

企業向けソーシャルプラットフォーム WebAPI リファレンスマニュアル

第 5.0.0 版

NEC ソリューションイノベータ株式会社

改版履歴

版数	発行年月日	改版内容
初版	2014. 06. 09	新規
2. 0 版	2014. 12. 05	V2. 0 対応
2. 1 版	2014. 12. 15	記述不備の修正、メッセージとインボックスの削除 API を追加
2. 2 版	2014. 12. 24	記述不備の修正
2. 3 版	2015. 07. 29	プロフィール情報のメールアドレスの項目について追記
2. 4 版	2015. 08. 25	新着メッセージ取得 API を追加
2. 5 版	2015. 08. 31	タスクの追加・更新、Inbox の追加 API を追加
3. 0 版	2015. 09. 21	ログアウト API を追加、添付ファイルアップロード/ダウンロード API を追加 (V3. 0)
3. 1 版	2016. 01. 13	記述不備の修正
4. 0 版	2016. 05. 31	V4. 0 対応
4. 1 版	2017. 04. 27	未記述の API を追加
4. 2 版	2017. 06. 07	Filedownload API エラー時の記述更新
5. 0. 0 版	2018. 03. 06	V5. 0. 0 対応

改版による修正

バージョン	修正内容
4.2 版 ↓ 5.0.0 版	<p>3.1.1 ログイン API のレスポンスに tenantInfo を追加</p> <p>3.2.1 ログインユーザ情報取得のパラメータに extras を追加</p> <p>3.2.2 ログインユーザのプロフィール変更のパラメータに extras を追加</p> <p>3.2.2 ログインユーザのプロフィール更新 API の異常系処理として LDAP データ更新不可の場合を記載</p> <p>3.2.3 ログインユーザのパスワード更新 API の異常系処理として LDAP データ更新不可の場合、ポリシー不正の場合を記載</p> <p>3.9.1 チャットルーム作成のパラメータ parentRoomId を追加</p> <p>3.9.2 チャットルーム一覧取得に parentRoomId を追加</p> <p>3.10.1 コミュニティの作成のパラメータ MemberEntryType の指定を修正</p> <p>3.10.2 参加コミュニティ一覧の取得のレスポンスパラメータ MemberEntryType の指定を修正</p> <p>3.10.3 コミュニティの詳細情報取得のレスポンスパラメータ MemberEntryType をの指定を修正</p> <p>3.10.7 コミュニティ情報更新パラメータ MemberEntryType の指定を修正</p> <p>3.11.4 メッセージ検索を追加</p> <p>3.11.5 検索リクエスト Json ブロック解説を追加</p> <p>3.11.6 メッセージ応答を追加</p>

互換性履歴

バージョン	互換性
V4.0 ↓ V5.0.0	変更なし。 V4.0 と V5.0.0 で互換は保たれている。

目次

1	はじめに	7
1.1	WebAPI でできること	7
1.2	注意事項	7
2	WebAPI 概要	8
2.1	WebAPI アクセス先	8
2.2	API 一覧	8
2.3	凡例	10
3	API 詳細	11
3.1	認証	11
3.1.1	ログイン	11
3.1.2	ログアウト	13
3.2	ユーザ情報取得更新	14
3.2.1	ログインユーザ情報取得	14
3.2.2	ログインユーザのプロフィール変更	16
3.2.3	ログインユーザのパスワード更新	19
3.2.4	プレゼンス変更	20
3.2.5	ユーザ情報の取得	21
3.3	コンタクトリスト	23
3.3.1	ログインユーザのコンタクトリスト取得	23
3.3.2	コンタクトリストメンバ追加	24
3.3.3	コンタクトリストメンバ削除	25
3.4	フィード（取得、投稿）	27
3.4.1	フィード取得	27
3.4.2	フィード投稿	30
3.4.3	ウォッチフィード（取得）	31
3.5	チャット（取得、投稿）	36
3.5.1	チャット取得	36
3.5.2	チャット投稿	39
3.6	タスク（取得、追加、変更）	40
3.6.1	タスク取得	40
3.6.2	タスク追加	43
3.6.3	タスク更新	47
3.6.4	タスク削除	50
3.6.5	タスク催促要求	51
3.6.6	タスク催促解除要求	52
3.7	インボックス（取得、追加）	53
3.7.1	インボックス取得	53
3.7.2	インボックス追加	57
3.7.3	インボックスの削除	60
3.8	Good Job	61
3.8.1	Good Job 要求	61
3.8.2	Good job したユーザ取得	62
3.9	グループチャットチャットルーム	63
3.9.1	チャットルーム作成	63

3.9.2	チャットルーム一覧取得	66
3.9.3	チャットのルーム情報取得	68
3.9.4	チャットのメンバ追加	70
3.9.5	チャットのメンバ削除	72
3.9.6	チャットのルーム情報更新	74
3.9.7	グループチャットメッセージ取得	75
3.9.8	グループチャットメッセージ投稿	78
3.10	プロジェクト	81
3.10.1	プロジェクトの作成	81
3.10.2	参加プロジェクト一覧の取得	83
3.10.3	プロジェクトの詳細情報取得	85
3.10.4	プロジェクトメンバ情報取得	86
3.10.5	プロジェクトメンバ追加	88
3.10.6	プロジェクトメンバ削除	90
3.10.7	プロジェクト情報更新	91
3.10.8	プロジェクトオーナー変更	93
3.10.9	プロジェクトフィードメッセージ取得	95
3.10.10	プロジェクトフィードメッセージ投稿	98
3.11	メッセージ	100
3.11.1	新着メッセージ取得	100
3.11.2	メッセージの削除（フィード、チャット、グループチャット、プロジェクトフィード）	105
3.11.3	メッセージの検索	106
3.11.4	検索リクエストサンプルと Json ブロック解説	107
3.11.5	メッセージの応答	109
3.12	スレッド	110
3.12.1	スレッドメッセージ一覧取得（フィード、グループチャット、プロジェクトフィード）	110
3.13	メッセージの既読	111
3.13.1	既読情報の取得	111
3.13.2	既読要求	112
3.14	スマートデバイス向け端末情報（Android : Registration ID / iOS : Device Token）	113
3.14.1	端末情報の登録	113
3.14.2	端末情報の削除	114
4	添付ファイル API	115
4.1	添付ファイルアップロード	115
4.1.1	アクセス先	115
4.1.2	リクエスト	115
4.1.3	レスポンス	115
4.2	添付ファイルダウンロード	116
4.2.1	アクセス先	116
4.2.2	リクエスト	116
4.2.3	レスポンス	116
5	短縮 URL	117
5.1	短縮 URL リダイレクト	117
5.1.1	アクセス先	117
5.1.2	リクエスト	117
5.1.3	レスポンス	117

1 はじめに

この文書は「企業向けソーシャルプラットフォーム」の WebAPI について記載したものです。

1.1 WebAPI でできること

- フィードやチャットにメッセージを投稿することなど、企業向けソーシャルプラットフォームに対する操作を行うことができますので、他システムからの情報を取り込むことができます。
- フィードやチャットのメッセージを参照することができます。

1.2 注意事項

本記載の情報はバージョンアップ等により、予告なく変更される可能性があります。

2 WebAPI 概要

2.1 WebAPI アクセス先

以下の URL に POST メソッドでアクセスする。

URL でテナント名を指定するパターンとしないパターンがあるが、どちらでも API は動作する。

- `http(s)://<cubee サイトのホスト名 or IP アドレス>/<cubee_root>/asynchronous/`
 - テナント名を指定しない URL の記述
この URL を使用する場合、ログイン API 実行時には `tenantName` パラメータの指定が必要。
- `http(s)://<cubee サイトのホスト名 or IP アドレス>/<cubee_root>/t/<テナント名>/asynchronous/`
 - テナント名を指定する URL の記述

※https は SSL をサポートしている時のみ

API のデータを POST メソッドのパラメータに設定して送信する

2.2 API 一覧

- WebAPI
 - 認証
 - ◇ ログイン
 - ◇ ログアウト
 - ユーザ情報取得/更新
 - ◇ ログインユーザ情報取得
 - ◇ ログインユーザのプロフィール変更
 - ◇ ログインユーザのパスワード変更
 - ◇ ログインユーザのプレゼンス変更
 - ◇ ユーザ情報の検索
 - コンタクトリスト
 - ◇ ログインユーザのコンタクトリスト取得
 - ◇ コンタクトリストメンバ追加
 - ◇ コンタクトリストメンバ削除
 - フィード
 - ◇ フィード取得
 - ◇ フィード投稿
 - ◇ ウォッチフィード取得
 - チャット
 - ◇ チャット取得
 - ◇ チャット投稿
 - タスク
 - ◇ タスク取得
 - ◇ タスク追加
 - ◇ タスク更新
 - ◇ タスク削除
 - ◇ タスク催促要求
 - ◇ タスク催促要求解除
 - インボックス

Copyright © NEC Solution Innovators, Ltd. 2018 All rights reserved.

- ◇ インボックス取得
 - ◇ インボックス追加
 - ◇ インボックス削除
- Good job
 - ◇ Good job 要求
 - ◇ Good job したユーザ取得
- グループチャットルーム
 - ◇ グループチャットルーム作成
 - ◇ グループチャットルーム一覧表示
 - ◇ グループチャットルーム情報取得
 - ◇ グループチャットルームメンバ追加
 - ◇ グループチャットルームメンバ削除
 - ◇ グループチャットルーム情報更新
 - ◇ グループチャットルームメッセージ取得
 - ◇ グループチャットルームメッセージ投稿
- プロジェクト
 - ◇ プロジェクトの作成
 - ◇ 参加プロジェクト一覧の取得
 - ◇ プロジェクト詳細情報取得
 - ◇ プロジェクトメンバ情報取得
 - ◇ プロジェクトメンバ追加
 - ◇ プロジェクトメンバ削除
 - ◇ プロジェクト情報更新
 - ◇ プロジェクトオーナー変更
 - ◇ プロジェクトフィードメッセージ取得
 - ◇ プロジェクトフィードメッセージ投稿
- メッセージ
 - ◇ 新着メッセージ取得
 - ◇ メッセージ削除
- スレッド
 - ◇ スレッドメッセージ一覧取得
- 既読
 - ◇ 既読情報の取得
 - ◇ 既読要求
- スマートデバイス向け端末情報 (Android : Registration ID / iOS : Device Token)
 - ◇ 端末情報の登録
 - ◇ 端末情報の削除
- 添付ファイル WebAPI
 - 添付ファイル操作
 - ◇ 添付ファイルアップロード
 - ◇ 添付ファイルダウンロード
- 短縮 URL
 - 短縮 URL リダイレクト

2.3 凡例

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "<リクエスト名>",
  "id": "<任意の文字列>",
  "version": "<API バージョン番号>",
  "content": {
    <API ごとに異なる>
  }
}
```

■ レスポンス

```
{
  "accessToken": "<リクエスト時のアクセストークン>",
  "request": "<リクエスト時の request>",
  "id": "<リクエスト時の id>",
  "version": "<リクエスト時の API バージョン番号>",
  "errorCode": "<エラーコード>",
  "content": {
    <API ごとに異なる>
  }
}
```

3 API 詳細

リクエスト、レスポンスとも JSON 形式のメッセージである。

3.1 認証

3.1.1 ログイン

■ リクエスト

```
{
  "request": "Login",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "user": "<ログインアカウント名>",
    "password": "<パスワード>",
    "tenantName": "<テナント名>"
  }
}
```

■ レスポンス

```
{
  "request": "Login",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": "<ログイン失敗時の理由(result が false の場合)>",
    "accessToken": "<アクセストークン(result が true の場合)>",
    "userInfo": {
      "jid": "<JID>",
      "name": "<ログインアカウント名>",
      "nickName": "<ニックネーム(UTF-8 で URI エンコード)>",
      "avatarType": "imagepath(アバター添付時)/(空の値の時は添付なし)",
      "avatarData": "</cubee/からの相対ファイルパス>",
      "mailCooperationCount": 1,
      "mailCooperationItems": [
        {
          "id": "<id>",
          "serverId": "<サーバ ID>",
          "jid": "<JID>",
          "branchNumber": "<枝番>",
          "mailAddress": "<メールアドレス>"
        }
      ]
    }
  }
}
```

```

        "mailAccount": "<メールのアカウント>",
        "mailPassword": "<パスワード>",
    },
    "mailCooperationType": 0(連携しない)/1(SMTP)/2(POP)/3(IMAP)/4(SMTP+POP)
},
"status": 0(通常)/1(削除済)/2(休止(ロック))
"mailAddress": "<メールアドレス (プロフィール用) >"
},
"tenantInfo": {
    "ldap": {},
    "passwordPolicy": {
        "complexityNumber": 32
    }
}
}
}

```

※ログイン時に払いだされる accessToken の有効期間は以下の状態の短い方となります。

1. ログイン後 30 分有効です。また、この accessToken が有効中に何等かのリクエストを行うと、その時点から 30 分間有効時間が伸びます。
2. ログイン後またはプレゼンス変更リクエスト後の 60 分間にプレゼンス変更リクエストを行うと、その時点から 60 分間有効時間が伸びます。

※ログイン API 以外の API でアクセストークンが無効の場合、それぞれのレスポンスの errorCode は 9 で返します。

※tenantInfo はテナント毎の設定が格納されている。

※tenantInfo の情報を取得できなかった場合、レスポンスで tenantInfo の value は {} を返す。

※complexityNumber はそのテナントのパスワードポリシーを示す。

※complexityNumber が 0 の場合、デフォルトのパスワード複雑性チェックを行う(半角英数・記号で 8 文字以上 32 文字以内)。また、32 の場合は半角文字、半角数字、記号をそれぞれ最低 1 つ以上利用した 8 文字以上 32 文字以下の文字列であること制限します。

3.1.2 ログアウト

■ リクエスト

```
{  
  "accessToken" : "<アクセストークン>",  
  "request" : "Logout",  
  "id" : "<任意の文字列>",  
  "version" : 1,  
  "content" : {  
  }  
}
```

■ レスポンス

```
{  
  "accessToken" : "<アクセストークン>",  
  "request" : " Logout ",  
  "id" : "<要求時の文字列>",  
  "version" : 1,  
  "errorCode" : 0,  
  "content" : {  
  }  
}
```

3.2 ユーザ情報取得更新

3.2.1 ログインユーザ情報取得

■ リクエスト

```
{  
  "accessToken": "<アクセストークン>",  
  "request": "GetLoginPersonData",  
  "id": "<任意の文字列>",  
  "version": 1,  
  "content": {}  
}
```

■ レスポンス

