Redact_phi

The `redact_phi` endpoint redacts all protected health information (PHI) in your text so that you can safely use the data within your organization or outside of it.

When you submit text to the ScienceIO API via this endpoint, our AI analyzes the query text, identifies all protected health information (PHI), and redacts it in a secure way. **Remember that only information *with identifiers tying it to an individual* is considered PHI.**

How to Call the Endpoint

For additional help with API calls, see Make an API Call (Python SDK) or Make an API Call (HTTP).

Web App

You can call the `redact_phi` endpoint from the **Analyze - Web App** without needing to use code. See Analyze - Web App for more details.

Python SDK

First, make sure you are using the latest version of the SDK; the endpoint will not work on versions prior to 2.0.0. You can use this command to upgrade:

```
pip install scienceio --upgrade
```

After initializing the ScienceIO client, use `scio.redact_phi()` to submit the request:

```
from scienceio import ScienceIO
scio = ScienceIO()

query_text = """Patient: John Doe
Address: 112 First Ave, New York, NY
Phone: 555-555-1212
Admission date: December 13, 2022
Diagnosis: UTI
Physician: Dr. Jane Smith
NPI: 1234567890
Physician number: 555-555-9876
Clinical note:
Mr. Doe is a 75-year-old male with a history of urinary tract infections.
He presented to the Pearson clinic today with symptoms of dysuria and frequency.
A urine culture was performed by Dr. Jones and showed significant growth of Escheri
The patient was started on a course of oral antibiotics and will follow up with
the clinic in one week for a repeat urine culture.
If no improvement, patient will be referred to St. Joseph Hospital."""

#call the redact_phi endpoint
response = scio.redact_phi(query_text)

print(response)
```

Optional:

Format the response to be more readable, as is seen in the sample JSON response on this page. Use the following code instead of `print(response)` :

```
# Format the JSON response and print
# Use instead of print(response)
import json
print(json.dumps(response, indent=2))
```

HTTP

After configuring your environment variables, you can submit a POST request to the `redact-phi` endpoint with your PHI text provided to the `input_text` keyword:

```
curl https://api.aws.science.io/v2/redact-phi \
  --request POST \
  --header "Content-type: application/json" \
  --header "x-api-id: $SCIENCEIO_KEY_ID" \
  --header "x-api-secret: $SCIENCEIO_KEY_SECRET" \
  --data '{ "input_text": "Patient: John Doe\nAddress: 112 First Ave, New York, NY\
```

> ### Note the use of `input_text` and not `text`.
>
> The use of `input_text` for this endpoint is part of a larger standardization of underlying schemas that is in progress.

Make sure your **GET** request also uses the `redact-phi` endpoint in line 1:

```
curl https://api.aws.science.io/v2/redact-phi/<REQUEST_ID> \
  --request GET \
  --header "x-api-id: $SCIENCEIO_KEY_ID" \
  --header "x-api-secret: $SCIENCEIO_KEY_SECRET"
```

For additional help with HTTP configuration, **POST** requests, or **GET** requests, see Make an API Call (HTTP).

