Infosys

Responsible AI Office

Infosys Responsible AI Toolkit

Questionnare Workbench

API usage Instructions

Contents

About Workbench

Artificial Intelligence (AI) offers numerous benefits across various domains, It is reshaping our world and driving innovation. It is necessary for an AI system to be developed and deployed responsibly, ensuring it is safe, private, transparent, and accountable. Essentially, it means that the system should be able to justify its operations in a way that aligns with ethical principles and builds user trust. Hence this adherence to responsible AI tenets is crucial for building trust, adoption of AI and ensuring accountability in high-risk AI applications.

Once the API swagger page is populated as per instructions given in the GitHub repository Readme file, click on ‘try it out’ to use the required endpoints. Details of endpoints associated with the responsible AI tenets are outlined below.

Dependencies

The Responsible AI Workbench depends on several external microservices for its core analysis capabilities. These include dedicated services for Privacy, Safety (Profanity), FM-Moderation, and Explainability. Please ensure the API endpoints for these services are correctly configured in the .env file as per the setup instructions. Ensure that all dependent microservices are up and running before interacting with the workbench APIs for file uploads and analysis

Privacy Repository (responsible-ai-privacy)

The privacy repository is used to detect and extract PII entities from input text using NLP models. It enables downstream analysis, redaction, or anonymization of sensitive data based on configurable parameters.

3.1: v1/privacy/text/analyze

Analyzes text content to identify and detect PII entities using NLP models.

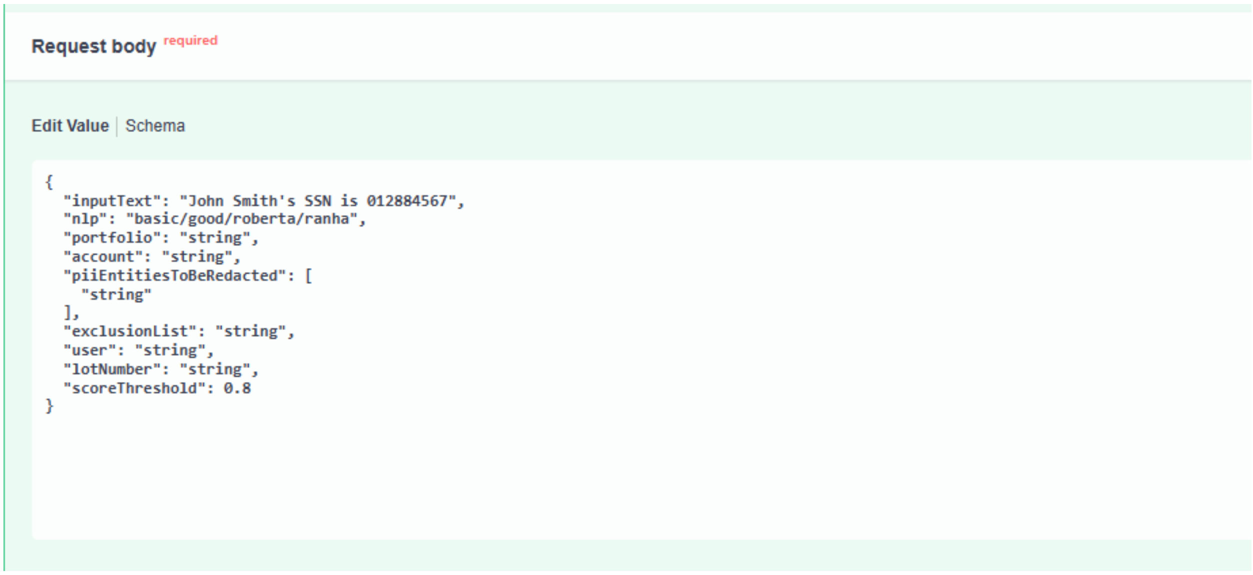
Provided below the details of payloads.

- **inputText :** The text content to be analyzed for PII detection. Mandatory string field.
- **portfolio :** Portfolio name for account validation and tracking. Optional string field with default None.
- **account :** Account name for validation and billing purposes. Optional string field with default None.
- **exclusionList :** Comma-separated list of PII entity types to exclude from analysis (e.g., "PERSON,EMAIL"). Optional string field with default None.
- **user :** User identifier for tracking and audit purposes. Optional string field with default None.
- **lotNumber :** Lot number for batch processing identification. Optional field with default None.