```
{
  "request": "GetLoginPersonData",
  "id": "<要求時の文字列>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "accessToken": "<アクセストークン>",
    "result": true/false,
    "reason": "<ログイン失敗時の理由(result が false の場合)>",
    "userInfo": {
      "jid": "<JID>",
      "name": "<ログインアカウント名>",
      "nickName": "<ニックネーム(UTF-8 で URI エンコード)>",
      "avatarType": "imagepath",
      "avatarData": "</cubee/からの相対ファイルパス>",
      "mailCooperationCount": 1,
      "mailCooperationItems": [
        {
          "id": "<id>",
          "serverId": "<サーバ ID>",
          "jid": "<JID>",
          "branchNumber": "<枝番>",
          "mailAddress": "<メールアドレス>"
          "settingInfo": {
            "popServer": {
              "mailAccount": "<メールのアカウント>",
              "mailPassword": "<パスワード>"
            }
          },
          "mailCooperationType": 0(連携しない)/1(SMTP)/2(POP)/3(IMAP)/4(SMTP+POP)
        }
      ],
      "status": 0(通常)/1(削除済)/2(休止(ロック))
      "mailAddress": "<メールアドレス (プロフィール用)>",
      "extras": "{
        "<ユーザ拡張情報 key>": "<ユーザ拡張情報 value>"
      }"
    }
  }
}
```

※extras はユーザの拡張情報を key-value 形式で格納している。V5 では、ログインしたことがあるかの情報を持つ。

3.2.2 ログインユーザのプロフィール変更

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "SetLoginPersonData",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "Profile",
    "nickName": "<ニックネーム(UTF-8 で URI エンコード)>",
    "avatarType": "<アバタータイプ>",
    "avatarData": "<アバターデータ>",
    "mailAddress": "<メールアドレス（プロフィール用）>",
    "extras": "{
      <ユーザ拡張情報 key>": "<ユーザ拡張情報 value>"
    }"
  }
}
```

※avatarType は「image/jpeg」、「image/gif」、「image/png」のいずれかでファイルの種類で変える

※avatarData は画像ファイルを Base64 エンコードした文字列

※mailAddress 項目がない API で要求した場合は API の下位互換性を保つため、古い API と認識してメールアドレスの更新はしない。メールアドレスを空で更新したい場合は、mailAddress の項目をつけ、値を空文字で要求する。

※extras は key:value 形式で複数登録可能

※extras に {} を指定することで中身が全てクリアされ、{}になる。

※extras は部分更新を許容している。一部の key-value だけを POST した場合、POST された key-value は更新され、それ以外のデータはそのまま残る。複数階層での更新、追加、削除はその階層そのものの置き換えによって実現する。

※extras 項目が無い API で要求した場合は API の下位互換性を保つため、レスポンスに extras 項目は返却しない。

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "SetLoginPersonData",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "type": "Profile"
    "nickName": "<ニックネーム(UTF-8 で URI エンコード)>",
    "avatarType": "<アバタータイプ>",
    "avatarData": "<アバターデータ>",
    "mailAddress": "<メールアドレス（プロフィール用）>"
    "extras": "{
      "<登録されている key>": "<登録されている value>"
    }"
  }
}
```

※extras はユーザの拡張情報を key-value 形式で格納している。V5 では、ログインしたことがあるかの情報を持つ。

■ 異常系レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": " SetLoginPersonData",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": " Profile",
    "reason": "<失敗時の理由(番号で表示)>",
    "result": false,
    "nickName": "",
    "avatarType": "",
    "avatarData": "",
    "mailAddress": ""
  }
}
```

異常系コード表

Key	Type	Description
errorCode	9	token が無効の場合
	1	パラメータが不正の場合
	0	上記以外の場合
result	True	処理成功
	Fales	処理失敗 & アクセス権がない場合も
reason	0	成功した場合
	10	パラメータに不正な値がある場合
	11	アバターの変更失敗の場合
	15	DB エラー（サーバ内部エラー）
	400000	失敗した場合（データ更新不可）
	404000	失敗した場合（NotFound）（複数ユーザの DB 処理でエラーになって失敗ユーザリストが返ってくる場合）

※ユーザ情報取得更新時にエラーが発生すると登録は行われない。

※LDAP データ更新可否 不可の場合の reason は”400000”を返す。

3.2.3 ログインユーザのパスワード更新

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "SetLoginPersonData",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "Password",
    "oldPassword": "<変更前パスワード>",
    "newPassword": "<新しいパスワード>"
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "SetLoginPersonData",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "type": "Password"
  }
}
```

異常系コード表

Key	Type	Description
reason	0	成功した場合
	1	パラメータに不正な値がある場合
	4	文字種別が不正な場合
	8	新旧のパスワードが不一致
	16	パスワードの長さが不正
	32	複雑性チェックに違反(半角文字、半角数字、記号をそれぞれ最低 1 つ以上利用した 8 文字以上 32 文字以下の文字列であること)

3.2.4 プレゼンス変更

■ リクエスト

```
{
  "accessToken" : "<アクセストークン>",
  "request" : "SetLoginPersonData",
  "id" : "<任意の文字列>",
  "version" : 1,
  "content" : {
    "type" : "Presence",
    "presence" : (プレゼンス(1:チャット可能(=オンライン)、2:離席中、3:出張中、4:取り込み中)),
    "myMemo" : "<マイメモ(UTF-8 で URI エンコード)>"
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "SetLoginPersonData",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "type": "Presence"
  }
}
```

3.2.5 ユーザ情報の取得

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetPersonList",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "Search",
    "subType": "AllUsers",
    "condition": {
      "filter": {
        "type": "item",
        "name": "login_account",
        "value": "<取得したいアカウント名>"
      },
      "sort": {
        "item": "login_account",
        "order": "1"
      }
    },
    "startId": "<取得開始の id 番号(最新を取得する場合は 0 を指定)>",
    "count": 1
  }
}
```

※filter 内の name 項目を「jid」に変更し value 項目に JID を設定することで JID を指定しての情報取得も可能になります。

■ レスポンス

```
{
  "accessToken" : "<アクセストークン>",
  "request" : "GetPersonList",
  "id" : "<要求時の id>",
  "version" : 1,
  "errorCode" : 0,
  "content" : {
    "result" : true/false,
    "reason" : <失敗時の理由(result が false の場合)>,
    "type" : "Search",
    "subType" : "AllUsers",
    "allItemCount" : 1,
    "count" : 1,
    "items" : [
      {
        "id" : "<コンタクトリストのユーザの ID>",
        "jid" : "<コンタクトリストのユーザの JID>",
        "userName" : "<ログインアカウント名>",
        "subscription" : "<ログインユーザとの関わり情報    ("both":コンタクトリストに追加済み、"none":
コンタクトリストに未追加) >",
        "groupCount" : 2,
        "groupItems" : [
          "<グループ名(UTF-8 で URI エンコード)>",
          "<グループ名(UTF-8 で URI エンコード)>"
        ],
        "nickName" : "<ニックネーム(UTF-8 で URI エンコード)>",
        "avatarType" : "imagepath",
        "avatarData" : "<アバターデータ>",
        "presence" : (プレゼンス(0:オフライン、1:チャット可能(=オンライン)、2:離席中、3:出張中、4:取り込
み中)),
        "myMemo" : "<マイメモ(UTF-8 で URI エンコード)>",
        "status" : 0(通常)/1(削除済)/2:(休止(ロック))
      }
    ]
  }
}
```

3.3 コンタクトリスト

3.3.1 ログインユーザのコンタクトリスト取得

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetPersonList",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "ContactList"
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetPersonList",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "type": "ContactList",
    "extras": {},
    "count": 13,
    "items": [
      {
        "jid": "<コンタクトリストのユーザの JID>",
        "userName": "<ログインアカウント名>",
        "subscription": "both/to/from/none",
        "groupCount": 2,
        "groupItems": [
          "<グループ名>",
          "<グループ名>"
        ],
        "nickName": "<ニックネーム(UTF-8 で URI エンコード)>",
        "avatarType": "imagepath",
        "avatarData": "</cubee/からの相対ファイルパス>",
        "presence": (プレゼンス(0:オフライン、1:チャット可能(=オンライン)、2:離席中、3:出張中、4:取り込み中)),
        "myMemo": "<マイメモ(UTF-8 で URI エンコード)>",
        "status": 0(通常)/1(削除済)/2(休止(ロック))
      },
      {<2 個目のデータ>},
      ...
    ]
  }
}
```

```

    {<13 個目のデータ>
  }
}

```

3.3.2 コンタクトリストメンバ追加

■ リクエスト

```

{
  "accessToken": "<アクセストークン>",
  "request": "AddMember",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "ContactList",
    "count": 2,
    "members": [
      {
        "jid": "<追加対象のユーザ jid>",
        "contactListGroup": ""
      },
      {<2 個目のデータ>}...
    ]
  }
}

```

※group は指定がない場合、NoGroup とする
 ※追加する位置は末尾とする

■ レスポンス

```

{
  "accessToken": "<アクセストークン>",
  "request": "AddMember",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": "ContactList",
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "extras": {},
    "count": 1,
    "items": [
      {
        "addedBy": "<追加した人の JID>",
        "count": 4,
        "members": {

```



```

    "successMembers": [
      {
        "id": "<コンタクトリストのユーザの ID>",
        "jid": "<コンタクトリストのユーザの JID>",
        "userName": "<ログインアカウント名>",
        "nickName": "<ニックネーム(UTF-8 で URI エンコード)>",
        "avatarType": "<アバタータイプ>",
        "avatarData": "<アバターデータ>",
        "status": 0(通常)/1(削除済)/2:(休止(ロック)),
        "contactListGroup": "",
        "position": <グループ内での表示順(※V2.0 では常に「-1」となる。本来は contactlistgroup
の要素と同じ数要素を持つ配列。要素のインデックスが対応付く)
      },{<2 個目のデータ>...}
    ],
    "failureMembers": [
      {
        "jid": "<追加失敗ユーザ jid>"
        "contactListGroup": ""
      },{<2 個目のデータ>...}
    ]
  }
}
}
}

```

3.3.3 コンタクトリストメンバ削除

■ リクエスト

```

{
  "accessToken": "<アクセストークン>",
  "request": "RemoveMember",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "ContactList",
    "count": 10<members の個数>,
    "members": [
      {
        "jid": "<削除対象のユーザ jid>"
      },
      {<2 個目のデータ>},...
    ]
  }
}

```

※全てのグループにおいて、指定ユーザを削除する。

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "RemoveMember",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": "ContactList",
    "result": true/false, (全員失敗した場合のみ false)
    "reason": <失敗時の理由(result が false の場合)>,
    "extras": {},
    "count": 1,
    "items": [
      {
        "removedBy": "<削除操作を行ったユーザ JID>",
        "count": 1,
        "members": [
          successMembers : [
            {
              "jid": "<削除成功ユーザ jid>"
            },
            {<2 個目のデータ>},...
          ],
          failureMembers : [
            {
              "jid": "<削除失敗ユーザ jid>"
            },
            {<2 個目のデータ>},...
          ]
        ]
      }
    ]
  }
}
```

※items は result が true の場合のみ。それ以外の場合は空[]となる。

3.4 フィード（取得、投稿）

3.4.1 フィード取得

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetMessage",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "MyFeed",
    "condition": {},
    "startId": "<取得開始のメッセージの ID(最新を取得する場合は 0 を指定)>",
    "count": 20
  }
}
```

※count は取得件数なので 1 以上であれば 20 以外でもよい

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetMessage",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "type": "MyFeed",
    "extras": {},
    "count": 20,
    "items": [
      {
        "id": 125,
        "itemId": "<メッセージの ItemID>",
        "from": "<投稿者の JID>",
        "to": "<送信先>",
        "type": 1(=公開メッセージ)/6(システムメッセージ)
        "body": "<メッセージ内容(UTF-8 で URI エンコード)>"
        "triggerAction": "<システムメッセージの場合はそのトリガーアクション>",
        "nodeName": "ノード名",
        "createdAt": "<メッセージ作成日時(yyyy/MM/dd HH:mm:ss 形式)>",
        "replyId": "<リプライメッセージはリプライ元メッセージの ItemID>",
        "replyTo": "リプライの宛先 JID",
        "goodJobCount": 2,
        "goodJobItems": [
```

Copyright © NEC Solution Innovators,Ltd.2018 All rights reserved.

```

{
  "fromJid" : "<Good Job した人の JID>",
  "fromName" : "<Good Job した人のニックネーム(UTF-8 で URI エンコード)>",
  "date" : "<Good Job した日時(yyyy/MM/dd HH:mm:ss 形式)>",
  "userName" : "<Good Job した人のログインアカウント名>",
  "nickName" : "<Good Job した人のニックネーム(UTF-8 で URI エンコード)>",
  "avatarData" : "<Good Job した人のアバターファイルのパス (/cubee/からの相対ファイルパス)
>",

  "avatarType" : "imagepath"
  "status" : <Good Job した人のアカウント状態 0(通常)/1(削除済)/2:(休止(ロック))>
},
{ ... }
],
"attachedCount" : 0,
"attachedItems" : [],
"context" : "",
"deleteFlag" : 0-2 の数値 (0:未削除, 1:完全削除(画面上に見えない), 2:論理削除(画面上に見える))
"readFlg" : 0/1,(※ログインユーザがそのメッセージを読んだかどうか 0:未読 1:既読)
"readAllCount" : 100, (※件数は全既読者数)
"readItems" : [
{
  "jid" : "<メッセージを既読にした人の JID>",
  "account" : "<メッセージを既読にした人のログインアカウント(UTF-8 で URI エンコード)>",
  "nickName" : "<メッセージを既読にした人のニックネーム(UTF-8 で URI エンコード)>",
  "avatarData" : "<メッセージを既読にした人のアバターファイルのパス (/cubee/からの相対ファ
イルパス) >",
  "avatarType" : "imagepath",
  "date" : "<メッセージを既読にした日時(yyyy/MM/dd HH:mm:ss 形式)>"
},
{ ... }
],
"shortenUrlCount" : <shortenItems の配列のサイズ>,
"shortenItems" : [
{
  "DisplayedURL" : "<パス部分を…に置換した URL>",
  "ShortenPath" : "<短縮 URI の相対パス>",
  "OriginalURL" : "<オリジナル URL>",
}
],
"personInfo" : {
  "<jid>" : {
    "userName" : "<jid に対応するログインアカウント(UTF-8 で URI エンコード)>",
    "nickName" : "<jid に対応するニックネーム(UTF-8 で URI エンコード)>",
    "avatarData" : "<jid に対応するアバターファイルのパス (/cubee/からの相対ファイルパス)
>",

    "avatarType" : "imagepath",
    "status" : <jid に対応するアカウント状態 0(通常)/1(削除済)/2:(休止(ロック))>,
  },

```

```

    "<jid>": { ... }
  }
  "convertFileCount": <convertFileItems の配列のサイズ>
  "convertFileItems": [
    {
      "originalURL": "<オリジナル画像の URL>",
      "convertURL": "<サムネイル画像の URL>",
    }
  ]
},
{<2 個目のデータ>},
...,
{<20 個目のデータ>}
]
}
}

```

※count, items は result が true の場合のみ

※nodeName はメッセージタイプがパブリックメッセージの場合のみ

※triggerAction はメッセージタイプがシステムメッセージの場合のみ

※例は items の 1 個目のみ詳細に記述

※readItems は直近で既読にした上位 1 名のユーザデータを表示（本 API に限らず以降に出てくる readItems も同様）

※personInfo はメッセージの投稿者やチャットの相手やタスクの担当者等の JID に対応するアカウント情報（本 API に限らず以降に出てくる personInfo も同様）

3.4.2 フィード投稿

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "SendMessage",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "Public",
    "body": "<メッセージ内容(UTF-8 で URI エンコード)>",
    "replyId": "<リプライ時はリプライ元メッセージの ItemID>",
    "attachedCount": 0,
    "attachedItems": [],
    "context": ""
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "SendMessage",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "type": "Public"
  }
}
```

3.4.3 ウォッチフィード（取得）

■ リクエスト

```
{
  "accessToken":"<アクセストークン>",
  "request":"GetMessage",
  "id":"<任意の文字列>",
  "version":1,
  "content":{
    "type":"Search",
    "condition":{
      "filter":{
        "type":"or"
        "value":[
          {
            "type":"and"
            "value":[
              {
                "type":"item",
                "name":"msgtype",
                "value":1
              },
              {
                "type":"keyword",
                "value":"%40<ログインユーザのアカウント名>",
                "include":[]
              }
            ]
          },
          {
            "type":"and"
            "value":[
              {
                "type":"item",
                "name":"msgtype",
                "value":4
              },
              {
                "type":"item",
                "name":"owner",
                "value":"<ログインユーザの JID>"
              }
            ]
          },
          {
            "type":"and"
            "value":[
              {
                "type":"not",
                "value":{
```

Copyright © NEC Solution Innovators,Ltd.2018 All rights reserved.

