
Rindap

Release 2020

Oct 13, 2020

Contents:

1	Introduction	1
2	Authentication	3
2.1	Workspaces	3
2.2	Workflows	16
2.3	Workers	33
2.4	Tasks	46
2.5	Rate Limit Profiles	60
2.6	Task Queues	72
2.7	Reservations	85
	HTTP Routing Table	105

CHAPTER 1

Introduction

Welcome to the Rindap API documentation! Rindap helps companies to increase their work efficiency by offering developers a low-code BPM platform with RESTful API in order to automate business processes based on the requirements set by the management.

The reason Rindap was developed was the reason to help developers manage to automate business processes by using APIs without the need for high-cost out-of-the-box programs or complex custom developments.

Developers and business analysts can cooperate to automate business processes on Rindap with ease. Developers can design workflows via a user-friendly drag&drop visual modeller, easily set business rules by filters that use visual code blocks and utilize Rindap's RESTful API to connect to any platform, device or system without any boundaries.

By using Rindap's RESTful API you can achieve end-to-end digital process automation across people, platforms and devices without being limited by company or department data silos. Rindap is fully compatible with any legacy software and can be used without any problems across the entire enterprise technology stack.

You can use our API documentation to get familiarized with Rindap API endpoints, which can be related to Tasks, Workspaces, Workflows, Workers and etc.

CHAPTER 2

Authentication

Rindap uses API keys to allow access to the API. You can register a new Rindap API key at our developer portal.

Rindap expects for the API key to be included in all API requests to the server in a header that looks like the following:

```
Authorization:Bearer [Account SID].[Auth token]
```

Note: You must replace **[Account SID]** and **[Auth token]** with your personal API credentials found on rindap.com developer console.

2.1 Workspaces

Rindap (Process) Workspace is a customizable place enabling you to create your Tasks, Workers, TaskQueues and Workflows elements in it to access and manage your business processes. The elements defined in one Workspace are specific to it and cannot be shared with other Workspace.

Various tasks with different attributes, specific and modifiable workflow, various skilled workers can be defined in one Workspace to orchestrate your business processes efficiently in the lowest time. Business Process Workspace helps organizations to improve their internal processes management by providing a single container to manage, control and monitor the whole elements needed in one place. With Rindap, managing the inner operational processes of a company in a purpose-built Workspace and defining your strategies and policies for the company in a single Workspace can be done.

2.1.1 Workspace Properties

Table 1: Properties

field	description
sid	The unique string that we created to identify the Workspace resource.
account_sid	The SID of the Account that created the Workspace resource.
friendly_name	The string that you assigned to describe the resource.
event_callback_url	The URL we call when an event occurs. If provided, the Workspace will publish events to this URL, for example, to collect data for reporting. See Workspace Events for more information.
event_callback_method	The HTTP Request Method for calling the EventCallbackUrl
default_activity	The Activity that will be used when new Workers are created in the Workspace.
timeout_activity	The Activity that will be assigned to a Worker when a Task reservation times out without a response
date_created	The date and time in GMT when the resource was created, specified in ISO 8601 format.
date_updated	The date and time in GMT when the resource was last updated, specified in ISO 8601 format.
url	The absolute URL of the resource
links	The URLs of related resources.

2.1.2 Create A Workspace

POST /v1/rindap-rest-gw/Workspaces/

You can create a Workspace by simply providing a friendly name

Table 2: Query Parameters

Parameter	Type	Default	Description
FriendlyName	String	""	A descriptive string that you create to describe the Workspace resource. It can be up to 512 characters long
EventCallbackUrl	URL	""	(Optional) The URL we call when an event occurs. If provided, the Workspace will publish events to this URL, for example, to collect data for reporting. See Workspace Events for more information.
EventCallbackMethod	String	POST	(Optional) The HTTP Request Method for calling the EventCallbackUrl
DefaultActivity	String	offline	(Optional) The Activity that will be used when new Workers are created in the Workspace.
TimeoutActivity	String	offline	(Optional) The Activity that will be assigned to a Worker when a Task reservation times out without a response

Example code pieces using SDKs

Shell

```
curl -X POST https://api.rindap.com/v1/rindap-rest-gw/Workspaces \
--data-urlencode 'FriendlyName=my test workspace' \
--data-urlencode 'EventCallbackUrl=https://my-backbone.mydomain.org' \
--data-urlencode 'EventCallMethod=GET' \
```

(continues on next page)

(continued from previous page)

```
--data-urlencode 'DefaultActivity=busy' \
--data-urlencode 'TimeoutActivity=offline'
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"
```

Java

```
// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.Task;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        Workspace ws = Workspace.creator("my test workspace")
            .setEventCallbackUrl("https://my-backbone.mydomain.org")
            .setEventCallbackMethod(HttpMethod.GET)
            .setDefaultActivity("busy")
            .setTimeoutActivity("idle")
            .create();

        System.out.println(ws);
    }
}
```

Phyton

```
from rindap.rest import Client
from rindap.rest import Rindap

client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

ws = rindap.workspaces.create("my test workspace",
                              event_callback_url="https://my-backbone.mydomain.org",
                              event_callback_method="GET",
                              default_activity="busy",
                              timeout_activity="idle")

print("FriendlyName: {}".format(ws.friendly_name))
```

JS

```
var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Create a workspace
rindap.workspaces.create({
  friendlyName: "Friendly Name",
  eventCallbackUrl: "https://my-backbone.mydomain.org",
```

(continues on next page)

(continued from previous page)

```

eventCallbackMethod: "GET",
defaultActivity: "busy",
timeoutActivity: "idle"
}, function(err, workspace) {
// Print workspace content
console.log('Workspace Created');
console.log(workspace.sid);
console.log(workspace.friendlyName);
console.log(workspace.eventCallbackUrl);
console.log(workspace.eventCallbackMethod);
console.log(workspace.defaultActivity);
console.log(workspace.timeoutActivity);
});

```

CSharp

```

using System;
using Rindap;
using Rindap.Rest.V1;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // Create a workspace
        WorkspaceResource ws = WorkspaceResource.Create(
            friendlyName: "Very Friendly Name From C#",
            eventCallbackUrl: new Uri("https://my-backbone.mydomain.org"),
            eventCallbackMethod: "GET",
            defaultActivity: "busy",
            timeoutActivity: "offline"
        );

        Console.WriteLine("Workspace Friendly Name      : " + ws.FriendlyName);
        Console.WriteLine("Workspace Sid      : " + ws.Sid);
        Console.WriteLine("Workspace Default Activity : " + ws.DefaultActivity);
        Console.WriteLine("Workspace Timeout Activity : " + ws.TimeoutActivity);
        Console.WriteLine("Workspace Event Calllback Url : " + ws.EventCallbackUrl);
        Console.WriteLine("Workspace Event callback Method: " + ws.
↪EventCallbackMethod);
    }
}

```

The above command returns JSON structured like this:

```

{
  "sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "friendly_name": "my test workspace",
  "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "default_activity": "busy",
  "timeout_activity": "idle",
  "date_created": "2020-05-04T01:36:02+03:00",
  "date_updated": "2020-05-04T01:36:02+03:00",
  "event_callback_url": "https:my-backbone.mydomain.org",

```

(continues on next page)

(continued from previous page)

```

    "event_callback_method": "GET",
    "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "links": {
        "tasks": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Tasks",
        "workers": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workers",
        "workflows": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workflows",
        "task_queues": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/TaskQueues"
    }
}

```

2.1.3 Get All Workspaces

This endpoint retrieves all Workspaces

Table 3: Query Parameters

Parameter	Type	Default	Description
FriendlyName	String	""	Human readable friendly name
PageSize	Integer	50	Page size for paging
FriendlyName	Integer	0	Page number for paging

Example code pieces using SDKs

CSharp

```

using System;
using Rindap;
using Rindap.Rest.V1;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // Fetch all workspaces
        var workspaces = WorkspaceResource.Read(limit: 100, pageSize: 100);

        // Iterate all workspaces
        foreach (var ws in workspaces)
        {
            // Print workspace content
            Console.WriteLine("Workspace Friendly Name      : " + ws.FriendlyName);
            Console.WriteLine("Workspace Sid      : " + ws.Sid);
            Console.WriteLine("Workspace Default Activity : " + ws.
↪DefaultActivity);
            Console.WriteLine("Workspace Timeout Activity : " + ws.
↪TimeoutActivity);
        }
    }
}

```

(continues on next page)

(continued from previous page)

```
        Console.WriteLine("Workspace Event Calllback Url  : " + ws.
↪EventCallbackUrl);
        Console.WriteLine("Workspace Event callback Method: " + ws.
↪EventCallbackMethod);
    }
}
```

Phyton

```
from rindap.rest import Client
from rindap.rest import Rindap

client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

lists = rindap.workspaces.list(friendly_name=None, limit=10, page_size=5)
workspace = lists.pop()
print(workspace.friendly_name)
```

JS

```
var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Get all workspaces
rindap.workspaces.list({
  limit: 100,
  page_size: 100
}, function(err, result) {
  result.forEach(function(workspace) {

    console.log(workspace.sid);
    console.log(workspace.friendlyName);
    console.log(workspace.eventCallbackUrl);
    console.log(workspace.eventCallbackMethod);
    console.log(workspace.defaultActivity);
    console.log(workspace.timeoutActivity);
  });
});
```

The above command returns JSON structured like this:

```
{
  "meta": {
    "page_size": 50,
    "page": 0,
    "first_page_url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/?Page=0&
↪PageSize=50",
    "previous_page_url": null,
    "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/?Page=0&PageSize=50",
    "key": "workspaces",
    "next_page_url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/?Page=1&
↪PageSize=50"
  },
}
```

(continues on next page)

(continued from previous page)

```

"workspaces": [
  {
    "sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "friendly_name": "my test workspace",
    "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "default_activity": "busy",
    "timeout_activity": "idle",
    "date_created": "2020-05-04T01:36:02+03:00",
    "date_updated": "2020-05-04T01:36:02+03:00",
    "event_callback_url": "https://my-backbone.mydomain.org",
    "event_callback_method": "GET",
    "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "links": {
      "tasks": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Tasks",
      "workers": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workers",
      "workflows": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workflows",
      "task_queues": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/TaskQueues"
    }
  }
]
}

```

2.1.4 Fetch A Workspace

This endpoint fetches a single Workspace with all its details

GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}

Table 4: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the Workspace

Example code pieces using SDKs

Shell

```

curl -X GET https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type: application/x-www-form-urlencoded"

```

Java

```

// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.Task;

```

(continues on next page)

(continued from previous page)

```
public class Example {  
    // Find your Account Sid and Token at rindap.com/console  
  
    public static void main(String[] args) {  
  
        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");  
  
        Workspace ws = Workspace.fetcher("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")  
            .fetch();  
  
        System.out.println(ws);  
    }  
}
```

Phyton

```
from rindap.rest import Client  
from rindap.rest import Rindap  
  
client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")  
rindap = Rindap(client)  
  
ws_fetcher = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")  
workspace = ws_fetcher.fetch()  
print(workspace.friendly_name)
```

JS

```
var Rindap = require('rindap');  
  
// Authenticate  
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");  
  
// Get a workspaces with SID  
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxx').fetch(  
function(err, workspace) {  
    console.log(err);  
    console.log(workspace.sid);  
    console.log(workspace.friendlyName);  
    console.log(workspace.eventCallbackUrl);  
    console.log(workspace.eventCallbackMethod);  
    console.log(workspace.defaultActivity);  
    console.log(workspace.timeoutActivity);  
});
```

CSharp

```
using System;  
using Rindap;  
using Rindap.Rest.V1;  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        // Authenticate
```

(continues on next page)

(continued from previous page)

```

RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Get a workspace with SID
var ws = WorkspaceResource.Fetch(
    pathSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
);

// Print workspace content
Console.WriteLine("Workspace Friendly Name      : " + ws.FriendlyName);
Console.WriteLine("Workspace Sid                : " + ws.Sid);
Console.WriteLine("Workspace Default Activity    : " + ws.DefaultActivity);
Console.WriteLine("Workspace Timeout Activity    : " + ws.TimeoutActivity);
Console.WriteLine("Workspace Event Callback Url  : " + ws.EventCallbackUrl);
Console.WriteLine("Workspace Event callback Method: " + ws.
↪EventCallbackMethod);
    }
}

```

The above command returns JSON structured like this:

```

{
  "sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "friendly_name": "my test workspace",
  "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "default_activity": "busy",
  "timeout_activity": "idle",
  "date_created": "2020-05-04T01:36:02+03:00",
  "date_updated": "2020-05-04T01:36:02+03:00",
  "event_callback_url": "https://my-backbone.mydomain.org",
  "event_callback_method": "GET",
  "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "links": {
    "tasks": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Tasks",
    "workers": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workers",
    "workflows": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workflows",
    "task_queues": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/TaskQueues"
  }
}

```

2.1.5 Update a Workspace

PUT /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}

Table 5: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the Workspace
FriendlyName	String	""	(Optional) A descriptive string that you create to describe the Workspace resource. It can be up to 512 characters long
EventCallbackUrl	URL	""	(Optional) The URL we call when an event occurs. If provided, the Workspace will publish events to this URL, for example, to collect data for reporting. See Workspace Events for more information.
EventCallbackMethod	String	POST	(Optional) The HTTP Request Method for calling the EventCallbackUrl
DefaultActivity	String	""	(Optional) The Activity that will be used when new Workers are created in the Workspace.
TimeoutActivity	String	""	(Optional) The Activity that will be assigned to a Worker when a Task reservation times out without a response

Example code pieces using SDKs

Shell

```
curl -X PUT https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
--data-urlencode 'FriendlyName=my test workspace updated' \
--data-urlencode 'EventCallbackUrl=https://my-backbone.my-new-domain.org' \
--data-urlencode 'EventCallMethod=POST' \
--data-urlencode 'DefaultActivity=idle' \
--data-urlencode 'TimeoutActivity=busy'
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"
```

Java

```
// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.Task;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        Workspace ws = Workspace
            .updater("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
            .setFriendlyName("my test workspace updated")
            .setEventCallbackUrl("https://my-backbone.my-new-domain.org")
            .setEventCallbackMethod("POST")
            .setDefaultActivity("idle")
            .setTimeoutActivity("offline")
            .update();

        System.out.println(ws);
    }
}
```


Phyton

```

from rindap.rest import Client
from rindap.rest import Rindap

client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

ws_fetcher = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
ws_fetcher.update(friendly_name="New Friendly Name", default_activity="offline",
                  timeout_activity="offline", event_callback_url="https://domain.com",
                  event_callback_method="POST")