JSON Response

The JSON Response includes `output_text` because the PHI has been redacted, so the original `input_text` has changed. It also includes the type of PHI found (`phi_type`), its location, the broader PHI category (`category`) that each piece of redacted PHI was assigned to, and a `score` (which is the confidence our API's model has in selecting the appropriate label; 1.0 is a perfect confidence score).

```
{
  "output_text": "Patient: [PATIENT]\nAddress: [STREET], [CITY], [STATE]\nPhone: [P
  "annotations": [
    {
      "labels": {
        "phi_type": {
          "label": "[PATIENT]",
          "score": 1.0
        },
        "category": {
          "label": "[PERSON]"
        }
      },
      "text": "[PATIENT]",
      "span": {
        "start": 9,
        "end": 18
      }
    },
    {
      "labels": {
        "phi_type": {
          "label": "[STREET]",
          "score": 0.998
        },
        "category": {
          "label": "[LOCATION]"
        }
      },
      "text": "[STREET]",
      "span": {
        "start": 28,
        "end": 36
      }
    },
    {
      "labels": {
        "phi_type": {
          "label": "[CITY]",
          "score": 0.94
        },
        "category": {
          "label": "[LOCATION]"
        }
      },
      "text": "[CITY]",
      "span": {
        "start": 38,
        "end": 44
      }
    },
    {
      "labels": {
        "phi_type": {
          "label": "[STATE]",
          "score": 0.999
        },
        "category": {
          "label": "[LOCATION]"
        }
      },
      "text": "[STATE]",
      "span": {
        "start": 46,
        "end": 53
      }
    },
    {
      "labels": {
        "phi_type": {
          "label": "[PHONE]",
          "score": 0.983
        },
        "category": {
          "label": "[CONTACT]"
        }
      },
      "text": "[PHONE]",
      "span": {
        "start": 61,
        "end": 68
      }
    },
    {
      "labels": {
        "phi_type": {
          "label": "[DATE]",
          "score": 1.0
        },
        "category": {
          "label": "[DATE]"
        }
      },
      "text": "[DATE]",
      "span": {
        "start": 85,
        "end": 91
      }
    },
    {
      "labels": {
        "phi_type": {
          "label": "[DOCTOR]",
          "score": 0.999
        },
        "category": {
          "label": "[PERSON]"
        }
      },
      "text": "[DOCTOR]",
      "span": {
        "start": 122,
        "end": 130
      }
    },
    {
      "labels": {
        "phi_type": {
          "label": "[MEDICALRECORD]",
          "score": 0.829
        },
        "category": {
          "label": "[IDENTIFIER]"
        }
```

```
  },
  "text": "[MEDICALRECORD]",
  "span": {
    "start": 136,
    "end": 151
  }
},
{
  "labels": {
    "phi_type": {
      "label": "[PHONE]",
      "score": 0.843
    },
    "category": {
      "label": "[CONTACT]"
    }
  },
  "text": "[PHONE]",
  "span": {
    "start": 170,
    "end": 177
  }
},
{
  "labels": {
    "phi_type": {
      "label": "[PATIENT]",
      "score": 1.0
    },
    "category": {
      "label": "[PERSON]"
    }
  },
  "text": "[PATIENT]",
  "span": {
    "start": 197,
    "end": 206
  }
},
{
  "labels": {
    "phi_type": {
      "label": "[AGE]",
      "score": 0.999
    },
    "category": {
      "label": "[DEMOGRAPHICS]"
    }
  },
  "text": "[AGE]",
  "span": {
    "start": 212,
    "end": 217
  }
},
{
  "labels": {
    "phi_type": {
      "label": "[HOSPITAL]",
      "score": 0.996
    },
    "category": {
          "label": "[INSTITUTION]"
        }
      },
      "text": "[HOSPITAL]",
      "span": {
        "start": 296,
        "end": 306
      }
    },
    {
      "labels": {
        "phi_type": {
          "label": "[DOCTOR]",
          "score": 0.999
        },
        "category": {
          "label": "[PERSON]"
        }
      },
      "text": "[DOCTOR]",
      "span": {
        "start": 390,
        "end": 398
      }
    },
    {
      "labels": {
        "phi_type": {
          "label": "[HOSPITAL]",
          "score": 0.999
        },
        "category": {
          "label": "[INSTITUTION]"
        }
      },
      "text": "[HOSPITAL]",
      "span": {
        "start": 628,
        "end": 638
      }
    }
  ]
}
```

