## Data Pools Services:

1. getDatapool Service:
   * Description: Get the availables data pool in the system:
   * Parameter:
     + numResults 🡪 Limitation
   * Example:
     + Get the availables data pools:
       - …/CaptureREST/rest/datapool?numResults=3
       - Resultado:datapool.xml

## Aggregated Services

Below see the aggregated services description for consuming the analyzed data.

1. getTagCloud Service.
   * + Description: Get the most repeated words for a concrete(s) data pool(s) for a concrete time and with a concrete periodicity
     + Parameters:
       - dataPoolIdL 🡪 Desired data pool Id. To get for multiples data pools include additional “dataPoolIdL” parameters.
       - date 🡪 Desired date. Following pattern: “Thu, 19 May 2015 00:00:00 GMT” codificado en la url
       - per 🡪 Periodicity (frequency measurement): {minutes1, hourly, daily}. Nowadays for search just hourly and daily. For streaming minutes1.
       - Result: It will get as much result as data pools desired. For each data pool will get the 20 most repeated words (with the number of word apparitions).
     + Example:
       - Get the most repeated words for the data pool “08593b4” and “1ddfbd05” for the whole day 20 July 2016.
         * …/AnalyticREST/rest/analytic/tagCloud?dataPoolIdL=08593b49&dataPoolIdL=1ddfbd05&date=Wed%2C%2027%20Jul%202016%2000%3A00%3A00%20GMT&per=daily
         * Resultado:tagCloud1.xml
       - Get the most repeated words for the data pool “[4ce776b8](http://154.48.153.6/CaptureREST/detailsDataPool?id=4ce776b8)” for the whole minute Sun, 10 Jul 2016 19:18:00 GMT.
         * …/AnalyticREST/rest/analytic/tagCloud?dataPoolIdL=4ce776b8&date=Sun%2C%2010%20Jul%202016%2019%3A18%3A00%20GMT%0A&per=minutes1
         * Resultado:tagCloud2.xml
2. getTagCloudAccumulated Service.
   * + Description: Get the most repeated words for a concrete(s) data pool(s) for a specific period of time and with a concrete periodicity
     + Parameters:
       - dataPoolIdL 🡪 Desired data pool Id. To get for multiples data pools include additional “dataPoolIdL” parameters.
       - initDate 🡪 Init date from the period of time. Following pattern: “Thu, 19 May 2015 00:00:00 GMT”
       - endDate 🡪 End date from the period of time. Following pattern: “Thu, 19 May 2015 00:00:00 GMT”
       - per 🡪 Periodicity (frequency measurement): {minutes1, hourly, daily}. Nowadays for search just hourly and daily. For streaming minutes1.
       - top (optional) 🡪 Return top N words. Default 25.
       - Result: It will get as much result as data pools desired. For each data pool will get the 20 most repeated words (with the number of word apparitions).
     + Example:
       - Get the most repeated words for the data pool “08593b4” and “1ddfbd05” from 20 July 2016 to 22 July 2016 and daily.
         * …/AnalyticREST/rest/analytic/tagCloudAccumulated?dataPoolIdL=08593b49&dataPoolIdL=1ddfbd05&initDate=Wed%2C%2027%20Jul%202016%2000%3A00%3A00%20GMT&endDate=Wed%2C%2027%20Jul%202016%2000%3A00%3A00%20GMT&per=daily
         * Resultado:tagCloud1.xml
       - Get the most repeated words for the data pool “[4ce776b8](http://154.48.153.6/CaptureREST/detailsDataPool?id=4ce776b8)” from Sun, 10 Jul 2016 19:18:00 GMT to Sun, 10 Jul 2016 19:35:00 GMT and by minute1 and top 20.
         * …/AnalyticREST/rest/analytic/tagCloudAccumulated?dataPoolIdL=4ce776b8&initDate=Sun%2C%2010%20Jul%202016%2019%3A18%3A00%20GMT%0A&endDate=Sun%2C%2010%20Jul%202016%2019%3A35%3A00%20GMT%0A&per=minutes1&top=20
         * Resultado:tagCloud2.xml
   1. getVolume Service.
      * Description: Get the number of tweets for a concrete(s) data pool(s) for a specific period of time and with a concrete periodicity
      * Parameters:
        + dataPoolIdL 🡪 Desired data pool Id. To get for multiples data pools include additional “dataPoolIdL” parameters.
        + initDate 🡪 Init date from the period of time. Following pattern: “Thu, 19 May 2015 00:00:00 GMT”
        + endDate 🡪 End date from the period of time. Following pattern: “Thu, 19 May 2015 00:00:00 GMT” codificado en la url
        + per 🡪 Periodicity (frequency measurement): {seconds15, minutes1, minutes5, minutes15, minutes30, hourly, daily}. Nowadays for search just hourly and daily. For streaming minutes1.
        + Result: A time series with as much data points as slots between initDate y endDate for the desired periodicity. (Ej.: 8 days, hourly will get 192 data point). For each data point, will get as much result as data pools desired. For each data pool will get the number of tweets.
      * Example:
        + Retrieve number of tweets for following data pools: “08593b49”. 29805cac7” and “a60723a6”. Between 00:00:00 (GMT/UTC) 11th July to 23:59:59 (GMT/UTC) 18th July with a “hourly” periodicity.
          - …AnalyticREST/rest/analytic/newVolume?dataPoolIdL=08593b49&dataPoolIdL=9805cac7&dataPoolIdL=a60723a6&initDate=Sun%20Jul%2010%202016%2022%3A00%3A00%20GMT&endDate=Mon%20Jul%2018%202016%2021%3A59%3A59%20GMT&per=hourly
          - Resultado: newVolume.xml
   2. getSentiment Service.
      * Description: Get the number of positive and negative tweets for a concrete(s) data pool(s) for a specific period of time and with a concrete periodicity
      * Parameters:
        + dataPoolIdL 🡪 Desired data pool Id. To get for multiples data pools include additional “dataPoolIdL” parameters.
        + initDate 🡪 Init date from the period of time. Following pattern: “Thu, 19 May 2015 00:00:00 GMT” codificado en la url
        + endDate 🡪 End date from the period of time. Following pattern: “Thu, 19 May 2015 00:00:00 GMT” codificado en la url
