Identify_phi

The `identify_phi` endpoint identifies all protected health information (PHI) in your text and categorizes it.

When you submit text to the ScienceIO API via this endpoint, our AI analyzes the query text, identifies and categorizes all protected health information (PHI), and returns a JSON response with that information. **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 `identify_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.identify_phi()` to submit the request:

```python
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 identify_phi endpoint
response = scio.identify_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)`:

```python
# 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 `identify-phi` endpoint with your PHI text provided to the `input_text` keyword:

```
curl https://api.aws.science.io/v2/identify-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 `identify-phi` endpoint in line 1:

```
curl https://api.aws.science.io/v2/identify-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 resulting JSON response includes each piece of PHI found, its location, the [type of PHI](#) it is (`phi_type`), and the broader PHI [category](#) (`category`) it was assigned to. It also includes a `score` , which is the confidence our API's model has in selecting the appropriate label (1.0 is a perfect confidence score).

```json
{
  "input_text": "Patient: John Doe\nAddress: 112 First Ave, New York, NY\nPhone: 55
  "annotations": [
    {
      "labels": {
        "phi_type": {
          "label": "[PATIENT]",
          "score": 1.0
        },
        "category": {
          "label": "[PERSON]"
        }
      },
      "text": "John Doe",
      "span": {
        "start": 9,
        "end": 17
      }
    },
    {
      "labels": {
        "phi_type": {
          "label": "[STREET]",
          "score": 0.998
        },
        "category": {
          "label": "[LOCATION]"
        }
      },
      "text": "112 First Ave",
      "span": {
        "start": 27,
        "end": 40
      }
    },
    {
      "labels": {
        "phi_type": {
          "label": "[CITY]",
          "score": 0.94
        },
        "category": {
          "label": "[LOCATION]"
        }
      },
      "text": "New York",
      "span": {
        "start": 42,
        "end": 50
      }
    },
    {
      "labels": {
        "phi_type": {
          "label": "[STATE]",
          "score": 0.999
        },
        "category": {
          "label": "[LOCATION]"
        }
      },
      "text": "NY",
```

```
      "span": {
        "start": 52,
        "end": 54
      }
    },
    {
      "labels": {
        "phi_type": {
          "label": "[PHONE]",
          "score": 0.983
        },
        "category": {
          "label": "[CONTACT]"
        }
      },
      "text": "555-555-1212",
      "span": {
        "start": 62,
        "end": 74
      }
    },
    {
      "labels": {
        "phi_type": {
          "label": "[DATE]",
          "score": 1.0
        },
        "category": {
          "label": "[DATE]"
        }
      },
      "text": "December 13, 2022",
      "span": {
        "start": 91,
        "end": 108
      }
    },
    {
      "labels": {
        "phi_type": {
          "label": "[DOCTOR]",
          "score": 0.999
        },
        "category": {
          "label": "[PERSON]"
        }
      },
      "text": "Jane Smith",
      "span": {
        "start": 139,
        "end": 149
      }
    },
    {
      "labels": {
        "phi_type": {
          "label": "[MEDICALRECORD]",
          "score": 0.829
        },
        "category": {
          "label": "[IDENTIFIER]"
        }
```

```
    },
    "text": "1234567890",
    "span": {
      "start": 155,
      "end": 165
    }
  },
  {
    "labels": {
      "phi_type": {
        "label": "[PHONE]",
        "score": 0.843
      },
      "category": {
        "label": "[CONTACT]"
      }
    },
    "text": "555-555-9876",
    "span": {
      "start": 184,
      "end": 196
    }
  },
  {
    "labels": {
      "phi_type": {
        "label": "[PATIENT]",
        "score": 1.0
      },
      "category": {
        "label": "[PERSON]"
      }
    },
    "text": "Doe",
    "span": {
      "start": 216,
      "end": 219
    }
  },
  {
    "labels": {
      "phi_type": {
        "label": "[AGE]",
        "score": 0.999
      },
      "category": {
        "label": "[DEMOGRAPHICS]"
      }
    },
    "text": "75",
    "span": {
      "start": 225,
      "end": 227
    }
  },
  {
    "labels": {
      "phi_type": {
        "label": "[HOSPITAL]",
        "score": 0.996
      },
      "category": {
```

```json
          "label": "[INSTITUTION]"
        }
      },
      "text": "Pearson clinic",
      "span": {
        "start": 306,
        "end": 320
      }
    },
    {
      "labels": {
        "phi_type": {
          "label": "[DOCTOR]",
          "score": 0.999
        },
        "category": {
          "label": "[PERSON]"
        }
      },
      "text": "Jones",
      "span": {
        "start": 404,
        "end": 409
      }
    },
    {
      "labels": {
        "phi_type": {
          "label": "[HOSPITAL]",
          "score": 0.999
        },
        "category": {
          "label": "[INSTITUTION]"
        }
      },
      "text": "St. Joseph Hospital",
      "span": {
        "start": 639,
        "end": 658
      }
    }
  ]
}
```

Key:Value Pairs

- Go to Schema: Identify_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.

```python
# 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	John Doe	[PATIENT]	1.000	[PERSON]	9	17
1	112 First Ave	[STREET]	0.998	[LOCATION]	27	40
2	New York	[CITY]	0.940	[LOCATION]	42	50
3	NY	[STATE]	0.999	[LOCATION]	52	54
4	555-555-1212	[PHONE]	0.983	[CONTACT]	62	74
5	December 13, 2022	[DATE]	1.000	[DATE]	91	108
6	Jane Smith	[DOCTOR]	0.999	[PERSON]	139	149
7	1234567890	[MEDICALRECORD]	0.829	[IDENTIFIER]	155	165
8	555-555-9876	[PHONE]	0.843	[CONTACT]	184	196
9	Doe	[PATIENT]	1.000	[PERSON]	216	219
10	75	[AGE]	0.999	[DEMOGRAPHICS]	225	227
11	Pearson clinic	[HOSPITAL]	0.996	[INSTITUTION]	306	320
12	Jones	[DOCTOR]	0.999	[PERSON]	404	409
13	St. Joseph Hospital	[HOSPITAL]	0.999	[INSTITUTION]	639	658

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

`category`

The following categories are possible:

CONTACT	LOCATION
DATE	ORGANIZATION
DEMOGRAPHICS	PERSON
IDENTIFIER	WEBADDRESS
INSTITUTION	

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

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