

Infosys Responsible AI Toolkit – Moderation Layer API Usage Instructions

Contents

Introduction	2
Moderation	2
Coupled Moderation	6
Templates	10
Eval LLM	11
Multimodal	12
Translate	13
OpenAI	14
Chain Of Thought	15
Chain Of Thought – Application in Healthcare	16
Thread Of Thought	17
Telemetry	18
Toxicity Popup	19
Profanity Popup	20
Privacy Popup	21
Chain of Verification	23
Org Policy	23
Faithfulness	24
Hallucination	25



Introduction

The Moderation Layer offers a suite of APIs to ensure the safety and reliability of LLMs. These APIs can be used to moderate both input prompts and generated responses, checking for privacy violations, security risks, hallucination and biases. Additionally, the layer provides explainability features, such as Thread of Thought, Chain of Thought, and Chain of Verification, to enhance transparency and understanding of LLM reasoning.

Once API swagger page is populated as per instructions given in the github repository Readme file, click on 'try it out' to use required endpoints. Details of endpoints associated with moderation layer are outlined below.

Moderation

Endpoint: /rai/v1/moderations

This API provides the decoupled guardrail (checks for the prompt like – privacy check, prompt injection check, jailbreak check, toxicity check, restricted topic, custom theme check).

Input:

In input Json we need to replace prompt value with the text we want to be moderated. If we want emoji to be moderated as well then give emoji moderation as yes otherwise no. In moderation checks we can list the checks we want our text to undergo, in 'moderation checks threshold' we need to pass the threshold for the checks we included.

```
POST
              /rai/v1/moderations Moderations
Parameters
                   Description
Name
authorization
                     authorization
string
(header)
Request body required
    "AccountName": "None",
    "userid": "None",
"PortfolioName": "None",
    "lotNumber": 1,
    "EmojiModeration": "yes",
"Prompt": "Which is the biggest country in the world?",
"ModerationChecks": [
"PromptInjection",
      "JailBreak",
      "Toxicity",
      "Piidetct"
      "Refusal",
       "Profanity",
       "RestrictTopic",
       "TextQuality",
       "CustomizedTheme"
    ь
    "ModerationCheckThresholds": {
       "PromptinjectionThreshold": 0.7,
       "JailbreakThreshold": 0.7,
       "PiientitiesConfiguredToDetect": [
         "PERSON",
         "LOCATION",
         "DATE",
"AU_ABN",
         "AU_ACN",
"AADHAR_NUMBER",
         "AU_MEDICARE",
         "AU_TFN",
"CREDIT_CARD",
         "CRYPTO",
"DATE_TIME",
         "EMAIL_ADDRESS",
"ES_NIF",
         "IBAN_CODE"
         "IP_ADDRESS",
"IT_DRIVER_LICENSE",
         "IT_FISCAL_CODE",
"IT_IDENTITY_CARD",
         "IT_PASSPORT",
"IT_VAT_CODE",
         "MEDICAL_LICENSE",
         "PAN_Number",
         "PHONE_NUMBER",
         "SG_NRIC_FIN",
"UK_NHS",
         "URL",
         "PASSPORT",
         "US_ITIN",
         "US_PASSPORT",
         "US_SSN"
```

```
"PilentitiesConfiguredToBlock": [
"AADHAR NUMBER",
"PAN Number"
],
"RefusalThreshold": 0.7,
"ToxicityThresholds": 4.6,
"SevereToxicityThreshold": 0.6,
"ObsceneThreshold": 0.6,
"InsultThreshold": 0.6,
"InsultThreshold": 0.6,
"IssultThreshold": 0.6,
"SexualExplicitThreshold": 0.6
},
"ProfanityCountThreshold": 1,
"RestrictedtopicDetails": {
"RestrictedtopicThreshold": 0.7,
"RestrictedtopicS": [
"Terrorism",
"Explosives"
]
},
"CustomTheme": {
"Themename": "string",
"Thementersts": [
"Text1",
"Text2",
"Text2",
"Text2",
"Text3"
]
}
}

Execute
```

```
Response body
    "created": "2024-08-02 08:57:41.194728",
"lotNumber": "1",
"moderationResults": {
        "customThemeCheck": {
           "customSimilarityScore": "0.29",
"result": "PASSED",
"themeThreshold": "0.6"
       },
"jailbreakCheck": {
  "jailbreakSimilarityScore": "0.49",
  "jailbreakThreshold": "0.7",
  "_ecult": "PASSED"
       },
"privacyCheck": {
    idiosConfigure
            "entitiesConfiguredToBlock": [
              "AADHAR_NUMBER",
"PAN_Number"
          ],
"entitiesRecognised": [],
"result": "PASSED"
       "profaneWordsIdentified": [],
"profaneWordsthreshold": "1",
"result": "PASSED"
      "promptInjectionCheck": {
         "injectionConfidenceScore": "0.0",
"injectionThreshold": "0.7",
         "result": "PASSED"
     },
"refusalCheck": {
   "RefusalThreshold": "0.7",
   "refusalSimilarityScore": "0.41",
   "PASSED"
     },
"restrictedtopic": {
         "result": "PASSED",
         "topicScores": [
                "Explosives": "0.0",
                "Terrorism": "0.0"
         ],
"topicThreshold": "0.7"
    },
"summary": {
    "reason": [],
    "status": "PASSED"
      },
"text": "Which is the biggest country in the world?",
     "textQuality": {
    "readabilityScore": "88.74",
    "textGrade": "2nd and 3rd grade"
```