- **piiEntitiesToBeRedacted :** List of specific PII entity types to focus analysis on (e.g., ["US_SSN", "PHONE_NUMBER"]). Optional list field with default None.

Once all fields are filled in, click "**execute**" to proceed.

Returns a object containing an array of PIIEntities with detected PII information including entity type, character positions (beginOffset/endOffset), confidence score, and the actual detected text value



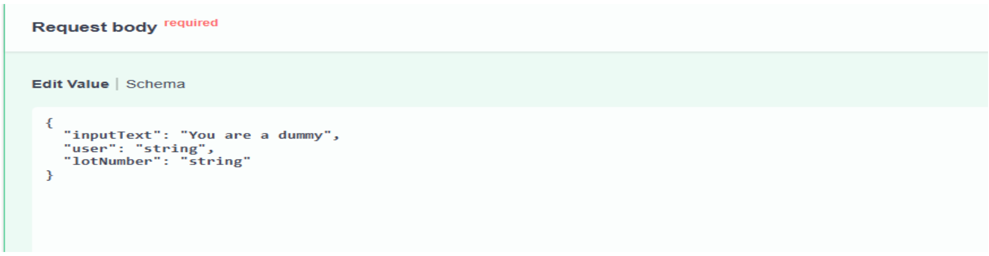
```
{
  "inputText": "John Smith's SSN is 012884567",
  "nlp": "basic/good/roberta/ranha",
  "portfolio": "string",
  "account": "string",
  "piiEntitiesToBeRedacted": [
    "string"
  ],
  "exclusionList": "string",
  "user": "string",
  "lotNumber": "string",
  "scoreThreshold": 0.8
}
```

Safety Repository (responsible-ai-safety)

This repo is used to analyze input text for profane or harmful language using specialized detection models. It supports responsible AI by detecting and evaluating offensive language, enabling the system to block or control unsuitable content.

4.1: api/v1/safety/profanity/analyze

Analyzes input text to detect profane content and returns profanity analysis results with detailed scoring metrics.



```
{
  "inputText": "You are a dummy",
  "user": "string",
  "lotNumber": "string"
}
```

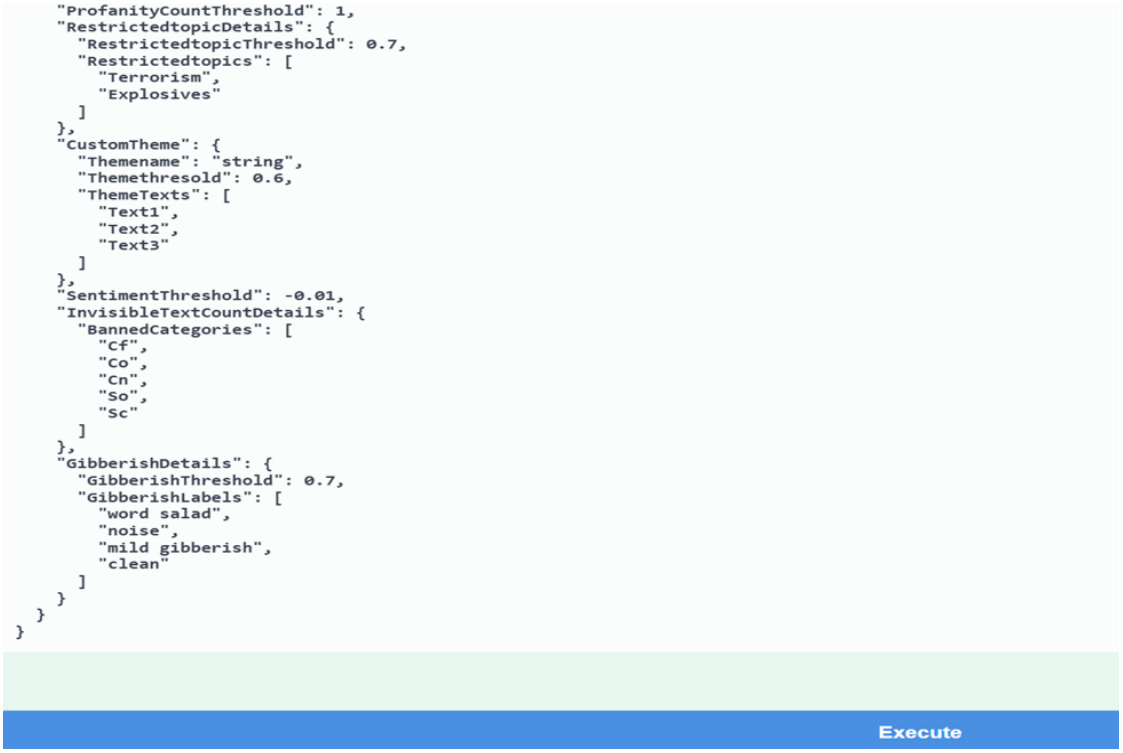
- **inpuText :** The text content to be analyzed for profanity.
- **User :** User identifier for tracking purposes.
- **lotNumber :** Lot number for batch processing identification.

