

# Annotate content

- Annotate
  - Introduction
  - Notation
  - Annotation request
  - Supported input formats
  - Supported output formats
  - Typical mention features
  - Examples
    - Posting plain text
    - Posting and receiving generic document
  - Simple mentions format
    - JSON

## Annotate

### Introduction

This section describes how to annotate documents with CES (Concept Extraction Service).

**Annotating** a document is the process of adding a set of meta data about words or phrases in an unstructured text.

A **mention** is a slice of text with attached meta data features. Mentions always have:

- **type** - the type of annotation, usually *Person*, *Organization* or *Location*, but other types can be returned
- **startOffset**, **endOffset** - 0-based offsets in the original text
- **features** map containing any number of properties/features depending on the mention origin

A **mention** is usually (but not necessarily) associated with a **concept** - a concept is a real-world entity that we recognized, a mention is a reference to that concept.

For example, annotating the text `Hello London` will yield a mention similar to the one below. The only mention has offsets within the original text and is associated with the concept <http://dbpedia.org/resource/London>

```
{
  "name": "London",
  "startOffset": 9,
  "endOffset": 15,
  "type": "Location",
  "features": {
    "inst": "http://dbpedia.org/resource/London",
    "class": "http://www.ontotext.com/proton/protontop#Location",
    "string": "London",
    "id": 392
  }
}
```

### Notation

All URLs in this document are of the form <http://worker-base/endpoint> where <http://worker-base> is the host:port/context of a deployed CES worker and [endpoint](http://worker-base/endpoint) is the specific worker call. For example, if you worker is deployed at <http://192.168.0.1/extractor-web> and this guide mentions <http://worker-base/extract> then the URL to query will be <http://192.168.0.1/extractor-web/extract>

### Annotation request

Annotation requests go to <http://worker-base/extract>. There are two ways to invoke annotation:

- **GET** request with a **url** parameter (e.g. <http://worker-base/extract?url=http://www.bbc.com/culture/story/20141020-the-plane-that-changed-air-travel>)

- **POST** request with meaningful **Content-type** header and body of the specified type  
The content type of the input document, whether specified by a URL or a request header, should be one of the supported formats.

It's also advisable to specify **Accept** header with the desired output mime type. The default will usually be `application/vnd.ontotext.ces+json`, see [output formats](#) for more.

## Supported input formats

- the standard web text formats such as `text/xml`, `text/html`, `text/plain`
- Ontotext's generic document schema in either JSON (`application/vnd.ontotext.ces.document+json`) or XML (`application/vnd.ontotext.ces.document+xml`)
- formats supported by [Apache Tika](#) should also work fine most of the time

## Supported output formats

✓ If **Accept** header is not specified, the simple mentions JSON format is returned (`application/vnd.ontotext.ces+json`)

- Ontotext's generic document schema in either JSON (`application/vnd.ontotext.ces.document+json`) or XML (`application/vnd.ontotext.ces.document+xml`)
- the "simple mentions" JSON format (`application/vnd.ontotext.ces` or `application/vnd.ontotext.ces+json`). Described in more details below

## Typical mention features

Mention features can vary wildly depending on the subsystem that generated the mention. Most mentions however will have

- **inst** - a URI for this mention's concept. This might "point out" to a concept database (freebase, dbpedia, etc) or be generated by machine learning subsystems
- **class** - generally related to the **type** of the mention, the **class** is a URI of class name within the concept database
- **string** - the slice of text associated with this mention, it is the text between **startOffset** and **endOffset**
- **id** - numeric id of the mention, unique within the document

Other returned features may include **confidence** (how sure the annotator feels about this mention), **ambiguityRank**, etc.

Other features are database and type dependant, for example locations such as *London* can have a **featClass**, **featCode**, **countryCode**, etc, giving more information about the concept

## Examples

### Posting plain text

Request:

```
POST /extractor-web/extract
Content-length: 14
Content-type: text/plain
Accept: application/vnd.ontotext.ces+json

Hello London!
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.ontotext.ces+json;charset=UTF-8

{
  "mentions": [{
    "name": "London",
    "startOffset": 6,
    "endOffset": 12,
    "type": "Location",
    "features": {
      "inst": "http://dbpedia.org/resource/London",
      "class": "http://www.ontotext.com/proton/protontop#Location",
      "string": "London",
      "id": 392
    }
  }]
}
```

### Posting and receiving generic document

Request:

```
POST /extractor-web/extract
Content-type: application/vnd.ontotext.ces.document+xml
Accept: application/vnd.ontotext.ces.document+json

<?xml version="1.0" encoding="utf-8"?>
<tns:document xmlns:tns="http://www.ontotext.com/DocumentSchema"
xmlns="http://www.w3.org/1999/xhtml" id="222-222">
  <tns:document-parts>
    <tns:document-part id="1" part="BODY">
      <tns:content>Hello, London!</tns:content>
    </tns:document-part>
  </tns:document-parts>
</tns:document>
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.ontotext.ces.document+json;charset=UTF-8

{
  "id": "222-222",
  "feature-set": [],
  "document-parts": {
    "feature-set": [{
      "name": {
        "type": "XS_STRING",
        "name": "encoding"
      },
      "value": {
        "type": "XS_STRING",
```

```

        "lang": null,
        "value": "UTF-8"
    }
  ],
  "document-part": [{
    "id": "1",
    "part": "BODY",
    "content": {
      "text": "Hello, London!",
      "node": [{
        "id": "0",
        "offset": 0
      }, {
        "id": "7",
        "offset": 7
      }, {
        "id": "13",
        "offset": 13
      }, {
        "id": "14",
        "offset": 14
      }
    ]
  },
  "feature-set": []
  ]
},
"annotation-sets": [{
  "name": "",
  "ref": null,
  "annotation": []
}, {
  "name": "Final",
  "ref": null,
  "annotation": [{
    "id": "51",
    "startnode": "7",
    "endnode": "13",
    "type": "Location",
    "status": "Suggested",
    "generated": false,
    "feature-set": [{
      "name": {
        "type": "XS_STRING",
        "name": "inst"
      },
      "value": {
        "type": "XS_STRING",
        "lang": null,
        "value": "http://dbpedia.org/resource/London"
      }
    }
  ], {
    "name": {
      "type": "XS_STRING",
      "name": "class"
    },
    "value": {
      "type": "XS_STRING",
      "lang": null,
      "value": "http://www.ontotext.com/proton/protontop#Location"
    }
  }
}

```

```
    }  
  }, {  
    "name": {  
      "type": "XS_STRING",  
      "name": "string"  
    },  
    "value": {  
      "type": "XS_STRING",  
      "lang": null,  
      "value": "London"  
    }  
  }, {  
    "name": {  
      "type": "XS_STRING",  
      "name": "id"  
    },  
    "value": {  
      "type": "XS_INTEGER",  
      "lang": null,  
      "value": "392"  
    }  
  }  
}]
```

```
}
  }]
}]
```

## Simple mentions format

### JSON

```
{
  "mentions": [{
    "name": "London", // the name of the mention, usually the string between
startOffset and endOffset
    "startOffset": 6, // start offset of the mention
    "endOffset": 12, // end offset of the mention
    "type": "Location", // the mention type
    "features": {
      "inst": "http://dbpedia.org/resource/London", // the instance id (URI) of
the concept associated with this mention
      "class": "http://www.ontotext.com/proton/protontop#Location", // the class
of the mentioned concept within the knowledge base
      "string": "London", // the sting between startOffset and endOffset
      "id": 392 // a unique id within the document
      // ... other features
    }
  }, {
    // ... more annotations
  }]
}
```