```

        "type":"item",
        "name":"status",
        "value":1
    }
},
{
    "type":"or"
    "value":[
        {
            "type":"not",
            "value":{
                "type":"item",
                "name":"status",
                "value":2
            }
        },
        {
            "type":"item",
            "name":"parent_item_id",
            "value":""
        }
    ],
}
],
},
],
},
{
    "type":"and"
    "value":[
        {
            "type":"item",
            "name":"msgtype",
            "value":4
        },
        {
            "type":"item",
            "name":"owner",
            "value":"<ログインユーザの JID>"
        },
    ],
    {
        "type":"or"
        "value":[
            {
                "type":"item",
                "name":"status",
                "value":1
            },

```



```

        {
            "type":"and"
            "value":[
                {
                    "type":"item",
                    "name":"status",
                    "value":2
                },
                {
                    "type":"not",
                    "value":{"
                        "type":"item",
                        "name":"parent_item_id",
                        "value":""
                    }
                }
            ],
        },
    ],
},
{
    "type":"item",
    "name":"msgtype",
    "value":9
},
{
    "type":"and"
    "value":[
        {
            "type":"item",
            "name":"msgtype",
            "value":5
        },
        {
            "type":"keyword",
            "value":"%40<ログインユーザのアカウント名>",
            "include":[]
        }
    ],
},
{
    "type":"and"
    "value":[
        {
            "type":"item",
            "name":"msgtype",

```

※2 count は取得件数であるため 1 以上であれば 20 以外でもよい

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetMessage",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "type": "Search",
    "extras": {※1
      "allItemCount": "<ウォッチフィードに表示できる全メッセージ件数>",
      "childrenCount": "<items 中のタスクの子タスクの数>",
      "childrenItems": [子タスクの情報（仕様は下の items の中と同じで内容が子タスクである(ただし、兄弟タスク情報は含まれない)）]
    },
    "count": 20,
    "items": [
      ※2
    ]
  }
}
```

※1 : extras の中身

childrenCount : 取得新着メッセージのうち、タスクについてそのタスクを親にもつ子タスクの一覧の件数

childrenItems : 新着メッセージのうち、タスクについてそのタスクを親にもつ子タスクの一覧。形式は items のタスク 1 件の形式の配列

※2 : items の中身

メッセージの種類によって異なる。1 つのメッセージに対する詳細は各種類のメッセージ取得 API のレスポンスの items の中身のものと同等

3.5 チャット（取得、投稿）

3.5.1 チャット取得

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetMessage",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "Chat",
    "condition": {
      "partner": "<チャット相手の JID>"
    },
    "startId": "<取得開始のメッセージの ID(最新を取得する場合は 0 を指定)>",
    "count": 20
  }
}
```

※count は取得件数なので 1 以上であれば 20 以外でもよい

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetMessage",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "type": "Chat",
    "extras": {},
    "count": 20,
    "items": [
      {
        "id": 125,
        "itemId": "<メッセージの ItemID>",
        "from": "<投稿者の JID>",
        "to": "<送信先の JID>",
        "type": 2(=チャットメッセージ),
        "body": "<メッセージ内容(UTF-8 で URI エンコード)>",
        "createdAt": "<メッセージ作成日時(yyyy/MM/dd HH:mm:ss 形式)>",
        "replyId": "<引用元メッセージの ItemID>",
        "replyTo": "<引用元メッセージの発言者 JID>",
        "goodJobCount": 2,
        "goodJobItems": [
```

Copyright © NEC Solution Innovators,Ltd.2018 All rights reserved.

```

{
  "fromJid" : "<Good Job した人の JID>",
  "fromName" : "<Good Job した人のニックネーム(UTF-8 で URI エンコード)>",
  "date" : "<Good Job した日時(yyyy/MM/dd HH:mm:ss 形式)>",
  "userName" : "<Good Job した人のログインアカウント名>",
  "nickName" : "<Good Job した人のニックネーム(UTF-8 で URI エンコード)>",
  "avatarData" : "<Good Job した人のアバターファイルのパス (/cubee/からの相対ファイルパス)
>",
  "avatarType" : "imagepath"
  "status" : <Good Job した人のアカウント状態 0(通常)/1(削除済)/2:(休止(ロック))>
},
{ ... }
],
"attachedCount" : 0,
"attachedItems" : [],
"context" : "",
"deleteFlag" : 0-2 の数値 (0:未削除, 1:完全削除(画面上に見えない), 2:論理削除(画面上に見える))
"readFlg" : 0/1,(※ログインユーザがそのメッセージを読んだかどうか 0:未読 1:既読)
"readAllCount" : 100, (※件数は全既読者数)
"readItems" : [
{
  "jid" : "<メッセージを既読にした人の JID>",
  "account" : "<メッセージを既読にした人のログインアカウント(UTF-8 で URI エンコード)>",
  "nickName" : "<メッセージを既読にした人のニックネーム(UTF-8 で URI エンコード)>",
  "avatarData" : "<メッセージを既読にした人のアバターファイルのパス (/cubee/からの相対ファ
イルパス) >",
  "avatarType" : "imagepath",
  "date" : "<メッセージを既読にした日時(yyyy/MM/dd HH:mm:ss 形式)>"
},
{ ... }
],
"shortenUrlCount" : <shortenItems の配列のサイズ>,
"shortenItems" : [
{
  "DisplayedURL" : "<パス部分を…に置換した URL>",
  "ShortenPath" : "<短縮 URI の相対パス>",
  "OriginalURL" : "<オリジナル URL>",
}
],
"personInfo" : {
  "<jid>" : {
    "userName" : "<jid に対応するログインアカウント(UTF-8 で URI エンコード)>",
    "nickName" : "<jid に対応するニックネーム(UTF-8 で URI エンコード)>",
    "avatarData" : "<jid に対応するアバターファイルのパス (/cubee/からの相対ファイルパス)
>",
    "avatarType" : "imagepath",
    "status" : <jid に対応するアカウント状態 0(通常)/1(削除済)/2:(休止(ロック))>,
  },

```

```

    "<jid>": { ... }
  }
  "convertFileCount": <convertFileItems の配列のサイズ>
  "convertFileItems": [
    {
      "originalURL": "<オリジナル画像の URL>",
      "convertURL": "<サムネイル画像の URL>",
    }
  ]
},
{<2 個目のデータ>},
...,
{<20 個目のデータ>}
]
}
}

```

3.5.2 チャット投稿

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "SendMessage",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "Chat",
    "to": "<チャット相手の JID>",
    "body": "<タスクの内容(UTF-8 で URI エンコード)>",
    "replyId": "<引用元メッセージの ItemID>",
    "attachedCount": 0,
    "attachedItems": [],
    "context": ""
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "SendMessage",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "type": "Chat"
  }
}
```

3.6 タスク（取得、追加、変更）

3.6.1 タスク取得

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetMessage",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "Task",
    "condition": {
      "filter": {
        "owner": "<取得者本人の JID>",
        "status": "2,3,4"
      },
      "sort": {
        "item" : "demand_status,demand_date,complete_date IS NULL,complete_date,due_date IS NULL,due_date,priority,status",
        "order": "2,1,2,2,1,1,2,1"
      }
    },
    "startId": "<取得開始のメッセージの ID(最新を取得する場合は 0 を指定)>",
    "count": 20
  }
}
```

※count は取得件数なので 1 以上であれば 20 以外でもよい

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetMessage",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "type": "Task",
    "extras": {
      "unfinishedTaskCount": "<未完了タスクの数>",
      "childrenCount": "<items のタスクの子タスクの数>",
      "childrenItems": [子タスクの情報（仕様は下の items の中と同じで内容が子タスクである(ただし、兄弟タスク情報は含まれない)）]
    },
    "count": 20,
    "items": [
```



```

{
  "id": 125,
  "itemId": "<メッセージの ItemID>",
  "from": "<投稿者の JID>",
  "to": "<送信先>",
  "type": 4(=Task),
  "title": "<タスクのタイトル(UTF-8 で URI エンコード)>",
  "body": "<メッセージ内容(UTF-8 で URI エンコード)>",
  "progress": 0, ※未使用
  "spentTime": 0, ※未使用
  "estimatedTime": 0, ※未使用
  "remainingTime": 0, ※未使用
  "goal": "", ※未使用
  "alert": 0, ※未使用
  "parentItemId": "<親タスクの ItemID>", ※依頼タスクのみで依頼元のタスクの ItemID を表す
  "priority": "<優先度(1~4) 1:低、2:中、3:高、4:最優先>",
  "createdAt": "<メッセージ作成日時(yyyy/MM/dd HH:mm:ss 形式)>",
  "replyId": "<引用元メッセージの ItemID>",
  "replyTo": "<引用元メッセージの発言者 JID>",
  "startDate": "<開始日時(yyyy/MM/dd HH:mm:ss 形式)>",
  "dueDate": "<締切(終了)日時(yyyy/MM/dd HH:mm:ss 形式)>",
  "completeDate": "<完了日時(yyyy/MM/dd HH:mm:ss 形式)>",
  "owner": "<担当者の JID>",
  "group": "<プロジェクトの場合はコプロジェクトの roomId、その他は空文字>",
  "groupName": "<プロジェクトの場合はプロジェクト名(UTF-8 で URI エンコード)、その他は空文字>",
  "client": "<依頼者の JID>",
  "status": "<タスクの状態 2:ASSIGNING (割当中)、3:NEW (新規)、4:DOING (進行中)、7:FINISHED (終了)、8:REJECTED (却下)>",
  "updatedAt": "<最終更新日時(yyyy/MM/dd HH:mm:ss 形式)>",
  "updatedBy": "<最終更新者の JID>",
  "noteCount": 0, ※未使用
  "noteItems": [], ※未使用
  "reminderCount": 0, ※未使用
  "reminderItems": [], ※未使用
  "attachedCount": 0, ※未使用
  "attachedItems": [], ※未使用
  "siblingCount": 2,
  "siblingItems": [
    {
      "siblingItemId": "<兄弟タスク ItemID>",
      "ownerJid": "<担当者の JID>",
      "ownerName": "<担当者のニックネーム(UTF-8 で URI エンコード)>",
      "status": 3<タスクの状態(1~8)>,
      "userName": "<担当者のログインアカウント名>",
      "nickName": "<担当者のニックネーム(UTF-8 で URI エンコード)>",
      "avatarType": "imagepath",
      "avatarData": "<担当者のアバターファイルのパス (/cubee/からの相対ファイルパス)>",

```

```

        "userStatus": <担当者のアカウント状態 0(通常)/1(削除済)/2:(休止(ロック))>
    },
    { ... }
],
"context": "",
"demandStatus": "催促状態(0:催促なし / 1:催促あり)",
"demandDate": "<催促日時>",
"deleteFlag": 0-2 の数値 (0:未削除, 1:完全削除(画面上に見えない), 2:論理削除(画面上に見える))
"shortenUrlCount": <shortenItems の配列のサイズ>,
"shortenItems": [
    {
        "DisplayedURL": "<パス部分を…に置換した URL>",
        "ShortenPath": "<短縮 URI の相対パス>",
        "OriginalURL": "<オリジナル URL>",
    }
],
"personInfo": {
    "<jid>": {
        "userName": "<jid に対応するログインアカウント(UTF-8 で URI エンコード)>",
        "nickName": "<jid に対応するニックネーム(UTF-8 で URI エンコード)>",
        "avatarData": "<jid に対応するアバターファイルのパス (/cubee/からの相対ファイルパス)
>",
        "avatarType": "imagpath",
        "status": <jid に対応するアカウント状態 0(通常)/1(削除済)/2:(休止(ロック))>,
    },
    "<jid>": { ... }
}
"convertFileCount": <convertFileItems の配列のサイズ>
"convertFileItems": [
    {
        "originalURL": "<オリジナル画像の URL>",
        "convertURL": "<サムネイル画像の URL>",
    }
]
},
{<2 個目のデータ>},
...,
{<20 個目のデータ>}
]
}
}

```

※count, items は result が true の場合のみ

※タスクの状態：1:INBOX (INBOX)、2:ASSIGNING (割当中)、3:NEW (新規)、4:DOING (進行中)、7:FINISHED (終了)、8:REJECTED (却下)

※例は items の 1 個目のみ詳細に記述

3.6.2 タスク追加

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "SendMessage",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "Task",
    "title": "<タスクのタイトル(UTF-8 で URI エンコード)>",
    "body": "<タスクの内容(UTF-8 で URI エンコード)>",
    "progress": 0, ※未使用
    "spentTime": 0, ※未使用
    "estimatedTime": 0, ※未使用
    "remainingTime": 0, ※未使用
    "goal": "", ※未使用
    "alert": 0, ※未使用
    "parentItemId": "", ※親タスクの ItemID を指定するところであるが、新規追加時は空文字
    "priority": <優先度(1~4) 1:低、2:中、3:高、4:最優先>,
    "replyId": "<引用元メッセージの ItemID、メッセージからタスク化した場合のみ>",
    "replyTo": "<引用元メッセージの発言者 JID、メッセージからタスク化した場合のみ>",
    "startDate": "<開始日時(yyyy/MM/dd HH:mm:ss 形式)>",
    "dueDate": "<締切(終了)日時(yyyy/MM/dd HH:mm:ss 形式)>",
    "owner": "<担当者の JID ※複数人に同時に依頼する場合はカンマ (,) 区切り>",
    "group": "<プロジェクトの場合はプロジェクトの roomId、その他は空文字>",
    "client": "<依頼者の JID>",
    "status": 3<タスクの状態(2~8)>,
    "attachedCount": 0, ※未使用
    "attachedItems": [], ※未使用
    "context": "" ※未使用
  }
}
```