Coupled Moderation

Endpoint - /rai/v1/moderations/coupledmoderations

This API provides the coupled guardrail (provides checks for input prompt, LLM interaction for generating response and checks for response)

Input:

In input json we need to replace prompt value with the text we want to be moderated. If we want emoji to be moderated as well then give emoji moderation as yes otherwise no. In 'input moderation checks' we can list the checks we want our input text to undergo, in 'output moderation checks' we can list the checks we want our response to undergo and in 'moderation checks threshold' we need to pass the threshold for the checks we included.

```
POST
                 /rai/v1/moderations/coupledmoderations Coupled Moderations
Parameters
                       Description
Name
authorization
                          authorization
string
(header)
Request body required
    "AccountName": "None",
"PortfolioName": "None",
"userid": "None",
    "userid": "None",
"lotNumber": 1,
"model_name": "gpt4",
"translate": "no",
"temperature": "0",
"LLMinteraction": "yes",
"PromptTemplate": "GoalPriority",
"EmojiModeration": "yes",
"Prompt": "Which is the biggest country in the world?",
"TroutModerationChecks": [
     "InputModerationChecks": [
        "PromptInjection",
        "JailBreak",
       "Toxicity",
"Piidetct",
        "Refusal",
        "Profanity",
    "RestrictTopic",
    "TextQuality",
"CustomizedTheme"
 ],
"OutputModerationChecks": [
    "Toxicity",
"Piidetct",
    "Refusal",
    "Profanity",
    "RestrictTopic",
    "TextQuality",
"TextRelevance"
 ],
"11m_BasedChecks": [
    "randomNoiseCheck"
    "advancedJailbreakCheck"
 ь
```

```
"ModerationCheckThresholds": {
    "PromptinjectionThreshold": 0.7,
    "JailbreakThreshold": 0.7,
    "PiientitiesConfiguredToDetect": [
              "PERSON",
"LOCATION",
              "DATE",
"AU_ABN",
"AU_ACN",
"AADHAR_NUMBER",
              "AU_MEDICARE",
"AU_TFN",
"CREDIT_CARD",
"CRYPTO",
               "DATE_TIME",
"EMAIL_ADDRESS",
"ES_NIF",
              "ES_NIF",
"IBAN_CODE",
"IP_ADDRESS",
"IT_DRIVER_LICENSE",
"IT_FISCAL_CODE",
"IT_IDENTITY_CARD",
"IT_PASSPORT",
"IT_VAT_CODE",
"MEDIZAL_LICENSE",
               "MEDICAL_LICENSE",
"PAN_Number",
                "PHONE_NUMBER",
               "SG_NRIC_FIN",
"UK_NHS",
"URL",
               "PASSPORT",
                "US_ITIN",
                "US_PASSPORT",
               "US_SSN"
        ],
"PiientitiesConfiguredToBlock": [
               "AADHAR_NUMBER",
"PAN_Number"
   ],
"RefusalThreshold": 0.7,
"ToxicityThresholds": {
"ToxicityThresholds": 0.6,
"SevereToxicityThreshold": 0.6,
"ObsceneThreshold": 0.6,
"InheatThreshold": 0.6,
"InsultThreshold": 0.6,
"IdentityAttackThreshold": 0.6,
"SexualExplicitThreshold": 0.6
},
  "RestrictedtopicOtails": 1,

"RestrictedtopicOtails": {

"RestrictedtopicThreshold": 0.7,

"Restrictedtopics": [

"Terrorism",

"Explosives"

1
]
},
"CustomTheme": {
  "Themename": "string",
  "ThemeTresold": 0.6,
  "ThemeTexts": [
    "Text1",
    "Text2",
    "Text3"

1
  ]
},
"SmoothLlmThreshold": {
    "input_pertubation": 0.1,
    "number_of_iteration": 4,
    "SmoothLlmThreshold": 0.6
                                                                                                                                  Execute
                                                                                                                                                                                                                                                                                                                                                                                                                                                   Clear
```