Moderation Repository (responsible-ai-fm-ext-flask)

This repository provides comprehensive guardrail checks to detect and prevent unsafe or unwanted content in input prompts. It enables configurable moderation for privacy, toxicity, prompt injections, and other risks to ensure safer AI interactions.

5.1: 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; along with some other optional checks like gibberish check, invisible text check, ban code check, sentiment check).



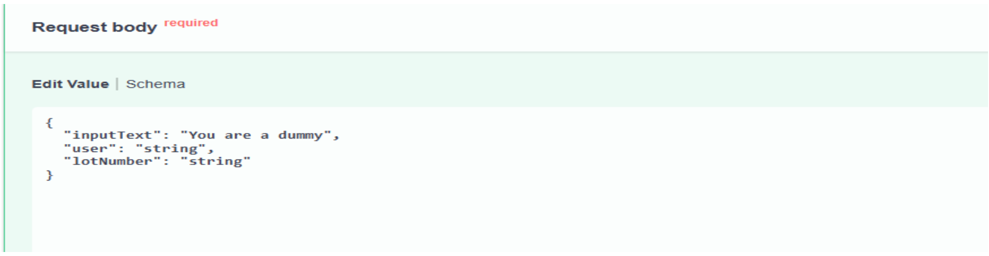
```
    "ProfanityCountThreshold": 1,
    "RestrictedtopicDetails": {
      "RestrictedtopicThreshold": 0.7,
      "Restrictedtopics": [
        "Terrorism",
        "Explosives"
      ]
    },
    "CustomTheme": {
      "Themename": "string",
      "Themethresold": 0.6,
      "ThemeTexts": [
        "Text1",
        "Text2",
        "Text3"
      ]
    },
    "SentimentThreshold": -0.01,
    "InvisibleTextCountDetails": {
      "BannedCategories": [
        "Cf",
        "Co",
        "Cn",
        "So",
        "Sc"
      ]
    },
    "GibberishDetails": {
      "GibberishThreshold": 0.7,
      "GibberishLabels": [
        "word salad",
        "noise",
        "mild gibberish",
        "clean"
      ]
    }
  }
}
```

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.

Explainability Repository (responsible-ai-explain)

This repository enables the AI system to generate clear, human-understandable explanations for its decisions or predictions. It supports transparency and builds user trust by providing insights into the model's reasoning process.

4.1: v1/explainability/local/explain :



Request body required

Edit Value | Schema

```
{
  "inputText": "You are a dummy",
  "user": "string",
  "lotNumber": "string"
}
```

