

Taskrow

API – 09/2025

Introdução

Este manual descreve os principais métodos da API para integração de usuários, cliente, projetos e tarefas. Bem como os detalhes de autenticação dos usuários

Versão 25.9.2

Conteúdo

Introdução.....	i
Chaves de acesso	2
Criando uma chave de acesso.....	2
Utilizando uma chave de acesso.....	3
Usuários	4
Listar usuários	4
Incluir um novo usuário	4
Obter detalhes de um usuário	8
Editar os dados de contato de um usuário	8
Editar as informações adicionais de um usuário	9
Editar os dados administrativos de um usuário.....	11
Listar itens que estão sob a responsabilidade de um usuário	12
Listar as etnias cadastradas na empresa	13
Clientes.....	14
Criar clientes	14
Criar endereço / CNPJ	15
Obter detalhes do cliente	17
Projetos.....	18
Lista projetos.....	18
Criar projetos	19
Obter detalhes de um projeto	20
Mudar status de um projeto.....	21
Tarefas.....	23
Criar uma nova tarefa	23
Obter detalhes de uma tarefa	24
Salvar uma subtarefa (item de checklist de uma tarefa)	25
Salvar uma subtarefa (tarefa subordinada a uma outra tarefa).....	26
Unidades Produzidas.....	28
Obter unidades produzidas de uma tarefa	28
Registrar unidade produzida.....	29
Timesheet	31



Buscas	33
Busca de Clientes	33
Busca de Projetos.....	34
Busca de Tarefas	35
Cadastros auxiliares	38
Listar Cidades	38
Listar Áreas de um cliente.....	38
Listar Subtipos de projeto	39



Chaves de acesso

Criando uma chave de acesso

Toda chamada da API é feita em nome de um usuário, de modo que eventualmente será necessário criar um usuário exclusivo para integrações. Você pode criar este usuário normalmente como qualquer outro usuário, associar um email, se logar na aplicação, e na página de perfil criar uma nova chave de API / Mobile:

Descrição	Chave	Data	IP
YWhQCHZziBMTIzamvnc2HjdnHub XZmIG1hcZQxMjNqa+dzYZN2cZ5tb HNzZmRmMjModMTIzlyRrcWVvJAGtdl UajjE83ZW5jaW8xMjMzZG5zZA		29/05/2015	200.153.167.14

Provedor	Serviço	Descrição	Autorizado
Google	Google Drive	Acesso ao google drive (antigo google docs)	<input checked="" type="checkbox"/>

Esta será a chave a ser incluída em cada uma das requisições que serão realizadas. É importante mantê-la em segredo da mesma forma que você age com suas senhas.



Utilizando uma chave de acesso

A chave de acesso deve ser incluída como um item do cabeçalho das requisições:

```
"__identifier": "YWohQCN2ZiBtMTIzamvnc2NRr...cWwjJGtxIUAjJEB3ZW5jaW8xMjMzbG5zZA=="
```



Usuários

Usuários são todos os colaboradores, terceiros e freelancers que façam uso do taskrow.

Abaixo estão os métodos suportados

Listar usuários

Listar usuários

URL: [https://\[HOST_NAME\].taskrow.com/api/v1/User/ListUsers](https://[HOST_NAME].taskrow.com/api/v1/User/ListUsers)

Método: GET

Retorno:

```
[
  {
    "UserID": 123,
    "FullName": "Ricardo Santos Silva",
    "MainEmail": "ricardo.silva@empresa.com.br",
    "UserLogin": "rsilva",
    "UserHashCode": "8523697410",
    "ExternalCode": null,
    "Inactive": false,
    "ProfileID": 12,
    "ProfileTitle": "Equipe",
    "ApprovalGroup": "Desenvolvimento",
    "ProfileRate": 9,
    "FunctionGroupName": "Produto",
    "UserFunctionTitle": "Arquiteto de Software"
    //...demais propriedades de um usuário
  }
]
```

Incluir um novo usuário

Inclusão de Usuário

URL: [https://\[HOST_NAME\].taskrow.com/api/v1/User/SaveNewUser](https://[HOST_NAME].taskrow.com/api/v1/User/SaveNewUser)

Método: POST



Campo	Detalhe
AppMainCompanyID *	99999 //ID da empresa no taskrow
AppMainLanguageID *	1 //Idioma do usuário - fixo 1
FullName *	"Nome completo" //Nome do usuário (string até 100 caracteres)
UserLogin *	"Nome de exibição" //Nome para exibição no sistema (string até 60 caracteres)
MainEmail *	"usuario@empresa.com.br" //Email do usuário, (string com até 100 caracteres)
ProfileID *	99 //Id do perfil de acesso, determina quais permissões o usuário terá no taskrow
selectedGroupID *	999 //Grupo em que o usuário será inserido (determina quem aprova as ações do usuário no Taskrow)
DateStart *	31/12/2015 //Data de início do usuário no sistema (string representando a data no formato dd/MM/aaaa)
UserRelationTypeID *	99 //Tipo do relacionamento do usuário com a empresa (determina a quantidade horas diárias do usuário)
UserFunctionID *	9999 //Função/cargo do usuário. Determina qual a área e qual função o usuário desempenhará na empresa
OfficelID *	99 //ID da unidade da empresa
MainPhoneNumber	
MainCellNumber	
Extension	
UserDetail.PhoneNumber2	
UserDetail.CellNumber2	
UserDetail.email2	
UserDetail.HomeAddress	