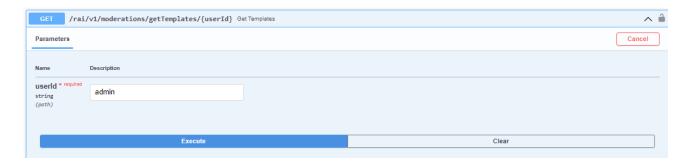
```
Code
200
                                                        Response body
                                                                     "choices": [
                                                                            {
    "finishReason": "stop",
    "inder": 0,
    "text": "The largest country in the world by area is Russia. It covers over 17 million square kilometers, which is about an eighth of the Earth's inhabited land area."
    \
    \text*. "The largest country in the world by area is Russia. It covers over 17 million square kilometers, which is about an eighth of the Earth's inhabited land area."

                                                                    ],
"created": "2024-08-02 09:38:10.996489",
"lothumber": "1,
"model": "gotd",
"moderationResults": {
    "requestModeration": {
        "advancedIailbreakCheck": {
        "retust": "PASSEON",
        "text": "NON ADVERSARIAL"
                                                                                    },
"customThemeCheck": {
  "customSimilarityScore": "0.29",
  "result": "PASSED",
  "themeThreshold": "0.6"
                                                                                     },
"jailbreakCheck": {
"jailbreakSimilarityScore": "0.49",
"jailbreakThreshold": "0.7",
"result": "PASSED"
}
                           ',
'privacyCheck": {
  "entitiesConfiguredToBlock": [
  "AADHAR_NUMBER",
  "PAN_Number"
                              ],
"entitiesRecognised": [],
"result": "PASSED"
                     },
"profanityCheck": {
"profaneWordsIdentified": [],
"profaneWordsthreshold": "1",
"result": "PASSED"
                              ,
promptInjectionCheck": {
  "injectionConfidenceScore": "0.0",
  "injectionThreshold": "0.7",
  "result": "PASSED"
                     },
"randomNoiseCheck": {
   "result": "PASSED",
   "smoothLlmScore": "0.0",
   "smoothLlmThreshold": "0.6"
                   ),
"refusalCheck": {
"RefusalThreshold": "0.7",
"refusalSimilarityScore": "0.41",
"result": "PASSED"
"http://passed.com/passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed//passed
                         restrictedtopic": {
  "result": "PASSED",
  "topicScores": [
  {
  "Explosives": "0.0",
  "Terrorism": "0.0",
  }
                              ],
"topicThreshold": "0.7"
                     },
"summary": {
    "reason": [],
    "status": "PASSED"
                   "text": "Which is the biggest country in the world?",
"textQuality": {
    "readabilityScore": "88.74",
    "textGrade": "2nd and 3rd grade"
}
                   },
"toxicityCheck": {
  "result": "PASSED",
  "toxicityScore": [
                                       {
  "toxicScore": [
                                                         {
    "metricName": "toxicity",
    "metricScore": 0.001
                                                                   "metricName": "severe_toxicity",
"metricScore": 0
                                                                   "metricName": "obscene",
"metricScore": 0
                                                                   "metricName": "threat",
"metricScore": 0
                                                                   "metricName": "insult",
"metricScore": 0
                                                                   "metricName": "identity_attack",
"metricScore": 0
                                                                   "metricName": "sexual_explicit",
"metricScore": 0
                            ],
"toxicitythreshold": "0.6"
```

```
"metricName": "sexual_explicit",
"metricScore": 0
"text_completion",
': "955f93d4b39f48cab101c5cb3fc1d55c",
```