```

JS

```

var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Update a workspaces with SID
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx').update({
  friendlyName: "New Friendly Name",
  eventCallbackUrl: "https://my-backbone.mydomain.com",
  defaultActivity: "offline",
  timeoutActivity: "offline"
}, function(err, workspace) {
  console.log(workspace.sid);
  console.log(workspace.friendlyName);
  console.log(workspace.eventCallbackUrl);
  console.log(workspace.eventCallbackMethod);
  console.log(workspace.defaultActivity);
  console.log(workspace.timeoutActivity);
});

```

CSharp

```

using System;
using System.ComponentModel.DataAnnotations;
using Rindap;
using Rindap.Rest.V1;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // Update a workspace with SID
        var ws = WorkspaceResource.Update(
            pathSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            friendlyName: "New Workspace Name",
            eventCallbackUrl: new Uri("https://my-backbone.mydomain.com"),
            eventCallbackMethod: "POST",
            defaultActivity: "busy",
            timeoutActivity: "busy"
        );
    }
}

```

(continues on next page)

(continued from previous page)

```

// Print workspace content
Console.WriteLine("Workspace Friendly Name      : " + ws.FriendlyName);
Console.WriteLine("Workspace Sid                : " + ws.Sid);
Console.WriteLine("Workspace Default Activity    : " + ws.DefaultActivity);
Console.WriteLine("Workspace Timeout Activity    : " + ws.TimeoutActivity);
Console.WriteLine("Workspace Event Calllback Url   : " + ws.EventCallbackUrl);
Console.WriteLine("Workspace Event callback Method: " + ws.
↪EventCallbackMethod);
    }
}

```

The above command returns JSON structured like this:

```

{
  "sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "friendly_name": "my test workspace updated",
  "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "default_activity": "idle",
  "timeout_activity": "offline",
  "date_created": "2020-05-04T01:36:02+03:00",
  "date_updated": "2020-05-04T01:36:02+03:00",
  "event_callback_url": "https:my-backbone.my-new-domain.org",
  "event_callback_method": "POST",
  "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "links": {
    "tasks": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Tasks",
    "workers": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workers",
    "workflows": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workflows",
    "task_queues": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/TaskQueues"
  }
}

```

2.1.6 Delete a Workspace

DELETE /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}

Table 6: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the Workspace

Example code pieces using SDKs

Shell

```
curl -X DEL https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"
```

Java

```
// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.Task;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        Workspace.deleter("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx").delete();
    }
}
```

Phyton

```
from rindap.rest import Client
from rindap.rest import Rindap

client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

ww_fetcher = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
if ww_fetcher.delete():
    print("Workspaces have been deleted!")
```

JS

```
var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Delete a workspaces with SID
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx').remove();
```

CSharp

```
using System;
using System.ComponentModel.DataAnnotations;
using Rindap;
using Rindap.Rest.V1;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
```

(continues on next page)

(continued from previous page)

```
RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Delete a workspace with SID
var isDeleted = WorkspaceResource.Delete(
    pathSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
);

if (isDeleted)
{
    Console.WriteLine("Workspace has been deleted!");
}
}
```

2.2 Workflows

Workflows manage and control the distribution of tasks to workers based on [Workflows configurations](#) and capabilities of workers. Workflows configurations is the most important aspect of a workflow process which specifies the order in which tasks must be executed over time by simply described in JSON format and pair tasks into Queues. Each configuration includes a set of conditions which control and evaluate tasks attributes. The rules of condition were defined in JSONLogic format. Read more about [JSONLogic here](#) .

When a task is added to the workspace, its own workflow is executed in order to control the task to be channeled into proper Queues. The workflow managing process will continue until the task is either completed or cancelled. Afterwards, based on the evaluation process of Workflows configurations, a task is assigned to a TaskQueue. Workers listen to TaskQueues and reserve the task according to workers availability and capabilities. Assignment Callback mechanism will be enabled to execute the intended request. Workflow's Assignment Callback contains all the information necessary for the worker to perform the task (for example, emailing or texting the Worker or showing a Notification If your Worker is logged in some sort of portal or making an IVR call to your Worker and later updating the Task with the input from the call).

Main features of Workflow listed as following:

- Condition: Condition of each Workflow is defined in the filters field of Workflow Configuration. Condition determines whether Task execution happens or not.
- Fork: A Task can be triggered to fork out to create new Tasks for different Workflows, executed in parallel. Read more about use cases and details here at [Task Forking](#)
- Redirected to: A filter can redirect the flow of the Task to another Filter to run a different course , like a branch.
- Looping Filter : A looping filter that can be validated between time periods, till It turns true or It's validated enough

You can read about these features and more about [Workflows here](#) .

2.2.1 Workflow Properties

Table 7: Properties

field	description
sid	The unique string that we created to identify the Workflow resource.
account_sid	The SID of the Account that created the Workflow resource.
workspace_sid	The SID of the Workspace that contains the Workflow
friendly_name	The string that you assigned to describe the resource. Friendly names are case insensitive, and unique within the Rindap Workspace.
assignment_callback_url	The URL that we call when a task managed by the Workflow is assigned to a Worker.
fallback_assignment_callback_url	The URL that we call when a call to the AssignmentCallbackUrl fails.
task_reservation_timeout	How long Rindap will wait for a confirmation response from your application after notifying the AssignmentCallbackUrl about the reservation. Can be up to 86,400 (24* hours) , minimum 5 , and the default is 86,400..
configuration	A URL-encoded JSON string that contains the Workflow's configuration. See Configuring Workflows for more information.
date_created	The date and time in GMT when the resource was created, specified in ISO 8601 format.
date_updated	The date and time in GMT when the resource was last updated, specified in ISO 8601 format.
url	The absolute URL of the resource
links	The URLs of related resources.

2.2.2 Create A Workflow

POST /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Workflows

You can easily design and create your Workflow on our developer Portal with an intuitive drag&drop interface. For more information [Workflows](#)

Table 8: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the Workspace in which the Task is created
FriendlyName	String	""	Human readable friendly name. It can be 512 characters long
AssignmentCallbackUrl	URL	""	The URL that we call when a task managed by the Workflow is assigned to a Worker.
FallbackAssignmentCallbackUrl	URL	""	The URL that we call when a call to the assignment_callback_url fails.
Configuration	String	""	A URL-encoded JSON string that contains the Workflow's configuration. For more information , see Workflows .
TaskReservationTimeout	Integer	86400	(optional) How long Rindap will wait for a confirmation response from your application after notifying the <i>assignment_callback_url</i> about the reservation.

Example code pieces using SDKs

Shell

```

curl -X POST https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workflows \
--data-urlencode 'FriendlyName=my test workflow'
--data-urlencode 'Configuration={"first_step":"FS111","filters":{"FS111":{"true_queue_
↳ sid":"WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx","false_filter_sid":"FS222","name":"Hot",
↳ "conditions":{">":[{"var":"temp"},"300"]},"FS222":{"true_queue_sid":
↳ "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx","name":"Mild","conditions":{"!=":[1,1]}}}}'
--data-urlencode 'AssignmentCallbackUrl=https://assignment-callback.my-backbone.
↳ mydomain.org'
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"

```

Java

```

// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.Workflow;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        Workflow wf = Workflow.creator("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            "my test workflow",
            "https://assignment-callback.my-backbone.mydomain.org",
            "{\"first_step\":\"FS111\",\"filters\":{\"FS111\":{\"true_queue_sid\":\
↳ \"WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx\", \"false_filter_sid\": \"FS222\", \"name\": \"Hot\
↳ \", \"conditions\": {\">\": [{\"var\": \"temp\"}, \"300\"]}}, \"FS222\": {\"true_queue_sid\
↳ \": \"WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx\", \"name\": \"Mild\", \"conditions\": {\"!=\": [\
↳ \"1\", \"1\"]}}}}\",
            )
            .create();

        System.out.println(wf);
    }
}

```

Phyton

```

from rindap.rest import Client
from rindap.rest import Rindap

# Authenticate
client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

# Get Workspace with workspace_sid
workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

# create configuration
configuration = {
    "first_step": "FS111",
    "filters": {

```

(continues on next page)

(continued from previous page)

```

    "FS111": {
      "true_queue_sid": "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "false_filter_sid": "FS222",
      "name": "Hot",
      "conditions": {
        ">": [{"var": "temp"}, "300"]
      }
    },
    "FS222": {
      "true_queue_sid": "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "name": "Mild",
      "conditions": {
        "!=" : ["1", "1"]
      }
    }
  }
}

# Create a workflow
workflow = workspace.workflows.create("My Workflow",
                                       str(configuration),
                                       assignment_callback_url="https://assignment-
→callback.my-backbone.mydomain.org")

# Print workflow content
print("WorkflowSid: {}".format(workflow.sid))
print("FriendlyName: {}".format(workflow.friendly_name))

```

JS

```

var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Workflow configuration
var configuration = {
  "first_step": "FS111",
  "filters": {
    "FS111": {
      "true_queue_sid": "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "false_filter_sid": "FS222",
      "name": "Hot",
      "conditions": {
        ">": [{"var": "temp"}, "300"]
      }
    },
    "FS222": {
      "true_queue_sid": "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "name": "Mild",
      "conditions": {
        "!=" : ["1", "1"]
      }
    }
  }
}

```

(continues on next page)

(continued from previous page)

```
// Crate a workflow
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
.workflows
.create({
  friendlyName: "Friendly Name",
  configuration: JSON.stringify(configuration),
  assignmentCallbackUrl: "https://assignment-callback.my-backbone.mydomain.org"
}, function(err, workflow) {
  console.log(workflow.sid);
  console.log(workflow.friendlyName);
});
```

C#

```
using System;
using System.Collections.Generic;
using Newtonsoft.Json;
using Rindap;
using Rindap.Rest.V1.Workspace;

class Program
{
  static void Main(string[] args)
  {
    // Authenticate
    RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

    // Create a configuration
    var configuration = JsonConvert.SerializeObject(new Dictionary<string, Object>
    {
      { "first_step", "FS111"},
      { "filters",
        new Dictionary<string, Object>()
        {
          { "FS111", new Dictionary<string, Object>()
            {
              { "true_queue_sid", "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"},
              { "false_filter_sid", "FS222"},
              { "name", "Hot"},
              { "conditions",
                new Dictionary<string, Object>() {
                  { ">",
                    new object [] {
                      new Dictionary<string, Object>() { { "var
→", "temp" } }, 300 }
                }
              }
            }
          }
        }
      },
      { "FS222", new Dictionary<string, Object>()
        {
          { "true_queue_sid", "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" }
        }
      }
    }
  }
}
```

(continues on next page)

(continued from previous page)

```

        { "name", "Mild" },
        { "conditions",
          new Dictionary<string, Object>() {
            { "!=",
              new object [] { "1", "1" }
            }
          }
        }
      }
    }
  }
}, Formatting.None);

// Create a workflow
var workflow = WorkflowResource.Create(
  assignmentCallbackUrl: new Uri("https://example.com/"),
  fallbackAssignmentCallbackUrl: new Uri("https://example2.com/"),
  friendlyName: "New Friendly Name",
  configuration: configuration,
  pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
);

// Print workflow content
Console.WriteLine("WorkflowSID : " + workflow.Sid);
Console.WriteLine("Friendly Name : " + workflow.FriendlyName);
}
}

```

The above command returns JSON structured like this:

```

{
  "sid": "WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "friendly_name": "WF1-yesno",
  "configuration": {
    "first_step": "FS111",
    "filters": {
      "FS111": {
        "true_queue_sid": "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        "false_filter_sid": "FS222",
        "name": "Hot",
        "conditions": {
          ">": [
            {
              "var": "temp"
            },
            "300"
          ]
        }
      },
      "FS222": {
        "true_queue_sid": "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        "name": "Mild",
        "conditions": {

```

(continues on next page)

(continued from previous page)

```

    "!=": [
      "1",
      "1"
    ]
  }
}
},
"task_reservation_timeout": 86400,
"assignment_callback_url": "https://assignment-callback.my-backbone.mydomain.org",
"date_created": "2020-05-06T10:13:34+03:00",
"date_updated": "2020-05-06T10:13:34+03:00",
"url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WS2b948ebc07d14f5388eecbc92139e4c7/Workflows/WWc6187bd0d24a4365bc01ee340d3f7010",
"links": {
  "workspace": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WS2b948ebc07d14f5388eecbc92139e4c7"
}
}
}

```

2.2.3 Get All Workflows

GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Workflows`

This endpoint retrieves all Workflows

Table 9: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the Workspace
FriendlyName	String	""	(optional) Human readable friendly name. Can be used for filtering
PageSize	Integer	50	Page size for paging
Page	Integer	0	Page number for paging

Example code pieces using SDKs

Shell

```

curl -X GET https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workflows \
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"

```

Java

```

// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.Workflow;

public class Example {
  // Find your Account Sid and Token at rindap.com/console

```

(continues on next page)

(continued from previous page)

```

public static void main(String[] args) {

    Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

    Workflow.Reader reader = Workflow.reader("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
↪");

    for(Workflow wf:reader.read())
        System.out.println(wf);
}

```

Phyton

```

from rindap.rest import Client
from rindap.rest import Rindap

# Authenticate
client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

# Get Workspace with workspace_sid
workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

# Fetch all workflow and print content
workflow_fetcher = workspace.workflows.list(limit=10, page_size=5)
for workflow in workflow_fetcher:
    print("WorkflowSid: {}".format(workflow.sid))
    print("FriendlyName: {}".format(workflow.friendly_name))

```

JS

```

var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// List all workflows
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
    .workflows
    .list({
        limit: 100,
        pageSize: 100
    }, function(err, workflows) {
        workflows.forEach(function(workflow) {
            console.log(workflow.sid);
            console.log(workflow.friendlyName);
        })
    });

```

C#

```

using System;
using Rindap;
using Rindap.Rest.V1.Workspace;

```

(continues on next page)

(continued from previous page)

```

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // Fetch all workflows
        var workflows = WorkflowResource.Read(
            pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            limit: 100,
            pageSize: 100
        );

        // Iterate on workflows
        foreach (var workflow in workflows)
        {
            // Print workflow content
            Console.WriteLine("WorkflowSID : " + workflow.Sid);
            Console.WriteLine("Friendly Name : " + workflow.FriendlyName);
        }
    }
}

```

The above command returns JSON structured like this:

```

{
  "meta": {
    "page_size": 50,
    "page": 0,
    "first_page_url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
    ↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workflows?Page=0&PageSize=50",
    "previous_page_url": null,
    "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
    ↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workflows?Page=0&PageSize=50",
    "key": "workflows",
    "next_page_url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
    ↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workflows?Page=1&PageSize=50"
  },
  "workflows": [
    {
      "sid": "WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "friendly_name": "my test workflow",
      "configuration": {
        "first_step": "FS111",
        "filters": {
          "FS111": {
            "true_queue_sid": "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            "false_filter_sid": "FS222",
            "name": "Hot",
            "conditions": {
              ">": [
                {
                  "var": "temp"
                }
              ]
            }
          }
        }
      }
    }
  ]
}