UserDetail.HomeAddressDetail	
UserDetail.HomeAddressNumber	
UserDetail.HomeDistrict	
UserDetail.HomeCity	
UserDetail.HomeState	
UserDetail.HomeZipCode	
UserDetail.Information	
UserDetail.Gender	0 //Gênero não informado
UserDetail.CivilStatus	
UserDetail.CPF	
UserDetail.RG	
UserDetail.RGDate	
UserDetail.Reservist	
UserDetail.UserScholarityID	
UserDetail.CityBirth	
UserDetail.StateBirth	
UserDetail.DateBirth	
UserDetail.MotherName	
UserDetail.FatherName	
UserAdministrativeDetail.UnionName	
UserAdministrativeDetail.PIS Pasep	
UserAdministrativeDetail.NumberCTPS	



UserAdministrativeDetail.SerieCTPS	
UserAdministrativeDetail.UFCTPS	
UserAdministrativeDetail.DateCTPS	
UserAdministrativeDetail.TituloEleitor	
UserAdministrativeDetail.TituloEleitorSecao	
UserAdministrativeDetail.TituloEleitorZona	
UserAdministrativeDetail.TituloEleitorZona	
UserAdministrativeDetail.Information	
BankNumber	
BankName	
BankAgencyNumber	
BankAccountNumber	
BankAccountInfo	
ExternalCode	"XXXX" //String com código externo do usuário para futuras integrações

Retorno: JSON

```
{
  Success: true/false, //Indica se a criação foi bem sucedida ou não
  Message: "Msg de erro", //Mensagem descritiva de um eventual erro
  Entity: {
    UserID: 9999 //ID do usuário no sistema taskrow
  }
}
```



Obter detalhes de um usuário

Detalhe do usuário

URL:**https://[HOST_NAME].taskrow.com/api/v1/User/GetUserByExternalCode****Método: GET**

Campo	Detalhe
externalCode	"99999" //Código externo do usuário, para uso na integração.

Retorno JSON

```
{
  User: {
    UserID: 9999, //Id do usuário
    FullName, "XXXXX XXXX", //Nome completo do usuário
    UserLogin, "XXXX", //Nome de exibição do usuário
  }
}
```

Editar os dados de contato de um usuário

Edição dos dados de contato de um Usuário**URL:** [https://\[HOST_NAME\].taskrow.com/api/v1/User/SaveUser](https://[HOST_NAME].taskrow.com/api/v1/User/SaveUser)**Método: POST**

Campo	Valor
UserID	99999 //ID do usuário
Section	Contact
UserDetail.PhoneNumber2	Telefone fixo
UserDetail.CellNumber2	Celular



UserDetail.email2	Email adicional
UserDetail.HomeAddress	Endereço residencial
UserDetail.HomeAddressDetail	Complemento
UserDetail.HomeAddressNumber	Número (aceita letras e números)
UserDetail.HomeDistrict	Bairro
UserDetail.HomeCity	Cidade
UserDetail.HomeState	Estado
UserDetail.HomeZipCode	CEP
UserDetail.Information	

Retorno: JSON

```
{  
    Success: true/false, //Indica se a edição foi bem sucedida ou não  
    Message: "Msg de erro", //Mensagem descritiva de um eventual erro  
}
```

Editar as informações adicionais de um usuário

Edição das informações adicionais de um Usuário

URL: [https://\[HOST_NAME\].taskrow.com/api/v1/User/SaveUser](https://[HOST_NAME].taskrow.com/api/v1/User/SaveUser)

Método: POST

Campo	Descrição
UserID	99999 //ID do usuário
Section	Detail



UserDetail.Gender	0 //Gênero não informado ou ID segundo cadastro da empresa
UserDetail.CivilStatus	Estado civil: 1 - Solteiro(a), 2 - Casado(a), 3 - divorciado(a), 4 - União estável, 5 - Separado(a), 6 - Viúvo(a)
UserDetail.CPF	CPF
UserDetail.RG	RG (Número e Emissor)
UserDetail.RGDate	Data de emissão do RG
UserDetail.Reservist	Reservista
UserDetail.UserScholarityID	Grau de formação: 1 - Ensino Fundamental Incompleto, 2 - Ensino Fundamental, 3 - Ensino Médio Incompleto, 4 - Ensino Médio, 5 - Técnico, 6 - Ensino Superior Incompleto, 7 - Ensino Superior, 8 - Mestrado, 9 - Doutorado, 10 - Outro, 11 - Pós-Graduação, 12 - Pós-Graduação incompleta
UserDetail.CityBirth	Cidade de nascimento
UserDetail.StateBirth	Estado de nascimento
UserDetail.DateBirth	Data de nascimento (dd/mm/yyyy)
UserDetail.MotherName	Nome da mãe
UserDetail.FatherName	Nome do pai
UserDetail.EthnicityID	O ID de uma das etnias cadastradas na empresa
UserDetail.Dependents	Qtd de dependentes

Retorno: JSON

```
{  
    Success: true/false, //Indica se a edição foi bem sucedida ou não  
    Message: "Msg de erro", //Mensagem descritiva de um eventual erro  
}
```



