List of folders in the workspace: "list_folders()" does not need any arguments. The command returns the list of folders available in the workspace including the name and the folder ID.

List of folders where the user can create records: "list_permission_folders()" does not need any arguments. The command returns the list of folders in the workspace where the user can create records including the name and the folder ID.

List of fields in the folder: “list_fields(*folderID*)” takes one argument, the folder ID. It will return the list of fields in the specified folder.

Search for records: "query_search(*folderID, search_query, field, result_limit*)" takes four arguments: 1) the folder ID 2) the search term or query (e.g. John) 3) the field for performing the search (e.g. First name) 4) maximum number of results you would like to be returned. For performing searches based on the record ID, in the second argument you need to introduce the record ID and in the third argument you need to use “Name”.

Find new records in a folder: “recent_records(*folderID, date_updated, result_limit*)” takes three arguments: 1) the folder ID 2) date since the update should be considered in the format YYYY_MM_DD 3) maximum number of results you would like to be returned. Depending on the defined limit, a list with the most recently created records will be returned.

Find recently created comments: "recent_comments(*folderID, date_updated, result_limit*)" takes three arguments: 1) the folder DI 2) date since the comments created should be considered in the format YYYY_MM_DD 3) maximum number of results you would like to be returned. Depending on the defined limit, a list with the most recently created comments will be returned.

Find recently created call logs: "recent_call_logs(*folderID, date, result_limit*)" takes three arguments: 1) the folder DI 2) date since the call log added should be considered in the format YYYY_MM_DD 3) maximum number of results you would like to be returned. Depending on the defined limit, a list with the most recently logged calls will be returned.

Find recently uploaded files to google drive: "recent_drive_files(*folderID, date, result_limit*)" takes three arguments: 1) the folder DI 2) date since the file uploaded should be considered in the format YYYY_MM_DD 3) maximum number of results you would like to be returned. Depending on the defined limit, a list with the most recently uploaded files will be returned.

Find records that have been updated recently: "recently_updated(*folderID, updated_fields, date_updated, result_limit*)" takes four arguments: 1) folder ID 2) the field or fields that have been updated 3) date since the update should be considered in the format YYYY_MM_DD 4) maximum number of results you would like to be returned. It will return the records whose specified fields have been updated from the date specified as well

Recent record changes: “recent_updates(*folderID, updated_fields, date_updated, result_limit, recordID*)” takes five arguments: 1) folder ID 2) the field or fields that have been updated 3) date since the update should be considered in the format YYYY_MM_DD 4) maximum number of results you would like to be returned 5) record ID. Records that have been updated since the indicated date and for the indicated fields will be returned. The record ID can be specified and then only the changes in the indicated record will be returned, otherwise “0” can be specified so all the records in the specified folder can be returned.

Create a new comment in a record: “create_comment(*recordID, comment_data*)” takes two arguments 1) record ID and 2) comment. The comment should be provided as a set specifying the word “text” followed by the comment itself. A new comment will be created in the specified record.

```
data={
        "text": "It's a comment"
        }
```

Create a new call log: “create_call_log(*recordID, call_log_data*)” takes two arguments 1) record ID and 2) data for the call log. The data should be provided as a set specifying the word “text” followed by the summary of the call; “time” including the date and time in the format YYYY-MM-DDTHH:MM:SSZ and the "duration" of the call. A new call log will be created in the specified record. Sample data:

```
data={
        “text”:"this is a new call log",
        "time": "2021-06-01T16:35:00Z",
        "duration": 4
        }
```

Update a record in a folder: "updte_record(*recordID, new_data*)" takes two arguments: 1) The record ID 2) The data to be updated. The data to be updated should be provided as a set, in the case of fields with multiple values the data should be provided as an array. The fields to be updated should be preceded by “fieldAction” in the set. There are three options for this command as for the information to be updated: 1) overwrite, should be specified as true in the set 2) replace should indicate the current value as well as the new one. Data example:

```
data={
        "fieldActions": {
                "Name": {"overwrite": "true","add": "John Doe"},
                "Phone": {"remove": "111","add": "777"},
                "Email": {"remove": ["user1@example.com"],"add": ["user2@example.com"]}
                }
        }
```

Create a new record in a folder: "create_record(*folderID, record_data*)" takes four arguments: 1) The folder ID 2) The data for the new record. The folder ID can be retrieved by running the command list_folders. The data for the new record can contain fields with multiple values such as for example email or telephone and should be provided as a set including the name of the field and the data. In those cases when the field can have multiple values, the information should be input as an array even if the new record will have only one value. Sample data (“Other Emails” corresponds to a multiple value field, while “Email” instead corresponds to a single value field):

```
data={
        "timeZone": "Europe/London",
                "Fields":{
                        "Name":"John Doe",
                        "Email":"name@example.com",
                        "Address":"London",
                        "Phone":"777",
                        "Other Emails":["example@gmail.com","email2@gmail.com"]
                        }
        }
```

Link email conversation/thread to a record: “link_thread(*recordID, threadID*)” takes two arguments 1) record ID 2) the email thread ID. The email thread ID should be included in the request as a set preceded by “gmailThreadId”. The command will link the indicated email thread and the indicated record. Sample data:

```
data={
        "gmailThreadId": "1234567890abcdef"
        }
```