Key:Value Pairs

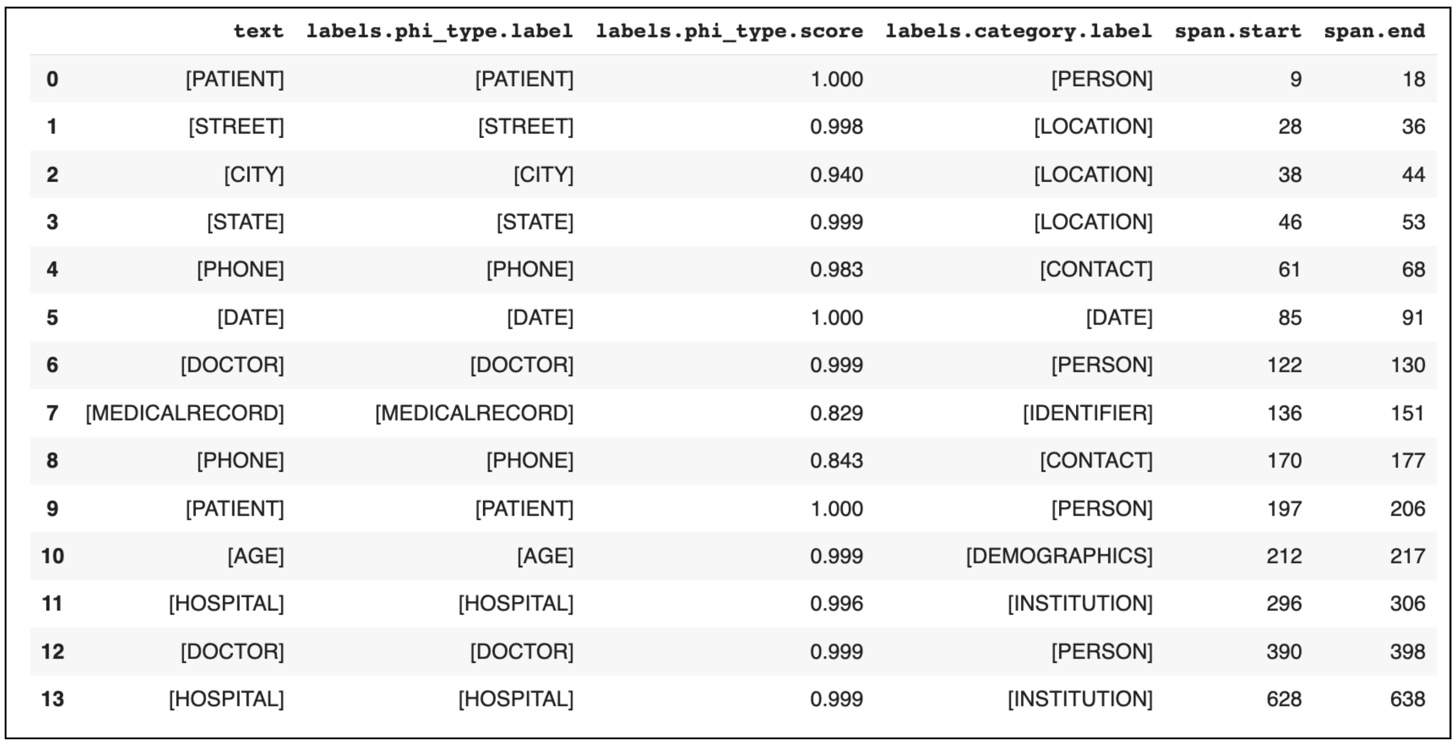
- Go to Schema: Redact_phi

View the Results in a Table

A table view can make it easier to interpret the results. Simply use pandas to create a DataFrame.

```
# Use pandas to view the results in a table.
import pandas as pd
df = pd.json_normalize(response['annotations'])
df
```

The resulting table looks like this:



	text	labels.phi_type.label	labels.phi_type.score	labels.category.label	span.start	span.end
0	[PATIENT]	[PATIENT]	1.000	[PERSON]	9	18
1	[STREET]	[STREET]	0.998	[LOCATION]	28	36
2	[CITY]	[CITY]	0.940	[LOCATION]	38	44
3	[STATE]	[STATE]	0.999	[LOCATION]	46	53
4	[PHONE]	[PHONE]	0.983	[CONTACT]	61	68
5	[DATE]	[DATE]	1.000	[DATE]	85	91
6	[DOCTOR]	[DOCTOR]	0.999	[PERSON]	122	130
7	[MEDICALRECORD]	[MEDICALRECORD]	0.829	[IDENTIFIER]	136	151
8	[PHONE]	[PHONE]	0.843	[CONTACT]	170	177
9	[PATIENT]	[PATIENT]	1.000	[PERSON]	197	206
10	[AGE]	[AGE]	0.999	[DEMOGRAPHICS]	212	217
11	[HOSPITAL]	[HOSPITAL]	0.996	[INSTITUTION]	296	306
12	[DOCTOR]	[DOCTOR]	0.999	[PERSON]	390	398
13	[HOSPITAL]	[HOSPITAL]	0.999	[INSTITUTION]	628	638

PHI Labels

PHI labels are broken down by `phi_type` (the PHI identifier assigned to the text) and `category` (the broader PHI category assigned to the text).

`phi_type`

The following PHI types are possible:

AGE
BIOID
CITY
COUNTRY
DATE
DEVICE
DOCTOR
EMAIL
FAX
HEALTHPLAN
HOSPITAL
IDNUM
LOCATION-OTHER
MEDICALRECORD
ORGANIZATION
PATIENT
PHONE
PROFESSION
STATE
STREET
URL
USERNAME
ZIP

`category`

The following categories are possible:

CONTACT
DATE
DEMOGRAPHICS
IDENTIFIER
INSTITUTION
LOCATION
ORGANIZATION
PERSON
WEBADDRESS

Mappings

Current mappings between each `category` (the broader PHI category assigned to the text) and its included `phi_type` (the PHI identifier assigned to the text) are as follows:

Category	PHI Type
CONTACT	EMAIL, FAX, PHONE, USERNAME
DATE	DATE
DEMOGRAPHICS	AGE, PROFESSION

Category	PHI Type
IDENTIFIER	BIOID, DEVICE, HEALTHPLAN, IDNUM, MEDICAL RECORD
INSTITUTION	ORGANIZATION
LOCATION	CITY, COUNTRY, STATE, STREET, ZIP, LOCATION-OTHER
ORGANIZATION	ORGANIZATION
PERSON	DOCTOR, PATIENT
WEBADDRESS	URL