```

(continues on next page)

(continued from previous page)

```

        "300"
      ]
    }
  },
  "FS222": {
    "true_queue_sid": "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "name": "Mild",
    "conditions": {
      "!=": [
        "1",
        "1"
      ]
    }
  }
},
"task_reservation_timeout": 86400,
"assignment_callback_url": "https://assignment-callback.my-backbone.mydomain.org",
"date_created": "2020-05-06T10:13:34+03:00",
"date_updated": "2020-05-06T10:13:34+03:00",
"url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
→WS2b948ebc07d14f5388eecbc92139e4c7/Workflows/WWc6187bd0d24a4365bc01ee340d3f7010",
"links": {
  "workspace": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
→WS2b948ebc07d14f5388eecbc92139e4c7"
}
}
]
}

```

2.2.4 Fetch a Workflow

GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSid}/Workflows/{WorkflowSID}

This endpoint fetches a single Workflow with all Its details

Table 10: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the Workspace
WorkflowSID	String	""	The SID of the Workflow

Example code pieces using SDKs

Shell

```

curl -X GET https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
→WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
/Workflows/WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type: application/x-www-form-urlencoded"

```

Java

```
// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.Workflow;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        Workflow wf = Workflow
            .fetcher("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
→ "WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
            .fetch();

        System.out.println(wf);
    }
}
```

Phyton

```
from rindap.rest import Client
from rindap.rest import Rindap

# Authenticate
client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

# Get Workspace with workspace_sid
workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

# Get workflow with sid and print content
workflow = workspace.workflows.get("WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx").fetch()
print("WorkflowSid: {}".format(workflow.sid))
print("FriendlyName: {}".format(workflow.friendly_name))
```

JS

```
var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Get a workflows with SID
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
    .workflows("WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
    .fetch(function(err, workflow) {
        console.log(workflow.sid);
        console.log(workflow.friendlyName);
    });
```

C#

```
using System;
using Rindap;
```

(continues on next page)

(continued from previous page)

```

using Rindap.Rest.V1.Workspace;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // Get a workflow with SID
        var workflow = WorkflowResource.Fetch(
            pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            pathSid: "WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
        );

        // Print workflow content
        Console.WriteLine("WorkflowSID : " + workflow.Sid);
        Console.WriteLine("Friendly Name : " + workflow.FriendlyName);
    }
}

```

The above command returns JSON structured like this:

```

{
  "sid": "WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "friendly_name": "my test workflow",
  "configuration": {
    "first_step": "FS111",
    "filters": {
      "FS111": {
        "true_queue_sid": "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        "false_filter_sid": "FS222",
        "name": "Hot",
        "conditions": {
          ">": [
            {
              "var": "temp"
            },
            "300"
          ]
        }
      },
      "FS222": {
        "true_queue_sid": "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        "name": "Mild",
        "conditions": {
          "!=": [
            "1",
            "1"
          ]
        }
      }
    }
  },
  "task_reservation_timeout": 86400,

```

(continues on next page)

(continued from previous page)

```

"assignment_callback_url": "https://assignment-callback.my-backbone.mydomain.org",
"date_created": "2020-05-06T10:13:34+03:00",
"date_updated": "2020-05-06T10:13:34+03:00",
"url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workflows/WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
"links": {
  "workspace": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
}
}

```

2.2.5 Update a Workflow

PUT /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Workflows/{WorkflowSID}

Table 11: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the Workspace
WorkflowSID	String	""	The SID of the Workflow
FriendlyName	String	""	(optional) Human readable friendly name. It can be 512 characters long
AssignmentCallbackURL	URL	""	(optional) The URL that we call when a task managed by the Workflow is assigned to a Worker.
FallbackAssignmentCallbackUrl	URL	""	(optional) The URL that we call when a call to the assignment_callback_url fails.
Configuration	String	""	(optional) A URL-encoded JSON string that contains the Workflow's configuration. For more information, see Workflows .
TaskReservationTimeout	Integer		(optional) How long Rindap will wait for a confirmation response from your application after notifying the <i>assignment_callback_url</i> about the reservation.

Example code pieces using SDKs

Shell

```

curl -X PUT https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
/Workflows/WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
--data-urlencode 'FriendlyName=my newly named workflow' \
--data-urlencode 'TaskReservationTimeout=3600' \
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"

```

Java

```

// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.Workflow;

public class Example {

```

(continues on next page)

(continued from previous page)

```
// Find your Account Sid and Token at rindap.com/console

public static void main(String[] args) {

    Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

    Workflow wf = Workflow
        .updater("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        ↪ "WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
        .setFriendlyName("my newly named workflow")
        .setTaskReservationTimeout(3600)
        .update();

    System.out.println(w);
}
}
```

Phyton

```
from rindap.rest import Client
from rindap.rest import Rindap

# Authenticate
client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

# Get Workspace with workspace_sid
workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

# Update Workflow
updated_workflow = workflow.update("New Workflow Name", task_reservation_timeout=3600)

print("WorkflowSid: {}".format(updated_workflow.sid))
print("FriendlyName: {}".format(updated_workflow.friendly_name))
```

JS

```
var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Update a workflows with SID
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
    .workflows("WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
    .update({
        friendlyName: "New Workflow Name",
        taskReservationTimeout: 3600
    }, function(err, workflow) {
        console.log(workflow.sid);
        console.log(workflow.friendlyName);
    });
```

C#

```
using System;
using Rindap;
```

(continues on next page)

(continued from previous page)

```

using Rindap.Rest.V1.Workspace;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // Update a workflow
        var workflow = WorkflowResource.Update(
            pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            pathSid: "WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            assignmentCallbackUrl: new Uri("https://example.org"),
            friendlyName: "New workflow name"
        );

        // Print workflow content
        Console.WriteLine("WorkflowSID : " + workflow.Sid);
        Console.WriteLine("Friendly Name : " + workflow.FriendlyName);
        Console.WriteLine("Callback URI : " + workflow.AssignmentCallbackUrl);
    }
}

```

The above command returns JSON structured like this:

```

{
  "sid": "WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "friendly_name": "my newly named workflow",
  "configuration": {
    "first_step": "FS111",
    "filters": {
      "FS111": {
        "true_queue_sid": "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        "false_filter_sid": "FS222",
        "name": "Hot",
        "conditions": {
          ">": [
            {
              "var": "temp"
            },
            "300"
          ]
        }
      },
      "FS222": {
        "true_queue_sid": "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        "name": "Mild",
        "conditions": {
          "!=": [
            "1",
            "1"
          ]
        }
      }
    }
  }
}

```

(continues on next page)

Phyton

```
from rindap.rest import Client
from rindap.rest import Rindap

# Authenticate
client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

# Get Workspace with workspace_sid
workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

# Delete workflow with sid
if workspace.workflows.get("WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx").delete():
    print("Workflow has been deleted!")
```

JS

```
var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Delete a workflows with SID
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
    .workflows("WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
    .remove();
```

C#

```
using System;
using Rindap;
using Rindap.Rest.V1.Workspace;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // Update a workflows workflow
        var isDeleted = WorkflowResource.Delete(
            pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            pathSid: "WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
        );

        if (isDeleted)
        {
            Console.WriteLine("Workflow has been deleted!");
        }
    }
}
```

2.3 Workers

In order to perform more tasks, perform tasks more precisely and in-time, Workers interact with TaskQueues. Workers always listen to multiple TaskQueues or no Queue at all. When a task is feeded to TaskQueue, based on eligibility, capability and role of a Worker, a task is assigned to Worker to perform it. For example, a Worker can be a person in a call center, a web service endpoint that listens to instructions, or an expert fixing the system failure.

2.3.1 Worker Properties

Table 13: Properties

field	description
sid	The unique string that we created to identify the Worker resource.
account_sid	The SID of the Account that created the Worker resource.
workspace_sid	The SID of the Workspace that contains the Worker
available	Whether the Worker is available to perform tasks.
friendly_name	The string that you assigned to describe the resource. Friendly names are case insensitive, and unique within the Rindap Workspace.
attributes	A JSON String for describing the Worker and its features such as skills and information. This value is used for deciding which TaskQueues the Worker can receive Tasks from
activity	<p>This Activity determines the Worker's current state in the system, as well as whether the</p> <ol style="list-style-type: none"> 1. idle: the Worker is available and is waiting for Task Assignments 2. busy: the Worker is not available. The Worker is set to this Activity when a Reservation for this Worker is accepted. 3. offline: the Worker is not available. 4. reserved: This is a system Activity. At this stage, the Worker is paired with a Task and a reservation is created.
date_created	The date and time in GMT when the resource was created, specified in ISO 8601 format.
date_status_changed	The date and time in GMT when the Activity of the Worker was last updated, specified in ISO 8601 format.
date_updated	The date and time in GMT when the resource was last updated, specified in ISO 8601 format.
url	The absolute URL of the resource
links	The URLs of related resources.

2.3.2 Worker Attributes

A worker's attributes are used to simply and accurately represent a working entity on Rindap. These attributes can be skills and information about a person, in the case of a human Worker. Or it can be some identifying data and hard-coded parameters for a digital entity like a web service. When populating this field, you should keep in mind that the value in some of these fields might be used for creating the TaskQueue relations of this Worker, according to the *worker_requirements* field of TaskQueues, which is simply a JsonLogic rule. For example, a worker with attributes such as the one below:

```
{
  "name" : "john doe",
```

(continues on next page)

(continued from previous page)

```

"age": 44,
"department" : "support",
"location": "Utah"
}

```

would be receiving Tasks from a TaskQueue with worker_requirements as such:

```

{
  "==": [{ "var": "department" }, "support" ]
}

```

Because the requirement rules for the TaskQueue would be matching the attributes of the Worker.

Note: Understanding the “requirements vs attributes” model is fundamental.

With this **requirements vs attributes** model, you do not need to point out which TaskQueues a Worker needs to receive Tasks from or which Workers are suitable for the Tasks in a TaskQueue. When you create a Worker, the Worker will automatically be receiving Tasks from appropriate TaskQueues. And when attributes of a Worker are updated, the Workers related TaskQueues are updated accordingly.

The same goes for TaskQueues : When requirements for a TaskQueue are updated, the related Workers for the TaskQueue are updated.

2.3.3 Create A Worker

POST /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Workers

You can create a Worker by simply providing a FriendlyName

Table 14: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the <i>Workspace</i>
FriendlyName	String	""	Human readable friendly name. It can be 512 characters long
Activity	String	""	(optional) Activity for the created Worker to be assigned to. If not provided, The default_activity for the Workspace will be used
Attributes	String	""	A URL-encoded JSON string with the features and skills of a Worker. This value is used for deciding which TaskQueues the Worker can receive Task s from.
RateLimitProfileSid	String	""	(optional) The SID of the RateLimitProfile for overseeing the maximum rate of reservations this Worker can have hourly. For more information see Rate Limit Profiles

Example code pieces using SDKs

Shell

```

curl -X POST https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX/Workers \
--data-urlencode 'FriendlyName=my test worker'

```

(continues on next page)

(continued from previous page)

```
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"
```

Java

```
// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.Worker;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        Worker w = Worker.creator("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
            .setFriendlyNameAttributes("my test worker")
            .create();

        System.out.println(w);
    }
}
```

Phyton

```
from rindap.rest import Client
from rindap.rest import Rindap

# Authenticate
client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

# Get Workspace with workspace_sid
workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

# Create a worker
worker = workspace.workers.create("My New Worker", activity="idle",
                                   task_queues=["WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"],
                                   rate_limit_profile_sid=
↳ "RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

# Print content of worker
print("FriendlyName: {}".format(worker.friendly_name))
print("WorkerSid: {}".format(worker.sid))
```

JS

```
var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Crate a worker
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
    .workers
```

(continues on next page)

(continued from previous page)

```

.create({
  friendlyName: "Friendly Worker Name" ,
  activity: "idle",
  rateLimitProfileSid: "RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  taskQueues: ["WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
↪ "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"]
}, function(err, worker) {
  console.log(worker.sid);
  console.log(worker.friendlyName);
  console.log(worker.activity);
  console.log(worker.rateLimitProfileSid);
  console.log(worker.taskQueues);
});

```

C#

```

using System;
using System.ComponentModel.DataAnnotations;
using Rindap;
using Rindap.Rest.V1;
using Rindap.Rest.V1.Workspace;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_AUTH_TOKEN", "YOUR_AUTH_TOKEN");

        // Create a worker
        var worker = WorkerResource.Create(
            pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            friendlyName: "New Worker",
            activity: "offline",
            taskQueues: new String[] { "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" },
            rateLimitProfileSid: "RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
        );

        // Print content of worker
        Console.WriteLine("Friendly Name      : " + worker.FriendlyName);
        Console.WriteLine("Activity          : " + worker.Activity);
        Console.WriteLine("TaskQueues       : " + string.Join(", ", worker.
↪ TaskQueues));
        Console.WriteLine("RateLimitProfileSid: " + worker.RateLimitProfileSid);
    }
}

```

The above command returns JSON structured like this:

```

{
  "sid": "WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "queues": [
    "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  ],
  "activity": "offline",

```

(continues on next page)

(continued from previous page)

```

        System.out.println(w);
    }
}

```

Phyton

```

from rindap.rest import Client
from rindap.rest import Rindap

client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

workspace = rindap.workspaces.get("WSb9d8cf8597f64f77a45666c4c0263862")
worker_fetcher = workspace.workers.list(limit=10, page_size=5)
worker = worker_fetcher.pop()
print("FriendlyName: {}".format(worker.friendly_name))
print("WorkerSid: {}".format(worker.sid))

```

JS

```

var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Get all workers
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
    .workers
    .list({
        limit: 100,
        pageSize: 100,
    }, function(err, workers) {
        workers.forEach(function(worker) {
            console.log(worker.sid);
            console.log(worker.friendlyName);
            console.log(worker.activity);
            console.log(worker.rateLimitProfileSid);
            console.log(worker.taskQueues);
        });
    });

```

C#

```

using System;
using System.ComponentModel.DataAnnotations;
using Rindap;
using Rindap.Rest.V1;
using Rindap.Rest.V1.Workspace;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_AUTH_TOKEN", "YOUR_AUTH_TOKEN");

        // Get all workers
    }
}

```

(continues on next page)

(continued from previous page)

```

var workers = WorkerResource.Read(
    pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    limit: 100,
    pageSize: 100
);

foreach (var worker in workers)
{
    // Print content of worker
    Console.WriteLine("Worker SID           : " + worker.Sid);
    Console.WriteLine("Friendly Name       : " + worker.FriendlyName);
    Console.WriteLine("Activity            : " + worker.Activity);
    Console.WriteLine("TaskQueues          : " + string.Join(", ", worker.
↪TaskQueues));
    Console.WriteLine("RateLimitProfileSid: " + worker.RateLimitProfileSid);
}
}
}