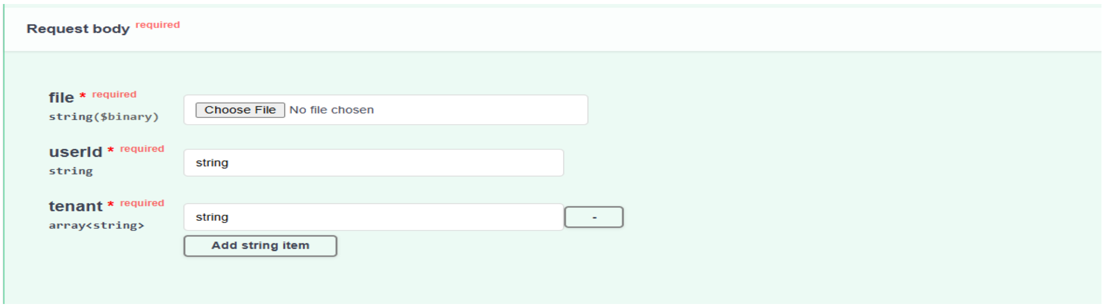
Features & API End Points:

• Workbench APIs

We have four APIs available in the Workbench repository, each serving a different purpose as detailed below:

5.1: /questionnaire/workbench/uploadFile - Uploads a CSV file and performs analysis across selected responsible AI tenets.

Payload



Request body required

file * required
string($binary)
Choose File No file chosen

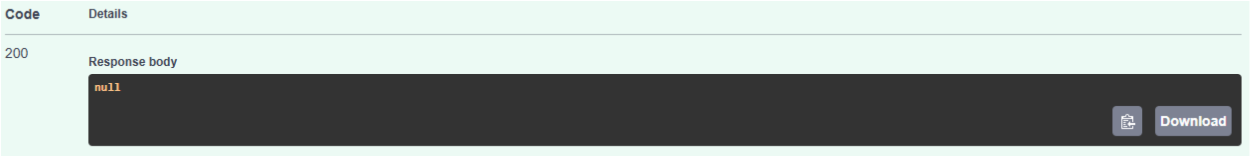
userId * required
string
string

tenant * required
array<string>
string

- **file** : Upload CSV file containing text data (one text entry per row, no headers).
- **userId** : User identifier for tracking and lot allocation.
- **Tenant** : List of analysis types to perform. Available options:
  - "Privacy" - Privacy and PII detection analysis
  - "Safety" - Profanity and toxicity analysis
  - "FM-Moderation" - Comprehensive foundation model moderation
  - "Explainability" - AI model explainability analysis

Returns a list of applicable methods as the response

Output

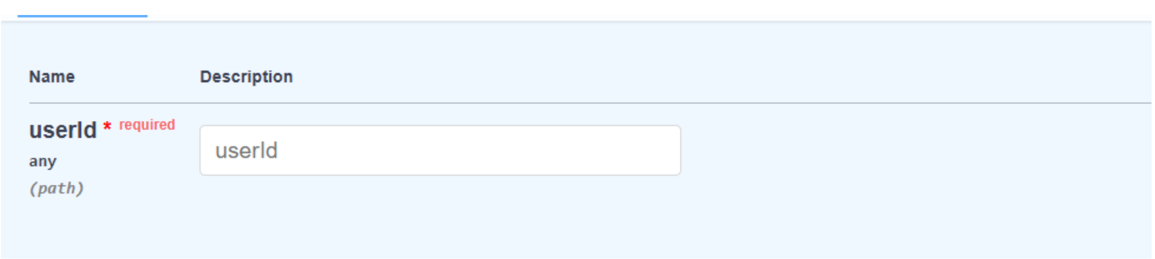


Code	Details
200	Response body

```
null
```

5.2: /questionnaire/allLotDetails/{userId} - Retrieves all analysis lot details for a specific user**.**