Editar os dados administrativos de um usuário

Edição dos dados administrativos de um Usuário

URL: [https://\[HOST_NAME\].taskrow.com/api/v1/User/SaveUser](https://[HOST_NAME].taskrow.com/api/v1/User/SaveUser)

Método: POST

Campo	Detalhe
UserID	99999 //ID do usuário
Section	administrative
UserAdministrativeDetail.UnionName	Sindicato (nome, região-uf)
UserAdministrativeDetail.PIS Pasep	Número do PIS-PASEP
UserAdministrativeDetail.NumberCTPS	Número da CTPS
UserAdministrativeDetail.SerieCTPS	Série da CTPS
UserAdministrativeDetail.UF CTPS	UF de emissão da CTPS
UserAdministrativeDetail.DateCTPS	Data de emissão da CTPS
UserAdministrativeDetail.TitleEleitor	Número do título eleitoral
UserAdministrativeDetail.TitleEleitorSecao	Seção do título eleitoral
UserAdministrativeDetail.TitleEleitorZona	Zona do título eleitoral
UserAdministrativeDetail.Information	Informações adicionais
UserAdministrativeDetail.VacationPeriodDateStart	Data inicial do controle de férias



UserAdministrativeDetail.VacationPeriodInitialBalance	Dias de férias adquiridos na data inicial
UserAdministrativeDetail.CompTimeDateStart	Data inicial do controle de horas adicionais
UserAdministrativeDetail.CompTimeInitialBalanceHours	Saldo de horas na data inicial

Retorno: JSON

```
{
  Success: true/false, //Indica se a edição foi bem sucedida ou não
  Message: "Msg de erro", //Mensagem descritiva de um eventual erro
}
```

Listar itens que estão sob a responsabilidade de um usuário

Para que um usuário seja inativado, é necessário que todos os itens que estejam sob sua responsabilidade sejam transferidas para um outro usuário. Para isso, este método lista todos estes itens.

Listar clientes, projetos e tarefas que estão sob a responsabilidade de um usuário

URL:

[https://\[HOST_NAME\].taskrow.com/api/v1/User/GetUserPendingEntities](https://[HOST_NAME].taskrow.com/api/v1/User/GetUserPendingEntities)

Método: GET

Campo	Detalhe
userID	99 //ID do usuário

Retorno JSON

```
{
  Success: true,
  Entity: {
    Clients: [],
    Jobs: [],
    Tasks: [],
    NeedNewOwner: false //Indica se este usuário ao ser inativado
                        //necessita que seja designado um novo responsável
  }
}
```



```
        // pelos seus itens
    }
}
```

Listar as etnias cadastradas na empresa

Detalhe do usuário

URL:

[https://\[HOST_NAME\].taskrow.com/api/v1/Administrative/ListCompanyEthnicity](https://[HOST_NAME].taskrow.com/api/v1/Administrative/ListCompanyEthnicity)

Método: GET

Retorno JSON

```
{
  "EthnicityList": [
    {
      "EthnicityID": 99,
      "Name": "XXXX"
    },
    {
      "EthnicityID": 200,
      "Name": "XXXX"
    }
  ]
}
```




Clientes

Empresas são todos os clientes ou prospects ou fornecedores, para as quais serão criadas tarefas, contratos, orçamentos, etc. Todo item do taskrow é relacionado a um cliente. Sua própria empresa é incluída como “uma empresa” de modo que seja possível a criação de projetos internos.

Abaixo estão os métodos suportados:

Criar clientes

Inclusão de Cliente

URL: [https://\[HOST_NAME\].taskrow.com/api/v1/Client/SaveClient](https://[HOST_NAME].taskrow.com/api/v1/Client/SaveClient)

Método: POST

Campo	Detalhe
CNPJ	
CityID ²	7352 //ID DA CIDADE – ver lista de cidades nos cadastros auxiliares
ClientID	0 //Fixo, sempre 0
ClientName *	NOME DO CLIENTE //Nome do cliente em string com até 80 caracteres
Complement ²	
ContactEmail	
ContactMainPhone	
ContactName	
CountryID: *	31
DisplayName: *	NOME DO CLIENTE //Nome do cliente em string com até 80 caracteres
District ²	
InscrEstad ²	
InscrMunic ²	
IsSupplier	false //Fixo, sempre False
Location ²	



MemberListString:	9999,9999,9999 //Lista dos usuários participantes deste cliente ('UserID's do Taskrow)
Memo	
NoCNPJ	false //Fixo, sempre False
Number ²	
OwnerID:	99999 //UserID do usuário responsável pelo Cliente (UserID do Taskrow)
SegmentListString	
SocialContractName	
State ²	
Street ²	
ZipCode ²	
ExternalCode:	[CODIGO EXTERNO] //Campo string (máximo 40), com código de sistema externo para viabilizar integrações

OBS:

- Campos com * são obrigatórios
- Campos com ² são opcionais, mas caso algum deles seja preenchido, todos os outros deverão

Retorno: JSON

```
{
  Success: true/false, //Indica se a criação foi bem sucedida ou não
  Message: "Msg de erro", //Mensagem descritiva de um eventual erro
  Entity: {
    ClientID: 9999 //ID do cliente no sistema taskrow
    ClientNickname: 'XXXXXXX' //Nome da 'pasta' do cliente no Taskrow - será
    utilizado em uma série de outras requisições
  }
}
```