```

The above commands return JSON structured like this:

```

{
  "meta": {
    "page_size": 50,
    "page": 0,
    "first_page_url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workers?Page=0&PageSize=50",
    "previous_page_url": null,
    "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workers?Page=0&PageSize=50",
    "key": "workers",
    "next_page_url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workers?Page=1&PageSize=50"
  },
  "workers": [
    {
      "sid": "WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "queues": [
        "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
      ],
      "activity": "offline",
      "available": false,
      "friendly_name": "my test worker",
      "date_created": "2020-05-04T11:03:41+03:00",
      "date_updated": "2020-05-04T11:03:41+03:00",
      "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workers/WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "links": {
        "workspace": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
      }
    }
  ]
}

```

2.3.5 Fetch A Worker

GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Workers/{WorkerSID}

This endpoint fetches a single Worker with all Its details

Table 16: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the <i>Workspace</i>
WorkerSid	String	""	The SID of the Worker

Example code pieces using SDKs

Shell

```
curl -X GET https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
/Workers/WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"
```

Java

```
// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.Worker;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        Worker w = Worker
            .fetcher("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
↳"WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
            .fetch();

        System.out.println(w);
    }
}
```

Phyton

```
from rindap.rest import Client
from rindap.rest import Rindap

client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

workspace = rindap.workspaces.get("WSb9d8cf8597f64f77a45666c4c0263862")
worker = workspace.workers.get("WKdc8c97f1e1c24410ae08605ff6e5d946").fetch()
print("FriendlyName: {}".format(worker.friendly_name))
print("WorkerSid: {}".format(worker.sid))
```

JS

```

var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Get a workers with sid
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
  .workers("WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx").fetch(
    function(err, worker) {
      console.log(worker.sid);
      console.log(worker.friendlyName);
      console.log(worker.activity);
      console.log(worker.rateLimitProfileSid);
      console.log(worker.taskQueues);
    });

```

C#

```

using System;
using Rindap;
using Rindap.Rest.V1.Workspace;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_AUTH_TOKEN", "YOUR_AUTH_TOKEN");

        // Get a worker with SID
        var worker = WorkerResource.Fetch(
            pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            pathSid: "WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
        );

        // Print content of worker
        Console.WriteLine("Worker SID           : " + worker.Sid);
        Console.WriteLine("Friendly Name       : " + worker.FriendlyName);
        Console.WriteLine("Activity            : " + worker.Activity);
        Console.WriteLine("TaskQueues          : " + string.Join(", ", worker.
→TaskQueues));
        Console.WriteLine("RateLimitProfileSid: " + worker.RateLimitProfileSid);
    }
}

```

The above command returns JSON structured like this:

```

{
  "sid": "WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "queues": [
    "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  ],
  "activity": "offline",
  "available": false,

```

(continues on next page)

(continued from previous page)

```

{
  "friendly_name": "my test worker",
  "date_created": "2020-05-04T11:03:41+03:00",
  "date_updated": "2020-05-04T11:03:41+03:00",
  "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workers/WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "links": {
    "workspace": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  }
}

```

2.3.6 Update A Worker

```
PUT /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Workers/{WorkerSID}
```

Table 17: Query Parameters

Parameter	Type	Default	Description
WorkspaceSid	String	""	The SID of the <i>Workspace</i>
WorkerSid	String	""	The SID of the <i>Workspace</i> with the <i>Worker</i> to update
FriendlyName	String	""	(optional) Human readable friendly name. It can be 512 characters long
Activity	String	""	(optional) Activity for <i>Worker</i> . A worker needs to be in <i>idle</i> activity to receive a suitable <i>Task</i>
Attributes	String	""	A URL-encoded JSON string with the features and skills of a <i>Worker</i> . This value is used for deciding which <i>TaskQueues</i> the <i>Worker</i> can receive <i>Tasks</i> from. When you update this value , the related <i>TaskQueues</i> of this <i>Worker</i> are updated
RateLimitProfileSid	String	""	(optional) The SID of the new <i>RateLimitProfile</i> for the <i>Worker</i>

Example code pieces using SDKs

Shell

```
curl -X PUT https://api.rindap.com/v1/rindap-rest-gw/Workspaces/  
WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \\  
/Workers/WKxxxxxxxxxxxxxxxxxxxxxxxx \\  
--data-urlencode 'Activity=idle' \\  
--data-urlencode 'TaskQueues=WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx, \\  
WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx, WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx' \\  
--data-urlencode 'AssignmentStatus=pending' \\  
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \\  
-H "Content-Type:application/x-www-form-urlencoded"
```

Java

```
// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.Worker;

public class Example {
```

(continues on next page)

(continued from previous page)

```
// Find your Account Sid and Token at rindap.com/console

public static void main(String[] args) {

    Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

    Worker w = Worker
        .updater("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        ↪ "WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
        .setActivity("idle")
        .addTaskQueue("WQ11xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
        .addTaskQueue("WQ22xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
        .update();

    System.out.println(w);
}
}
```

Phyton

```
from rindap.rest import Client
from rindap.rest import Rindap

client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

workspace = rindap.workspaces.get("WSb9d8cf8597f64f77a45666c4c0263862")
worker = workspace.workers.get("WKdc8c97f1e1c24410ae08605ff6e5d946")
wk = worker.update(friendly_name="New Worker Name", activity="idle")
print("FriendlyName: {}".format(wk.friendly_name))
print("WorkerSid: {}".format(wk.sid))
```

JS

```
var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Update a workers
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
    .workers("WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx").update({
        friendlyName: "New Friendly Worker Name" ,
        activity: "offline",
        rateLimitProfileSid: "RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        taskQueues: ["WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"]
    } , function(err, worker) {
        console.log(worker.sid);
        console.log(worker.friendlyName);
        console.log(worker.activity);
        console.log(worker.rateLimitProfileSid);
        console.log(worker.taskQueues);
    });
```

C#

```

using System;
using Rindap;
using Rindap.Rest.V1.Workspace;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_AUTH_TOKEN", "YOUR_AUTH_TOKEN");

        // Update a worker with SID
        var worker = WorkerResource.Update(
            pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            pathSid: "WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            rateLimitProfileSid: "RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            friendlyName: "New Friendly Name",
            taskQueues: new string[] { "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
↪ "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" }
        );

        // Print content of worker
        Console.WriteLine("Worker SID           : " + worker.Sid);
        Console.WriteLine("Friendly Name       : " + worker.FriendlyName);
        Console.WriteLine("Activity            : " + worker.Activity);
        Console.WriteLine("TaskQueues          : " + string.Join(", ", worker.
↪ TaskQueues));
        Console.WriteLine("RateLimitProfileSid: " + worker.RateLimitProfileSid);
    }
}

```

The above command returns JSON structured like this:

```

{
  "sid": "WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "queues": [
    "WQ00xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "WQ11xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "WQ22xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  ],
  "activity": "idle",
  "available": true,
  "friendly_name": "my test worker",
  "date_created": "2020-05-04T11:03:41+03:00",
  "date_updated": "2020-05-04T11:03:41+03:00",
  "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workers/WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "links": {
    "workspace": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  }
}

```


2.3.7 Delete A Worker

DELETE /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Workers/{WorkerSID}

Table 18: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the <i>Workspace</i>
WorkerSID	String	""	The SID of the Worker

Example code pieces using SDKs

Shell

```
curl -X DEL https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
  /Workers/WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"
```

Java

```
// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.Worker;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        Worker.deleter("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
↪ "WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
            .delete();
    }
}
```

Phyton

```
from rindap.rest import Client
from rindap.rest import Rindap

client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

workspace = rindap.workspaces.get("WSb9d8cf8597f64f77a45666c4c0263862")
if workspace.workers.get("WK369cfb99700446deal8fcabbd9f1a68c").delete():
    print("Worker has been deleted")
```

JS

```
var Rindap = require('rindap');
```

(continues on next page)

(continued from previous page)

```
// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Delete a workers with sid
rindap.workspaces('WS70cdccad050a41619162db0f32b7fc43')
    .workers("WK2b9748b550ee40a2bd010f06c78511e3").remove();
```

C#

```
using System;
using Rindap;
using Rindap.Rest.V1.Workspace;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_AUTH_TOKEN", "YOUR_AUTH_TOKEN");

        // Delete a worker with SID
        var isDeleted = WorkerResource.Delete(
            pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            pathSid: "WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
        );

        if (isDeleted)
        {
            Console.WriteLine("Worker has been deleted!");
        }
    }
}
```

2.4 Tasks

In process management, a task is an activity that needs to be done, leading to the final deliverable. Tasks are a small essential piece of work that must be carried out to progress the whole project. For example, in business process systems, a process could consist of several tasks or activities to be completed in a defined period of time. CRM or ticketing systems allow organizations to generate tasks such as creating reminders in customer accounts that are synced with the calendar.

2.4.1 Task Attributes

Task attributes are the core features of Tasks that are evaluated in the Workflow process. Consequently, based on the evaluation results, the path a task will follow in the Workflow is determined. Attributes are expressed in JSON data, for example:

2.4.2 Task Lifecycle

A Task has a Lifecycle to manage a task. When a task is created, either manually or automatically by a system, it can pass through several states it is completed by its own or is closed manually. A Task's lifecycle starts when the task

is generated, then the task is assigned, is processed, is completed and finally is verified. In the process management system, a user submits a task to a Workflow. Afterwards, the Task's attributes are evaluated against the Filters in the Workflow, starting from the starting Filter defined in the Workflow until a match is found. When a match is found, the Task is pushed to the TaskQueue designated by the matching Filter. Task waits in the queue till a capable and available Worker is handled the Task.

2.4.3 Task Properties

Table 19: Properties

field	description
sid	The unique string that we created to identify the Task resource.
account_sid	The SID of the Account that created the Task resource.
workspace_sid	The SID of the Workspace that contains the Task
workflow_sid	The SID of the Workflow that is controlling the Task
workflow_friendly_name	The friendly name of the Workflow that is controlling the Task
initial_attributes	A JSON String for the attributes of the task. This value cannot be changed, for showing the original state of the Task
attributes	A JSON String for the attributes of the task. This field can be updated throughout the Lifecycle of the Task and will be used for evaluating filters of the Task's Workflow
assignment_status	The current status of the Task's assignment. Can be: pending, awaiting_reservation,reserved, postponed, accepted, cancelled or completed.
task_queue_sid	The SID of the TaskQueue that the Task was last sent to
task_queue_sid_friendly_name	The friendly name of the TaskQueue that the Task was last sent to
age	The number of seconds since the Task was created.
step_history	A JSON Array of JSON Objects, showing the steps which were selected for the Task throughout its lifecycle. The JSON Objects hold the Step Id and the Date it was selected for the Task
task_queue_history	A JSON Array of JSON Objects, showing the Task Queues to which the Task was sent, throughout its lifecycle. The JSON Objects hold the Task Queue SID, Task Queue friendly name and the Date
last_charge_date	The date The Task was last charged
next_charge_date	The date The Task will be charged again, If It has not ended Its lifecycle
total_cost	The total cost of charges for The Task so far
forked_from	The SID of the Parent Task, if this Task is forked from another Task. For more information, see Task Forking
postponed_till	The date and time in GMT, till the task is postponed, valid when The Task AssignmentStatus is <i>postponed</i> .
postponing_reason	The explanation of the reason for postponing, valid when The Task AssignmentStatus is <i>postponed</i> .
loop_retries_left	The number of retries left for the current Loop Filter in the Workflow, valid when The Task AssignmentStatus is <i>postponed</i> . For more information, see Workflow Configuration .
date_created	The date and time in GMT when the resource was created, specified in ISO 8601 format.
date_updated	The date and time in GMT when the resource was last updated, specified in ISO 8601 format.
url	The absolute URL of the resource
links	The URLs of related resources.

2.4.4 Create A Task

POST /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Tasks

You can create a task by simply providing the Task Attributes and the Workflow for the task

Table 20: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the Workspace in which the Task is created
WorkflowSID	String	""	The SID of the Workflow that you would like Task to be handled by
Attributes	String	""	A URL-encoded JSON string with the attributes of the new task. This value is passed to the Workflow's <code>assignment_callback_url</code> when the Task is assigned to a Worker.
Timeout	Integer	""	(Optional) (Max=2073600) (Min=60) Timeout value in seconds, for the Task to be closed after, if the Task <code>assignment_status</code> is not <i>completed</i> or <i>cancelled</i> at the time of control. Task will be "cancelled" with reason "Task TTL Exceeded"

Example code pieces using SDKs

Shell

```
curl -X POST https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX/Tasks \
--data-urlencode 'WorkflowSid=WWb898bae42dec49e792f37f9639bf625c'
--data-urlencode 'Attributes={"some_integer": 55,"some_text": "hello world","some_
↳boolean" : true}' \
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"
```

Java

```
// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.Task;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        Task t = Task.creator("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
            .setAttributes("{\"some_integer\": 55, \"some_text\": \"hello world\", \"
↳some_boolean\" : true}")
            .setWorkflowSid("WWXXXXXXXXXXXXXXXXXXXXXXXXXXXX")
            .create();

        System.out.println(t);
    }
}
```

Python

```

from rindap.rest import Client
from rindap.rest import Rindap

# Authenticate
client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

# Get Workspace with workspace_sid
workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

# Create a task
task = workspace.tasks.create(workflow_sid="WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
                              attributes='{"some_integer": 55, "some_text": "hello world"
↪      , "some_boolean" : true}')

# Print content of task
print("TaskSid: {}".format(task.sid))

```

JS

```

var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Create a task
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
  .tasks
  .create({
    attributes: '{"some_integer": 55, "some_text": "hello world", "some_boolean" :_
↪true}',
    workflowSid: 'WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'
  }, function(err, task) {
    console.log(task.sid);
    console.log(task.attributes);
    console.log(task.workflowSid);
  });

```

CSharp

```

using System;
using System.Collections.Generic;
using Newtonsoft.Json;
using Rindap;
using Rindap.Rest.V1.Workspace;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // Create an attributes
        var attributes = JsonConvert.SerializeObject(new Dictionary<string, Object>()
        {

```

(continues on next page)

(continued from previous page)

```

        {"type", "support"}
    }, Formatting.Indented);

    // Create a task
    var task = TaskResource.Create(
        pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        workflowSid: "WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        attributes: attributes,
        timeout: 100
    );

    // Print task content
    Console.WriteLine("Workspace Sid : " + task.Sid);
    foreach (var attribute in task.Attributes)
    {
        Console.WriteLine("Attributes: ");
        Console.WriteLine("\tKey    : " + attribute.Key);
        Console.WriteLine("\tValue  : " + attribute.Value);
    }
    Console.WriteLine("Timeout      : " + task.Timeout);
    Console.WriteLine("WorkflowSid   : " + task.WorkflowSid);
}

