Examples of custom infotype detectors

ProtocolJava (#java)

The sample JSON objects on this page are a few examples of Cloud DLP custom infoType detectors (/dlp/docs/concepts-infotypes) that you can use during inspection and de-identification.

For more information about the different kinds of custom infoType detectors and how to create them, see Creating custom infoType detectors (/dlp/docs/creating-custom-infotypes).

Example `customInfoTypes` contents

The JSON given in the following table can be inserted into any inspection (/dlp/docs/creating-job-triggers) or de-identification (/dlp/docs/deidentify-sensitive-data) job definition, inside the `"customInfoTypes"` (/dlp/docs/reference/rest/v2/InspectConfig#CustomInfoType) attribute. The following complete JSON request, which sends a short string to Cloud DLP, includes a custom infoType detector definition that defines a medical record number format.

JSON input:

```
POST https://dlp.googleapis.com/v2/projects/[PROJECT_ID]/content:inspect?key={YOUR_A

{
  "item":{
    "value":"Patients MRN 444-5-22222"
  },
  "inspectConfig":{
    "customInfoTypes":[
      {
        "infoType":{
          "name":"C_MRN"
        },
        "regex":{
          "pattern":"[1-9]{3}-[1-9]{1}-[1-9]{5}"
        },
        "likelihood":"POSSIBLE"
      }
    ]
```

```
  }
}
```

Example description	Custom infoType detector
Generic alpha-numeric identifier	`{ "infoType":{ "name":"GENERIC_ID2" }, "regex":{ "pattern":"\\b(([0-9]|([a-zA-Z()]{1,2}[0-9]{1}))[0-9-\\[\\]:(), ._]+) }, "likelihood":"POSSIBLE" }`
Generic currency	`{ "infoType":{ "name":"GENERIC_CURRENCY" }, "regex":{ "pattern":"((\\p{Sc}{1})( ){0,1}[0-9,()]+)(.){0,1}[0-9]{0,}" }, "likelihood":"VERY_LIKELY" }`
Generic percentage	`{ "infoType":{ "name":"GENERIC_PERCENT" }, "regex":{ "pattern":"\\b([0-9,.()]+)( ){0,1}(%){1}" }, "likelihood":"VERY_LIKELY" }`
Generic unit of measurement	`{ "infoType":{ "name":"GENERIC_MEASURE" }, "regex":{ "pattern":"(?i)([0-9])([0-9,.]+)( ){0,1}(?i)(°C|°F|K|°Ré|°N|°Ra|m³|dm }, "likelihood":"VERY_LIKELY" }`

Generic Requests for Comments (RFC) identifier	<pre>{
	  "infoType":{
	  "name":"GENERIC_RFC"
	  },
	  "regex":{
	  "pattern":"\\b(?i)(rfc)( ){0,1}(20|768|783|791|792|793|826|854|855|86
	  },
	  "likelihood":"LIKELY"
	}</pre>

Generic RJ45 network connector identifier	<pre>{
	  "infoType":{
	  "name":"GENERIC_RJ_NETWORK"
	  },
	  "regex":{
	  "pattern":"\\b(?i)(rj)(-){0,1}(?i)(A1X|A2X|A3X|2MB|11|12|13|14|15C|18
	  },
	  "likelihood":"VERY_LIKELY"
	}</pre>

Generic data size	<pre>{
	  "infoType":{
	  "name":"GENERIC_DATA_SIZE"
	  },
	  "regex":{
	  "pattern":"\\b([0-9,.]+)(?i)(byte|Kilobyte|KiB|Kilobit|kbit|bit|Mega
	  },
	  "likelihood":"VERY_LIKELY"
	}</pre>

Numerical identifier	<pre>{
	  "infoType":{
	  "name":"GENERIC_ID1"
	  },
	  "regex":{
	  "pattern":"\\w*[0-9][0-9-()\\[\\].:/]+[0-9]\\w*"
	  },
	  "likelihood":"POSSIBLE"
	}</pre>

Except as otherwise noted, the content of this page is licensed under the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the Apache 2.0 License (https://www.apache.org/licenses/LICENSE-2.0). For details, see the Google Developers Site Policies (https://developers.google.com/site-policies). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2020-08-13 UTC.