※タスクの状態：2:ASSIGNING（割当中）、3:NEW（新規）、4:DOING（進行中）、7:FINISHED（終了）、8:REJECTED（却下）

※注意事項：client と owner が異なる場合（複数人に割り当てる場合を含む）は必ずタスクの状態(status)の値は 2:ASSIGNING（割当中）とすること

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "SendMessage",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": "Task",
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "extras": {
      "childrenCount": "<items のタスクの子タスクの数>",
      "childrenItems": [子タスクの情報（仕様は下の items の中と同じで内容が子タスクである(ただし、兄弟タスク情報は含まれない)）]
    },
    "count": 1,
    "items": [
      {
        "id": 125,
        "itemId": "<メッセージの ItemID>",
        "from": "<投稿者の JID>",
        "to": "<送信先>",
        "type": 4(=Task),
        "title": "<タスクのタイトル(UTF-8 で URI エンコード)>",
        "body": "<メッセージ内容(UTF-8 で URI エンコード)>",
        "progress": 0, ※未使用
        "spentTime": 0, ※未使用
        "estimatedTime": 0, ※未使用
        "remainingTime": 0, ※未使用
        "goal": "", ※未使用
        "alert": 0, ※未使用
        "parentItemId": "<親タスクの ItemID>",
        "priority": "<優先度(1~4) 1:低、2:中、3:高、4:最優先>",
        "createdAt": "<メッセージ作成日時(yyyy/MM/dd HH:mm:ss 形式)>",
        "replyId": "<引用元メッセージの ItemID>",
        "replyTo": "<引用元メッセージの発言者 JID>",
        "startDate": "<開始日時(yyyy/MM/dd HH:mm:ss 形式)>",
        "dueDate": "<締切(終了)日時(yyyy/MM/dd HH:mm:ss 形式)>",
        "completeDate": "<完了日時(yyyy/MM/dd HH:mm:ss 形式)>",
        "owner": "<担当者の JID>",
        "group": "<プロジェクトの場合はプロジェクトの roomId、その他は空文字>",
        "groupName": "<プロジェクトの場合はプロジェクト名(UTF-8 で URI エンコード)、その他は空文字>",
        "client": "<依頼者の JID>",
        "status": 3<タスクの状態(1~8)>,
        "updatedAt": "<最終更新日時(yyyy/MM/dd HH:mm:ss 形式)>",

```

Copyright © NEC Solution Innovators,Ltd.2018 All rights reserved.

```

"updatedBy" : "<最終更新者の JID>",
"noteCount" : 2, ※未使用
"noteItems" : [], ※未使用
"reminderCount" : 0, ※未使用
"reminderItems" : [], ※未使用
"attachedCount" : 0, ※未使用
"attachedItems" : [], ※未使用
"siblingCount" : 2,
"siblingItems" : [
  {
    "siblingItemId" : "<兄弟タスク ItemID>",
    "ownerJid" : "<担当者の JID>",
    "ownerName" : "<担当者のニックネーム(UTF-8 で URI エンコード)>",
    "status" : 3<タスクの状態(1~8)>,
    "userName" : "<担当者のログインアカウント名>",
    "nickName" : "<担当者のニックネーム(UTF-8 で URI エンコード)>",
    "avatarType" : "imagepath",
    "avatarData" : "<担当者のアバターファイルのパス (/cubee/からの相対ファイルパス) >",
    "userStatus" : <担当者のアカウント状態 0(通常)/1(削除済)/2:(休止(ロック))>
  },
  { ... }
],
"context" : "", ※未使用
"demandStatus" : "催促状態(0:催促なし / 1:催促あり)",
"demandDate" : "<催促日時>",
"deleteFlag" : 0-2 の数値 (0:未削除, 1:完全削除(画面上に見えない), 2:論理削除(画面上に見える))
"shortenUrlCount" : <shortenItems の配列のサイズ>,
"shortenItems" : [
  {
    "DisplayedURL" : "<パス部分を…に置換した URL>",
    "ShortenPath" : "<短縮 URI の相対パス>",
    "OriginalURL" : "<オリジナル URL>",
  }
],
"personInfo" : {
  "<jid>" : {
    "userName" : "<jid に対応するログインアカウント(UTF-8 で URI エンコード)>",
    "nickName" : "<jid に対応するニックネーム(UTF-8 で URI エンコード)>",
    "avatarData" : "<jid に対応するアバターファイルのパス (/cubee/からの相対ファイルパス)
>",
    "avatarType" : "imagepath",
    "status" : <jid に対応するアカウント状態 0(通常)/1(削除済)/2:(休止(ロック))>,
  },
  "<jid>" : { ... }
}
"convertFileCount" : <convertFileItems の配列のサイズ>
"convertFileItems" : [
  {
    "originalURL" : "<オリジナル画像の URL>",

```

```
        "convertURL": "<サムネイル画像の URL>",
    }
    ]
}
]
```

※タスクの状態：2:ASSIGNING（割当中）、3:NEW（新規）、4:DOING（進行中）、7:FINISHED（終了）、8:REJECTED（却下）

3.6.3 タスク更新

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "UpdateMessage",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "Task",
    "itemId": "<更新するタスクの ItemID>",
    "title": "<タスクのタイトル(UTF-8 で URI エンコード)>",
    "body": "<タスクの内容(UTF-8 で URI エンコード)>",
    "progress": 0, ※未使用
    "spentTime": 0, ※未使用
    "estimatedTime": 0, ※未使用
    "remainingTime": 0, ※未使用
    "goal": "", ※未使用
    "alert": 0, ※未使用
    "parentItemId": "<親タスクの ItemID>",
    "priority": "<優先度(1~4) 1:低、2:中、3:高、4:最優先>",
    "replyId": "<引用元メッセージの ItemID>",
    "replyTo": "<引用元メッセージの発言者 JID>",
    "startDate": "<開始日時(yyyy/MM/dd HH:mm:ss 形式)>",
    "dueDate": "<締切(終了)日時(yyyy/MM/dd HH:mm:ss 形式)>",
    "owner": "<担当者の JID>",
    "group": "プロジェクトの場合はプロジェクトの roomId、その他は空文字",
    "client": "<依頼者の JID>",
    "status": 3<タスクの状態(1~8)>,
    "attachedCount": 0,
    "attachedItems": [],
    "context": ""
  }
}
```

※タスクの状態：1:INBOX (INBOX)、2:ASSIGNING (割当中)、3:NEW (新規)、4:DOING (進行中)、7:FINISHED (終了)、8:REJECTED (却下)

※注意事項：親タスク(依頼元タスク)に対しては、この WebAPI は使用しなはならない

※注意事項：client、owner のそれぞれの値は変更前の値から変更はしてはならない

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "UpdateMessage",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
```

```

"type": "Task",
"result": true/false,
"reason": <失敗時の理由(result が false の場合)>,
"extras": {
  "childrenCount": <items のタスクの子タスクの数>,
  "childrenItems": [子タスクの情報（仕様は下の items の中と同じで内容が子タスクである(ただし、兄弟タスク情報は含まれない)）]
},
"count": 1,
"items": [
{
  "id": 125,
  "itemId": "<メッセージの ItemID>",
  "from": "<投稿者の JID>",
  "to": <送信先>
  "type": 4(=Task),
  "title": "<タスクのタイトル(UTF-8 で URI エンコード)>",
  "body": "<メッセージ内容(UTF-8 で URI エンコード)>",
  "progress": 0, ※未使用
  "spentTime": 0, ※未使用
  "estimatedTime": 0, ※未使用
  "remainingTime": 0, ※未使用
  "goal": "", ※未使用
  "alert": 0, ※未使用
  "parentItemId": "<親タスクの ItemID>",
  "priority": <優先度(1~4) 1:低、2:中、3:高、4:最優先>,
  "createdAt": "<メッセージ作成日時(yyyy/MM/dd HH:mm:ss 形式)>",
  "replyId": "<引用元メッセージの ItemID>",
  "replyTo": "<引用元メッセージの発言者 JID>",
  "startDate": "<開始日時(yyyy/MM/dd HH:mm:ss 形式)>",
  "dueDate": "<締切(終了)日時(yyyy/MM/dd HH:mm:ss 形式)>",
  "completeDate": "<完了日時(yyyy/MM/dd HH:mm:ss 形式)>",
  "owner": "<担当者の JID>",
  "group": "<プロジェクトの場合はプロジェクトの roomId、その他は空文字>",
  "groupName": "<プロジェクトの場合はプロジェクト名(UTF-8 で URI エンコード)、その他は空文字>",

  "client": "<依頼者の JID>",
  "status": 3<タスクの状態(1~8)>,
  "updatedAt": "<最終更新日時(yyyy/MM/dd HH:mm:ss 形式)>",
  "updatedBy": "<最終更新者の JID>",
  "noteCount": 2, ※未使用
  "noteItems": [], ※未使用
  "reminderCount": 0, ※未使用
  "reminderItems": [], ※未使用
  "attachedCount": 0, ※未使用
  "attachedItems": [], ※未使用
  "siblingCount": 2,

```



```

"siblingItems": [
  {
    "siblingItemId": "<兄弟タスク ItemID>",
    "ownerJid": "<担当者の JID>",
    "ownerName": "<担当者のニックネーム(UTF-8 で URI エンコード)>",
    "status": 3<タスクの状態(1~8)>,
    "userName": "<担当者のログインアカウント名>",
    "nickName": "<担当者のニックネーム(UTF-8 で URI エンコード)>",
    "avatarType": "imagepath",
    "avatarData": "<担当者のアバターファイルのパス (/cubee/からの相対ファイルパス) >",
    "userStatus": <担当者のアカウント状態 0(通常)/1(削除済)/2:(休止(ロック))>
  },
  { ... }
],
"context": "", ※未使用
"demandStatus": "催促状態(0:催促なし / 1:催促あり)",
"demandDate": "<催促日時>",
"deleteFlag": 0-2 の数値 (0:未削除, 1:完全削除(画面上に見えない), 2:論理削除(画面上に見える))
"shortenUrlCount": <shortenItems の配列のサイズ>,
"shortenItems": [
  {
    "DisplayedURL": "<パス部分を…に置換した URL>",
    "ShortenPath": "<短縮 URI の相対パス>",
    "OriginalURL": "<オリジナル URL>",
  }
],
"personInfo": {
  "<jid>": {
    "userName": "<jid に対応するログインアカウント(UTF-8 で URI エンコード)>",
    "nickName": "<jid に対応するニックネーム(UTF-8 で URI エンコード)>",
    "avatarData": "<jid に対応するアバターファイルのパス (/cubee/からの相対ファイルパス) >",
    "avatarType": "imagepath",
    "status": <jid に対応するアカウント状態 0(通常)/1(削除済)/2:(休止(ロック))>,
  },
  "<jid>": { ... }
}
"convertFileCount": <convertFileItems の配列のサイズ>
"convertFileItems": [
  {
    "originalURL": "<オリジナル画像の URL>",
    "convertURL": "<サムネイル画像の URL>",
  }
]
}
}
]
}
}

```

3.6.4 タスク削除

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "DeleteMessage",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "Delete",
    "itemId": "<削除するメッセージの ItemID>",
    "deleteFlag": 1(削除フラグタイプ 0~2)
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "DeleteMessage",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": "Delete",
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>"
  }
}
```

3.6.5 タスク催促要求

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "MessageOption",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "DemandTask",
    "itemId": "<催促をするタスクメッセージの ItemID>",
    "clearCondition": "<催促解除条件>"
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "MessageOption",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "type": "DemandTask"
  }
}
```

3.6.6 タスク催促解除要求

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "MessageOption",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "ClearDemandedTask",
    "itemId": "<催促をするタスクメッセージの ItemID>"
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "MessageOption",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "type": "ClearDemandedTask"
  }
}
```

3.7 インボックス（取得、追加）

3.7.1 インボックス取得

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetMessage",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "Search",
    "condition": {
      "filter": {
        "type": "and",
        "value": [
          {
            "type": "item",
            "name": "msgtype",
            "value": "4"
          },
          {
            "type": "item",
            "name": "owner",
            "value": "<取得者本人の JID>"
          }
        ],
        "type": "or",
        "value": [
          {
            "type": "item",
            "name": "status",
            "value": 1
          },
          {
            "type": "and",
            "value": [
              {
                "type": "item",
                "name": "status",
                "value": 2
              },
              {
                "type": "not",
                "value": {
                  "type": "item",
                  "name": "parent_item_id",
                  "value": ""
                }
              }
            ]
          }
        ]
      }
    }
  }
}
```

```

        }
      }
    ]
  }
]
},
"sort": {
  "item": "demand_status,demand_date,updated_at",
  "order": "2,1,2"
},
"startId": <取得開始のメッセージの ID(最新を取得する場合は 0 を指定)>,
"count": 20 ※
}
}

```

※ count は取得件数なので 1 以上であれば 20 以外でもよい

■ レスポンス

```

{
  "accessToken": "<アクセストークン>",
  "request": "GetMessage",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": <失敗時の理由(result が false の場合)>,
    "type": "Search",
    "extras": {
      "allItemCount": <インボックスの全ての数>,
      "childrenCount": 0,
      "childrenItems": []
    },
    "count": 20,
    "items": [
      {
        "id": メッセージを一意に表す ID(数値),
        "itemId": "<メッセージを一意に表す文字列の ID、ItemID>",
        "from": "<作成者の JID>",
        "to": "<保存ホスト>",
        "type": 4(=Task),
        "title": "<タスクのタイトル(UTF-8 で URI エンコード)>", ※依頼タスクのみ、自分でインボックス
        へ入れたものは空文字
        "body": "<インボックスの内容(UTF-8 で URI エンコード)>",
        "progress": 0, ※未使用
      }
    ]
  }
}

```

Copyright © NEC Solution Innovators,Ltd.2018 All rights reserved.