Criar endereço / CNPJ

Todo cliente pode ter 1 ou mais endereços. Cada endereço tem um CNPJ

Inclusão de endereço / CNPJ de um cliente

**URL:****https://[HOST_NAME].taskrow.com/api/v1/Client/SaveClientAddress****Método: POST**

Campo	Detalhe
ClientID	9999 //ID do cliente retornado pelo método SaveClient
ClientNickName	"XXXXX" //Nome da 'pasta' do cliente no Taskrow retornado pelo método SaveClient
ClientAddressID	0 //Fixo zero
CNPJ	00.000.000/0001-01
CountryID	31 //Fixo 31 = Brasil
Street	"Avenida Paulista" //1ª linha do endereço
Number	1337 //Número
Complement	51 //Complemento
District	"Cerqueira César" //Bairro
CityID	7352 //Cidade
State	"São Paulo" //Estado
ZipCode	"01311-200" //CEP
Location	"-23.5634,-46.654743" //Coordenadas
ProductTag	"Tag do produto"
NoCNPJ	FALSE
ExternalCode	"XXXXX" //Código externo

Retorno JSON

```
{
  Success: true, //
  Entity: {
    ClientAddressID: 999, //ID do endereço/CNPJ
    ClientID: 999 //ID do cliente
  }
}
```



Obter detalhes do cliente

Detalhe do cliente

URL:

https://[HOST_NAME].taskrow.com/api/v1/Client/GetClientByExternalCode

Método: GET

Campo	Detalhe
externalCode	"99999" //Código externo do cliente, para uso na integração.

Retorno JSON

```
{
  Client: {
    ClientID: 99999, //ID do cliente
    ClientNickname: "XXXXXXX", //Pasta do cliente
    //... demais campos
  }
}
```



Projetos

No Taskrow, um projeto é a maior unidade de trabalho que pode ser realizado. Pode ser um projeto no sentido tradicional da palavra, isto é, com orçamento, início e término preestabelecidos, mas também pode ser uma conta de serviços com um orçamento mensal recorrente, um projeto de prospecção sem fonte de receita, ou por fim um projeto de atividades internas, também sem receita prevista.

Lista projetos

Lista de projetos (Job / Campanha)

URL: [https://\[HOST_NAME\].taskrow.com/api/v2/core/job/list](https://[HOST_NAME].taskrow.com/api/v2/core/job/list)

Método: GET

Parâmetro (Query String)	Detalhe
includeInactives	true / false (opcional)
orderBy	Title / TitleDesc / Number / NumberDesc (opcional)
nextToken	Token de continuação (opcional)
ClientID	ID do cliente (opcional)

Retorno:

```
{
  "items": [
    {
      "jobID": 9999,
      "jobNumber": 888,
      "jobTitle": "Projeto de contratos simples",
      "inactive": false,
      "externalCode": "XX95",
      "client": {
        "clientID": 879,
        "clientNickname": "EmpresaXyz",
        "displayName": "Empresa XYZ",
        "inactive": false,
        "colorID": 0
      }
    }
  ],
}
```



```
"nextToken": "XXXX989912" //Valor a ser fornecido na chamada seguinte para listagem dos próximos registros
}
```

Criar projetos

Inclusão de Projeto (Job / Campanha)

URL: [https://\[HOST_NAME\].taskrow.com/api/v1/Job/SaveJob](https://[HOST_NAME].taskrow.com/api/v1/Job/SaveJob)

Método: POST

Campo	Detalhe
ClientID *	9999 //Id do cliente no sistema taskrow
HealthOftenRequired	
HealthReference	
InternalNumber	
IsPrivate	false //Indicando se é um job sigiloso
JobDetail.AdmMemo	
JobDetail.Memo	
JobTitle *	"Job do Natal" //Título do job
JobTypeID *	1 Atividade Geral 2 Valor fixo 3 Recorrente 4 Prospecção 7 Suporte interno
MemberListString *	999,999,999 //Lista de Ids dos usuários participantes do job (Ids do taskrow)
MonthEnd	
MonthStart	
OwnerUserID *	999 //ID do usuário responsável pelo job



TagListString	Produto1,Online //Lista de tags para classificação do job. OBS: Não requer cadastro anterior
clientNickName *	"NomeDaPastaDoCliente" //Nome da pasta do cliente. Este é o nome retornado pelo taskrow na criação do cliente.
ExternalCode	99999 //Código da empresa para o job
ClientAreaID *	999 //ID da área do cliente
JobSubTypeID *	9999 //ID do subtipo de projeto

OBS:

- * campos obrigatórios

Retorno: JSON

```
{
  Success: true/false, //Indica se a criação foi bem sucedida ou não
  Message: "Msg de erro", //Mensagem descritiva de um eventual erro
  Entity: {
    JobID: 9999 //ID do usuário no sistema taskrow
    JobNumber: 99999 //Número sequencia do projeto. Este valor será requisitado
em futuras chamadas
  }
}
```

Obter detalhes de um projeto

Detalhe do projeto pelo código de integração

URL:

[https://\[HOST_NAME\].taskrow.com/api/v1/Job/GetJobByExternalCode](https://[HOST_NAME].taskrow.com/api/v1/Job/GetJobByExternalCode)

Método: GET

Campo	Detalhe
externalCode	"99999" //Código externo do projeto, para uso na integração.