```

The above command returns JSON structured like this:

```

{
  "sid": "WT9a69287d97fe4110a76ef9db8cf728f8",
  "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workflow_sid": "WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workflow_friendly_name": "my first workflow",
  "initial_attributes": {
    "some_integer": 55,
    "some_text": "hello world"
    "some_boolean" : true
  },
  "attributes": {
    "some_integer": 55,
    "some_text": "hello world"
    "some_boolean" : true
  },
  "assignment_status": "pending",
  "step": -1,
  "reason": null,
  "date_created": "2020-01-01T16:11:35+03:00",
  "date_updated": "2020-01-01T16:11:35+03:00",
  "last_charge_date": "2020-01-01T16:11:35+03:00",
  "next_charge_date": "2020-02-01T16:11:35+03:00",
  "total_cost": "0.01",
  "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Tasks/WT9a69287d97fe4110a76ef9db8cf728f8",
  "links": {
    "workspace": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "workflow": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workflows/WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  }
}

```

(continues on next page)

(continued from previous page)

```

},
"age": 0
}

```

2.4.5 Get All Tasks

GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Tasks

This endpoint retrieves all Tasks

Table 21: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the Workspace in which the Task is created
FriendlyName	String	""	Human readable friendly name
PageSize	Integer	50	Page size for paging
Page	Integer	0	Page number for paging

Example code pieces using SDKs

C#

```

using System;
using Rindap;
using Rindap.Rest.V1.Workspace;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // Fetch all tasks
        var tasks = TaskResource.Read(
            pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            limit: 100,
            pageSize: 100
        );

        // Iterate on all tasks
        foreach (var task in tasks)
        {
            // Print task content
            Console.WriteLine("Workspace Sid : " + task.Sid);
            foreach (var attribute in task.Attributes)
            {
                Console.WriteLine("Attributes: ");
                Console.WriteLine("\tKey : " + attribute.Key);
                Console.WriteLine("\tValue : " + attribute.Value);
            }
            Console.WriteLine("Timeout : " + task.Timeout);
            Console.WriteLine("WorkflowSid : " + task.WorkflowSid);
        }
    }
}

```

(continues on next page)

(continued from previous page)

```

    }
}

```

Python

```

from rindap.rest import Client
from rindap.rest import Rindap

# Authenticate
client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

# Get Workspace with workspace_sid
workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

# Fetch all tasks
task_fetcher = workspace.tasks.list(limit=10, page_size=5)
for task in task_fetcher:
    # Print content of task
    print("TaskSid: {}".format(task.sid))

```

JS

```

var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// List all tasks
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
    .tasks
    .list({
        limit: 100,
        pageSize: 100
    }, function(err, tasks) {
        tasks.forEach(function(task) {
            console.log(task.sid);
            console.log(task.attributes);
            console.log(task.workflowSid);
        })
    });

```

The above command returns JSON structured like this:

```

{
  "meta": {
    "page_size": 50,
    "page": 0,
    "first_page_url": "string",
    "previous_page_url": "string",
    "url": "string",
    "key": "string",
    "next_page_url": "string"
  },
  "workspaces": [
    {

```

(continues on next page)

(continued from previous page)

```

    "sid": "string",
    "friendly_name": "string",
    "event_callback_url": "string",
    "account_sid": "string",
    "date_created": "2020-04-02T12:30:58.083Z",
    "date_updated": "2020-04-02T12:30:58.083Z",
    "url": "string",
    "links": {
      "tasks": "string",
      "workers": "string",
      "workflows": "string",
      "task_queues": "string"
    }
  }
}
]
}

```

2.4.6 Fetch a Task

This endpoint fetches a single Task with all its details

GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Tasks/{TaskSID}

Table 22: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the Workspace in which the Task is created
TaskSID	String	""	TaskSID (Secure Identifier) - a unique ID for the Task

Example code pieces using SDKs

Shell

```

curl -X GET https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
/Tasks/WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"

```

Java

```

// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.Task;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        Task t = Task

```

(continues on next page)

(continued from previous page)

```

        .fetcher("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        ↪ "WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
        .fetch();

        System.out.println(t);
    }
}

```

Python

```

from rindap.rest import Client
from rindap.rest import Rindap

# Authenticate
client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

# Get Workspace with workspace_sid
workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

# Get a task with sid
task = workspace.tasks.get("WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx").fetch()

# Print content of task
print("TaskSid: {}".format(task.sid))

```

JS

```

var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Get a tasks with SID
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
    .tasks("WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
    .fetch(function(err, task) {
        console.log(task.sid);
        console.log(task.attributes);
        console.log(task.workflowSid);
    });

```

C#

```

using System;
using Rindap;
using Rindap.Rest.V1.Workspace;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // Get a tasks with SID
    }
}

```

(continues on next page)

(continued from previous page)

```

var task = TaskResource.Fetch(
    pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    pathSid: "WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
);

// Print task content
Console.WriteLine("Workspace Sid : " + task.Sid);
foreach (var attribute in task.Attributes)
{
    Console.WriteLine("Attributes: ");
    Console.WriteLine("\tKey    : " + attribute.Key);
    Console.WriteLine("\tValue  : " + attribute.Value);
}
Console.WriteLine("Timeout      : " + task.Timeout);
Console.WriteLine("WorkflowSid   : " + task.WorkflowSid);
}
}

```

The above command returns JSON structured like this:

```

{
  "sid": "WT9a69287d97fe4110a76ef9db8cf728f8",
  "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workflow_sid": "WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workflow_friendly_name": "my first workflow",
  "initial_attributes": {
    "some_integer": 55,
    "some_text": "hello world",
    "some_boolean": true
  },
  "attributes": {
    "some_integer": 55,
    "some_text": "hello world",
    "some_boolean": true
  },
  "assignment_status": "pending",
  "step": -1,
  "reason": null,
  "date_created": "2020-01-01T16:11:35+03:00",
  "date_updated": "2020-01-01T16:11:35+03:00",
  "last_charge_date": "2020-01-01T16:11:35+03:00",
  "next_charge_date": "2020-02-01T16:11:35+03:00",
  "total_cost": "0.01",
  "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Tasks/WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "links": {
    "workspace": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "workflow": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workflows/WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  },
  "age": 0
}

```

2.4.7 Update a Task

PUT /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Tasks/{TaskSID}

Table 23: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the Workspace in which the Task is created
TaskSID	String	""	TaskSID (Secure Identifier) - a unique ID for the Task
Attributes	String	""	(Optional) A URL-encoded JSON string that will replace the <i>Attributes</i> field of your Task. You should use this parameter for updating the data related to this Task.
AssignmentStatus	String	""	(Optional) you can update the assignment status of your task by one of these options: <i>pending</i> , 'completed', 'cancelled'. If you set your task's <i>assignment_status</i> to <i>pending</i> , it will be processed by Its Workflow once again, from the Workflow Step that's shown at the <i>step</i> field of the Task. If you set this field to <i>cancelled</i> , you can also send the <i>Reason</i> parameter for telling what reason the Task is cancelled for.
Reason	String	""	(Optional) Can only be sent when the <i>AssignmentStatus</i> parameter is set as <i>cancelled</i> as well.

Example code pieces using SDKs

Shell

```
curl -X PUT https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
/Tasks/WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
--data-urlencode 'Attributes={"result_of_some_query": 55,"some_text": "new related_
↳ info"}' \
--data-urlencode 'AssignmentStatus=pending' \
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"
```

Java

```
// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.Task;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        Task t = Task
            .updater("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
↳ "WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
            .setAttributes("{\"result_of_some_query\": 55, \"some_text\": \"new_
↳ related info\"}")
            .setAssignmentStatus("pending")
```

(continues on next page)

(continued from previous page)

```

        .update();

        System.out.println(t);
    }
}

```

Python

```

from rindap.rest import Client
from rindap.rest import Rindap

# Authenticate
client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

# Get Workspace with workspace_sid
workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

# Get a task with sid
task = workspace.tasks.get("WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

# Update task
updated_task = task.update(assignment_status="completed")

# Print content of task
print("TaskSid: {}".format(updated_task.sid))

```

JS

```

var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Update a tasks
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
    .tasks("WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
    .update({
        assignmentStatus: "cancelled"
    }, function(err, task) {
        console.log(err);
        console.log(task.sid);
        console.log(task.attributes);
        console.log(task.workflowSid);
    });

```

C#

```

using System;
using Rindap;
using Rindap.Rest.V1.Workspace;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
    }
}

```

(continues on next page)

(continued from previous page)

```

RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Update a tasks with SID
var task = TaskResource.Update(
    pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    pathSid: "WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    assignmentStatus: "cancelled"
);

// Print task content
Console.WriteLine("Workspace Sid : " + task.Sid);
foreach (var attribute in task.Attributes)
{
    Console.WriteLine("Attributes: ");
    Console.WriteLine("\tKey    : " + attribute.Key);
    Console.WriteLine("\tValue  : " + attribute.Value);
}
Console.WriteLine("Timeout      : " + task.Timeout);
Console.WriteLine("WorkflowSid   : " + task.WorkflowSid);
}

```

The above command returns JSON structured like this:

```

{
  "sid": "WT9a69287d97fe4110a76ef9db8cf728f8",
  "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workflow_sid": "WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workflow_friendly_name": "my first workflow",
  "initial_attributes": {
    "some_integer": 55,
    "some_text": "hello world"
    "some_boolean" : true
  },
  "attributes": {
    "result_of_some_query": 55,
    "some_text": "new related info"
  },
  "assignment_status": "pending",
  "step": -1,
  "reason": null,
  "date_created": "2020-01-01T16:11:35+03:00",
  "date_updated": "2020-01-01T16:11:35+03:00",
  "last_charge_date": "2020-01-01T16:11:35+03:00",
  "next_charge_date": "2020-02-01T16:11:35+03:00",
  "total_cost": "0.01",
  "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
  ↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Tasks/WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "links": {
    "workspace": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
  ↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "workflow": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
  ↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workflows/WWxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  },
  "age": 0
}

```

2.4.8 Delete a Task

DELETE /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Tasks/{TaskSID}

Table 24: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the Workspace in which the Task is created
TaskSID	String	""	TaskSID (Secure Identifier) - a unique ID for the Task

Example code pieces using SDKs

Shell

```
curl -X DEL https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
/Tasks/WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"
```

Java

```
// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.Task;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        Task.deleter("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
↳"WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
            .delete();
    }
}
```

Python

```
from rindap.rest import Client
from rindap.rest import Rindap

# Authenticate
client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

# Get Workspace with workspace_sid
workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

# Get a task with sid and delete it
if workspace.tasks.get("WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx").delete():
    print("Task has been deleted!")
```

JS

```
var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Get a tasks with SID
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
  .tasks("WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
  .remove();
```

C#

```
using System;
using Rindap;
using Rindap.Rest.V1.Workspace;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // Get a tasks with SID
        var isDeleted = TaskResource.Delete(
            pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            pathSid: "WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
        );

        if (isDeleted)
        {
            Console.WriteLine("Task has been deleted!");
        }
    }
}
```

2.5 Rate Limit Profiles

RateLimitProfiles define a limit for the maximum number of Reservations that can be created Hourly.

When a Worker has a `rate_limit_profile` set, you guarantee that the Worker will not be assigned more than a certain number of Tasks **hourly**, regardless of the Task's nature , or which Workflow It's processed through or which TaskQueue it comes from. You can read more about [Rate Limits and Rate Limit Profiles](#) here .

Using Rate Limit Profiles helps you group similar Workers together and change their pace of Task consumption from a single point of setting. For example , a Call Center may set a rate limit profile for all the Workers in the support department, which lets them all have 12 calls per hour in the morning session , and then update the Rate Limit Profile and let them have 6 calls per hour in the afternoon session, halving their workload and redirecting it to some other department, with a single Rate Limit Profile update.

- **With utilizing Rate Limit Profiles , you can :**
 - change the pace of Task consumption or
 - enforce quotas on limited resources and budgets or

- prevent bottlenecks on your application side

You can read more about [Rate Limits](#) and [Rate Limit Profiles](#) here

2.5.1 RateLimitProfile Properties

Table 25: Properties

field	description
sid	The unique string that we created to identify the Rate Limit Profile resource.
account_sid	The SID of the Account that created the RateLimitProfile resource.
workspace_sid	The SID of the Workspace that contains the RateLimitProfile
friendly_name	The string that you assigned to describe the resource.
reservations_per_hour	The maximum number of Reservations that can be created hourly, for a Worker with this RateLimitProfile
date_created	The date and time in GMT when the resource was created, specified in ISO 8601 format.
date_updated	The date and time in GMT when the resource was last updated, specified in ISO 8601 format.
url	The absolute URL of the RateLimitProfile resource
links	The URLs of related resources.