```

"spentTime": 0, ※未使用
"estimatedTime": 0, ※未使用
"remainingTime": 0, ※未使用
"goal": "", ※未使用
"alert": 0, ※未使用
"parentItemId": "<親タスクの ItemID>", ※依頼タスクのみで依頼元のタスクの ItemID を表す
"priority": <優先度(1~4)>,
"createdAt": "<メッセージ作成日時(yyyy/MM/dd HH:mm:ss 形式)>",
"replyId": "<引用元メッセージの ItemID>",
"replyTo": "<引用元メッセージの発言者 JID>",
"startDate": "<開始日時(yyyy/MM/dd HH:mm:ss 形式)>",
"dueDate": "<締切(終了)日時(yyyy/MM/dd HH:mm:ss 形式)>",
"completeDate": "<完了日時(yyyy/MM/dd HH:mm:ss 形式)>",
"owner": "<担当者の JID>",
"group": "<プロジェクトフィードのメッセージをインボックス化した場合はそのプロジェクトの
roomId、その他は空文字>",
"groupName": "<プロジェクトフィードのメッセージをインボックス化した場合はそのプロジェクト
名(UTF-8 で URI エンコード) 、その他は空文字>",
"client": "<依頼者の JID>",
"status": 1(インボックス) or 2(依頼タスク(依頼中)),
"updatedAt": "<最終更新日時(yyyy/MM/dd HH:mm:ss 形式)>",
"updatedBy": "<最終更新者の JID>",
"noteCount": 0, ※未使用
"noteItems": [], ※未使用
"reminderCount": 0, ※未使用
"reminderItems": [], ※未使用
"attachedCount": 0, ※未使用
"attachedItems": [], ※未使用
"siblingCount": 2,
"siblingItems": [
  {
    "siblingItemId": "<兄弟タスク ItemID>",
    "ownerJid": "<担当者の JID>",
    "ownerName": "<担当者のニックネーム(UTF-8 で URI エンコード)>",
    "status": 2<タスクの状態(2~8)>
  },
  { ... }
],
"context": "", ※未使用
"demandStatus": "催促状態(0:催促なし / 1:催促あり)",
"demandDate": "<催促日時>",
"deleteFlag": 0-2 の数値 (0:未削除, 1:完全削除(画面上に見えない), 2:論理削除(画面上に見える))
"shortenUrlCount": <shortenItems の配列のサイズ>,
"shortenItems": [
  {
    "DisplayedURL": "<パス部分を…に置換した URL>",
    "ShortenPath": "<短縮 URI の相対パス>",
    "OriginalURL": "<オリジナル URL>",

```

```

    }
  ],
  "personInfo": {
    "<jid>": {
      "userName": "<jid に対応するログインアカウント(UTF-8 で URI エンコード)>",
      "nickName": "<jid に対応するニックネーム(UTF-8 で URI エンコード)>",
      "avatarData": "<jid に対応するアバターファイルのパス (/cubee/からの相対ファイルパス)
>",
      "avatarType": "imagpath",
      "status": "<jid に対応するアカウント状態 0(通常)/1(削除済)/2:(休止(ロック))>",
    },
    "<jid>": { ... }
  }
},
"convertFileCount": <convertFileItems の配列のサイズ>
"convertFileItems": [
  {
    "originalURL": "<オリジナル画像の URL>",
    "convertURL": "<サムネイル画像の URL>",
  }
]
{<2 個目のデータ>},
...,
{<20 個目のデータ>}
]
}
}

```


3.7.2 インボックス追加

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "SendMessage",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "Task",
    "title": "",
    "body": "<インボックス内容(UTF-8 で URI エンコード)>",
    "progress": 0, ※未使用
    "spentTime": 0, ※未使用
    "estimatedTime": 0, ※未使用
    "remainingTime": 0, ※未使用
    "goal": "", ※未使用
    "alert": 0, ※未使用
    "parentItemId": "",
    "priority": 1,
    "replyId": "<引用元メッセージの ItemID、メッセージからインボックス化した場合のみ>",
    "replyTo": "<引用元メッセージの発言者 JID、メッセージからインボックス化した場合のみ>",
    "startDate": "",
    "dueDate": "",
    "owner": "<ログインユーザの JID>",
    "group": "<プロジェクトフィードのメッセージをインボックス化する場合はそのプロジェクトの roomId、
    その他は空文字>",
    "client": "<メッセージからインボックス化した場合はメッセージの投稿者の JID、それ以外はログインユーザの JID >",
    "status": 1,
    "attachedCount": 0, ※未使用
    "attachedItems": [], ※未使用
    "context": "" ※未使用
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "SendMessage",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": "Task",
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "extras": {
      "childrenCount": 0,
      "childrenItems": []
    },
  },
  "count": 1,
  "items": [
    {
      "id": 125,
      "itemId": "<メッセージの ItemID>",
      "from": "<投稿者の JID>",
      "to": "<送信先>",
      "type": 4(=Task),
      "title": "",
      "body": "<インボックス内容(UTF-8 で URI エンコード)>",
      "progress": 0, ※未使用
      "spentTime": 0, ※未使用
      "estimatedTime": 0, ※未使用
      "remainingTime": 0, ※未使用
      "goal": "", ※未使用
      "alert": 0, ※未使用
      "parentItemId": "<親タスクの ItemID>",
      "priority": 1,
      "createdAt": "<作成日時(yyyy/MM/dd HH:mm:ss 形式)>",
      "replyId": "<引用元メッセージの ItemID>",
      "replyTo": "<引用元メッセージの発言者 JID>",
      "startDate": "",
      "dueDate": "",
      "completeDate": "",
      "owner": "<ログインユーザの JID>",
      "group": "<プロジェクトフィードのメッセージをインボックス化した場合はそのプロジェクトの roomId、その他は空文字>",
      "groupName": "<プロジェクトフィードのメッセージをインボックス化した場合はそのプロジェクト名(UTF-8 で URI エンコード)、その他は空文字>",
      "client": "<メッセージからインボックス化した場合はメッセージの投稿者の JID、それ以外はログインユーザの JID>",
      "status": 1,
    }
  ]
}
```

```

"updatedAt": "<最終更新日時(yyyy/MM/dd HH:mm:ss 形式)>",
"updatedBy": "<最終更新者の JID>",
"noteCount": 0, ※未使用
"noteItems": [], ※未使用
"reminderCount": 0, ※未使用
"reminderItems": [], ※未使用
"attachedCount": 0, ※未使用
"attachedItems": [], ※未使用
"siblingCount": 0,
"siblingItems": [],
"context": "", ※未使用
"demandStatus": "催促状態(0:催促なし / 1:催促あり)",
"demandDate": "<催促日時>",
"deleteFlag": 0-2 の数値 (0:未削除, 1:完全削除(画面上に見えない), 2:論理削除(画面上に見える))
"shortenUrlCount": <shortenItems の配列のサイズ>,
"shortenItems": [
  {
    "DisplayedURL": "<パス部分を…に置換した URL>",
    "ShortenPath": "<短縮 URI の相対パス>",
    "OriginalURL": "<オリジナル URL>",
  }
],
"personInfo": {
  "<jid>": {
    "userName": "<jid に対応するログインアカウント(UTF-8 で URI エンコード)>",
    "nickName": "<jid に対応するニックネーム(UTF-8 で URI エンコード)>",
    "avatarData": "<jid に対応するアバターファイルのパス (/cubee/からの相対ファイルパス)>",
    "avatarType": "imagpath",
    "status": <jid に対応するアカウント状態 0(通常)/1(削除済)/2:(休止(ロック))>,
  },
  "<jid>": { ... }
}
"convertFileCount": <convertFileItems の配列のサイズ>
"convertFileItems": [
  {
    "originalURL": "<オリジナル画像の URL>",
    "convertURL": "<サムネイル画像の URL>",
  }
]
}
}
]
}
}
}

```

3.7.3 インボックスの削除

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "DeleteMessage",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "Delete",
    "itemId": "<削除するインボックスの ItemID>",
    "deleteFlag": 1
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "DeleteMessage",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": "Delete",
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>"
  }
}
```

※要求者が作成したインボックスのみ可能

3.8 Good Job

3.8.1 Good Job 要求

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "MessageOption",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "AddGoodJob",
    "itemId": "<Good Job をするメッセージの ItemID>"
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "MessageOption",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "type": "AddGoodJob"
  }
}
```

3.8.2 Good job したユーザ取得

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "MessageOption",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "GetGoodJobList",
    "itemId": "<GoodJob 者一覧を取得したいメッセージの ItemID>"
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "MessageOption",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "type": "GetGoodJobList",
    "itemCount": 2 (GoodJob 数) ,
    "items": [
      {
        "itemId": "<メッセージの ItemID>",
        "fromJid": "<Good Job した人の JID>",
        "fromName": "<Good Job した人のニックネーム(UTF-8 で URI エンコード)>",
        "date": "<Good Job した日時(yyyy/MM/dd HH:mm:ss 形式)>",
        "userName": "<Good Job した人のログインアカウント名>",
        "nickName": "<Good Job した人のニックネーム(UTF-8 で URI エンコード)>",
        "avatarData": "<Good Job した人のアバターデータ>",
        "avatarType": "<Good Job した人のアバタータイプ>",
        "status": "<Good Job した人のアカウント状態 0(通常)/1(削除済)/2:(休止(ロック))>"
      },
      {...}
    ]
  }
}
```

※"avatarType"について：「imagepath」固定

※"avatarData"について：cubee ドキュメントルート（通常/cubee/）からの相対ファイルパス

※例は items の 1 個目のみ詳細に記述

3.9 グループチャットチャットルーム

3.9.1 チャットルーム作成

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "CreateGroup",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "GroupChatRoom",
    "roomName": "<チャットルームの名前(UTF-8 URL エンコード済み)>",
    "parentRoomId": "<このルームの親ルーム ID（プロジェクト ID のみ許容）>",
    "memberCount": 3,
    "memberItems": [
      "<参加者 1 の jid>",
      "<参加者 2 の jid>",
      "<参加者 3 の jid>"
    ],
    "notifyType": "<0(すべて通知する)、1(すべて通知しない)>",
    "manageGroupchat": [ "<管理権限を持つユーザーID>" ],
    "sendMessageToGroupchat": [ "<投稿権限を持つユーザーID>" ],
    "viewMessageInGroupchat": [ "<閲覧権限を持つユーザーID>" ]
  }
}
```

※parentRoomId はプロジェクトに紐付く場合に、該当のプロジェクト ID を指定する。紐付がない場合は空文字。

※manageGroupchat はグループチャットの管理権限が紐付くユーザを指定する。

※sendMessageToGroupchat はグループチャットのかんり投稿/閲覧権限が紐付くユーザを指定する。

※viewMessageInGroupchat はグループチャットの閲覧権限のみが紐付くユーザを指定する。

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "CreateGroup",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": "GroupChatRoom",
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "extras": {},
    "count": 1,
    "items" [
      {

```


■ 異常系レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "CreateGroup",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": "GroupChatRoom",
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "extras": {}
  }
}
```

異常系コード表

Key	Type	Description
errorCode	9	token が無効の場合
	1	必須パラメータがない場合
	0	上記以外の場合
result	True	処理成功
	Fales	処理失敗 & アクセス権がない場合も
reason	200000	成功した場合
	5	失敗した場合 (BadRequest)
	404000	失敗した場合 (NotFound) (複数ユーザの DB 処理でエラーになって失敗ユーザリストが返ってくる場合)
	403000	アクセス権がない場合 (Forbidden)
	500000	DB エラー (サーバ内部エラー)

グループチャット作成時にエラーが起きると、チャットルームやユーザなども登録されず。上記のエラーコードが返される。

3.9.2 チャットルーム一覧取得

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetGroup",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "GroupChatRoomList",
    "parentRoomId": "<このルームの親ルーム ID（プロジェクト ID のみ許容）>",
    "startId": "<グループチャット一覧の取得開始番号(最新を取得する場合は 0 を指定)>",
    "count": 20,
    "condition": {
      "sort": {
        "item": "updated_at",
        "order": "2"
      }
    }
  }
}
```

※count は取得件数なので 1 以上であれば 20 以外でもよい

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetGroup",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": "GroupChatRoomList",
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "count": 20,
    "extras": {},
    "items" [
      {
        "id": 1234,
        "roomId": "<グループチャットの RoomID>",
        "roomName": "<チャットルームの名前(UTF-8 URL エンコード済み)>",
        "parentRoomId": "<このルームの親ルーム ID（プロジェクト ID のみ許容）>",
        "memberCount": 3,
        "memberItems": [
          "<参加者 1 の jid>",
          "<参加者 2 の jid>",
          "<参加者 3 の jid>"
        ]
      }
    ]
  }
}
```

Copyright © NEC Solution Innovators,Ltd.2018 All rights reserved.

```
],
"notifyType": <0(すべて通知する)、1(すべて通知しない)>,
"createdAt": "<チャットルーム作成日時(yyyy/MM/dd HH:mm:ss 形式)>",
"createdBy": "<チャットルームを作成した人の JID>",
"updatedAt": "<更新日時(yyyy/MM/dd HH:mm:ss 形式)>",
"updatedBy": "<最終更新者の JID>"
},
{<2 個目のデータ>},
...,
{<20 個目のデータ>}
]
}
}
```

3.9.3 チャットのルーム情報取得

■ リクエスト

```
{  
  "accessToken": "<アクセストークン>",  
  "request": "GetGroup",  
  "id": "<任意の文字列>",  
  "version": 1,  
  "content": {  
    "type": "GroupChatRoomInfo",  
    "roomId": "<グループチャットルーム ID>"  
  }  
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetGroup",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": "GroupChatRoomInfo",
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "extras": {},
    "count": 1,
    "items" [
      {
        "id": 1234,
        "roomId": "<グループチャットの RoomID>",
        "roomName": "<チャットルームの名前(UTF-8 URL エンコード済み)>",
        "parentRoomId": "<このチャットの親になるルーム ID (プロジェクトルーム ID) >",
        "memberCount": 3,
        "memberItems": [
          "<参加者 1 の jid>",
          "<参加者 2 の jid>",
          "<参加者 3 の jid>"
        ],
        "notifyType": "<0(すべて通知する)、1(すべて通知しない)>",
        "createdAt": "<チャットルーム作成日時(yyyy/MM/dd HH:mm:ss 形式)>",
        "createdBy": "<チャットルームを作成した人の JID>",
        "updatedAt": "<更新日時(yyyy/MM/dd HH:mm:ss 形式)>",
        "updatedBy": "<最終更新者の JID>",
        "personInfo": {
          "<jid>": {
            "userName": "<jid に対応するログインアカウント(UTF-8 で URI エンコード)>",
            "nickName": "<jid に対応するニックネーム(UTF-8 で URI エンコード)>",
            "avatarData": "<jid に対応するアバターファイルのパス (/cubee/からの相対ファイルパス)
>",
            "avatarType": "imagpath",
            "status": "<jid に対応するアカウント状態 0(通常)/1(削除済)/2:(休止(ロック))>",
          },
          "<jid>": { ... }
        }
      }
    ]
  }
}
```

3.9.4 チャットのメンバ追加

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "AddMember",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "GroupChatRoom",
    "roomId": "<グループチャットルーム ID>",
    "member": "<追加したい人の JID>"
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "AddMember",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": "GroupChatRoom",
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "extras": {},
    "count": 1,
    "items": [
      {
        "roomId": "<グループチャットルーム ID>",
        "roomName": "<チャットルームの名前(UTF-8 URL エンコード済み)>",
        "parentRoomId": "<このルームの親ルーム ID (プロジェクト ID のみ許容) >",
        "addedBy": "<追加した人の JID>",
        "count": 1,
        "members": [
          "<追加された人の jid>"
        ],
        "notifyType": 0,
        "personInfo": {
          "jid": {
            "userName": "<jid に対応するログインアカウント(UTF-8 で URI エンコード)>",
            "nickName": "<jid に対応するニックネーム(UTF-8 で URI エンコード)>",
            "avatarData": "<jid に対応するアバターファイルのパス (/cubee/からの相対ファイルパス)"
          },
          "avatarType": "imagpath",
          "status": "<jid に対応するアカウント状態 0(通常)/1(削除済)/2:(休止(ロック))>",
          "jid": { ... }
        }
      }
    ]
  }
}
```

3.9.5 チャットのメンバ削除

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "RemoveMember",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "GroupChatRoom",
    "removeType": "own" 自身の退会/"member":メンバーの退会
    "roomId": <GroupChat の RoomId>
    "count": 1,
    "members": [
      "<削除したい人 1 の jid>",
      "<削除したい人 2 の jid>",
      "<削除したい人 3 の jid>"
    ]
  }
}
```