Retorno JSON

```
{
  Job: {
    Client: {
```



```
ClientID: 99999, //ID do cliente
ClientNickname: "XXXXXXX", //Pasta do cliente
//... demais campos do cliente
},
JobID: 99999, //ID do job
JobNumber: 999999, //Número do job
JobTitle: "PROJETO XXXXXX", //Nome do projeto
//... demais campos do projeto
}
}
```

Informações do projeto pelo ID

URL: [https://\[HOST_NAME\].taskrow.com/api/v1/Job/GetJobInfoByJobID](https://[HOST_NAME].taskrow.com/api/v1/Job/GetJobInfoByJobID)

Método: GET

Campo	Detalhe
jobID	99999 //ID do projeto

Retorno JSON

```
{
  JobID: 99999, //ID do job
  JobNumber: 999999, //Número do job
  JobTitle: "PROJETO XXXXXX", //Nome do projeto
  ClientID: 9999, //ID do cliente
  ClientNickname: 'CLIENTEXYZ', //Parte da URL do cliente
  ClientDisplayName: 'CLIENTE X.Y.Z', //Nome de exibição do cliente
  //... demais campos do projeto
}
```

Mudar status de um projeto

Mudança de status de projeto

URL: [https://\[HOST_NAME\].taskrow.com/api/v1/Job/UpdateJobStatus](https://[HOST_NAME].taskrow.com/api/v1/Job/UpdateJobStatus)

Método: GET

Campo	Detalhe
clientNickName	Xpto //Nome da pasta do cliente



jobNumber	Número do projeto
status	4 = bloqueado, 3 = liberado

Retorno JSON

```
{  
  Success: true,  
  Message: ""  
}
```



Tarefas

Tarefas são o coração do taskrow, representam algo que deva ser realizado, tem um ciclo de vida curto, isto é, são criadas, direcionadas a alguém, eventualmente encaminhadas, assim que o trabalho é realizado, e considerado finalizado, o solicitante pode – e deve – fechar a tarefa.

Criar uma nova tarefa

Inclusão de Tarefa (ação)

URL: [https://\[HOST_NAME\].taskrow.com/api/v1/Task/SaveTask](https://[HOST_NAME].taskrow.com/api/v1/Task/SaveTask)

Método: POST

Campo	Detalhe
DueDate	2015-09-03T00:00:00 //Prazo (data em formato canônico, o sistema não suporta horas diferente de zero)
EffortEstimation	0 //Estimativa inicial de esforço em minutos
EffortUnitListString	
JobID	99999 //ID do job em que a tarefa será criada
MemberListString	
OwnerUserID	89999 //ID do usuário responsável pela tarefa
RequestContactID	
TagListString	
TaskItemComment	"FAZER PEÇA" //Texto inicial da tarefa (briefing) - string sem limite de tamanho, pode conter algumas tags HTML como font, p, div, b, em e outras que não representem riscos de segurança
TaskTitle	"Título da tarefa" //Título da tarefa
UserAllocationID	



jobNumber	999 //Número do job, retornado na criação do mesmo
ExternalCode	"99999" //Código externo da tarefa, para uso na integração

Retorno: JSON

```
{
  Success: true/false, //Indica se a criação foi bem sucedida ou não
  Message: "Msg de erro", //Mensagem descritiva de um eventual erro
  Entity: {
    TaskID: 9999 //ID da tarefa no sistema taskrow
    TaskNumber: 99999 //Número sequencial da tarefa. Este valor será requisitado
em futuras chamadas
  }
}
```

Obter detalhes de uma tarefa

Detalhe da tarefa, buscando pelo external code

URL:

[https://\[HOST_NAME\].taskrow.com/api/v1/Task/GetTaskByExternalCode](https://[HOST_NAME].taskrow.com/api/v1/Task/GetTaskByExternalCode)

Método: GET

Campo	Detalhe
externalCode	"99999" //Código externo da tarefa, para uso na integração.

Retorno JSON

```
{
  Task: {
    TaskNumber: 99999, //Número sequencial da tarefa
    TaskID: 99999, //ID da tarefa
    TaskTitle: "XXXXX". //Título da tarefa
    JobID: 99999, //ID do job
    //... demais informações da tarefa
  }
}
```

**Detalhe da tarefa, buscando pelo número da tarefa****URL:** [https://\[HOST_NAME\].taskrow.com/api/v1/Task/TaskDetail](https://[HOST_NAME].taskrow.com/api/v1/Task/TaskDetail)**Método:** GET

Campo	Detalhe
clientNickname	"URL_DO_CLIENTE" //Parte da URL do cliente
jobNumber	999 //Número do projeto
taskNumber	999 //Número da tarefa
connectionID	"GUID" //Que identificar a conexão do usuário

Retorno JSON

```
{
  TaskData: {
    TaskNumber: 99999, //Número sequencial da tarefa
    TaskID: 99999, //ID da tarefa
    TaskTitle: "XXXXX". //Título da tarefa
    JobID: 99999, //ID do job
    //... demais informações da tarefa
  }
}
```