2.5.2 Create A RateLimitProfile

POST /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/RateLimitProfiles

You can easily create a RateLimitProfile by providing a friendly name and the number of maximum hourly reservations

Table 26: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the <i>Workspace</i>
FriendlyName	String	""	Human readable friendly name. It can be 512 characters long
ReservationsPerHour	Integer	""	The maximum number of Reservations that can be created hourly, for a Worker with this RateLimitProfile

Example code pieces using SDKs

Shell

```
curl -X POST https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/RateLimitProfiles \
--data-urlencode 'FriendlyName=my rate limit profile for 6 reservations an Hour'
--data-urlencode 'ReservationsPerHour=6'
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"
```

Java

```
// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.RateLimitProfile;
```

(continues on next page)

(continued from previous page)

```

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        RateLimitProfile rl = RateLimitProfile.creator(
        ↪ "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            "my rate limit profile for 6 reservations an Hour",
            6
        )
        .create();

        System.out.println(rl);
    }
}

```

Python

```

from rindap.rest import Client
from rindap.rest import Rindap

client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

workspace = rindap.workspaces.get("WSb9d8cf8597f64f77a45666c4c0263862")
rtp = workspace.rate_limit_profiles.create("My Rate Limit", reservation_per_hour=7)
print(rtp)

```

JS

```

var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Crate a reservation
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
.rateLimitProfiles
.create({
    friendlyName: 'New Rate Limit',
    reservationsPerHour: 4
}, function(err, rateLimitProfile) {
    console.log(rateLimitProfile.sid);
    console.log(rateLimitProfile.friendlyName);
    console.log(rateLimitProfile.reservationsPerHour);
});

```

C#

```

using System;
using Rindap;
using Rindap.Rest.V1.Workspace;

class Program

```

(continues on next page)

(continued from previous page)

```

{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // Create a RateLimitProfile
        var rateLimitProfile = RateLimitProfileResource.Create(
            pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            friendlyName: "New RateLimitProfile",
            reservationsPerHour: 7
        );

        // Print RateLimitProfile Content
        Console.WriteLine("SID           : " + rateLimitProfile.Sid);
        Console.WriteLine("FriendlyName       : " + rateLimitProfile.FriendlyName);
        Console.WriteLine("ReservationPerHour: " + rateLimitProfile.
↪ReservationPerHour);
    }
}

```

The above command returns JSON structured like this:

```

{
  "sid": "RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "friendly_name": "my rate limit profile for 6 reservations an Hour",
  "reservations_per_hour": 6,
  "date_created": "2020-05-06T16:24:36+03:00",
  "date_updated": "2020-05-06T16:24:36+03:00",
  "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/RateLimitProfiles/
↪RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "links": {
    "workspace": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  }
}

```

2.5.3 Get All RateLimitProfiles

GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/RateLimitProfiles

This endpoint retrieves all RateLimitProfiles

Table 27: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the <i>Workspace</i>
PageSize	Integer	50	Page size for paging
Page	Integer	0	Page number for paging

Example code pieces using SDKs

Shell

```
curl -X GET https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/RateLimitProfiles \
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"
```

Java

```
// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.RateLimitProfile;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        RateLimitProfile.Reader reader = RateLimitProfile.reader(
↳"WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx");

        for(RateLimitProfile rl:reader.read())
            System.out.println(rl);
    }
}
```

Python

```
from rindap.rest import Client
from rindap.rest import Rindap

client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

workspace = rindap.workspaces.get("WSb9d8cf8597f64f77a45666c4c0263862")
rate_limit_profiles = workspace.rate_limit_profiles.list(limit=10, page_size=5)
for rate_limit_profile in rate_limit_profiles:
    print("RateLimitProfileSid: {}".format(rate_limit_profile.sid))
    print("FriendlyName: {}".format(rate_limit_profile.friendly_name))
```

JS

```
var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Get all ratelimits
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
    .rateLimitProfiles
    .list({
        limit: 100,
```

(continues on next page)

(continued from previous page)

```

    pageSize: 100
  }, function(err, rateLimitProfiles) {
    rateLimitProfiles.forEach(function(rateLimitProfile) {
      console.log(rateLimitProfile.sid);
      console.log(rateLimitProfile.friendlyName);
      console.log(rateLimitProfile.reservationsPerHour);
    });
  });
});

```

C#

```

using System;
using Rindap;
using Rindap.Rest.V1.Workspace;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // List All RateLimitProfiles
        var rateLimitProfiles = RateLimitProfileResource.Read(
            pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            limit: 100,
            pageSize: 100
        );

        foreach (var rateLimitProfile in rateLimitProfiles)
        {
            // Print RateLimitProfile Content
            Console.WriteLine("SID                : " + rateLimitProfile.Sid);
            Console.WriteLine("FriendlyName       : " + rateLimitProfile.FriendlyName);
            Console.WriteLine("ReservationPerHour: " + rateLimitProfile.
↵ReservationPerHour);
        }
    }
}

```

The above command returns JSON structured like this:

```

{
  "meta": {
    "page_size": 50,
    "page": 0,
    "first_page_url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↵WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/RateLimitProfiles?Page=0&PageSize=50",
    "previous_page_url": null,
    "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↵WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/RateLimitProfiles?Page=0&PageSize=50",
    "key": "rate_limit_profiles",
    "next_page_url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↵WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/RateLimitProfiles?Page=1&PageSize=50"
  },
  "rate_limit_profiles": [
    {

```

(continues on next page)

(continued from previous page)

```

    "sid": "RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "friendly_name": "my rate limit profile for 6 reservations an Hour",
    "reservations_per_hour": 6,
    "date_created": "2020-05-06T16:24:36+03:00",
    "date_updated": "2020-05-06T16:24:36+03:00",
    "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/RateLimitProfiles/
↪RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "links": {
        "workspace": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
    }
}
]
}

```

2.5.4 Fetch a RateLimitProfile

GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSid}/RateLimitProfiles/{RateLimitProfileSID}

This endpoint fetches a single RateLimitProfile with all its details

Table 28: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the <i>Workspace</i>
RateLimitProfileSID	String	""	The SID of the RateLimitProfile

Example code pieces using SDKs

Shell

```

curl -X GET https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
/RateLimitProfiles/RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"

```

Java

```

// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.RateLimitProfile;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");
    }
}

```

(continues on next page)

(continued from previous page)

```

        RateLimitProfiles rl = RateLimitProfile
            .fetcher("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
↪ "RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
            .fetch();

        System.out.println(rl);
    }
}

```

Python

```

from rindap.rest import Client
from rindap.rest import Rindap

client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
rate_limit_profile = workspace.rate_limit_profiles.get(
↪ 'RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx').fetch()
print("RateLimitProfileSid: {}".format(rate_limit_profile.sid))
print("FriendlyName: {}".format(rate_limit_profile.friendly_name))

```

JS

```

var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Get a ratelimit with SID
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
.rateLimitProfiles("RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
.fetch(function(err, rateLimitProfile) {
    console.log(rateLimitProfile.sid);
    console.log(rateLimitProfile.friendlyName);
    console.log(rateLimitProfile.reservationsPerHour);
});

```

C#

```

using System;
using Rindap;
using Rindap.Rest.V1.Workspace;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // Get a RateLimitProfile with SID
        var rateLimitProfile = RateLimitProfileResource.Fetch(
            pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            pathSid: "RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
        );
    }
}

```

(continues on next page)

(continued from previous page)

```
// Print RateLimitProfile Content
Console.WriteLine("SID           : " + rateLimitProfile.Sid);
Console.WriteLine("FriendlyName   : " + rateLimitProfile.FriendlyName);
Console.WriteLine("ReservationPerHour: " + rateLimitProfile.
↪ReservationPerHour);
    }
}
```

The above command returns JSON structured like this:

```
{
  "sid": "RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "friendly_name": "my rate limit profile for 6 reservations an Hour",
  "reservations_per_hour": 6,
  "date_created": "2020-05-06T16:24:36+03:00",
  "date_updated": "2020-05-06T16:24:36+03:00",
  "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/RateLimitProfiles/
↪RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "links": {
    "workspace": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  }
}
```

2.5.5 Update a RateLimitProfile

PUT /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/RateLimitProfiles/{RateLimitProfileSID}

Warning: When you update the *reservations_per_hour* field, all the Workers with this RateLimitProfile will be affected immediately, Thus receiving Reservations according to this new value.

Table 29: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the Workspace with the Rate Limit Profile to update
RateLimitProfileSID	String	""	The SID of the Rate Limit Profile to be updated
FriendlyName	String	""	(optional) Human readable friendly name. It can be 512 characters long
ReservationsPerHour	Integer	""	(optional) The maximum number of Reservations that can be created hourly, for a Worker with this RateLimitProfile

Example code pieces using SDKs

Shell


```
curl -X PUT https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
/RateLimitProfiles/RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
--data-urlencode 'FriendlyName=my newly named rate limit profile' \
--data-urlencode 'ReservationsPerHour=30' \
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"
```

Java

```
// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.RateLimitProfile;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        RateLimitProfile rl = RateLimitProfile
            .updater("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
↳"RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
            .setFriendlyName("my newly named rate limit profile")
            .setReservationsPerHour(30)
            .update();

        System.out.println(rl);
    }
}
```

Python

```
from rindap.rest import Client
from rindap.rest import Rindap

client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

workspace = rindap.workspaces.get("WSb9d8cf8597f64f77a45666c4c0263862")
rate_limit_profile_fetcher = workspace.rate_limit_profiles.get(
↳'RL8e22315c14294c749db7eee2d9d7bc27')
rate_limit_profile = rate_limit_profile_fetcher.update(friendly_name="New New Rate_
↳Limit", reservation_per_hour=9)
print("RateLimitProfileSid: {}".format(rate_limit_profile.sid))
print("FriendlyName: {}".format(rate_limit_profile.friendly_name))
```

JS

```
var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Update a ratelimit with SID
```

(continues on next page)

(continued from previous page)

```

rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
.rateLimitProfiles("RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
.update({
  friendlyName: "New Friendly Name",
  reservationsPerHour: 7
}, function(err, rateLimitProfile) {
  console.log(rateLimitProfile.sid);
  console.log(rateLimitProfile.friendlyName);
  console.log(rateLimitProfile.reservationsPerHour);
});

```

C#

```

using System;
using Rindap;
using Rindap.Rest.V1.Workspace;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // Update RateLimitProfile
        var rateLimitProfile = RateLimitProfileResource.Update(
            pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            pathSid: "RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            friendlyName: "New Friendly Name",
            reservationsPerHour: 6
        );

        // Print RateLimitProfile Content
        Console.WriteLine("SID                : " + rateLimitProfile.Sid);
        Console.WriteLine("FriendlyName         : " + rateLimitProfile.FriendlyName);
        Console.WriteLine("ReservationPerHour: " + rateLimitProfile.
↪ReservationPerHour);
    }
}

```

The above command returns JSON structured like this:

```

{
  "sid": "RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "friendly_name": "my newly named rate limit profile",
  "reservations_per_hour": 30,
  "date_created": "2020-05-06T16:24:36+03:00",
  "date_updated": "2020-05-06T16:24:36+03:00",
  "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/RateLimitProfiles/
↪RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "links": {
    "workspace": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  }
}

```

(continues on next page)

(continued from previous page)

}

2.5.6 Delete a RateLimitProfile

DELETE /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/RateLimitProfiles/{RateLimitProfileSID}

Warning: When you DELETE a RateLimitProfile, all the Workers with this RateLimitProfile will be affected immediately, Thus receiving Reservations with NO LIMITS

Table 30: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the Workspace with the Rate Limit Profile
RateLimitProfileSID	String	""	The SID of the Rate Limit Profile

Example code pieces using SDKs

Shell

```
curl -X DEL https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
  ↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
  /RateLimitProfiles/RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"
```

Java

```
// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.RateLimitProfile;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        RateLimitProfile.deleter("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  ↳ "RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
            .delete();
    }
}
```

Python

```
from rindap.rest import Client
from rindap.rest import Rindap

client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
```

(continues on next page)

(continued from previous page)

```
rindap = Rindap(client)

workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
if workspace.rate_limit_profiles.get('RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx').delete():
    print("RateLimitProfile has been deleted!")
```

JS

```
var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Delete a ratelimit with SID
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
.rateLimitProfiles("RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
.remove();
```

C#

```
using System;
using Rindap;
using Rindap.Rest.V1.Workspace;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // Update RateLimitProfile
        var isDeleted = RateLimitProfileResource.Delete(
            pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            pathSid: "RLxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
        );

        if (isDeleted)
        {
            Console.WriteLine("RateLimitProfile has been deleted!");
        }
    }
}
```

2.6 Task Queues

TaskQueue helps the process management system to manage and control how tasks are executed and determine capable Workers to accomplish those Tasks. Tasks are evaluated through predefined Workflow and based on the evaluation results, Tasks are given to TaskQueue to handle by an appropriate Worker.

2.6.1 TaskQueue Properties

Table 31: Properties

field	description
sid	The unique string that we created to identify the TaskQueue resource.
account_sid	The SID of the Account that created the TaskQueue resource.
workspace_sid	The SID of the Workspace that contains the TaskQueue
friendly_name	The string that you assigned to describe the resource.
worker_requirements	A JSON String for the JsonLogic rule , deciding the Worker relations for the TaskQueue
date_created	The date and time in GMT when the resource was created, specified in ISO 8601 format.
date_updated	The date and time in GMT when the resource was last updated, specified in ISO 8601 format.
url	The absolute URL of the RateLimitProfile resource
links	The URLs of related resources.

TaskQueue worker requirements

A TaskQueue's worker requirements is a JsonLogic rule, used to simply and accurately represent the necessary features or skills required for handling a Task in this TaskQueue

For example , a worker with attributes such as the one below:

```
{
  "name" : "john doe"
  "age": 44
  "department" : "support"
  "location": "Utah"
}
```

would be receiving Tasks from a TaskQueue with worker_requirements as such:

```
{
  "==": [ { "var": "department" }, "support" ]
}
```

Because the requirement rules for the TaskQueue would be matching the attributes of the Worker.

Note: Understanding the “requirements vs attributes” model is fundamental.

With this **requirements vs attributes** model, you do not need to point out which TaskQueues a Worker needs to receive Tasks from or which Workers are suitable for the Tasks in a TaskQueue.

When you create a Worker , the Worker will automatically be receiving Tasks from appropriate TaskQueues. And when attributes of a Worker are updated, the Workers related TaskQueues are updated accordingly.

The same goes for TaskQueues : When requirements for a TaskQueue are updated , the related Workers for the TaskQueue are updated.

2.6.2 Create A TaskQueue

POST /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/TaskQueues

You can create a TaskQueue by simply providing a *FriendlyName*

Table 32: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the <i>Workspace</i>
FriendlyName	String	""	Human readable friendly name. It can be 512 characters long
WorkerRequirements	JSON Object	{"=="(boolean)}	(Optional) A URL-encoded JSON string , representing a JsonLogic rule for Worker relation

Example code pieces using SDKs

Shell

```
curl -X POST https://api.rindap.com/v1/rindap-rest-gw/Workspaces/  
WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/TaskQueues \\  
--data-urlencode 'FriendlyName=my test task queue' \\  
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \\  
-H "Content-Type:application/x-www-form-urlencoded"
```

Java

```
// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.TaskQueue;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        TaskQueue tq = TaskQueue.creator("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
            .setFriendlyName("my test task queue")
            .create();

        System.out.println(tq);
    }
}
```

Python

```
from rindap.rest import Client
from rindap.rest import Rindap

# Authenticate
client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

# Get Workspace with workspace_sid
workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

# Create a task queue
```

(continues on next page)

(continued from previous page)

```
task_queue = workspace.task_queues.create("My new task queue")

# Print task queue content
print("TaskQueue FriendlyName: {}".format(task_queue.friendly_name))
print("TaskQueue Sid: {}".format(task_queue.sid))
```

JS

```
var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Create a task queue
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
.taskQueues
.create({
  friendlyName: "Friendly Name"
}, function(err, taskQueue) {
  console.log(taskQueue.sid);
  console.log(taskQueue.friendlyName);
});
```

C#

```
using System;
using Rindap;
using Rindap.Rest.V1.Workspace;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // Create a taskQueue
        var taskQueue = TaskQueueResource.Create(
            pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            friendlyName: "New Task Queue"
        );

        // Print taskQueue content
        Console.WriteLine("TaskQueue Sid : " + taskQueue.Sid);
        Console.WriteLine("Friendly Name : " + taskQueue.FriendlyName);
    }
}
```

The above command returns JSON structured like this:

```
{
  "sid": "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "friendly_name": "my test task queue",
  "date_created": "2020-05-04T09:38:26+03:00",
```

(continues on next page)

(continued from previous page)

```

"date_updated": "2020-05-04T09:38:26+03:00",
"url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/TaskQueues/WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
"links": {
  "workspace": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
}
}