■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "RemoveMember",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": "GroupChatRoom",
    "result": true/false, (全員失敗した場合のみ false)
    "reason": <失敗時の理由(result が false の場合)>,
    "extras": {
      removeType : "own" 自身の退会/"member":メンバーの退会 (各々chatroom_member_table の state
2(退出),3 (強制退出) に相当)
    },
    "count": 1,
    "items": [
      {
        "roomId": "<グループチャットルーム ID>",
        "roomName": "<チャットルームの名前(UTF-8 URL エンコード済み)>",
        "parentRoomId": "<このルームの親ルーム ID (プロジェクト ID のみ許容) >",
        "removedBy": "<削除操作を行ったユーザ JID>",
        "count": 1,
        "members": [
          "successMembers": [
            "<削除された人 1 の jid>",
            "<削除された人 2 の jid>"
          ],
          "failureMembers": [
            "<削除に失敗した人 1 の jid>",
            "<削除に失敗した人 2 の jid>"
          ]
        ]
      }
    ]
  }
}
```

3.9.6 チャットのルーム情報更新

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "UpdateGroup",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "GroupChatRoomInfo",
    "extras": {
      "subType": ["ChangeRoomName" or "ChangeImage" or "ChangeNotify" ],
    },
    "roomId": "<グループチャットルーム ID>",
    "roomName": "<チャットルームの名前(UTF-8 URL エンコード済み)>",
    "notifyType": <0(すべて通知する)、1(すべて通知しない)>
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "UpdateGroup",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": "GroupChatRoomInfo",
    "result": true/false,
    "reason": <失敗時の理由(result が false の場合)>
  }
}
```

3.9.7 グループチャットメッセージ取得

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetMessage",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "Search",
    "condition": {
      "filter": {
        "type": "and",
        "value": [
          {
            "type": "item",
            "name": "msgtype",
            "value": 3
          },
          {
            "type": "item",
            "name": "msgto",
            "value": "<グループチャットルーム ID>"
          }
        ]
      },
      "sort": {
        "item": "id",
        "order": "2"
      }
    },
    "startId": "<取得開始のメッセージの ID(最新を取得する場合は 0 を指定)>",
    "count": 20
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetMessage",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "type": "Search",
    "extras": {
```

```

    "allItemCount": "<グループチャットメッセージの全ての数>",
    "childrenCount": 0,
    "childrenItems": [],
  },
  "count": 20,
  "items": [
    {
      "id": メッセージの(DB上の)ID,
      "itemId": "<メッセージの ItemID>",
      "from": "<投稿者の JID>",
      "to": "<グループチャットルーム ID>",
      "type": 3(=グループチャットメッセージ),
      "roomName": "<グループチャットのルームの名前(UTF-8 で URI エンコード)>",
      "body": "<メッセージ内容(UTF-8 で URI エンコード)>",
      "createdAt": "<投稿日時(yyyy/MM/dd HH:mm:ss 形式)>",
      "replyId": "<リプライメッセージはリプライ元メッセージの ItemID>",
      "replyTo": "リプライの宛先 JID",
      "attachedCount": 0,
      "attachedItems": [],
      "context": "",
      "goodJobCount": 2,
      "goodJobItems": [
        {
          "fromJid": "<Good Job した人の JID>",
          "fromName": "<Good Job した人のニックネーム(UTF-8 で URI エンコード)>",
          "date": "<Good Job した日時(yyyy/MM/dd HH:mm:ss 形式)>",
          "userName": "<Good Job した人のログインアカウント名>",
          "nickName": "<Good Job した人のニックネーム(UTF-8 で URI エンコード)>",
          "avatarData": "<Good Job した人のアバターファイルのパス (/cubee/からの相対ファイルパス)
>",
          "avatarType": "imagepath"
          "status": <Good Job した人のアカウント状態 0(通常)/1(削除済)/2:(休止(ロック))>
        },
        { ... }
      ],
      "deleteFlag": 0-2 の数値 (0:未削除, 1:完全削除(画面上に見えない), 2:論理削除(画面上に見える)),
      "readFlg": 0/1,(※ログインユーザがそのメッセージを読んだかどうか 0:未読 1:既読)
      "readAllCount": 100, (※件数は全既読者数)
      "readItems": [
        {
          "jid": "<メッセージを既読にした人の JID>",
          "account": "<メッセージを既読にした人のログインアカウント(UTF-8 で URI エンコード)>",
          "nickName": "<メッセージを既読にした人のニックネーム(UTF-8 で URI エンコード)>",
          "avatarData": "<メッセージを既読にした人のアバターファイルのパス (/cubee/からの相対ファ
イルパス) >",
          "avatarType": "imagpath",
          "date": "<メッセージを既読にした日時(yyyy/MM/dd HH:mm:ss 形式)>"
        },
      ],

```

```

    { ... }
  ],
  "shortenUrlCount" : <shortenItems の配列のサイズ>,
  "shortenItems" : [
    {
      "UrlId" : "<URL の ID>",
      "DisplayedURL" : "<パス部分を…に置換した URL>",
      "ShortenPath" : "<短縮 URI の相対パス>",
      "OriginalURL" : "<オリジナル URL>",
      "_expandedURL" : null
    }
  ],
  "personInfo" : {
    "<jid>" : {
      "userName" : "<jid に対応するログインアカウント(UTF-8 で URI エンコード)>",
      "nickName" : "<jid に対応するニックネーム(UTF-8 で URI エンコード)>",
      "avatarData" : "<jid に対応するアバターファイルのパス (/cubee/からの相対ファイルパス)>",
      "avatarType" : "imagpath",
      "status" : <jid に対応するアカウント状態 0(通常)/1(削除済)/2:(休止(ロック))>,
    },
    "<jid>" : { ... }
  }
},
"convertFileCount" : <convertFileItems の配列のサイズ>
"convertFileItems" : [
  {
    "originalURL" : "<オリジナル画像の URL>",
    "convertURL" : "<サムネイル画像の URL>",
  }
]
{<2 個目のデータ>},
...,
{<20 個目のデータ>}
]
}
}

```

※count, items は result が true の場合のみ

※例は items の 1 個目のみ詳細に記述

3.9.8 グループチャットメッセージ投稿

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "SendMessage",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "GroupChat",
    "roomId": "<グループチャットの RoomID>",
    "body": "<メッセージ内容(UTF-8 で URI エンコード)>",
    "replyId": "<リプライ時はリプライ元メッセージの ItemID>",
    "replyTo": "<リプライの宛先 JID>",
    "attachedCount": 0,
    "attachedItems": [],
    "context": ""
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "SendMessage",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": "GroupChat",
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "extras": {},
    "count": 1,
    "items": [
      {
        "id": メッセージの(DB上の)ID,
        "itemId": "<メッセージの ItemID>",
        "from": "<投稿者の JID>",
        "to": "<グループチャットルーム ID>",
        "type": 3(=グループチャットメッセージ),
        "roomName": "<グループチャットのルームの名前(UTF-8 で URI エンコード)>",
        "parentRoomId": "<このルームの親ルーム ID (プロジェクト ID のみ許容)>",
        "body": "<メッセージ内容(UTF-8 で URI エンコード)>",
        "createdAt": "<投稿日時(yyyy/MM/dd HH:mm:ss 形式)>",
        "replyId": "<リプライメッセージはリプライ元メッセージの ItemID>",
        "replyTo": "リプライの宛先 JID",
        "attachedCount": 0,
        "attachedItems": [],
        "context": "",
        "goodJobCount": 2,
        "goodJobItems": [],
        "deleteFlag": 0-2 の数値 (0:未削除, 1:完全削除(画面上に見えない), 2:論理削除(画面上に見える)),
        "readFlg": 0/1,(※ログインユーザがそのメッセージを読んだかどうか 0:未読 1:既読)
        "readAllCount": 100, (※件数は全既読者数)
        "readItems": [],
        "shortenUrlCount": <shortenItems の配列のサイズ>,
        "shortenItems": [
          {
            "UrlId": "<URL の ID>",
            "DisplayedURL": "<パス部分を…に置換した URL>",
            "ShortenPath": "<短縮 URI の相対パス>",
            "OriginalURL": "<オリジナル URL>",
            "_expandedURL": null
          }
        ],
        "personInfo": {
          "<jid>": {
```

```
"userName": "<jid に対応するログインアカウント(UTF-8 で URI エンコード)>",  
"nickName": "<jid に対応するニックネーム(UTF-8 で URI エンコード)>",  
"avatarData": "<jid に対応するアバターファイルのパス (/cubee/からの相対ファイルパス)  
>",  
  
"avatarType": "imagpath",  
"status": <jid に対応するアカウント状態 0(通常)/1(削除済)/2:(休止(ロック))>,  
},  
"<jid>": { ... }  
"convertFileCount": <convertFileItems の配列のサイズ>  
"convertFileItems": [  
  {  
    "originalURL": "<オリジナル画像の URL>",  
    "convertURL": "<サムネイル画像の URL>",  
  }  
]  
}  
}  
]  
}
```


3.10 プロジェクト

3.10.1 プロジェクトの作成

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "CreateGroup",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "CommunityRoom",
    "roomName": "<プロジェクトの名前(UTF-8 URL エンコード済み)>",
    "description": "<プロジェクトの説明(UTF-8 URL エンコード済み)>",
    "privacyType": 2(秘密),
    "memberEntryType": "<RGB カラー(16 進数で指定)>",
    "logoUrl": "<プロジェクトロゴの URL>",
    "notifyType": <0(すべて通知する)、1(すべて通知しない)>
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "CreateGroup",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": "CommunityRoom",
    "result": true/false,
    "reason": <失敗時の理由(result が false の場合)>,
    "extras": {},
    "count": 1,
    "items": [
      {
        "id": 2222,
        "roomId": "<プロジェクトの RoomID>",
        "roomName": "<プロジェクトの名前(UTF-8 URL エンコード済み)>",
        "description": "<プロジェクトの説明(UTF-8 URL エンコード済み)>",
        "memberCount": 1,
        "privacyType": 2(秘密),
        "memberEntryType": "<RGB カラー(16 進数で指定) FFFFFFFF 形式>",
        "logoUrl": "<プロジェクトロゴの URL>",
        "notifyType": <0(すべて通知する)、1(すべて通知しない)>,
        "createdAt": "<プロジェクト作成日時(yyyy/MM/dd HH:mm:ss 形式)>",
        "createdBy": "<プロジェクトを作成した人の JID>",
        "updatedAt": "<更新日時(yyyy/MM/dd HH:mm:ss 形式)>"
      }
    ]
  }
}
```

Copyright © NEC Solution Innovators,Ltd.2018 All rights reserved.

```

        "updatedBy" : "<最終更新者の JID>"
    }
}
}
}

```

■ 異常系レスポンス

```

{
  "accessToken" : "<アクセストークン>",
  "request" : "CreateGroup",
  "id" : "<要求時の id>",
  "version" : 1,
  "errorCode" : 0,
  "content" : {
    "type" : " CommunityRoom ",
    "result" : true/false,
    "reason" : <失敗時の理由(result が false の場合)>,
    "extras" : {}
  }
}

```

異常系コード表

Key	Type	Description
errorCode	9	token が無効の場合
	1	パラメータが不正の場合
	0	上記以外の場合
result	True	処理成功
	Fales	処理失敗 & アクセス権がない場合も
reason	200000	成功した場合
	5	失敗した場合 (BadRequest)
	404000	失敗した場合 (NotFound) (複数ユーザの DB 処理でエラーになって失敗ユーザリストが返ってくる場合)
	403000	アクセス権がない場合 (Forbidden)
	500000	DB エラー (サーバ内部エラー)

3.10.2 参加プロジェクト一覧の取得

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetGroup",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "MyCommunityList",
    "startId": "<プロジェクト一覧の取得開始の ID(最新を取得する場合は 0 を指定)>",
    "count": 20,
    "condition": {
      "sort": {
        "item": "updated_at",
        "order": "2"
      }
    }
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetGroup",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": "MyCommunityList",
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "extras": {},
    "count": 20,
    "items": [
      {
        "id": 2222,
        "roomId": "<プロジェクトの RoomID>",
        "roomName": "<プロジェクトの名前(UTF-8 URL エンコード済み)>",
        "description": "<プロジェクトの説明(UTF-8 URL エンコード済み)>",
        "memberCount": 1,
        "privacyType": 2(秘密),
        "memberEntryType": "<RGB カラー(16 進数で指定) FFFFFFFF 形式>",
        "logoUrl": "<プロジェクトロゴの URL>",
        "notifyType": "<0(すべて通知する)、1(すべて通知しない)>",
        "createdAt": "<プロジェクト作成日時(yyyy/MM/dd HH:mm:ss 形式)>",
        "createdBy": "<プロジェクトを作成した人の JID>",
        "updatedAt": "<更新日時(yyyy/MM/dd HH:mm:ss 形式)>",
        "updatedBy": "<最終更新者の JID>"
      },
      {<2 個目のデータ>},
      ...,
      {<20 個目のデータ>}
    ]
  }
}
```

※この API ではメンバー一覧を返さない。別途プロジェクトメンバー一覧取得 API にて取得する

3.10.3 プロジェクトの詳細情報取得

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetGroup",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "CommunityInfo",
    "roomId": "<プロジェクトルーム ID>"
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetGroup",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": "CommunityInfo",
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "extras": {},
    "count": 1,
    "items": [
      {
        "id": 2222,
        "roomId": "<プロジェクトの RoomID>",
        "roomName": "<プロジェクトの名前(UTF-8 URL エンコード済み)>",
        "description": "<プロジェクトの説明(UTF-8 URL エンコード済み)>",
        "memberCount": 1,
        "privacyType": 2(秘密),
        "memberEntryType": "<RGB カラー(16 進数で指定) FFFFFFFF 形式>",
        "logoUrl": "<プロジェクトロゴの URL>",
        "notifyType": "<0(すべて通知する)、1(すべて通知しない)>",
        "createdAt": "<プロジェクト作成日時(yyyy/MM/dd HH:mm:ss 形式)>",
        "createdBy": "<プロジェクトを作成した人の JID>",
        "updatedAt": "<更新日時(yyyy/MM/dd HH:mm:ss 形式)>",
        "updatedBy": "<最終更新者の JID>"
      }
    ]
  }
}
```

※この API ではメンバー一覧を返さない。必要であれば、別途プロジェクトメンバ情報取得 API にて取得すること

3.10.4 プロジェクトメンバ情報取得

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetGroup",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "CommunityMemberInfo",
    "roomId": "<プロジェクトルーム ID>"
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetGroup",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": "CommunityMemberInfo",
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "extras": {},
    "count": 1,
    "items": [
      {
        "id": 2222,
        "roomId": "<プロジェクトの RoomID>",
        "memberCount": 10,
        "memberItems": {
          "ownerItems": [
            {
              "jid": "<コンタクトリストのユーザの JID>",
              "userName": "<ログインアカウント名>",
              "nickName": "<ニックネーム(UTF-8 で URI エンコード)>",
              "avatarType": "imagpath",
              "avatarData": "<アバターファイルのパス (/cubee/からの相対ファイルパス) >",
              "status": 0(通常)/1(削除済)/2:(休止(ロック)),
              "role": 2(Owner)
            },
            {<2 個目のデータ>},
            ...
          ]
        },
        "generalMemberItems": [
```