Salvar uma subtarefa (item de checklist de uma tarefa)**Salvar uma subtarefa****URL:** [https://\[HOST_NAME\].taskrow.com/api/v1/Task/SaveSubtask](https://[HOST_NAME].taskrow.com/api/v1/Task/SaveSubtask)**Método:** POST

Campo	Detalhe
subtask.TaskID	"99999" //ID da tarefa.
subtask.SubtaskID	0
subtask.Title	"Título da subtarefa"
subtask.Done	"false"

Retorno JSON



```
{
  Entity: {
    SubtaskID: 9999, //ID da subtarefa criada
    //... demais informações da subtarefa
    Task1: {
      RowVersion: "999999" //versão atualizada da tarefa principal
      //....demais informações da tarefa principal
    }
  }
}
```

Salvar uma subtarefa (tarefa subordinada a uma outra tarefa)

Salvar uma subtarefa

URL: [https://\[HOST_NAME\].taskrow.com/api/v1/Task/SaveTask](https://[HOST_NAME].taskrow.com/api/v1/Task/SaveTask)

Método: POST

Campo	Detalhe
DueDate	2015-09-03T00:00:00 //Prazo (data em formato canônico, o sistema não suporta horas diferente de zero)
EffortEstimation	0 //Estimativa inicial de esforço em minutos
EffortUnitListString	
JobID	99999 //ID do job em que a tarefa será criada
MemberListString	99999,77777 //Ids dos participantes da tarefam incluído o responsável
OwnerUserID	89999 //ID do usuário responsável pela tarefa
RequestContactID	
TagListString	
TaskItemComment	"FAZER PEÇA" //Texto inicial da tarefa (briefing) - string sem limite de tamanho, pode conter algumas tags HTML como font, p, div, b, em e outras que não representem riscos de segurança



jobNumber	999 //Número do job, retornado na criação do mesmo
ExternalCode	"99999" //Código externo da tarefa, para uso na integração
TaskID	999 //ID da tarefa
TaskNumber	9999 //Número da tarefa
RowVersion	"99999999" //versão atualizada da tarefa principal
createChildTask	true //fixo
SubtaskID	99999 //ID da subtarefa criada no passo anterior
Title	"Título da nova tarefa"
RequestTypeID	9999 //ID do tipo de solicitação da nova tarefa

Retorno: JSON

```
{
  Success: true/false, //Indica se a criação foi bem sucedida ou não
  Message: "Msg de erro", //Mensagem descritiva de um eventual erro
  Entity: {
    TaskID: 9999 //ID da tarefa no sistema taskrow
    TaskNumber: 99999 //Número sequencial da tarefa. Este valor será requisitado
em futuras chamadas
  }
}
```



Unidades Produzidas

Obter unidades produzidas de uma tarefa

Unidades produzidas em uma tarefa, buscando pelo taskNumber

URL: `https://[HOST_NAME].taskrow.com/api/v1/`

RequestDelivery/ListTaskUnitDeliveries

Método: GET

Campo	Detalhe
taskNumber	999 //Número da tarefa

Retorno JSON

```
{
  "unitDeliveryTrack": [ //lista de unidades
    { //unidade com registro de produção já efetuado
      "effortUnitID": 99999,
      "effortUnitName": "Nome da unidade",
      "effortUnitGroupID": 99999,
      "effortUnitGroupName": "Nome do grupo",
      "inactive": false,
      "estimatedCount": 0, //estimado da unidade
      "effortUnitGroupApprovalListString": "",
      "unitDeliveryTrackRequest": [ //registros de unidades já efetuados
        {
          "requestDeliveryID": 99999, //ID
          "effortUnitID": 9999, //ID da unidade
          "year": 2022, //ano
          "month": 3, //mes
          "requestID": 99999, //id da solicitação
          "requestUserGroupID": 690, //id do grupo do usuario que abriu a solictacao
          "qty": 5, //quantidade registrada no mês
          "userIDs": [
            5380 //usuarios marcados na produção da unidade
          ]
        }
      ]
    }
  ],
  //...
```



```
{
  //unidade sem registro de produção
  "effortUnitID": 99999,
  "effortUnitName": "Nome da unidade",
  "effortUnitGroupID": 99999,
  "effortUnitGroupName": "Nome do grupo",
  "inactive": false,
  "estimatedCount": 0, //estimado da unidade
  "effortUnitGroupApprovalListString": "",
  "unitDeliveryTrackRequest": []
},
"requests": [ //lista de solicitações
{
  "request": {
    "requestID": 99999,
    "changeDate": "/Date(\"2022-04-01T16:26:30.961-03:00\")/",
    "requestTypeID": 999,
    "acronymOrInitials": "SG",
    "requestTypeClassificationID": 1,
    "requestTypeName": "[SG] Solicitação Geral",
    "requestTypeColorID": 9,
    "hasRequestDelivery": false,
    "requestUserGroupID": 99999 //id do grupo do usuario que abriu a solicitação
  },
  "timesheetUsers": [
    99999 //usuários com apontamento de timesheet na solicitação
  ],
  "otherUsers": [
    9999 //outros usuários que eventualmente participaram da solicitação
  ]
}
]
}
```

Registrar unidade produzida

Unidades produzidas em uma tarefa, buscando pelo taskNumber

URL: [https://\[HOST_NAME\].taskrow.com/api/v1/](https://[HOST_NAME].taskrow.com/api/v1/RequestDelivery/SaveUnitDeliveryTrack)

