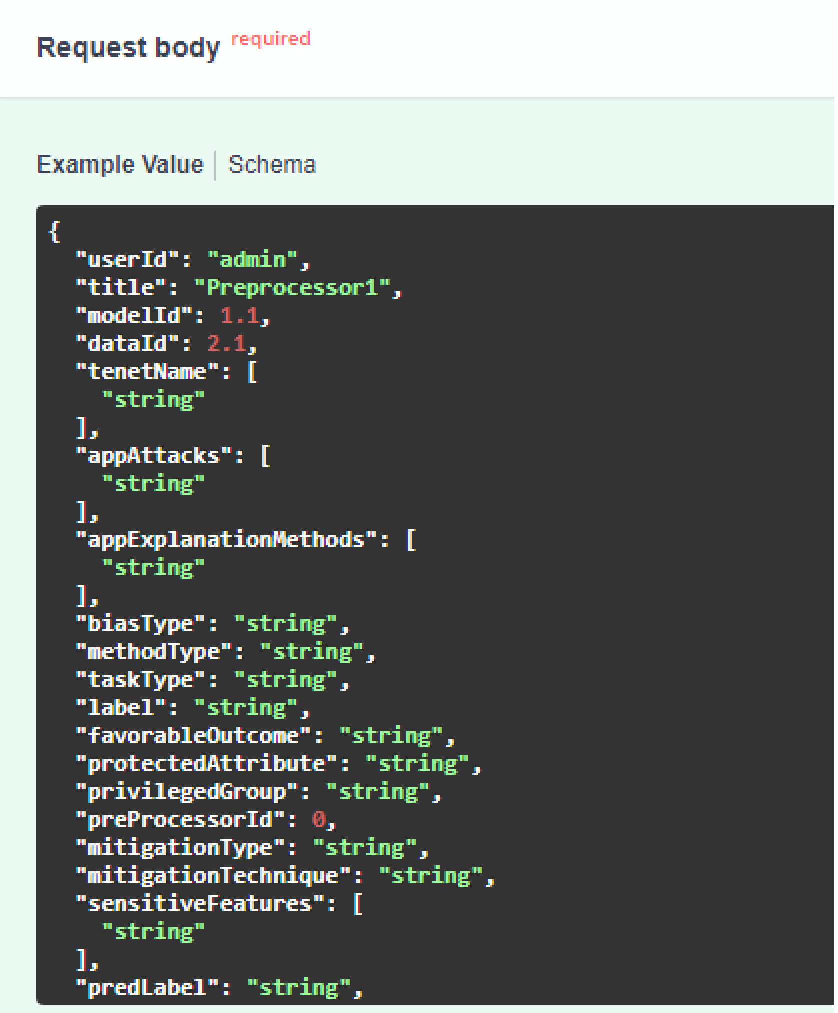
Batch Generation

Endpoint – /v1/security_workbench/ batchgeneration
Using this API, we can get the ID of our associated report.



POST /v1/workbench/batchgeneration Getbatch

The requested payload is attached below



```
{
  "userId": "admin",
  "title": "Preprocessor1",
  "modelId": 1.1,
  "dataId": 2.1,
  "tenetName": [
    "string"
  ],
  "appAttacks": [
    "string"
  ],
  "appExplanationMethods": [
    "string"
  ],
  "biasType": "string",
  "methodType": "string",
  "taskType": "string",
  "label": "string",
  "favorableOutcome": "string",
  "protectedAttribute": "string",
  "privilegedGroup": "string",
  "preProcessorId": 0,
  "mitigationType": "string",
  "mitigationTechnique": "string",
  "sensitiveFeatures": [
    "string"
  ],
  "predLabel": "string",
```

userId : The "userId" parameter, which corresponds to the "user_name."

modelId: You will obtain the "modelId" from the "get models" API. (/v1/workbench/model).

dataId: You will obtain the "dataId" from the "get datas" API. (/v1/workbench/data).
tenetName: "Security" is the tenet name.
appAttacks : It is the list of applicable attack which we run on our model. You will get it from above provided endpoint in step-3.
You can either remove these fields or leave them as they are:

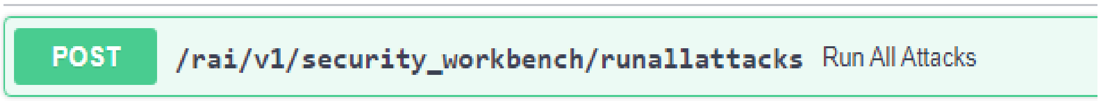
- title
- appExplanationMethods
- biasType
- methodType
- taskType
- label
- favorableOutcome
- protectedAttribute

- privilegedGroup
- preProcessorId
- mitigationType
- mitigationTechnique
- sensitiveFeatures
- predLabel
- knn

Once you have provided all the necessary data, click on "execute." You will receive the batchId and tenetId as a response.

Report Generation

Endpoint – /rai/v1/security_workbench/runallattacks
Using this API, we can add dataset to the configured DB on which model has been trained as well as we want to get the assessment report.



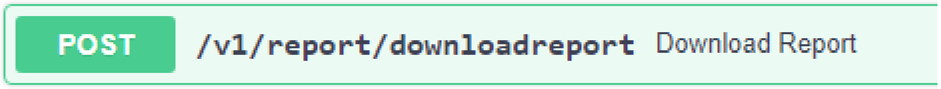
POST /rai/v1/security_workbench/runallattacks Run All Attacks

We need to pass the batchId as a request to above endpoint, and It will create the report and store it in zip format in the database.

Now that we've created the report as well. Next, we can proceed to download the report.

Download Report

Endpoint – /v1/report/downloadreport
Using this API, we can add dataset to the configured DB on which model has been trained as well as we want to get the assessment report.



POST /v1/report/downloadreport Download Report

To download the report, please refer to API Endpoint provided above. Use the same batchId that was used to generate the report.