Copyright © NEC Solution Innovators,Ltd.2018 All rights reserved.

```

{
  "jid" : "<コンタクトリストのユーザの JID>",
  "userName" : "<ログインアカウント名>",
  "nickName" : "<ニックネーム(UTF-8 で URI エンコード)>",
  "avatarType" : "imagpath",
  "avatarData" : "<アバターファイルのパス (/cubee/からの相対ファイルパス) >",
  "status" : 0(通常)/1(削除済)/2:(休止(ロック)),
  "role" : 1(GeneralMember)
},
{<2 個目のデータ>},
...
]
}
}
]
}
}

```

3.10.5 プロジェクトメンバ追加

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "AddMember",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "CommunityRoom",
    "roomId": "<プロジェクトの RoomID>",
    "count": <追加したい人数>,
    "members": ["<追加したい人の JID>",...]
  ]
}
```


■ レスポンス

```
{
  "accessToken" : "<アクセストークン>",
  "request" : "AddMember",
  "id" : "<要求時の id>",
  "version" : 1,
  "errorCode" : 0,
  "content" : {
    "type" : "CommunityRoom",
    "result" : true/false,
    "reason" : <失敗時の理由(result が false の場合)>,
    "extras" : {},
    "count" : 2,
    "items" : [
      {
        "roomId" : "<プロジェクトの RoomID>",
        "roomName" : "<プロジェクトの名前(UTF-8 URL エンコード済み)>",
        "addedBy" : "<追加した人の JID>",
        "count" : 5,
        "notifyType" : 0,
        "members" : [
          {
            "jid" : "<コンタクトリストのユーザの JID>",
            "userName" : "<ログインアカウント名>",
            "nickName" : "<ニックネーム(UTF-8 で URI エンコード)>",
            "avatarType" : "imagpath",
            "avatarData" : "<アバターファイルのパス (/cubee/からの相対ファイルパス) >",
            "status" : 0(通常)/1(削除済)/2(休止(ロック)),
            "role" : <役割 1:一般ユーザ、2:管理者>
          },
          {<2 個目のデータ>},
          ...,
          {<5 個目のデータ>}
        ]
      }
    ]
  }
}
```

3.10.6 プロジェクトメンバ削除

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "RemoveMember",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "CommunityRoom",
    "roomId": "<プロジェクトの RoomID>",
    "count": <削除したい人数>,
    "members": [
      "<削除したい人の JID>",
      ...
    ]
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "RemoveMember",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": "CommunityRoom",
    "result": true/false,
    "reason": <失敗時の理由(result が false の場合)>,
    "extras": {},
    "count": 1,
    "items": [
      {
        "roomId": "<プロジェクトの RoomID>",
        "roomName": "<プロジェクトの名前(UTF-8 URL エンコード済み)>",
        "removedBy": "<削除した人の JID>",
        "count": 5,
        "members": [
          "<削除された人 1 の jid>",
          "<削除された人 2 の jid>",
          ...,
          "<削除された人 5 の jid>",
        ]
      }
    ]
  }
}
```

3.10.7 プロジェクト情報更新

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "UpdateGroup",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "CommunityRoomInfo",
    "roomId": "<プロジェクトルーム ID>",
    "roomName": "<プロジェクトの名前(UTF-8 URL エンコード済み)>",
    "description": "<プロジェクトの説明(UTF-8 URL エンコード済み)>",
    "privacyType": 2(秘密),
    "memberEntryType": "<RGB カラー(16 進数で指定)>",
    "logoUrl": "<プロジェクトロゴの URL>",
    "notifyType": <0(すべて通知する)、1(すべて通知しない)>
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "UpdateGroup",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": "CommunityRoomInfo",
    "result": true/false,
    "reason": <失敗時の理由(result が false の場合)>,
    "extras": {
      "preInfo": {
        "id": 2222,
        "roomId": "<プロジェクトの RoomID>",
        "roomName": "<変更前のプロジェクトの名前(UTF-8 URL エンコード済み)>",
        "description": "<変更前のプロジェクトの説明(UTF-8 URL エンコード済み)>",
        "memberCount": 10,
        "privacyType": 2(秘密),
        "memberEntryType": "<RGB カラー(16 進数で指定) FFFFFFFF 形式>",
        "logoUrl": "<変更前のプロジェクトロゴの URL>",
        "notifyType": <0(すべて通知する)、1(すべて通知しない)>
      }
    },
    "count": 1,
    "items": [
      {
        "id": 2222,
```

```

    "roomId": "<プロジェクトの RoomID>",
    "roomName": "<プロジェクトの名前(UTF-8 URL エンコード済み)>",
    "description": "<プロジェクトの説明(UTF-8 URL エンコード済み)>",
    "memberCount": 10,
    "privacyType": 2(秘密),
    "memberEntryType": "<RGB カラー(16 進数で指定) FFFFFFFF 形式> ",
    "logoUrl": "<プロジェクトロゴの URL>",
    "notifyType": <0(すべて通知する)、1(すべて通知しない)>,
    "createdAt": "<プロジェクト作成日時(yyyy/MM/dd HH:mm:ss 形式)>",
    "createdBy": "<プロジェクトを作成した人の JID>",
    "updatedAt": "<更新日時(yyyy/MM/dd HH:mm:ss 形式)>",
    "updatedBy": "<最終更新者の JID>"
  }
]
}
}

```

■ 異常系レスポンス

```

{
  "accessToken": "<アクセストークン>",
  "request": " UpdateGroup ",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": <失敗時の理由(result が false の場合)>,
    "extras": {}
  }
}

```

異常系コード表

Key	Type	Description
errorCode	9	token が無効の場合
	1	パラメータが不正の場合
	0	上記以外の場合
result	True	処理成功
	Fales	処理失敗 & アクセス権がない場合も
reason	200000	成功した場合
	400000	失敗した場合 (BadRequest)
	404000	失敗した場合 (NotFound) (複数ユーザの DB 処理でエラーになって失敗ユーザリストが返ってくる場合)
	403000	アクセス権がない場合 (Forbidden)
	500000	DB エラー (サーバ内部エラー)

3.10.8 プロジェクトオーナー変更

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "UpdateMember",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "CommunityOwner",
    "roomId": "<プロジェクトルーム ID>",
    "ownerCount": <オーナーの数>,
    "ownerItems": ["<プロジェクトオーナー 1 の JID>",...]
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "UpdateMember",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": "CommunityOwner",
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "extras": {
      "preOwnerCount": "<変更前のプロジェクトオーナーの数>"
      "preOwnerItems": ["<変更前のプロジェクトオーナー1 の JID>",...]
    },
    "count": 1,
    "items": [
      {
        "roomId": "<プロジェクトルーム ID>",
        "ownerCount": "<オーナーの数>",
        "ownerItems": ["<プロジェクトオーナー 1 の JID>",...]
      }
    ]
  }
}
```

3.10.9 プロジェクトフィードメッセージ取得

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetMessage",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "Search",
    "condition": {
      "filter": {
        "type": "and",
        "value": [
          {
            "type": "item",
            "name": "msgtype",
            "value": 5(プロジェクトの種別)
          },
          {
            "type": "item",
            "name": "msgto",
            "value": "<プロジェクト ID>"
          }
        ]
      },
      "sort": {
        "item": "id",
        "order": "2"
      }
    },
    "startId": "<取得開始のメッセージの ID(最新を取得する場合は 0 を指定)>",
    "count": 20
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetMessage",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "type": "Search",
    "extras": {
```

```

    "allItemCount": "<プロジェクトフィードメッセージの全ての数>",
    "childrenCount": 0,
    "childrenItems": [],
  },
  "count": 20,
  "items": [
    {
      "id": メッセージの ID
      "itemId": "<メッセージの ItemID>",
      "from": "<投稿者の JID>",
      "to": "<プロジェクト ID>",
      "type": 5(メッセージタイプ),
      "roomName": "<プロジェクト名(UTF-8 で URI エンコード)>",
      "body": "<メッセージ内容(UTF-8 で URI エンコード)>",
      "createdAt": "<投稿日時(yyyy/MM/dd HH:mm:ss 形式)>",
      "replyId": "<リプライメッセージはリプライ元メッセージの ItemID>",
      "replyTo": "リプライの宛先 JID",
      "attachedCount": 0, ※未使用
      "attachedItems": [], ※未使用
      "context": "<コンテキスト(UTF-8 で URI エンコード)>",
      "goodJobCount": 2,
      "goodJobItems": [
        {
          "fromJid": "<Good Job した人の JID>",
          "fromName": "<Good Job した人のニックネーム(UTF-8 で URI エンコード)>",
          "date": "<Good Job した日時(yyyy/MM/dd HH:mm:ss 形式)>",
          "userName": "<Good Job した人のログインアカウント名>",
          "nickName": "<Good Job した人のニックネーム(UTF-8 で URI エンコード)>",
          "avatarData": "<Good Job した人のアバターファイルのパス (/cubee/からの相対ファイルパス)
>",
          "avatarType": "imagepath"
          "status": <Good Job した人のアカウント状態 0(通常)/1(削除済)/2:(休止(ロック))>
        },
        { ... }
      ],
      "deleteFlag": 0-2 の数値 (0:未削除, 1:完全削除(画面上に見えない), 2:論理削除(画面上に見える)),
      "readFlg": 0/1,(※ログインユーザがそのメッセージを読んだかどうか 0:未読 1:既読)
      "readAllCount": 100, (※件数は全既読者数)
      "readItems": [
        {
          "jid": "<メッセージを既読にした人の JID>",
          "account": "<メッセージを既読にした人のログインアカウント(UTF-8 で URI エンコード)>",
          "nickName": "<メッセージを既読にした人のログインアカウント(UTF-8 で URI エンコード)>",
          "avatarData": "<メッセージを既読にした人のアバターファイルのパス (/cubee/からの相対ファ
イルパス) >",
          "avatarType": "imagpath",
          "date": "<メッセージを既読にした日時(yyyy/MM/dd HH:mm:ss 形式)>"
        },
      ],

```



```

    { ... }
  ],
  "shortenUrlCount" : <shortenItems の配列のサイズ>,
  "shortenItems" : [
    {
      "UrlId" : "<URL の ID>",
      "DisplayedURL" : "<パス部分を…に置換した URL>",
      "ShortenPath" : "<短縮 URI の相対パス>",
      "OriginalURL" : "<オリジナル URL>",
      "_expandedURL" : null
    }
  ],
  "personInfo" : {
    "<jid>" : {
      "userName" : "<jid に対応するログインアカウント(UTF-8 で URI エンコード)>",
      "nickName" : "<jid に対応するニックネーム(UTF-8 で URI エンコード)>",
      "avatarData" : "<jid に対応するアバターファイルのパス (/cubee/からの相対ファイルパス)>",
      "avatarType" : "imagpath",
      "status" : <jid に対応するアカウント状態 0(通常)/1(削除済)/2:(休止(ロック))>,
    },
    "<jid>" : { ... }
  }
},
"convertFileCount" : <convertFileItems の配列のサイズ>
"convertFileItems" : [
  {
    "originalURL" : "<オリジナル画像の URL>",
    "convertURL" : "<サムネイル画像の URL>",
  }
]
{<2 個目のデータ>},
...,
{<20 個目のデータ>}
]
}
}

```

3.10.10 プロジェクトフィードメッセージ投稿

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "SendMessage",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "Community",
    "roomId": "<プロジェクトの RoomID>",
    "body": "<メッセージ内容(UTF-8 で URI エンコード)>",
    "replyId": "<リプライ時はリプライ元メッセージの ItemID>",
    "replyTo": "<リプライの宛先 JID>",
    "attachedCount": 0,
    "attachedItems": [],
    "context": "<コンテキスト(UTF-8 で URI エンコード)>"
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "SendMessage",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": "Community",
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "extras": {},
    "count": 1,
    "items": [
      {
        "id": メッセージの ID
        "itemId": "<メッセージの ItemID>",
        "from": "<投稿者の JID>",
        "to": "<プロジェクト ID>",
        "type": 5(メッセージタイプ),
        "roomName": "<プロジェクト名(UTF-8 で URI エンコード)>",
        "body": "<メッセージ内容(UTF-8 で URI エンコード)>",
        "createdAt": "<投稿日時(yyyy/MM/dd HH:mm:ss 形式)>",
        "replyId": "<リプライメッセージはリプライ元メッセージの ItemID>",
        "replyTo": "リプライの宛先 JID",
        "attachedCount": 0, ※未使用
        "attachedItems": [], ※未使用
        "context": "<コンテキスト(UTF-8 で URI エンコード)>",

```

Copyright © NEC Solution Innovators,Ltd.2018 All rights reserved.

```

"goodJobCount" : 0,
"goodJobItems" : [],
"deleteFlag" : 0-2 の数値 (0:未削除, 1:完全削除(画面上に見えない), 2:論理削除(画面上に見える)),
"readFlg" : 0/1,(※ログインユーザがそのメッセージを読んだかどうか 0:未読 1:既読)
"readAllCount" : 0, (※件数は全既読者数)
"readItems" : [],
"shortenUrlCount" : <shortenItems の配列のサイズ>,
"shortenItems" : [
  {
    "UrlId" : "<URL の ID>",
    "DisplayedURL" : "<パス部分を…に置換した URL>",
    "ShortenPath" : "<短縮 URI の相対パス>",
    "OriginalURL" : "<オリジナル URL>",
    "_expandedURL" : null
  }
],
"personInfo" : {
  "<jid>" : {
    "userName" : "<jid に対応するログインアカウント(UTF-8 で URI エンコード)>",
    "nickName" : "<jid に対応するニックネーム(UTF-8 で URI エンコード)>",
    "avatarData" : "<jid に対応するアバターファイルのパス (/cubee/からの相対ファイルパス)>",

    "avatarType" : "imagpath",
    "status" : <jid に対応するアカウント状態 0(通常)/1(削除済)/2:(休止(ロック))>,
  },
  "<jid>" : { ... }
"convertFileCount" : <convertFileItems の配列のサイズ>
"convertFileItems" : [
  {
    "originalURL" : "<オリジナル画像の URL>",
    "convertURL" : "<サムネイル画像の URL>",
  }
]
}
}
]
}
}