RequestDelivery/SaveUnitDeliveryTrack

Método: GET



Campo	Detalhe
CurrentUnitTrack.EffortUnitID	9999
CurrentUnitTrack.EffortUnitName	Banner+tipo+2
CurrentUnitTrack.EffortUnitGroupID	9999
CurrentUnitTrack.EffortUnitGroupName	Digital
CurrentUnitTrack.EstimatedCount	2
CurrentUnitTrack.CleanName	bannertipo2
CurrentUnitTrack.Inactive	false
CurrentUnitTrack.UnitDeliveryTrackRequest[0].RequestID	9999
CurrentUnitTrack.UnitDeliveryTrackRequest[0].RequestUserGroupID	99999
CurrentUnitTrack.UnitDeliveryTrackRequest[0].Month	11
CurrentUnitTrack.UnitDeliveryTrackRequest[0].Year	2023
CurrentUnitTrack.UnitDeliveryTrackRequest[0].UserIDs[0]	5380
CurrentUnitTrack.UnitDeliveryTrackRequest[0].Qty	1
TaskNumber	9999



Timesheet

Para tirar todo o proveito do taskrow é muito importante que toda atividade tenha suas horas apontadas em tarefas. Abaixo estão alguns métodos para obtenção dos tempos gastos

Horas lançadas

URL:

[https://\[HOST_NAME\].taskrow.com/api/v1/Task/GetTaskMinutesSpent](https://[HOST_NAME].taskrow.com/api/v1/Task/GetTaskMinutesSpent)

Método: GET

Campo	Detalhe
taskNumber	99999 //Número sequencial da tarefa.

Retorno: JSON

```
{
  Success: true/false, //Indica se a requisição foi bem sucedida ou não
  Message: "Msg de erro", //Mensagem descritiva de um eventual erro
  Entity: {
    MainTask: [
      //Lançamentos na tarefa principal - 1 objeto para cada usuário
      {
        User: {
          UserID: 999, //ID do usuário no Taskrow
          FullName: "XXXXXXXX", //Nome do usuário
          ExternalCode: "XXXXXX", //Código de integração
        },
        TotalMinutesSpent: 60, //Quantidade minutos apontados (inteiro)
      },
      {
        User: {
          UserID: 999, //ID do usuário no Taskrow
          FullName: "XXXXXXXX", //Nome do usuário
          ExternalCode: "XXXXXX", //Código de integração
        },
        TotalMinutesSpent: 60, //Quantidade minutos apontados (inteiro)
      },
      ...
    ]
  }
}
```





Buscas

Busca de Clientes

URL: `https://[HOST_NAME].taskrow.com/api/v1/Search/SearchClients`

Método: GET

Campo	Detalhe
q	Nome do cliente / CNPJ do cliente no formato 09.999.999/0000-00
showInactives	true / false (opcional)

Observações:

- O nome do cliente pode ser parte do nome
- O CNPJ precisa ser fornecido exatamente nesse formato, com o zero à esquerda, caso comece com zero

Retorno: JSON

```
[
  {
    "value": "Empresa teste 123123",
    "ClientName": "Empresa teste 123123",
    "ClientID": 99999,
    "ClientNickName": "Empresateste123123",
    "Inactive": false
  },
  {
    "value": "Empresa Inativa",
    "ClientName": "Empresa Inativa",
    "ClientID": 82828,
    "ClientNickName": "EmpresaInativa",
    "Inactive": true
  }
]
```



Busca de Projetos

URL: [https://\[HOST_NAME\].taskrow.com/api/v1/Search/SearchJobs](https://[HOST_NAME].taskrow.com/api/v1/Search/SearchJobs)

Método: GET

Campo	Detalhe
clientID	99999 (ID de um cliente, opcional)
q	Nome do projeto / Nome do cliente / Tag do projeto
showInactives	true / false (opcional)

Observações:

- A busca pode ser feita dentro de um cliente específico (fornecendo o clientID), ou na empresa toda (sem fornecer o clientID)
- O nome do projeto ou cliente pode ser fornecido parcialmente
- Para uma busca por tag, forneça-a com o prefixo '#'. Por exemplo listando todos os projetos com a tag "segundo semestre", faça q=#segundo+semestre

Retorno: JSON

```
[
  {
    "value": "#101 Projeto Exemplo 1 (ClienteX)",
    "JobDisplayTitle": "#101 Projeto Exemplo 1",
    "JobNumber": 101,
    "JobID": 100001,
    "ProductID": null,
    "ProductName": null,
    "ClientNickName": "ClienteX",
    "ClientID": 9999,
    "ClientName": "ClienteX",
    "ClientNumber": 10,
    "Closed": false
  },
  {
    "value": "#102 Projeto Exemplo 2 (ClienteX)",
    "JobDisplayTitle": "#102 Projeto Exemplo 2",
    "JobNumber": 102,
    "JobID": 100002,
    "ProductID": null,
    "ProductName": null,
    "ClientNickName": "ClienteX",
```



```
"ClientID": 9999,  
"ClientName": "ClienteX",  
"ClientNumber": 10,  
"Closed": false  
}  
]
```