        + per 🡪 Periodicity (frequency measurement): {seconds15, minutes1, minutes5, minutes15, minutes30, hourly, daily}. Nowadays for search just hourly and daily. For streaming minutes1.
        + Result: A time series with as much data points as slots between initDate y endDate for the desired periodicity. (Ej.: 8 days, hourly will get 192 data point). For each data point, will get as much result as data pools desired. For each data pool will get the number of positive tweets (position 0) and negative tweets (position 1).
      * Examples:
        + Retrieve positive and negative number of tweets for the data pool “08593b49 between 00:00:00 (GMT/UTC) 10th July to 23:59:59 (GMT/UTC) 10th July with a “hourly” periodicity
          - …/AnalyticREST/rest/analytic/sentiment?dataPoolIdL=08593b49&initDate=Sun%2C%2010%20Jul%202016%2000%3A00%3A00%20GMT%0A&endDate=Sun%2C%2010%20Jul%202016%2023%3A59%3A59%20GMT%0A&per=daily
          - Resultado:sentiment.xml
   3. getSentimentDegree
      * Description: Get the index sentiment (total positive - total negatives/total positives + total negatives) for a concrete(s) data pool(s) for a specific period of time and with a concrete periodicity
      * Parameters:
        + dataPoolIdL 🡪 Desired data pool Id. To get for multiples data pools include additional “dataPoolIdL” parameters.
        + initDate 🡪 Init date from the period of time. Following pattern: “Thu, 19 May 2015 00:00:00 GMT” codificado en la url
        + endDate 🡪 End date from the period of time. Following pattern: “Thu, 19 May 2015 00:00:00 GMT” codificado en la url
        + per 🡪 Periodicity (frequency measurement): {seconds15, minutes1, minutes5, minutes15, minutes30, hourly, daily}. Nowadays for search just hourly and daily. For streaming minutes1.
        + Results: A single data point (set to endDate) with as much result as data pools desired. For each data pool will get its corresponding sentiment degree.
      * Examples:
        + Retrieve the sentiment degree for following data pools: “08593b49”, “9805cac7”, “4a673994”, “cb52dc79” and “cb52dc79” between 00:00:00 (GMT/UTC) 10th July to 23:59:59 (GMT/UTC) 10th July.
          - …/AnalyticREST/rest/analytic/sentimentDegree?dataPoolIdL=08593b49&dataPoolIdL=9805cac7&dataPoolIdL=4a673994&dataPoolIdL=cb52dc79&dataPoolIdL=1ddfbd05&initDate=Sun%2C%2010%20Jul%202016%2000%3A00%3A00%20GMT%0A&endDate=Sun%2C%2010%20Jul%202016%2023%3A59%3A59%20GMT%0A&per=daily
          - Resultado: sentimentDegree.xml
   4. getVolumePer:
      * Description: Get the popularity (based on the number of tweets and expressed in %) for a concrete data pool from a set of data pools in a specific period of time and with a concrete periodicity. Popularity shows the percentage of tweets from the candidate data pool with respect the total number of tweets.
      * Parameters:
        + dpCandidateIdL 🡪 Desired data pool Id.
        + dataPoolIdL 🡪 Set of data pools to measure with. Each data pool id will be in a “dataPoolIdL” parameter.
        + initDate 🡪 Init date from the period of time. Following pattern: “Thu, 19 May 2015 00:00:00 GMT” codificado en la url
        + endDate 🡪 End date from the period of time. Following pattern: “Thu, 19 May 2015 00:00:00 GMT” codificado en la url
        + per 🡪 Periodicity (frequency measurement): {seconds15, minutes1, minutes5, minutes15, minutes30, hourly, daily}. Nowadays for search just hourly and daily. For streaming minutes1.
        + Result: Results: A single data point (set to endDate) showing a single data pool (that indicated in dpCandidateIdL) with its popularity.
      * Example:
        + Retrieve the popularity for the data pool “08593b49” from a set of following data channels: “08593b49”, “9805cac7” and “a60723a6” between 00:00:00 (GMT/UTC) July 11th to 23:59:59 (GMT/UTC) 18th July
          - …/AnalyticREST/rest/analytic/overalPopularity?dataPoolIdL=08593b49&dataPoolIdL=9805cac7&dataPoolIdL=a60723a6&dpCandidateIdL=08593b49&initDate=Sun%20Jul%2010%202016%2022%3A00%3A00%20GMT&endDate=Mon%20Jul%2018%202016%2021%3A59%3A59%20GMT&per=daily
          - Resultado: overalPopularity.xml
   5. getTopVolume
      * Description: Get the N data pools with higher number of tweets from a set of data pools in a specific period of time and with a concrete periodicity.
      * Parameters:
        + dataPoolIdL 🡪 Set of data pools to include in the computation. Each data pool id will be in a “dataPoolIdL” parameter
        + initDate 🡪 Init date from the period of time. Following pattern: “Thu, 19 May 2015 00:00:00 GMT” codificado en la url
        + endDate 🡪 End date from the period of time. Following pattern: “Thu, 19 May 2015 00:00:00 GMT” codificado en la url
        + per 🡪 Periodicity (frequency measurement): {seconds15, minutes1, minutes5, minutes15, minutes30, hourly, daily}. Nowadays for search just hourly and daily. For streaming minutes1.
        + Nor 🡪 number of result (N). By default is 3.
        + Results: A single data point (set to endDate) with the top N data pools
      * Example:
        + Get the 2 data pools with higher volume from a set of following data pools: “08593b49”, “9805cac7”, “4a673994”, “cb52dc79” and “cb52dc79” between 00:00:00 (GMT/UTC) 10th July to 23:59:59 (GMT/UTC) 10th July
          - …/AnalyticREST/rest/analytic/populars?dataPoolIdL=08593b49&dataPoolIdL=9805cac7&dataPoolIdL=4a673994&dataPoolIdL=cb52dc79&dataPoolIdL=1ddfbd05&initDate=Sun%2C%2010%20Jul%202016%2000%3A00%3A00%20GMT%0A&endDate=Sun%2C%2010%20Jul%202016%2023%3A59%3A59%20GMT%0A&per=daily&nor=2
          - Populars: populars.xml