```

2.6.3 Get All TaskQueues

GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/TaskQueues

Table 33: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the <i>Workspace</i>
FriendlyName	String	""	(optional) Human readable friendly name
PageSize	Integer	50	Page size for paging
Page	Integer	0	Page number for paging

Example code pieces using SDKs

Shell

```

curl -X GET https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/TaskQueues \
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"

```

Java

```

// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.TaskQueue;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        TaskQueue.Reader reader = TaskQueue.reader("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
↳");

        for(TaskQueue tq:reader.read())
            System.out.println(tq);
    }
}

```

Python


```

from rindap.rest import Client
from rindap.rest import Rindap

# Authenticate
client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

# Get Workspace with workspace_sid
workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

# Get all task queues
task_queue_fetcher = workspace.task_queues.list(limit=10, page_size=5)
for task_queue in task_queue_fetcher:
    # Print task queue content
    print("TaskQueue FriendlyName: {}".format(task_queue.friendly_name))
    print("TaskQueue Sid: {}".format(task_queue.sid))

```

JS

```

var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// List all task queues
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
.taskQueues
.list({
  limit: 100,
  pageSize: 100
}, function(err, taskQueues) {
  taskQueues.forEach(function(taskQueue) {
    console.log(taskQueue.sid);
    console.log(taskQueue.friendlyName);
  })
});

```

C#

```

using System;
using Rindap;
using Rindap.Rest.V1.Workspace;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // List all taskQueues
        var taskQueues = TaskQueueResource.Read(
            pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            limit: 100,
            pageSize: 100
        );
    }
}

```

(continues on next page)

(continued from previous page)

```

foreach (var taskQueue in taskQueues)
{
    // Print taskQueue content
    Console.WriteLine("TaskQueue Sid : " + taskQueue.Sid);
    Console.WriteLine("Friendly Name : " + taskQueue.FriendlyName);
}
}

```

The above command returns JSON structured like this:

```

{
  "meta": {
    "page_size": 50,
    "page": 0,
    "first_page_url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
    ↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/TaskQueues?Page=0&PageSize=50",
    "previous_page_url": null,
    "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
    ↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/TaskQueues?Page=0&PageSize=50",
    "key": "task_queues",
    "next_page_url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
    ↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/TaskQueues?Page=1&PageSize=50"
  },
  "task_queues": [
    {
      "sid": "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "friendly_name": "my test task queue",
      "date_created": "2020-05-04T09:38:26+03:00",
      "date_updated": "2020-05-04T09:38:26+03:00",
      "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
      ↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/TaskQueues/WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "links": {
        "workspace": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
        ↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
      }
    }
  ]
}

```

2.6.4 Fetch a TaskQueue

GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/TaskQueues/{TaskQueueSID}

This endpoint fetches a single TaskQueue with all its details

Table 34: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the <i>Workspace</i>
TaskQueueSID	String	""	The SID of the TaskQueue

Example code pieces using SDKs

Shell

```
curl -X GET https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
/TaskQueues/WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"
```

Java

```
// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.TaskQueue;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        TaskQueue tq = TaskQueue
            .fetcher("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
↳ "WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
            .fetch();

        System.out.println(tq);
    }
}
```

Python

```
from rindap.rest import Client
from rindap.rest import Rindap

# Authenticate
client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

# Get Workspace with workspace_sid
workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

# Get task with sid
task_queue = workspace.task_queues.get("WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx").fetch()

# Print task queue content
print("TaskQueue FriendlyName: {}".format(task_queue.friendly_name))
print("TaskQueue Sid: {}".format(task_queue.sid))
```

JS

```
var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");
```

(continues on next page)

(continued from previous page)

```
// Get a task queues with SID
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
.taskQueues("WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
.fetch(function(err, taskQueue) {
  console.log(taskQueue.sid);
  console.log(taskQueue.friendlyName);
});
```

C#

```
using System;
using Rindap;
using Rindap.Rest.V1.Workspace;

class Program
{
  static void Main(string[] args)
  {
    // Authenticate
    RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

    // Get a taskQueue with SID
    var taskQueue = TaskQueueResource.Fetch(
      pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      pathSid: "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
    );

    // Print taskQueue content
    Console.WriteLine("TaskQueue Sid : " + taskQueue.Sid);
    Console.WriteLine("Friendly Name : " + taskQueue.FriendlyName);
  }
}
```

The above command returns JSON structured like this:

```
{
  "sid": "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "friendly_name": "my test task queue",
  "date_created": "2020-05-04T09:38:26+03:00",
  "date_updated": "2020-05-04T09:38:26+03:00",
  "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
  ↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/TaskQueues/WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "links": {
    "workspace": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
    ↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  }
}
```

2.6.5 Update a TaskQueue

PUT /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/TaskQueues/{TaskQueueSID}

Table 35: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the <i>Workspace</i>
TaskQueueSID	String	""	The SID of the TaskQueue
FriendlyName	String	""	Human readable friendly name. It can be 512 characters long
WorkerRequirements	JSON Object	{ "==" : { "friendlyName" : "" } }	(Optional) A URL-encoded JSON string , representing a JsonLogic rule for deciding the Worker relations for the TaskQueue

Example code pieces using SDKs

Shell

```
curl -X PUT https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
/TaskQueues/WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
--data-urlencode 'FriendlyName=my new name of task queue' \
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"
```

Java

```
// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.TaskQueue;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        TaskQueue tq = TaskQueue
            .updater("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
↳ "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
            .setFriendlyName("my new name of task queue")
            .update();

        System.out.println(tq);
    }
}
```

Python

```
from rindap.rest import Client
from rindap.rest import Rindap

# Authenticate
client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

# Get Workspace with workspace_sid
workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
```

(continues on next page)

(continued from previous page)

```
# Get task with sid
task_queue = workspace.task_queues.get("WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
updated_task_queue = task_queue.update(friendly_name="My Task Queue New Name")

# Print task queue content
print("TaskQueue FriendlyName: {}".format(updated_task_queue.friendly_name))
print("TaskQueue Sid: {}".format(updated_task_queue.sid))
```

JS

```
var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Update a task queues with SID
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
.taskQueues("WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
.update({
  friendlyName: "New Friendly Name"
}, function(err, taskQueue) {
  console.log(taskQueue.sid);
  console.log(taskQueue.friendlyName);
});
```

C#

```
using System;
using Rindap;
using Rindap.Rest.V1.Workspace;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // Update a taskQueue with SID
        var taskQueue = TaskQueueResource.Update(
            pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            pathSid: "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            friendlyName: "New Friendly Name"
        );

        // Print taskQueue content
        Console.WriteLine("TaskQueue Sid : " + taskQueue.Sid);
        Console.WriteLine("Friendly Name : " + taskQueue.FriendlyName);
    }
}
```

The above command returns JSON structured like this:

```
{
  "sid": "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
```

(continues on next page)

(continued from previous page)

```

"workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
"friendly_name": "my new name of task queue",
"date_created": "2020-05-04T09:38:26+03:00",
"date_updated": "2020-05-04T09:38:26+03:00",
"url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/TaskQueues/WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
"links": {
  "workspace": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
}
}

```

2.6.6 Delete a TaskQueue

DELETE /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/TaskQueues/{TaskQueueSID}

Table 36: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the <i>Workspace</i>
TaskQueueSID	String	""	The SID of the <i>TaskQueue</i>

Example code pieces using SDKs

Shell

```

curl -X DEL https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
/TaskQueues/WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"

```

Java

```

// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
import com.rindap.rest.v1.workspace.TaskQueue;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        TaskQueue.deleter("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
↳"WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
            .delete();
    }
}

```

Python

```
from rindap.rest import Client
from rindap.rest import Rindap

# Authenticate
client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

# Get Workspace with workspace_sid
workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

# Get task with sid and delete it
if workspace.task_queues.get("WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx").delete():
    print("TaskQueue has been deleted!")
```

JS

```
var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Delete a task queues with SID
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
.taskQueues("WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
.remove();
```

C#

```
using System;
using Rindap;
using Rindap.Rest.V1.Workspace;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // Delete a taskQueue with SID
        var isDeleted = TaskQueueResource.Delete(
            pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            pathSid: "WQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
        );

        if (isDeleted)
        {
            Console.WriteLine("TaskQueue has been deleted!");
        }
    }
}
```


2.7 Reservations

Rindap creates a Reservation subresource whenever a Task is reserved for a Worker. Rindap will provide the details of this Reservation subresource in the Assignment Callback HTTP request it makes to your application server. You can read more about [Reservations here](#)

You have multiple options for handling a Reservation:

- Respond to the Assignment Callback with an Assignment Instruction.
- Call the REST API with how to handle it.

You can read more about [Reservations here](#)

2.7.1 Reservation Properties

Table 37: Properties

field	description
sid	The unique string that we created to identify the Reservation resource.
account_sid	The SID of the Account
workspace_sid	The SID of the Workspace
worker_sid	The SID of the reserved Worker resource
task_sid	The SID of the reserved Task resource
reservation_status	The current status of the reservation. Can be: pending, accepted, rejected.
date_created	The date and time in GMT when the resource was created, specified in ISO 8601 format.
date_updated	The date and time in GMT when the resource was last updated, specified in ISO 8601 format.
url	The absolute URL of the RateLimitProfile resource
links	The URLs of related resources.

Accessing Reservations

There are 3 ways to access a Reservation resource:

- For accessing a certain Reservation with Its SID , You can access It through Its Task or Worker.
- For accessing all Reservations of a certain Task, you need to access through the Task
- For accessing all Reservations of a certain Worker, you need to access through the Worker

2.7.2 Through a Task: Fetching a Reservation

GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Tasks/{TaskSID}/Reservations/{ReservationSID}

This endpoint fetches a single Reservation with all its details

Table 38: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the <i>Workspace</i>
TaskSID	String		The SID of the Task
ReservationSID	String		The SID of the Reservation

Example code pieces using SDKs

Shell

```
curl -X GET https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
/Tasks/WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Reservations/
↳ WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"
```

Java

```
// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
// IMPORTING from com.rindap.rest.v1.workspace.task package
import com.rindap.rest.v1.workspace.task.Reservation;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        Reservation r=Reservation.fetcher(
            "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            "WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx", // Task SID
            "WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
            .fetch();

        System.out.println(r);
    }
}
```

Python

```
from rindap.rest import Client
from rindap.rest import Rindap

# Authenticate
client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)
```

(continues on next page)

(continued from previous page)

```
# Get Workspace with workspace_sid
workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

# Get Task with SID
task = workspace.tasks.get("WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx").fetch()

# Get reservation with SID
reservation = task.reservations.get("WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx").fetch()
print("Reservation SID: {}".format(reservation.sid))
print("Reservation Status: {}".format(reservation.reservation_status))
```

JS

```
var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Get a reservation with task and reservation SID
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
  .tasks('WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
  .reservations('WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
  .fetch(function(err, reservation) {
    console.log(reservation.sid);
    console.log(reservation.reservationStatus);
  });
```

C#

```
using System;
using Rindap;
using Rindap.Rest.V1.Workspace.Task;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // Fetch a reservation with SID
        var reservation = ReservationResource.Fetch(
            pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            pathTaskSid: "WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            pathSid: "WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
        );

        // Print Reservation Content
        Console.WriteLine("Reservation SID : " + reservation.Sid);
        Console.WriteLine("Reservation Status: " + reservation.ReservationStatus);
    }
}
```

The above command returns JSON structured like this:

```
{
  "sid": "WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "worker_sid": "WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "task_sid": "WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "reservation_status": "rejected",
  "date_created": "2020-04-23T13:47:15+03:00",
  "date_updated": "2020-04-23T13:49:42+03:00",
  "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Tasks/WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/
↳ Reservations/WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "links": {
    "task": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Tasks/WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "worker": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workers/WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "workspace": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  }
}
```

2.7.3 Through a Task: Listing All Reservations of A Task

GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Tasks/{TaskSID}/Reservations

This endpoint retrieves all Reservations of a Task

Table 39: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String	""	The SID of the <i>Workspace</i>
TaskSID	String		The SID of the Task
ReservationStatus	String		(optional) The current status of the reservation. Can be: pending, accepted, rejected, completed. Can be used for filtering
WorkerSid	String		(optional) The SID of the reserved Worker. Can be used to filter the Reservations by Worker

Example code pieces using SDKs

Shell

```
curl -X GET https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
/Tasks/WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Reservations
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"
```

Java

```
// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
// IMPORTING from com.rindap.rest.v1.workspace.task package
import com.rindap.rest.v1.workspace.task.Reservation;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        Reservation.Reader reader = Reservation.reader(
            "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            "WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
            .setReservationStatus(Reservation.Status.ACCEPTED)
            ;

        for(Reservation r:reader.read())
            System.out.println(r);
    }
}
```

Python

```
from rindap.rest import Client
from rindap.rest import Rindap

# Authenticate
client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

# Get Workspace with workspace_sid
workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

# Get Task with SID
task = workspace.tasks.get("WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx").fetch()

# Get all reservation
reservations = task.reservations.list(limit=10, page_size=100)
for reservation in reservations:
    # Print Content of reservation
    print("Reservation SID: {}".format(reservation.sid))
    print("Reservation Status: {}".format(reservation.reservation_status))
```

JS

```
var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// List all reservations
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
    .tasks("WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
```

(continues on next page)

(continued from previous page)

```
.reservations
.list({
  limit: 100,
  pageSize: 100
}, function(err, reservations) {
  reservations.forEach(function(reservation) {
    console.log(reservation.sid);
    console.log(reservation.reservationStatus);
  });
});
```

C#

```
using System;
using Rindap;
using Rindap.Rest.V1.Workspace.Task;

class Program
{
  static void Main(string[] args)
  {
    // Authenticate
    RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

    // Get all reservations
    var reservations = ReservationResource.Read(
      pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      pathTaskSid: "WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      limit: 100,
      pageSize: 100
    );

    foreach (var reservation in reservations)
    {
      // Print Reservation Content
      Console.WriteLine("Reservation SID : " + reservation.Sid);
      Console.WriteLine("Reservation Status: " + reservation.ReservationStatus);
    }
  }
}
```

The above command returns JSON structured like this:

```
{
  "meta": {
    "page_size": 2147483647,
    "page": 0,
    "first_page_url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Tasks/WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/
↪Reservations?Page=0&PageSize=2147483647",
    "previous_page_url": null,
    "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Tasks/WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/
↪Reservations?Page=0&PageSize=2147483647",
    "key": "",
    "next_page_url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Tasks/WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/
↪Reservations?Page=1&PageSize=2147483647"
```

(continues on next page)

(continued from previous page)

```

},
"reservations": [
  {
    "sid": "WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "worker_sid": "WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "task_sid": "WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "reservation_status": "rejected",
    "date_created": "2020-04-23T13:47:15+03:00",
    "date_updated": "2020-04-23T13:49:42+03:00",
    "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Tasks/WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/
↪Reservations/WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "links": {
      "task": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Tasks/WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "worker": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workers/WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "workspace": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↪WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
    }
  }
]
}

```

2.7.4 Through a Task: Updating a reservation

PUT /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Tasks/{TaskSID}/Reservations/{ReservationSID}

Note: You can only update a Reservation while It's at the *pending* status

You can update a Reservation by updating Its ReservationStatus.