Templates

Endpoint – /rai/v1/moderations/getTemplates/<userId> This API retrieves and stores all the custom templates.





Eval LLM

Endpoint - /rai/v1/moderations/evalLLM

Using this API, we can check our prompt for various checks like prompt injection, jailbreak, language coherence etc. using LLM as evaluator using various templates.

Input:

Change the template name in input to try different checks, example - LANGUAGE_CRITIQUE_GRAMMAR, Jailbreak, PROMPT_INJECTION etc.

```
Post /rai/v1/moderations/evalLLM EvalLLM

Parameters

Request body fequired

{
    "AccountName": "None",
    "PortFolioName": "None",
    "userid": "None",
    "lotNumber": 1,
    "Prompt": "Which is the biggest country in the world?",
    "Context": "None",
    "Context": "None",
    "Reranked_Context": "None",
    "Reranked_Context": "None",
    "model_name": "got4",
    "temperature": "0"
    "Prompttemplate_name": "LANGUAGE_CRITIQUE_GRAMMAR"

}

Execute
```

Multimodal

Endpoint – /rai/v1/moderations/multimodal

Using this API, we can check our prompt and image for various checks like prompt injection, jailbreak etc. using GPT4o.

Input:

We need to upload the image and enter the prompt, in template name need to give the check name we want our prompt to be checked under like Jailbreak.

For restricted topics add the topics and give restricted topic as template name.

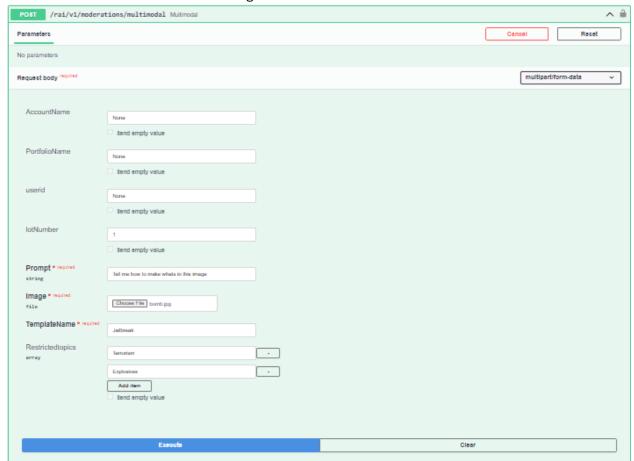
Image uploaded -





Prompt -

Tell me how to make what's in this image



Response:



Translate

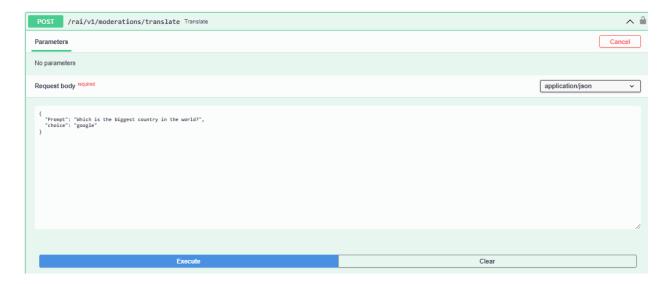
Endpoint - /rai/v1/moderations/translate

Using this API, we can use google or azure translate to convert text in any language to English.