## Tweets Services:

1. Tweets

* Description Get a list of tweets for a concrete(s) data pool(s) for a specific period of time and with a concrete periodicity:
* Parameters:
  + id => datapool ID.
  + filterExpression (optional) => query filters
    - Start with
      * (
    - Date => optional
      * createdAt:[2016-08-12T00:00:00.000Z TO 2016-08-13T00:00:00.000Z]
    - Text search with ponderation error (España - Espana) => optional
      * AND text:España~0.75
    - Only show original tweets (without retweets) => optional
      * AND originalTweetId:""
    - Only tweets with geo parameters (+- 1% of total volume) => optional
      * AND latitude:["" TO \*]
    - End with
      * )
    - Examples:
      * filterExpression=(createdAt:[2016-08-12T00:00:00.000Z TO 2016-08-13T00:00:00.000Z] AND text:España~0.75 AND originalTweetId:"" AND latitude:["" TO \*])
      * filterExpression=(createdAt:[2016-08-12T00:00:00.000Z TO 2016-08-13T00:00:00.000Z] AND originalTweetId:"")
  + sorter (optional) => default “retweetCount”.
  + mode (optional) => default “desc”.
  + fields (optional) => name of fields to return.
  + numResults (optional) => volumen to return, default “100”.
  + page (opcional) => page to return, default “1”.
  + Examples:
    - …/CaptureREST/rest/datachannel/data?id=2c6590ac& filterExpression=(createdAt:[2016-08-12T00:00:00.000Z TO 2016-08-13T00:00:00.000Z] AND originalTweetId:"")
    - …/CaptureREST/rest/datachannel/data?id=2c6590ac& filterExpression=(createdAt:[2016-08-12T00:00:00.000Z TO 2016-08-13T00:00:00.000Z] AND originalTweetId:"")&sorter=retweetCount&mode=desc&fields=text&numResults=10&page=1