Payload

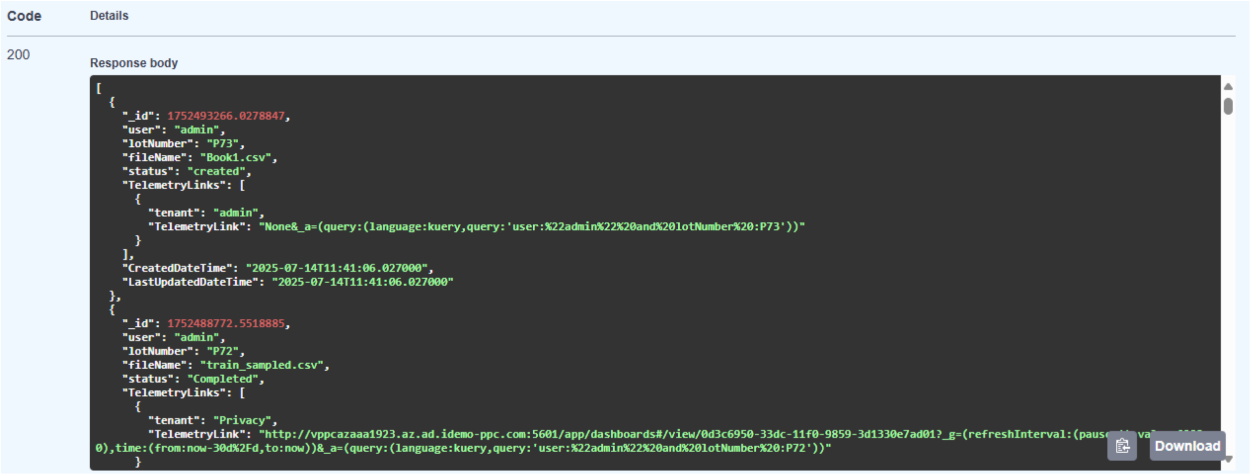


Name	Description
userId * required any *(path)*	userId

- **userId** : User identifier to fetch lot history.

Returns a List of all analysis lots with status, file names, and telemetry links (ordered by most recent first).

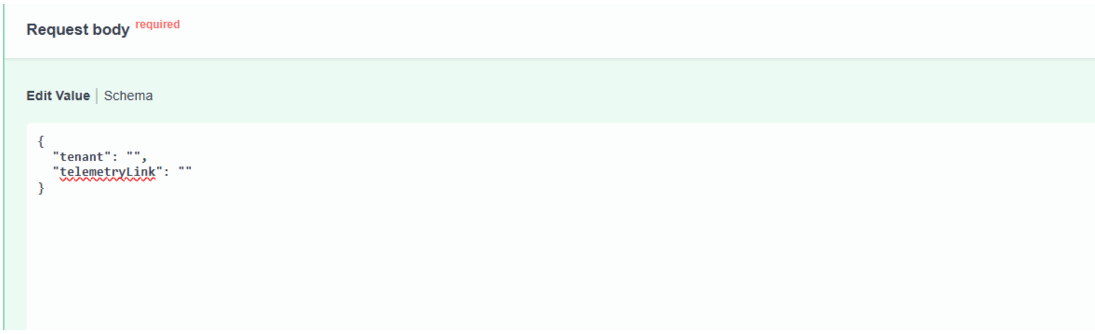
Output



Code	Details
200	Response body

```
[
  {
    "_id": 1752493266.0278847,
    "user": "admin",
    "lotNumber": "P73",
    "fileName": "Book1.csv",
    "status": "created",
    "TelemetryLinks": [
      {
        "tenant": "admin",
        "TelemetryLink": "None&_a=(query:(language:kuery,query:'user:%22admin%22%20and%20lotNumber%20:P73'))"
      }
    ],
    "CreatedDateTime": "2025-07-14T11:41:06.027000",
    "LastUpdatedDateTime": "2025-07-14T11:41:06.027000"
  },
  {
    "_id": 1752488772.5518885,
    "user": "admin",
    "lotNumber": "P72",
    "fileName": "train_sampled.csv",
    "status": "Completed",
    "TelemetryLinks": [
      {
        "tenant": "Privacy",
        "TelemetryLink": "http://vppcazaaa1923.az.ad.idemo-ppc.com:5601/app/dashboards#/view/0d3c6950-33dc-11f0-9859-3d1330e7ad01?_g=(refreshInterval:(paus
0),time:(from:now-30d%2Fd,to:now))&_a=(query:(language:kuery,query:'user:%22admin%22%20and%20lotNumber%20:P72'))"
      }
```

5.3: /questionnaire/telemetryUrlAdd- Adds telemetry URL configuration for monitoring.