```

3.11 メッセージ

3.11.1 新着メッセージ取得

■ リクエスト

```
{
  "accessToken":"<アクセストークン>",
  "request":"GetMessage",
  "id":"<任意の文字列>",
  "version":1,
  "content":{
    "type":"Search",
    "condition":{
      "filter":{
        "type":"and",
        "value":[
          {
            "type":"and",
            "value":[
              {
                "type":"item",
                "name":"msgtype",
                "value":1
              },
              {
                "type":"keyword",
                "value":"%40<ログインユーザのアカウント名>(例：アカウントが honda の場合は「%40hoinda」)",
                "include":[]
              }
            ]
          },
          {
            "type":"and",
            "value":[
              {
                "type":"item",
                "name":"msgtype",
                "value":2
              },
              {
                "type":"or",
                "value":[
                  {
                    "type":"item",
                    "name":"msgfrom",
                    "value":"<ログインユーザの JID>"
                  }
                ]
              }
            ]
          }
        ]
      }
    }
  }
}
```

```

    },
    {
      "type":"item",
      "name":"msgto",
      "value":"<ログインユーザの JID>"
    }
  ]
}
]
},
{
  "type":"item",
  "name":"msgtype",
  "value":3
},
{
  "type":"and",
  "value":[
    {
      "type":"item",
      "name":"msgtype",
      "value":4
    },
    {
      "type":"item",
      "name":"owner",
      "value":"<ログインユーザの JID>"
    },
    {
      "type":"and",
      "value":[
        {
          "type":"not",
          "value":{
            "type":"item",
            "name":"status",
            "value":1
          }
        },
        {
          "type":"or",
          "value":[
            {
              "type":"not",
              "value":{
                "type":"item",
                "name":"status",
                "value":2

```

```

        }
    },
    {
        "type": "item",
        "name": "parent_item_id",
        "value": ""
    }
]
}
]
}
]
},
{
    "type": "and",
    "value": [
        {
            "type": "item",
            "name": "msgtype",
            "value": 4
        },
        {
            "type": "item",
            "name": "owner",
            "value": "<ログインユーザの JID>"
        },
        {
            "type": "or",
            "value": [
                {
                    "type": "item",
                    "name": "status",
                    "value": 1
                },
                {
                    "type": "and",
                    "value": [
                        {
                            "type": "item",
                            "name": "status",
                            "value": 2
                        },
                        {
                            "type": "not",
                            "value": {
                                "type": "item",
                                "name": "parent_item_id",
                                "value": ""
                            }
                        }
                    ]
                }
            ]
        }
    ]
}

```

```

        }
    }
}
]
}
]
},
{
    "type":"item",
    "name":"msgtype",
    "value":5
},
{
    "type":"item",
    "name":"msgtype",
    "value":9
}
]
},
{
    "type":"particular",
    "name":"NotNotified",
    "value":"<ユーザ JID>"
}
]
},
"sort":{
    "item":"id",
    "order":"2"
}
},
"startId":"<取得開始のメッセージの ID(最新を取得する場合は 0 を指定)>",
"count":20
}
}

```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetMessage",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "type": "Search",
    "extras": {※1
      "allItemCount": "<新着メッセージの全ての数>",
      "childrenCount": "<items 中のタスクの子タスクの数>",
      "childrenItems": [子タスクの情報（仕様は下の items の中と同じで内容が子タスクである(ただし、兄弟タスク情報は含まれない)）]
    },
    "count": 20,
    "items": [
      ※2
    ]
  }
}
```

※1 : extras の中身

childrenCount : 取得新着メッセージのうち、タスクについてそのタスクを親にもつ子タスクの一覧の件数

childrenItems : 新着メッセージのうち、タスクについてそのタスクを親にもつ子タスクの一覧。形式は items のタスク 1 件の形式の配列

※2 : items の中身

メッセージの種類によって異なる。1 つのメッセージに対する詳細は各種類のメッセージ取得 API のレスポンスの items の中身のものと同等

3.11.2 メッセージの削除（フィード、チャット、グループチャット、プロジェクトフィード）

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "DeleteMessage",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "Delete",
    "itemId": "<削除するメッセージの ItemID>",
    "deleteFlag": "<完全削除:1 論理削除:2>"
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "DeleteMessage",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "type": "Delete",
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>"
  }
}
```

3.11.3 メッセージの検索

API 基本構造

```
{
  "accessToken":"<アクセストークン>",
  "request":"GetMessage",
  "id":"<任意の文字列>",
  "version":1,
  "content":{
    "type":"Search",
    "condition":{
      "filter":{
        ※ここの部分は 3.11.5 の検索サンプルを参照
      },
      "sort":{
        "item":"<<ソート項目>> ",
        "order":"<<昇順=1, 降順=2>>"
      }
    },
    "startId": <取得開始のメッセージの ID(最新を取得する場合は 0 を指定)>,
    "count":20
  }
}
```

3.11.4 検索リクエストサンプルと Json ブロック解説

メッセージを検索するサンプル

JID「**aoi@tenantid**」さんが、メッセージ作成日 (**created_at**) が「**2018/02/05 00:00:00**」から「**2018/02/06 00:00:00**」の間に投稿したメッセージのうち、「**nec**」「**test**」「**best**」を含むメッセージを検索する場合、以下の Json となる。

```
{
  "value": [
    {
      "type": "item",
      "name": "msgfrom",
      "value": "aoi@tenantid "
    },
    {
      "value": [
        {
          "type": "keyword",
          "value": "nec "
        },
        {
          "type": "keyword",
          "value": "test "
        },
        {
          "type": "keyword",
          "value": "best "
        }
      ],
      "type": "or"
    }
  ],
  {
    "type": "greaterthan",
    "name": "created_at",
    "value": "2018/02/05 00:00:00"
  },
  {
    "type": "lessthan",
    "name": "created_at",
    "value": "2018/02/06 00:00:00"
  },
  {
    "value": [
      {
        "type": "item",
        "name": "msgtype",
        "value": "<1=MyWorkplace 内, 2=チャット内, 3=GC 内 4=タスク内, 5=プロジェクト内>"
      }
    ],
    {

```

```

        {
            "type":"item",
            "name":"msgtype",
            "value":"<1=MyWorkplace 内, 2=チャット内, 3=GC 内 4=タスク内, 5=プ
プロジェクト内>"
        }
    ],
    "type":"or"
}
],
"type":"and"
}

```

Json ブロック解説

■構造ブロック

”type”:"and”の場合は同列の"value"の配列内が AND 検索

”type”:"or”の場合は同列の"value"の配列内が OR 検索

"value" の設定が配列 ([]) の時は最低 2 ブロックは必要

■値ブロック（値ブロックを構造ブロックの配列の中に使う）

“type”: ----- 検索の値のタイプ

例)

"type"の値が "item" のとき "name"の値を持ち、"name"に設定の値を"value"が持つ

"type"の値が "keyword" 文字列検索、文字列は URL エンコードされたものが必須。"name" 設定は不要

"type"の値が"greaterthan"の時は "name"を必須で持ちその値を"value"に数字、日時などを入れ、その値以上に該当するメッセージを検索する

"type"の値が"lessthan"の時は"name"を必須で持ちその値を"value"に数字、日時などを入れ、その値以下に該当するメッセージを検索する

“name”: ----- "value"の値が示すもの（type が keyword の時は使用しない）

"name"に使用できる値について

msgfrom = JID ユーザー指定の時利用

msgtype = メッセージのタイプ

1 = フィード

2 = チャット

3 = グループチャット

4 = タスク

5 = プロジェクトフィード

created_at = メッセージの作成日時（フォーマット：2018/02/06 00:00:00）

“value”: ----- name が指定されていればその値、name が無いときは type に関する値

3.11.5 メッセージの応答

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetMessage",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "type": "Search",
    "extras": {
      "allItemCount": 121,
      "childrenCount": "<items 中のタスクの子タスクの数>",
      "childrenItems": [子タスクの情報（仕様は下の items の中と同じで内容が子タスクである(ただし、兄弟タスク情報は含まれない)）]
    },
    "count": 20,
    "items": [
      ※
    ]
  }
}
```

- extras の中身
 - フィルター条件に当てはまった全ての数(allItemCount)
 - フィルター条件に当てはまったメッセージのうち、タスクについてそのタスクを親にもつ子タスクの一覧(childrenItems)
- items の中身
 - メッセージの種類によって異なる。1つのメッセージに対する詳細は各種別のメッセージ取得APIの items の中身のものと同等

3.12 スレッド

3.12.1 スレッドメッセージ一覧取得（フィード、グループチャット、プロジェクトフィード）

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetMessage",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "Thread",
    "itemId": "<取得したいスレッドに含まれるメッセージの ItemID>"
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "GetMessage",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "type": "Thread",
    "extras": {},
    "count": 13,
    "items": [
      ※
    ]
  }
}
```

※items の中身はフィード取得、チャット取得、グループチャットメッセージ取得、プロジェクトフィードメッセージ取得 API のレスポンスの items の中身のものと同等

3.13 メッセージの既読

3.13.1 既読情報の取得

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "MessageOption",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "GetExistingReaderList",
    "itemId": "<既読者を取得したいメッセージの ItemID>"
  }
}
```

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "MessageOption",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": "<失敗時の理由(result が false の場合)>",
    "type": "GetExistingReaderList",
    "itemCount": 2,
    "items": [
      {
        "jid": "<メッセージを既読にした人の JID>",
        "account": "<メッセージを既読にした人のログインアカウント(UTF-8 で URI エンコード)>",
        "nickName": "<メッセージを既読にした人のログインアカウント(UTF-8 で URI エンコード)>",
        "avatarData": "imagepath",
        "avatarType": "<メッセージを既読にした人のアバターファイルのパス (/cubee/からの相対ファイルパス) >",
        "date": "<メッセージを既読にした日時(yyyy/MM/dd HH:mm:ss 形式)>"
      },
      {...}
    ]
  }
}
```

3.13.2 既読要求

■ リクエスト

```
{
  "accessToken": "<アクセストークン>",
  "request": "MessageOption",
  "id": "<任意の文字列>",
  "version": 1,
  "content": {
    "type": "SetReadMessage",
    "itemCount": 1<同時に複数のメッセージを既読にしたい場合はその数>,
    "items": [
      {
        "itemId": "<既読にするメッセージの ItemID>"
      } ※
    ]
  }
}
```

※同時に複数のメッセージを既読にしたい場合はその数だけ{"itemId": "<既読にするメッセージの ItemID>"}を繰り返す

■ レスポンス

```
{
  "accessToken": "<アクセストークン>",
  "request": "MessageOption",
  "id": "<要求時の id>",
  "version": 1,
  "errorCode": 0,
  "content": {
    "result": true/false,
    "reason": <失敗時の理由(result が false の場合)>,
    "type": "SetReadMessage",
    "itemCount": 1<同時に複数のメッセージを既読にした場合はその数>,
    "items": [
      {
        "itemId": "<既読にするメッセージの ItemID>"
      } ※
    ]
  }
}
```

※同時に複数のメッセージを既読にした場合はその数だけ{"itemId": "<既読にするメッセージの ItemID>"}を繰り返す

3.14 スマートデバイス向け端末情報（Android : Registration ID / iOS : Device Token）

3.14.1 端末情報の登録

■ リクエスト

```
{
  "accessToken" : "<アクセストークン>",
  "request" : "SetLoginPersonData",
  "id" : "<任意の文字列>",
  "version" : 1,
  "content" : {
    "type" : "RegisterDeviceInfo",
    "deviceId" : "<Android : Registration ID / iOS : Device Token>",
    "notificationService" : "<通知サービス先( 1:GCM / 2:APNS )>"
  }
}
```

■ レスポンス

```
{
  "accessToken" : "<アクセストークン>",
  "request" : "SetLoginPersonData",
  "id" : "<要求時の id>",
  "version" : 1,
  "errorCode" : 0,
  "content" : {
    "type" : "RegisterDeviceInfo",
    "result" : true/false,
    "reason" : "<失敗時の理由(result が false の場合)>"
  }
}
```

3.14.2 端末情報の削除

■ リクエスト

```
{
  "accessToken" : "<アクセストークン>",
  "request" : "SetLoginPersonData",
  "id" : "<任意の文字列>",
  "version" : 1,
  "content" : {
    "type" : "DeleteDeviceInfo",
    "deviceId" : "<Android : Registration ID / iOS : Device Token>"
  }
}
```

■ レスポンス

```
{
  "accessToken" : "<アクセストークン>",
  "request" : "SetLoginPersonData",
  "id" : "<要求時の id>",
  "version" : 1,
  "errorCode" : 0,
  "content" : {
    "type" : "DeleteDeviceInfo",
    "result" : true/false,
    "reason" : "<失敗時の理由(result が false の場合)>"
  }
}
```

4 添付ファイル API

4.1 添付ファイルアップロード

4.1.1 アクセス先

以下の URL に POST メソッドでアクセスする。

- `http(s)://<cube サイトのホスト名 or IP アドレス>/<cube_root>/fileupload`

※https は SSL をサポートしている時のみ

4.1.2 リクエスト

WebAPI と異なり、form データとして以下を送信する。

Name	説明
accesstoken	ログイン時に取得するアクセストークン
uploadfile	アップロードするファイル

※20MB 以下のファイルのみアップロード可能(1MB=1024x1024B で計算)

4.1.3 レスポンス

➤ 成功時

```
{
  "result": success,
  "path": file/<格納ファイルパス>/<URL エンコードしたファイル名>,
  "filename": <URL エンコードしたファイル名>
}
```

➤ 失敗時

```
{
  "result": failed
}
```

4.2 添付ファイルダウンロード

4.2.1 アクセス先

以下の URL に POST メソッドでアクセスする。

- `http(s)://<cube サイトのホスト名 or IP アドレス>/<cube_root>/filedownload`

※https は SSL をサポートしている時のみ

4.2.2 リクエスト

WebAPI と異なり、form データとして以下を送信する

Name	説明
downloadURL	ダウンロード URL Ex.) <code>http(s)://<hostname>/<cube_root>/file/<格納ファイルパス>/<URL エンコードしたファイル名></code>
accesstoken	ログイン時に取得するアクセストークン
itemId	メッセージを一意に特定する ID

4.2.3 レスポンス

➤ 成功時

ファイルデータが返却される。

ResponseHeaders に以下の情報が格納される。

Name	説明
Content-Disposition	<code>Attachment;filename="<URL エンコードされたファイル名>"</code>
Content-Length	ファイルサイズ
Content-Type	ファイル種別

➤ 失敗時（必要なパラメータがない場合、またはアクセストークンが不正な場合）

```
{  
  "result": failed  
}
```

※この場合でも、200 の HTTP レスポンスコードを返却する。

5 短縮 URL

5.1 短縮 URL リダイレクト

5.1.1 アクセス先

以下の URL に POST メソッドでアクセスする。

- `http(s)://<cube サイトのホスト名 or IP アドレス>/<cube_root>/redir/<短縮 URL キー>`

※https は SSL をサポートしている時のみ

5.1.2 リクエスト

WebAPI と異なり、form データとして以下を送信する。

Name	説明
accesstoken	ログイン時に取得するアクセストークン
itemId	短縮 URL を掲載しているメッセージの ID

5.1.3 レスポンス

➤ 成功時

※HTTP レスポンスコード 301

※HTTP レスポンスヘッダとして下記を返却

```
{
  "Location": URL にて渡された <短縮 URL キー> に紐づく正規の URL
}
```

➤ 失敗時

※HTTP レスポンスコード 404