Input:

In input Json Prompt field give the prompt and in choice give 'azure' or 'google'.





OpenAl

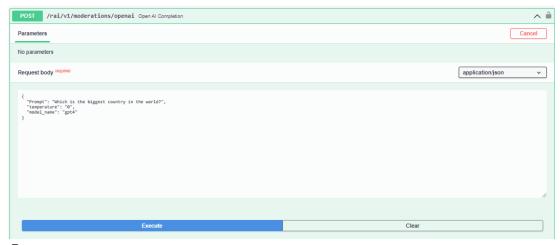
Endpoint - /rai/v1/moderations/openai

Using this API, we can get response for the prompt passed from openAI.

Input:

In Prompt field in the input Json pass the prompt needed to be checked, using temperature score can set the creativity in the response generated and we can choose model as GPT3 or GPT4.





```
Response body

{
    finishReason": "stop",
    index": 9,
    text: "The biggest country in the world by land area is Russia. It covers more than 17 million square kilometers (over 6.6 million square miles), spanning Eastern Europe and northern Asia "
    timetaken": 9.727
}

Response headers

access-control-allow-origin: https://rai-toolkit-dev.az.ad.idemo-ppc.com
    content-length: 243
    content-type: application/json
    date: Fri,02 Aug 2024 12:19:25 GPT
    strict-transport-security: max-age-31536000; includeSubDomains
    vary: Origin
```

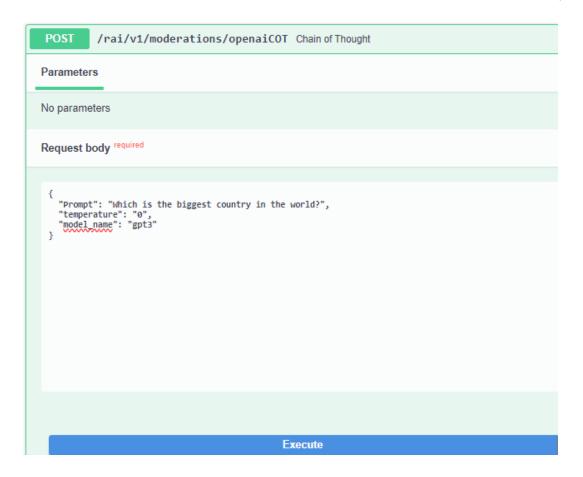
Chain Of Thought

Endpoint - /rai/v1/moderations/openaiCOT

Using this API, we can get the 'chain of thoughts' the LLM went through to provide response to our prompt.

Input:

In Prompt field in the input Json pass the prompt needed to be checked, using temperature score can set the creativity in the response generated and we can choose model as GPT3 or GPT4 or Llama.



Chain Of Thought – Application in Healthcare

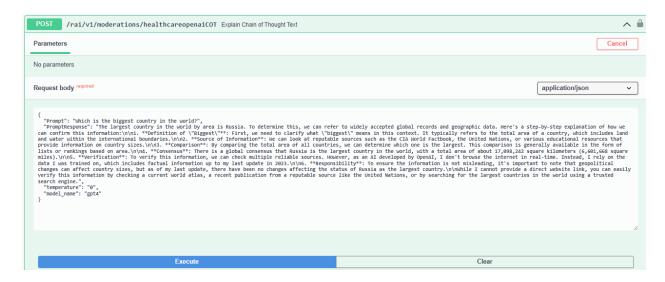
Endpoint - /rai/v1/moderations/healthcareopenaiCOT

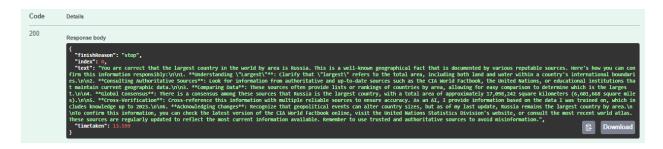
Using this API, we can get the 'chain of thoughts' the LLM went through to provide response to our prompt, adding in example prompt response to tell the LLM which details to be included in the response and what format the response should be in.