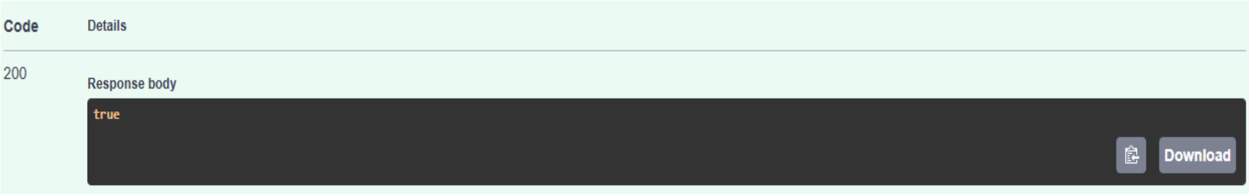
Request body required

```
{
  "tenant": "",
  "telemetryLink": ""
}
```

- **tenant :** tenant name (Privacy, Safety, FM-Moderation, Explainability)
- **telemetryLink :** Telemetry monitoring URL for the tenant.

Returns Configuration creation confirmation.

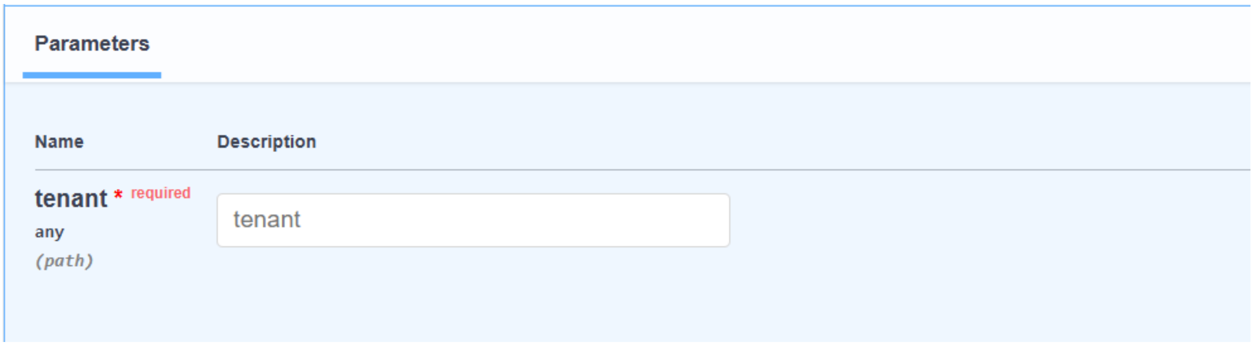
Output



Code	Details
200	Response body

```
true
```

5.4: /questionnaire/telemetryUrlGet/{tenant} - Retrieves telemetry URL for specific tenant.

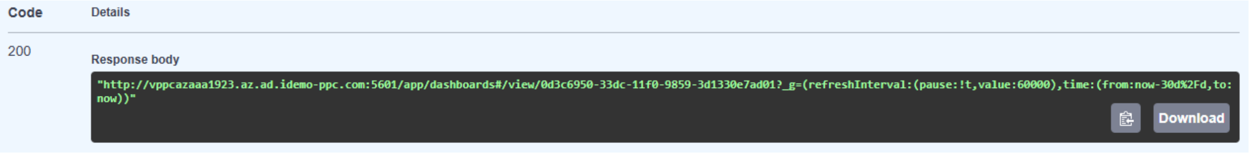


Name	Description
tenant * required any *(path)*	tenant

- **tenant :** Tenant name to get telemetry URL.

Returns telemetry URL configuration for the specified tenant.

Output



```
"http://vppcazaaa1923.az.ad.idemo-ppc.com:5601/app/dashboards#/view/0d3c6950-33dc-11f0-9859-3d1330e7ad01?_g=(refreshInterval:(pause:!t,value:60000),time:(from:now-30d%2Fd,to:
now))"
```

- ## Endpoints Usage Flow

Endpoint	Description
/questionnaire/allLotDetails/{userId}	Retrieves all analysis lot details and telemetry links for a specific user in reverse chronological order.
/questionnaire/telemetryUrlAdd	Adds or configures telemetry monitoring URL for a specific responsible AI tenant (Privacy, Safety, FM-Moderation, Explainability).
/questionnaire/telemetryUrlGet/{tenant}	Retrieves the configured telemetry monitoring URL for a specified tenant.
/questionnaire/workbench/uploadFile	Uploads a CSV file and performs responsible AI analysis across selected tenants, generating lot numbers and telemetry tracking.