- **accepted** - use this for accepting a reservation, which means that the Worker will process the Task
- **rejected** - use this for rejecting a reservation. At this situation, Rindap will reject the Reservation and try to assign the Task to another available Worker
- **completed** - This is a special status setting, a shortcut, for saying that the Reservation is accepted and the Task is complete

After receiving this status,

- the Reservation will be updated as *accepted*
- the Task will be updated as *completed*
- the Worker will be updated as *idle* and will be available for Reservations

This is useful for Workers with short-lived functions such as a WebService endpoint where your application does whatever needed to be done, instantaneously

Table 40: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String		• The SID of the <i>Workspace</i>
TaskSID	String		• The SID of the Task
ReservationStatus	String		• the status of the reservation. Can be “accepted”, “rejected” or “completed”
ReservationSID	String		• The SID of the Reservation

Example code pieces using SDKs

Shell

```
curl -X PUT https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
/Tasks/WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Reservations/
↳WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
--data-urlencode 'ReservationStatus=accepted' \
--data-urlencode 'AssignmentStatus=pending' \
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"
```

Java

```
// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
// IMPORTING from com.rindap.rest.v1.workspace.task package
import com.rindap.rest.v1.workspace.task.Reservation;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        Reservation r=Reservation
            .updater(
                "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
                "WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx", // Task SID
                "WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
            .setReservationStatus(Reservation.Status.ACCEPTED)
            .update();

        System.out.println(r);
    }
}
```

(continues on next page)

(continued from previous page)

```
}
}
```

Python

```
from rindap.rest import Client
from rindap.rest import Rindap

# Authenticate
client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

# Get Workspace with workspace_sid
workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

# Get Task with SID
task = workspace.tasks.get("WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx").fetch()

# Get reservation with SID
reservation = task.reservations.get("WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx").fetch()

updated_reservation = reservation.update(reservation_status='rejected')
print("Reservation SID: {}".format(updated_reservation.sid))
print("Reservation Status: {}".format(updated_reservation.reservation_status))
```

JS

```
var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Update a reservation with SID
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
  .tasks("WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
  .reservations("WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
  .update({
    reservationStatus: 'rejected'
  }, function(err, reservation) {
    console.log(err);
    console.log(reservation.sid);
  });
```

C#

```
using System;
using Rindap;
using Rindap.Rest.V1.Workspace.Task;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // Fetch a reservation with SID
```

(continues on next page)

(continued from previous page)

```

var reservation = ReservationResource.Update(
    pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    pathTaskSid: "WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    pathSid: "WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    reservationStatus: ReservationResource.StatusEnum.Rejected
);

// Print Reservation Content
Console.WriteLine("Reservation SID : " + reservation.Sid);
Console.WriteLine("Reservation Status: " + reservation.ReservationStatus);
}
}

```

The above command returns JSON structured like this:

```

{
  "sid": "WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "worker_sid": "WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "task_sid": "WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "reservation_status": "rejected",
  "date_created": "2020-04-23T13:47:15+03:00",
  "date_updated": "2020-04-23T13:49:42+03:00",
  "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
  ↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Tasks/WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/
  ↳Reservations/WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "links": {
    "task": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
    ↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workers/WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "worker": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
    ↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workers/WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "workspace": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
    ↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  }
}

```

2.7.5 Through a Worker: Fetching a Reservation

GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Workers/{WorkerSID}/Reservations/{ReservationSID}

This endpoint fetches a single Reservation with all its details

Table 41: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String		• The SID of the <i>Workspace</i>
WorkerSID	String		• The SID of the Worker
ReservationSID	String		• The SID of the Reservation

Example code pieces using SDKs

Shell

```
curl -X GET https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
/Workers/WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Reservations/
↳WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"
```

Java

```
// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;

// IMPORTING from com.rindap.rest.v1.workspace.worker package
import com.rindap.rest.v1.workspace.worker.Reservation;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        Reservation r=Reservation.fetcher(
            "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            "WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx", // Worker SID
            "WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
            .fetch();

        System.out.println(r);
    }
}
```

Python

```
from rindap.rest import Client
from rindap.rest import Rindap

# Authenticate
client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

# Get Workspace with workspace_sid
workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

# Get Worker with SID
worker = workspace.workers.get("WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx").fetch()

# Get reservation with SID
reservation = worker.reservations.get("WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx").fetch()
print("Reservation SID: {}".format(reservation.sid))
print("Reservation Status: {}".format(reservation.reservation_status))
```

JS

```

var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Get a reservation with worker and reservation SID
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
  .workers("WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
  .reservations("WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
  .fetch(function(err, reservation) {
    console.log(reservation.sid);
  });

```

C#

```

using System;
using Rindap;
using Rindap.Rest.V1.Workspace.Worker;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // Fetch a reservation with SID
        var reservation = ReservationResource.Fetch(
            pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            pathWorkerSid: "WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            pathSid: "WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
        );

        // Print Reservation Content
        Console.WriteLine("Reservation SID : " + reservation.Sid);
        Console.WriteLine("Reservation Status: " + reservation.ReservationStatus);
    }
}

```

The above command returns JSON structured like this:

```

{
  "sid": "WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "worker_sid": "WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "task_sid": "WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "reservation_status": "rejected",
  "date_created": "2020-04-23T13:47:15+03:00",
  "date_updated": "2020-04-23T13:49:42+03:00",
  "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
  ↪ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workers/WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/
  ↪ Reservations/WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "links": {
    "task": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
    ↪ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Tasks/WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "worker": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
    ↪ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workers/WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",

```

(continues on next page)

(continued from previous page)

```

    "workspace": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
    ↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  }
}

```

2.7.6 Through a Worker: Listing All Reservations of A Worker

GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Workers/{WorkerSID}/Reservations

This endpoint retrieves all Reservations of a Worker

Table 42: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String		• The SID of the <i>Workspace</i>
WorkerSID	String		• The SID of the Worker
ReservationStatus	String		• (optional) The current status of the reservation. Can be: pending, accepted, rejected. Can be used for filtering

Example code pieces using SDKs

Shell

```

curl -X GET https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
/Workers/WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Reservations
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"

```

Java

```

// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
//IMPORTING Reservation from com.rindap.rest.v1.workspace.worker
import com.rindap.rest.v1.workspace.worker.Reservation;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        Reservation.Reader reader = Reservation.reader(
            "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",

```

(continues on next page)

(continued from previous page)

```

        "WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx");

        for(Reservation r:reader.read())
            System.out.println(r);
    }
}

```

Python

```

from rindap.rest import Client
from rindap.rest import Rindap

# Authenticate
client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

# Get Workspace with workspace_sid
workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

# Get Worker with SID
worker = workspace.workers.get("WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx").fetch()

# Get all reservation
reservations = worker.reservations.list(limit=10, page_size=100)
for reservation in reservations:
    # Print Content of reservation
    print("Reservation SID: {}".format(reservation.sid))
    print("Reservation Status: {}".format(reservation.reservation_status))

```

JS

```

var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// List all reservations
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
    .workers("WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
    .reservations
    .list({
        limit: 100,
        pageSize: 100
    }, function(err, reservations) {
        reservations.forEach(function(reservation) {
            console.log(reservation.sid);
            console.log(reservation.reservationStatus);
        });
    });

```

C#

```

using System;
using Rindap;
using Rindap.Rest.V1.Workspace.Worker;

class Program

```

(continues on next page)

(continued from previous page)

```

{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // List all reservations
        var reservations = ReservationResource.Read(
            pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            pathWorkerSid: "WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            limit: 100,
            pageSize: 100
        );

        foreach (var reservation in reservations)
        {
            // Print Reservation Content
            Console.WriteLine("Reservation SID    : " + reservation.Sid);
            Console.WriteLine("Reservation Status: " + reservation.ReservationStatus);
        }
    }
}

```

The above command returns JSON structured like this:

```

{
  "meta": {
    "page_size": 2147483647,
    "page": 0,
    "first_page_url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
    ↪ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workers/WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/
    ↪ Reservations?Page=0&PageSize=2147483647",
    "previous_page_url": null,
    "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
    ↪ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workers/WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/
    ↪ Reservations?Page=0&PageSize=2147483647",
    "key": "",
    "next_page_url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
    ↪ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workers/WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/
    ↪ Reservations?Page=1&PageSize=2147483647"
  },
  "reservations": [
    {
      "sid": "WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "worker_sid": "WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "task_sid": "WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "reservation_status": "rejected",
      "date_created": "2020-04-23T13:47:15+03:00",
      "date_updated": "2020-04-23T13:49:42+03:00",
      "url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
      ↪ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workers/WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/
      ↪ Reservations/WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "links": {
        "task": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
        ↪ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Tasks/WTxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        "worker": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
        ↪ WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workers/WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
      }
    }
  ]
}

```

(continues on next page)

(continued from previous page)

```

    "workspace": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
    WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  }
}
]
}
```

2.7.7 Through a Worker: Updating a reservation

```
PUT /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Workers/{WorkerSID}/Reservations/{ReservationSID}
```

Note: You can only update a Reservation while It's at the pending status

You can update a Reservation by updating Its ReservationStatus.

- **accepted** - use this for accepting a reservation, which means that the Worker will process the Task
- **rejected** - use this for rejecting a reservation. At this situation, Rindap will reject the Reservation and try to assign the Task to another available Worker
- **completed** - This is a special status setting, a shortcut, for saying that the Reservation is accepted and the Task is complete

After receiving this status,

- the Reservation will be updated as *accepted*
- the Task will be updated as *completed*
- the Worker will be updated as *idle* and will be available for Reservations

This is useful for Workers with short-lived functions such as a `WebService` endpoint where your application does whatever needed to be done, instantaneously

Table 43: Query Parameters

Parameter	Type	Default	Description
WorkspaceSID	String		<ul style="list-style-type: none"> The SID of the <i>Workspace</i>
WorkerSID	String		<ul style="list-style-type: none"> The SID of the Worker
ReservationStatus	String		<ul style="list-style-type: none"> the status of the reservation. Can be “accepted”, “rejected” or “completed”
ReservationSID	String		<ul style="list-style-type: none"> The SID of the Reservation

Example code pieces using SDKs

Shell


```
curl -X PUT https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
/Workers/WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Reservations/
↳WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \
--data-urlencode 'ReservationStatus=accepted' \
-H "Authorization: Bearer {YOUR_ACCOUNT_SID}.{YOUR_AUTH_TOKEN}" \
-H "Content-Type:application/x-www-form-urlencoded"
```

Java

```
// Install the Java helper library from rindap.com/docs/java/install

import com.rindap.Rindap;
//IMPORTING Reservation from package com.rindap.rest.v1.workspace.worker
import com.rindap.rest.v1.workspace.worker.Reservation;

public class Example {
    // Find your Account Sid and Token at rindap.com/console

    public static void main(String[] args) {

        Rindap.init("YOUR_ACCOUNT_SID","YOUR_AUTH_TOKEN");

        Reservation r=Reservation
            .updater(
                "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
                "WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx", // Worker SID
                "WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
            .setReservationStatus(Reservation.Status.ACCEPTED)
            .update();

        System.out.println(r);
    }
}
```

Python

```
from rindap.rest import Client
from rindap.rest import Rindap

# Authenticate
client = Client("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN")
rindap = Rindap(client)

# Get Workspace with workspace_sid
workspace = rindap.workspaces.get("WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")

# Get Worker with SID
worker = workspace.workers.get("WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx").fetch()

# Get reservation with SID
reservation = worker.reservations.get("WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx").fetch()

# Update Reservation
updated_reservation = reservation.update(reservation_status='rejected')
print("Reservation SID: {}".format(updated_reservation.sid))
print("Reservation Status: {}".format(updated_reservation.reservation_status))
```

JS

```
var Rindap = require('rindap');

// Authenticate
var rindap = new Rindap("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

// Update a reservation with SID
rindap.workspaces('WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
  .workers("WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
  .reservations("WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
  .update({
    reservationStatus: 'rejected'
  }, function(err, reservation) {
    console.log(err);
    console.log(reservation.sid);
  });
```

C#

```
using System;
using Rindap;
using Rindap.Rest.V1.Workspace.Worker;

class Program
{
    static void Main(string[] args)
    {
        // Authenticate
        RindapClient.Init("YOUR_ACCOUNT_SID", "YOUR_AUTH_TOKEN");

        // Update a reservation with SID
        var reservation = ReservationResource.Update(
            pathWorkspaceSid: "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            pathWorkerSid: "WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            pathSid: "WRxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
            reservationStatus: ReservationResource.StatusEnum.Completed
        );

        // Print Reservation Content
        Console.WriteLine("Reservation SID : " + reservation.Sid);
        Console.WriteLine("Reservation Status: " + reservation.ReservationStatus);
    }
}
```

The above command returns JSON structured like this:

```
{
  "sid": "WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "account_sid": "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "workspace_sid": "WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "queues": [
    "WQ00xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "WQ11xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "WQ22xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  ],
  "activity": "idle",
  "available": true,
```

(continues on next page)

(continued from previous page)

```
"friendly_name": "my test worker",
"date_created": "2020-05-04T11:03:41+03:00",
"date_updated": "2020-05-04T11:03:41+03:00",
"url": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/Workers/WKxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
"links": {
  "workspace": "https://api.rindap.com/v1/rindap-rest-gw/Workspaces/
↳WSxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
}
```

HTTP Routing Table

/v1

	34
GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Workers,	17
9	
GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/RateLimitProfiles,	11
63	
GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/TaskQueues,	68
76	
GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Tasks,	80
78	
GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Tasks,	56
51	
GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Tasks/{TaskSID},	91
53	
GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Tasks/{TaskSID}/Reservations,	42
88	
GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Tasks/{TaskSID}/Reservations/{ReservationSID},	100
85	
GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Workers,	28
37	
GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Workers/{WorkerSID},	14
40	
GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Workers/{WorkerSID}/Reservations,	71
97	
GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Workers/{WorkerSID}/Reservations/{ReservationSID},	83
94	
GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Workflows,	59
22	
GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/RateLimitProfiles/{RateLimitProfilesSID},	45
66	
GET /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Workflows/{WorkflowsSID},	31
25	
POST /v1/rindap-rest-gw/Workspaces/,4	
POST /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/RateLimitProfiles,	
61	
POST /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/TaskQueues,	
73	
POST /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Tasks,	
48	
POST /v1/rindap-rest-gw/Workspaces/{WorkspaceSID}/Workers,	