Busca de Tarefas

URL: [https://\[HOST_NAME\].taskrow.com/api/v2/search/tasks/advancedsearch](https://[HOST_NAME].taskrow.com/api/v2/search/tasks/advancedsearch)

Método: **POST**

Campo	Detalhe
ClientID	ID do cliente para filtrar as tarefas
ProductID	ID do produto para filtrar as tarefas
JobID	ID do projeto (job) para filtrar as tarefas
Term	Termo de busca para filtrar por texto
OnlyMainTasks	Indica se deve buscar apenas tarefas principais
OnlySubtasks	Indica se deve buscar apenas subtarefas
StartDate	Data de início do período de busca
EndDate	Data de fim do período de busca
FilterUserID	ID do usuário para filtrar
FilterUserType	Tipo de usuário para filtrar
FilterExtranet	Filtro relacionado à extranet
Offset	Offset para paginação dos resultados
Closed	Indica se deve filtrar tarefas fechadas (true), abertas (false) ou todas (null)
Sort	Critério de ordenação (DueDateDesc ou Title)

Observações:



- Os campos são todos opcionais, para uma busca sem parâmetros envie um objeto vazio no body.
- O termo de busca pode ser fornecido parcialmente
- Para uma busca por tag, forneça-a com o prefixo '#'. Por exemplo listando todas as tarefas com a tag "segundo semestre", faça term=#segundo+semestre
- A busca retornar os primeiros 200 registros, para visualizar os próximos faça uma nova busca, enviando no parâmetro Offset o valor retornado em "NextOffset"

Exemplo de requisição:

```
{
  "ClientID": 123,
  "JobID": 456,
  "Term": "tarefa importante",
  "OnlyMainTasks": false,
  "OnlySubtasks": false,
  "StartDate": "2025-09-01T00:00:00Z",
  "EndDate": "2025-09-30T23:59:59Z",
  "FilterUserID": 789,
  "Offset": 0,
  "Closed": null,
  "Sort": 0
}
```

Exemplo de resposta de sucesso:

```
{
  "Data": [
    {
      "TaskID": 123,
      "TaskNumber": 456,
      "TaskTitle": "Título da tarefa",
      "CreationDate": "2025-09-26T10:00:00Z",
      "Closed": false,
      "CloseDate": null,
      "DueDate": "2025-09-30T23:59:59Z",
      "AbsoluteUserOrder": 1,
      "UrlData": "url-da-tarefa",
      "RequestTypeAcronym": "REQ",
      "RequestTypeName": "Requisição",
      "Tags": "tag1,tag2",
      "ParentTaskID": null,
      "ParentTaskNumber": null,
      "ParentTaskTitle": null,
      "PipelineStep": "Em andamento",
      "ColorID": 1,
    }
  ]
}
```



```
"HasExternalItems": false,
"ExtranetPipelineStep": null,
"ExtranetColorID": null,
"JobID": 789,
"JobNumber": 101,
"JobTypeID": 1,
"JobTitle": "Título do projeto",
"JobType": "Tipo do projeto",
"JobUrlData": "url-do-projeto",
"JobDisplayTitle": "Nome de exibição do projeto",
"ClientID": 555,
"ClientNickName": "cliente-nome",
"ClientDisplayName": "Nome do Cliente",
"ClientUrlData": "url-do-cliente",
"OwnerUserID": 333,
"OwnerUserLogin": "usuario.dono",
"OwnerUserHashCode": "hash-usuario-dono",
"CreationUserID": 222,
"CreationUserHashCode": "hash-usuario-criacao",
"CreationUserLogin": "usuario.criacao",
"ClientContactID": 444,
"ContactName": "Nome do Contato",
"ContactEmail": "contato@cliente.com",
"ContactRequestDate": "2025-09-25T09:00:00Z",
"CurrentlyOpen": true,
"OpenInPeriod": true,
"ClosedInPeriod": false,
"OpenAndClosedInPeriod": false
}
],
"NextOffset": 200
}
```




Cadastros auxiliares

Listar Cidades

URL: [https://\[HOST_NAME\].taskrow.com/api/v1/Client/ListCities](https://[HOST_NAME].taskrow.com/api/v1/Client/ListCities)

Método: GET

Campo	Detalhe
uf	SP //UF das cidades

Retorno: JSON

```
[
  {
    CityID: 9999,
    UF: "SP",
    Name: "São Paulo"
  },
  ...
]
```

Listar Áreas de um cliente

URL: [https://\[HOST_NAME\].taskrow.com/api/v1/Client/ListClientArea](https://[HOST_NAME].taskrow.com/api/v1/Client/ListClientArea)

Método: GET

Campo	Detalhe
clientNickName	"XXXXX" //Nome da 'pasta' do cliente no Taskrow retornado pelo método SaveClient

Retorno: JSON

```
{
  "ClientAreaList": [
    {
      "ClientAreaID": 9999,
      "Name": "XXXXX"
    },
    ....
  ]
}
```



Listar Subtipos de projeto

URL: [https://\[HOST_NAME\].taskrow.com/api/v1/Administrative/ListJobSubType](https://[HOST_NAME].taskrow.com/api/v1/Administrative/ListJobSubType)

Método: GET

Retorno: JSON

```
{
  "JobSubTypeList": [
    {
      "JobSubTypeID": 9999,
      "Name": "XXXXX"
    },
    .....
  ]
}
```