Named Entity Resolver

**Summary**

NER – Named Entity Resolver is a service that takes a text as input and identifies nouns (entities) such as Name, place, organization, date in it.

**API Documentation:**

Content for the documentation is available in **ner\_swagger.yaml** file.

**TIP:**  To get a better view of the documentation, copy the contents of the above yaml file in <https://editor.swagger.io/>. This will present the documentation in a neat UI.

NOTE: “Try it out” button in the swagger editor will not work as this document is just for information purpose.

**Brief description:**

NER is an API that takes an input text and identifies entities present in it. The service is capable of identifying the following list of entities or tags with the pretrained model:

|  |  |
| --- | --- |
| **Entity** | **Tag name** |
| Relative date | DATE\_REL |
| Approximate time | TIME\_APPROX |
| Email ID | EMAIL |
| URL | URL |
| Word contains only digit | DIGIT |
| US Social Security Numbers | SSN |
| Credit or debit card numbers | CARD\_NUMBER |
| Standard date formats and tokens like today, tomorrow and all | DATE |
| Standard time formats and tokens like morning, evening and all | TIME |
| Name of the people | PERSON |
| Name of the places | GEO |
| Name of the organization | ORG |
| Amount related | PRICE |
| Internet speed and data related | SPEED |
| US phone number patterns | PHONE\_NUMBER |
| US Zip codes | ZIPCODE |
| Other entities which didn’t fall under the aforesaid entity types and likely not to have any secured info | MISC |
| Other/Outside category. Mostly the words that are not categorized to any of the above will be categorized to this one | O |

Sample request/response for NER:

|  |  |
| --- | --- |
| Builtin Entities | |
| # | 1 |
| API | GET https://piiapi.genesysappliedresearch.com/api/v1/list\_tags |
| Purpose | To list down all the possible builtin entities |
| Authentication | Auth token to be included in request header (will be provided at the time of hackathon) |
| Payload sample | Not applicable |
| Response sample | {  "builtin\_entities": [  "DATE\_REL",  "TIME\_APPROX",  "EMAIL",  "URL",  "DIGIT",  "CARD\_NUMBER",  "SSN",  "PHONE\_NUMBER",  "ZIPCODE",  "DATE",  "TIME",  "PERSON",  "GEO",  "ORG",  "PRICE",  "SPEED",  "Misc",  "O"  ],  "bilstm\_tags": [  "PRICE",  "DATE",  "TIME\_APPROX",  "TIME",  "ORG",  "GEO",  "DATE\_REL",  "PERSON"  ]  } |

|  |  |
| --- | --- |
| Entity resolution | |
|  |  |
| API | POST https://piiapi.genesysappliedresearch.com/api/v1/resolve |
| Authentication | Auth token to be included in request header (will be provided at the time of hackathon) |
| Payload sample | {  "text": "Ram would like to book a flight from Chennai to Mumbai on 5th March"  } |
| Response sample | {  "masked": "<PERSON> likes to book a flight from <GEO> to <GEO> <DATE> <DATE> <DATE>",  "resolved\_result": [  {  "confidence": 0.8398494124412537,  "end": 3,  "start": 0,  "tag": "PERSON",  "text": "Ram",  "value": null  },  {  "confidence": 0.9999054869016012,  "end": 31,  "start": 4,  "tag": "O",  "text": "likes to book a flight from",  "value": null  },  {  "confidence": 0.9996507167816162,  "end": 39,  "start": 32,  "tag": "GEO",  "text": "Chennai",  "value": null  },  {  "confidence": 0.9997134804725647,  "end": 42,  "start": 40,  "tag": "O",  "text": "to",  "value": null  },  {  "confidence": 0.9994838237762451,  "end": 49,  "start": 43,  "tag": "GEO",  "text": "Mumbai",  "value": null  },  {  "confidence": 1.0,  "end": 62,  "start": 50,  "tag": "DATE",  "text": "on 5th March",  "value": {  "values": [  "2020-03-04 23:59:59.999996+00:00"  ]  }  }  ],  "result\_with\_probabilities": [  {  "confidence": 0.8398494124412537,  "end\_idx": 3,  "index": 0,  "start\_idx": 0,  "tag": "PERSON",  "word": "Ram"  },  {  "confidence": 0.9999785423278809,  "end\_idx": 9,  "index": 1,  "start\_idx": 4,  "tag": "O",  "word": "likes"  },  …  ],  "task\_id": "fac8c7b8-d45e-4a5a-bbbc-3d354cb3f1fc"  } |

**Things to note:**

The tags returned under bilstm\_tags (in the list\_tags API response) can only be used in expected\_tags of the resolve API. This will help in creating a bias in detecting the entities from the given text.

For example, in the below payload, you can set the expectation that the given text might contain ORG & PERSON to be specific so that system will try to map the words to these expected tags.

{

"text": "My name is Tony Bates and I work for Genesys",

"expected\_tags": [

"ORG",

"PERSON"

]

}