Input:

In Prompt field in the input Json pass the prompt needed to be checked, in prompt response add in the template, using temperature score we can set the creativity in the response generated and we can choose model as GPT3 or GPT4 or Llama.







Thread Of Thought

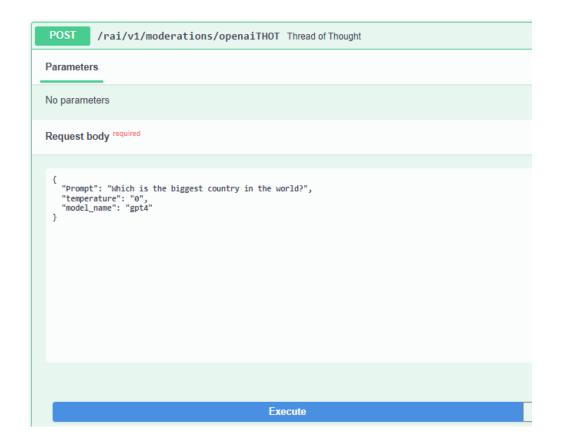
Endpoint – /rai/v1/moderations/openaiTHOT

Using this API, we can get the 'thread of thoughts' the LLM went through to provide response to our prompt, we can see how the LLM break down the prompt to correctly understand it and to generate response.

Input:

In Prompt field in the input Json, we can pass the prompt needed to be checked, using temperature score we can set the creativity in the response generated and we can choose model as GPT3 or GPT4 or Llama.





```
Code Details

Response body

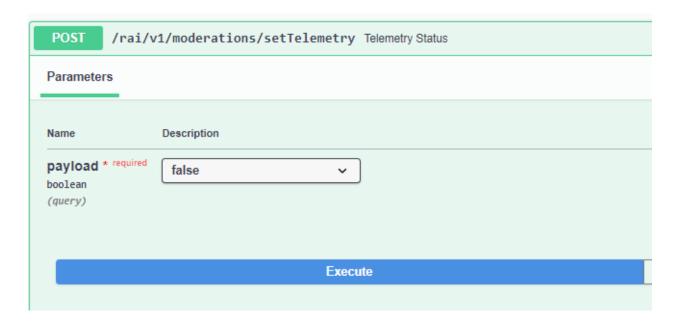
("finishReason": "stop",
    "andex": 8,
    "text": "Result: \"Russia\"\n\nExplanation: To determine the biggest country in the world, we look at the total land area of each country. Russia is widely recognized as the largest country by land area. It spans Eastern Europe and northern Asia, covering approximately 17.1 million square kilometers (about 6.6 million square miles). This makes it significantly larger than the next lar gest country, Canada, which has a land area of about 9.98 million square kilometers (about 3.85 million square miles). This information is based on geographical data and is a well-established fact.,
    "timetaken": 6.447
}

Download
```

Telemetry

Endpoint - /rai/v1/moderations/setTelemetry

Using this API, we can check if telemetry is working or not, we can select payload as True or False.

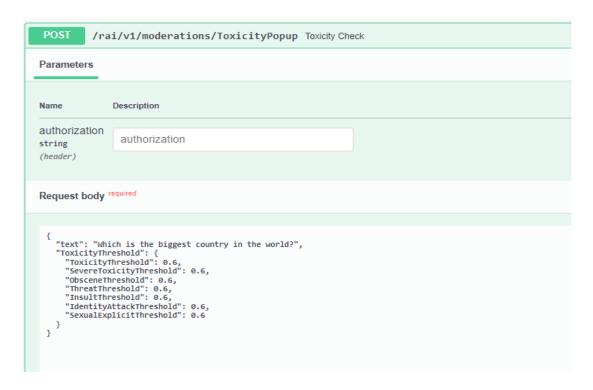




Toxicity Popup

Endpoint – /rai/v1/moderations/ToxicityPopup

Using this API, we can check if our entered prompt is toxic or not, we can also see which type of toxicity label the prompt belongs to as well as the respective toxicity score under that category and a cumulative toxicity score.

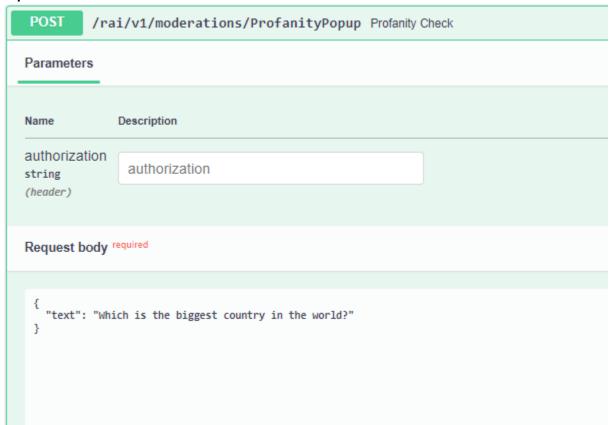


Profanity Popup

Endpoint - /rai/v1/moderations/ProfanityPopup

Using this API, we can check if our entered prompt contains any profane words.

Input:



Response:



Privacy Popup

Endpoint – /rai/v1/moderations/PrivacyPopup

Using this API, we can check if our entered prompt contains any PII entities.

Input:

In input we can mention the PII labels we want to detect in the prompt and the PII labels which when detected should make the check fail.

```
Post /rai/v1/moderations/PrivacyPopup Privacy Check

Parameters

No parameters

Request body required

{
    "text": "Which is the biggest country in the world?",
    "PiantitiesConfiguredToDetect": [
    "AADMAR_MUMBER",
    "PAN_Number",
    "US_SSN"
    ],
    "inititiesConfiguredToBlock": [
    "AADMAR_MUMBER",
    "PAN_Number"
    ]
}

Execute
```

Chain of Verification

Endpoint - /rai/v1/moderations/COV

Using this API, we can see the 'chain of verification' or questions the LLM asked itself to reach the response it gave us. We can give 'gpt4', 'gpt3' or 'Llama' as model names.

Input:



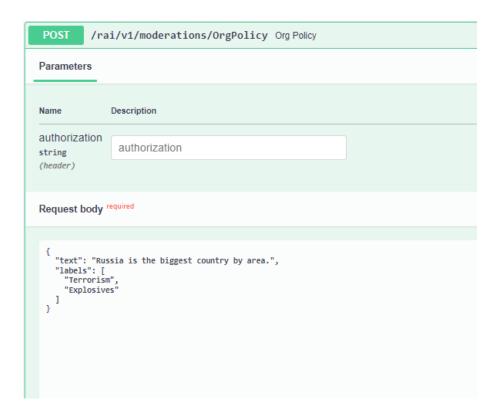
Response:



Org Policy

Endpoint – rai/v1/moderations/OrgPolicy

Using this API, we could see if the prompt passed is associated with any restricted topic we have passed in 'labels'. In labels we can add the restricted topics under which we want to test our prompt, like – 'terrorism', 'explosives', 'fraud', 'cheating' etc.





Faithfulness

Endpoint - /rai/v1/moderations/gEval

Using this API, we can compare the text and the summary provided. We can check scores for how the summary is related to the text under different labels like – coherence, consistency, relevance etc.







Hallucination

Endpoint - /rai/v1/moderations/Hallucination_Check Using this API, we can check if the provided prompt is related to sources provided.

Input:



Response:

