

API BUSINESS HUB

Consuming APIs with the SAP API Business Hub



45 min

Description

In this exercise, you'll learn how to consume API'S using API Hub.

System prerequisites

- Account on the SAP Cloud Platform (<https://account.hanatrial.ondemand.com>)

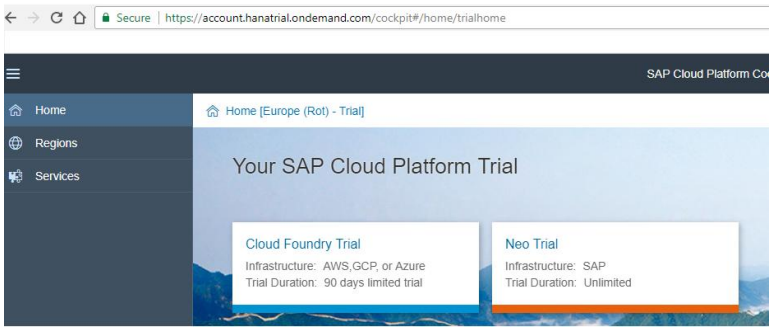
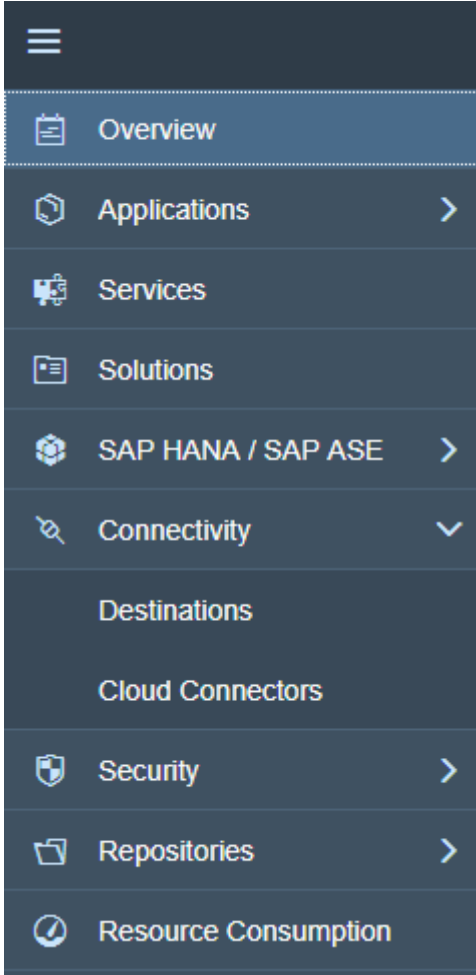
Target group

- Application developers
- People interested in the SAP Cloud Platform

Target group requirements

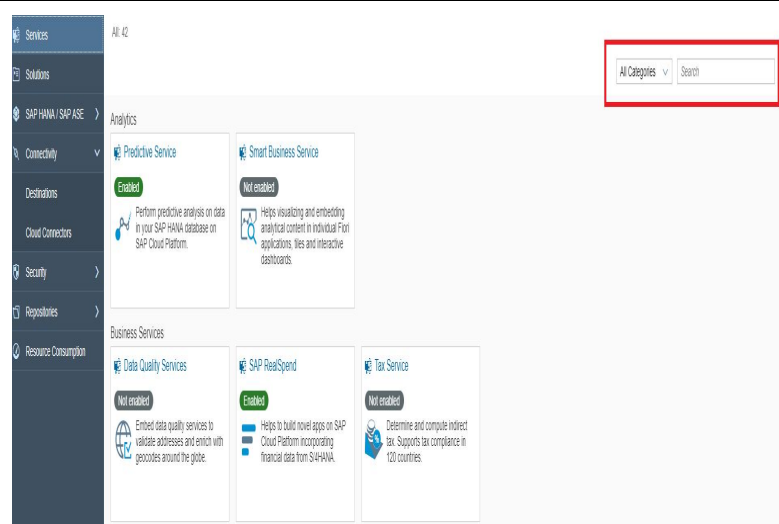
- Basic programming skills, ideally JavaScript

Scenario 1a: Onboarding – Register with the API Hub

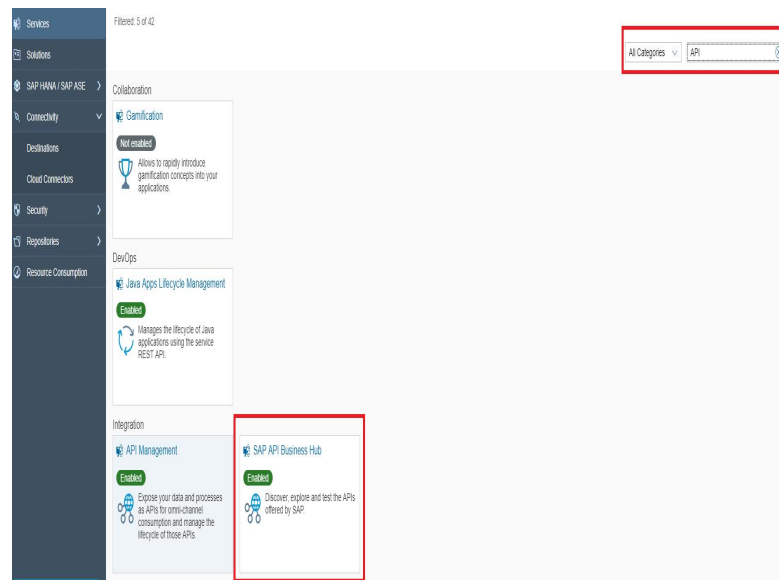
Explanation	Screenshot
<p>1. Connect to your SAP Cloud Platform trial account</p> <p>https://account.hanatrial.ondemand.com</p>	
<p>2. Choose Neo Trial. Then select Services in the left pane.</p>	

Consuming APIs with the SAP API Business Hub

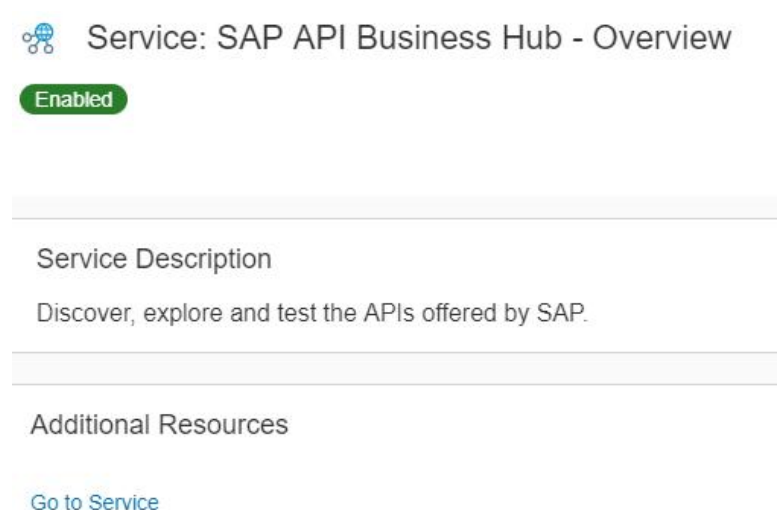
3. In the list of services, you can see on the top right a **Search Field**



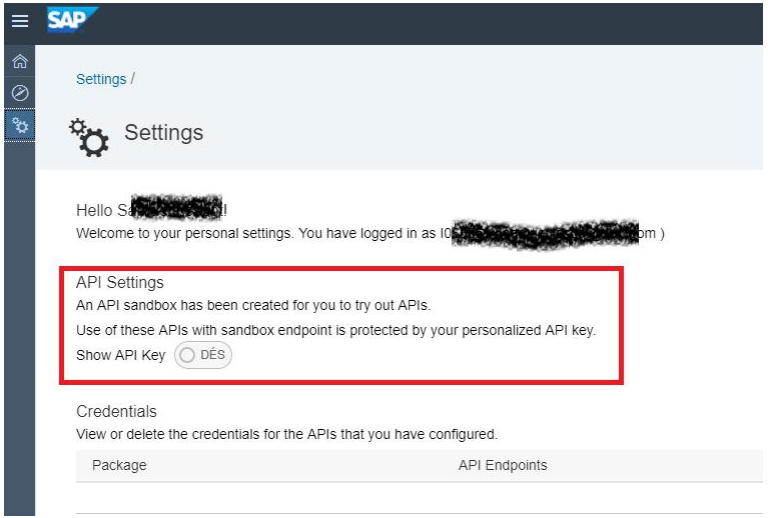
4. In the **Search Field**, enter **API**.
You can see the **SAP API Business Hub** returned in the list of services.



5. Select the SAP API Business Hub service and click on **Go to service**.
You can also access directly to the API Business Hub with this link:
<https://api.sap.com/>



Consuming APIs with the SAP API Business Hub

<p>6. Click on Classic Design to return to the classic design</p>	
<p>7. If you are not registered yet, click on Register. If you are already registered, click on Login</p>	
<p>8. Click on APIs</p>	
<p>9. On the left pane, click on Settings</p>	
<p>10. You can now click on the button to activate your API key</p>	

Consuming APIs with the SAP API Business Hub

11. Copy your API Key

API Settings

An API sandbox has been created for you to try out APIs.

Use of these APIs with sandbox endpoint is protected by your personalized API key.

Show API Key

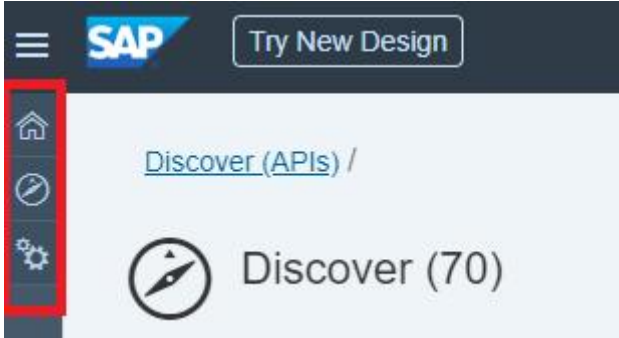
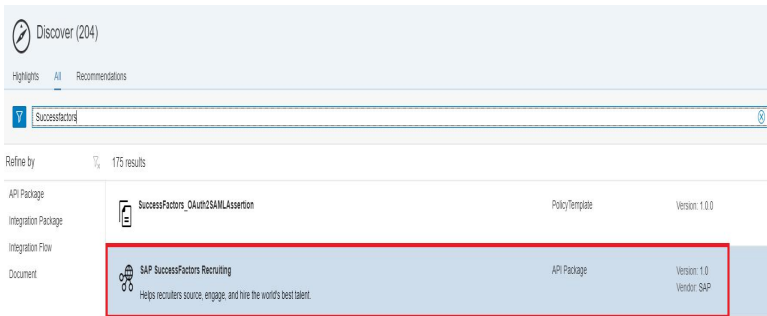


Use of these APIs with sandbox endpoint is protected by your personalized API key.

Copy API Key

Consuming APIs with the SAP API Business Hub

Scenario 1b: Exploration of SFSF Job Interview API

Explanation	Screenshot
1. From the Left Menu, click on the Home to go back to the homepage	
2. From the API Hub homepage, click on Browse and search for SuccessFactors	 <p>The top screenshot shows the 'Browse' button highlighted with a red box. The bottom screenshot shows the search results for 'SuccessFactors' with the search term highlighted in the search bar.</p>
3. Find the SAP SuccessFactors Recruiting Package	 <p>The screenshot shows the search results for 'SuccessFactors'. The 'SAP SuccessFactors Recruiting' package is highlighted with a red box.</p>

Consuming APIs with the SAP API Business Hub

- Find the **Job Application Interview API** in the **artifacts** tab

SAP SuccessFactors Recruiting

Helps recruiters source, engage, and hire the world's best talent.

Vendor: SAP
Version: 1.0
Published: 22 Jan 2016
Average User Rating: 4.0 out of 5 (1 Rating)

Overview **Artifacts (6)** Documents Tags Ratings

Filter Artifacts

Name	Type	Version	Actions
Candidate API to query and maintain candidate's profile and background information as defined in the candidate template.	ODATA API	1.0	
Job Application API to access and maintain job application of an applicant in Recruiting module.	ODATA API	1.0	
Job Application Interview API to access basic information about a job interview including notes, date and time and interviewer. It allows navigation to the related Job Application and Job Requestion	ODATA API	1.0	

- Explore the API and test using the GET request for **JobApplicationInterview({applicationinterviewid})**

Data:
Applicationinterviewid = 111
\$expand:
interviewoverallassessment

Job Application Interview

OVERVIEW **RESOURCE**

SAP API Business Hub allows you to select and configure an API Endpoint, so you can test APIs against the data available.

API Endpoint:

JobApplicationInterview

GET	/JobApplicationInterview
POST	/JobApplicationInterview
DELETE	/JobApplicationInterview({applicationinterviewid})
GET	/JobApplicationInterview({applicationinterviewid})
PUT	/JobApplicationInterview({applicationinterviewid})

Response Class (Status 200)

Model | Model Schema

```
{
  "d": {}
}
```

Response Content Type: application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
applicationinterviewid	<input type="text" value="111"/>	key: applicationinterviewid	path	long
\$select	<input type="text" value="applicationid, applicationinterviewid, candSlotMapId"/>	Select properties to be returned, see ODATA Select	query	Array[string]
\$expand	<input type="text" value="+ interviewOverallAssessment"/>	Expand related entities, see ODATA Expand	query	Array[string]

Response Messages

Consuming APIs with the SAP API Business Hub

6. Click on **Try it Out!**

GET /JobApplicationInterview(applicationInterviewId)

Get entity from JobApplicationInterview by key Code Snippet

Try it Out!

Response Class (Status 200)

Model Model Schema

```
{
  "d": {}
}
```

Response Content Type: application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type	Actions
applicationInterviewId	111	key: applicationInterviewId	path	long	
\$select	applicationId applicationInterviewId candSlotMapId	Select properties to be returned, see OData Select	query	Array(String)	
\$expand	,	Expand related entities, see OData Expand	query	Array(String)	

7. You should have a result like this.

Try it Out **View Response**

Request URL

`https://sandbox.api.sap.com/successfactors/odata/v2/JobApplicationInterview(111)?%24expand=InterviewOverallAssessment`

Response Body

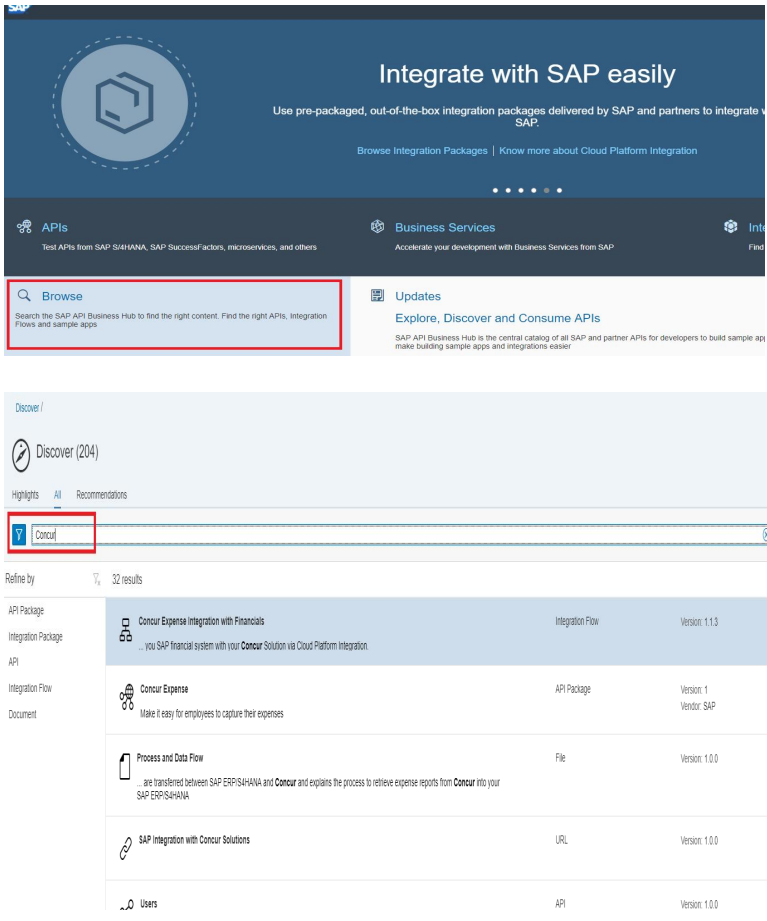
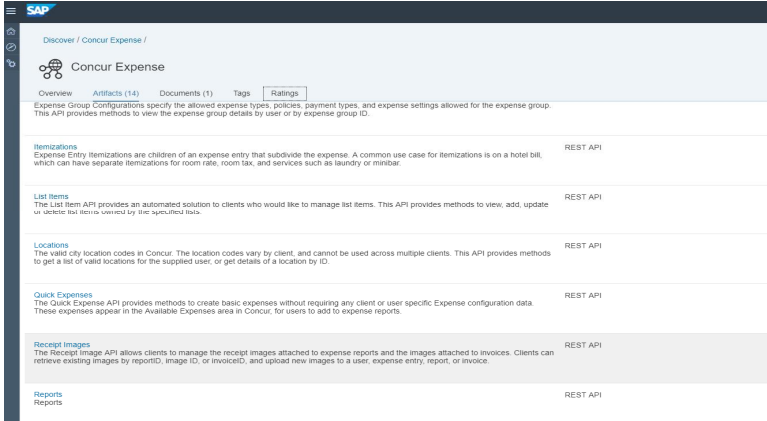
```
{
  "d": {
    "__metadata": {
      "uri": "https://sandbox.api.sap.com/successfactors/odata/v2/JobApplicationInterview(111L)",
      "type": "SFOData.JobApplicationInterview"
    },
    "applicationInterviewId": "111",
    "startDate": "/Date(1445875200000+0000)/",
    "templateType": "JOB_REQ",
    "candSlotMapId": "85",
    "source": "INTERNAL",
    "status": "RATED",
    "applicationId": "77",
    "endDate": "/Date(1445877000000+0000)/",
    "recruitEventStaffId": null,
    "notes": null,
    "isTimeSet": 1,
    "interviewNotes": {
      "__deferred": {
        "uri": "https://sandbox.api.sap.com/successfactors/odata/v2/JobApplicationInterview(111L)/interviewNotes"
      }
    }
  }
}
```

Response Code

200

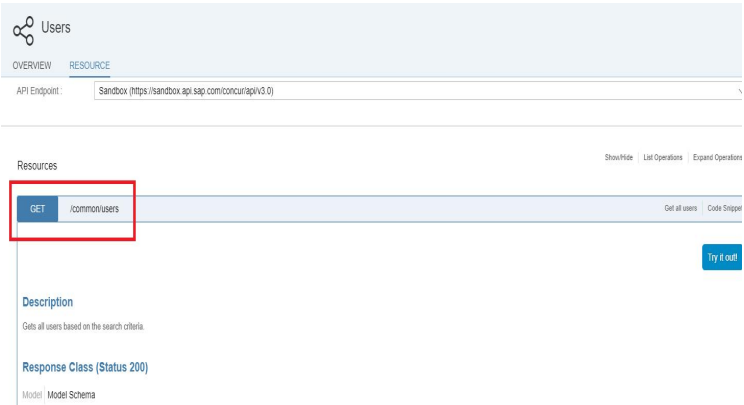
Consuming APIs with the SAP API Business Hub

Scenario 1c: Explore Concur User API

Explanation	Screenshot
1. From the API Hub homepage, click on Browse and search for Concur	 <p>The screenshot shows the SAP API Business Hub homepage. The 'Browse' button is highlighted with a red box. Below it, the search results for 'Concur' are shown, with the 'Concur Expense' integration flow highlighted.</p>
2. In the artifacts, select the Users API	 <p>The screenshot shows the SAP API Business Hub 'Concur Expense' page. The 'Users' API is highlighted in the list of artifacts.</p>

Consuming APIs with the SAP API Business Hub

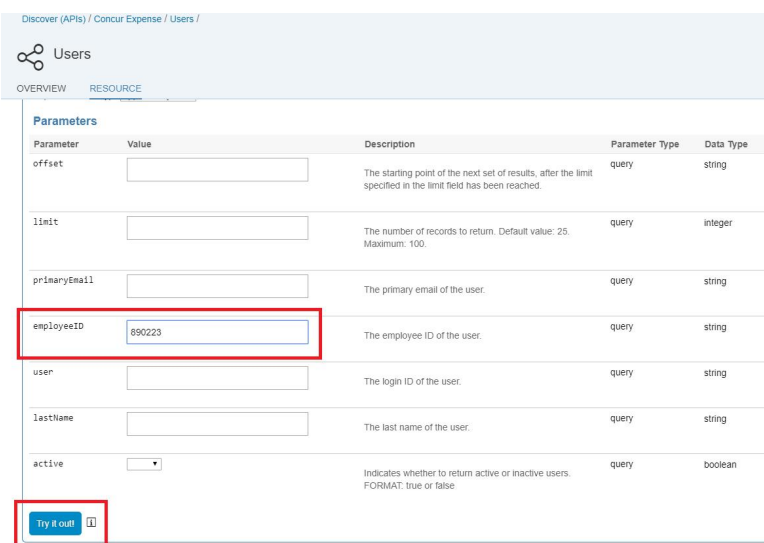
3. Test the API without any specific value, just a **GET**



The screenshot shows the 'Users' resource page in SAP API Business Hub. The 'Overview' tab is selected. The 'Resources' section shows a 'GET' button for the endpoint '/common/users', which is highlighted with a red box. The 'API Endpoint' is 'Sandbox (https://sandbox.api.sap.com/concur/api/v3.0)'. The 'Description' section states 'Gets all users based on the search criteria.' and the 'Response Class (Status 200)' is 'Model / Model Schema'. A 'Try it out!' button is visible in the bottom right corner.

4. Specify now an **Employeeid**.
Employeeid : 890223

Click on **Try it Out!**



The screenshot shows the 'Parameters' section of the 'Users' resource page. The 'employeeID' field is highlighted with a red box and contains the value '890223'. The 'Try it out!' button is also highlighted with a red box. The table below shows the parameters and their descriptions:

Parameter	Value	Description	Parameter Type	Data Type
offset		The starting point of the next set of results, after the limit specified in the limit field has been reached.	query	string
limit		The number of records to return. Default value: 25. Maximum: 100.	query	integer
primaryEmail		The primary email of the user.	query	string
employeeID	890223	The employee ID of the user.	query	string
user		The login ID of the user.	query	string
lastName		The last name of the user.	query	string
active		Indicates whether to return active or inactive users. FORMAT: true or false.	query	boolean

5. You should have a result like this.
Copy the **LoginID**, you will need this later.



The screenshot shows the 'Request URL' and 'Response Body' for the 'Users' resource. The 'Request URL' is 'https://sandbox.api.sap.com/concur/api/v3.0/common/users?employeeID=890223'. The 'Response Body' is a JSON object containing user information:

```
{
  "Items": [
    {
      "LoginID": "Mohan.Kumari@Bestrun.com",
      "EmployeeID": "890223",
      "FirstName": "Mohan",
      "LastName": "Kumar",
      "MiddleName": "",
      "PrimaryEmail": "Mohan.Kumar@Bestrun.com",
      "Active": true,
      "CellPhoneNumber": null,
      "OrganizationUnit": null,
      "ID": null,
      "URI": null
    }
  ],
  "NextPage": null
}
```

Scenario 1d: Explore the Concur Report API

- Go back to the **Concur Artifacts** and select the **Report**

Concur Expense		
Overview	Artifacts (14)	Documents (1) Tags Ratings
Expense Group Configurations		
Expense Group Configurations specify the allowed expense types, policies, payment types, and expense settings allowed for the expense group. This API provides methods to view the expense group details by user or by expense group ID.		
Itemizations		REST API 3.0
Expense Entry Itemizations are children of an expense entry that subdivide the expense. A common use case for itemizations is on a hotel bill, which can have separate itemizations for room rate, room tax, and services such as laundry or minibar.		
List Items		REST API 3.0
The List Item API provides an automated solution to clients who would like to manage list items. This API provides methods to view, add, update or delete list items owned by the specified lists.		
Locations		REST API 3.0
The valid city location codes in Concur. The location codes vary by client, and cannot be used across multiple clients. This API provides methods to get a list of valid locations for the supplied user, or get details of a location by ID.		
Quick Expenses		REST API 3.0
The Quick Expense API provides methods to create basic expenses without requiring any client or user specific Expense configuration data. These expenses appear in the Available Expenses area in Concur, for users to add to expense reports.		
Receipt Images		REST API 3.0
The Receipt Image API allows clients to manage the receipt images attached to expense reports and the images attached to invoices. Clients can retrieve existing images by reportID, image ID, or invoiceID, and upload new images to a user, expense entry, report, or invoice.		
Reports		REST API 3.0
Reports		

- Select the **GET** to list the report

Discover (APIs) / Concur Expense / Reports /

Reports

OVERVIEW RESOURCE

SAP API Business Hub allows you to select and configure an API Endpoint, so you can test APIs against the data available in y

API Endpoint :

Resources

GET	/expense/reports
POST	/expense/reports
GET	/expense/reports/{id}
PUT	/expense/reports/{id}

- Specify the user of your previous scenario. The user is the Login ID of your previous scenario
Click on **Try it Out!**

Discover / Concur Expense / Reports /

Reports

OVERVIEW RESOURCE

limit Number of records to return (default 25) query integer

user Optional. The login ID of the report owner(s) to use when searching for reports. If the value is set to LoginID, reports for the report owner with this login ID value are returned. If the value is set to ALL, reports for all report owners are returned. If this parameter is not specified, reports for the OAuth Consumer are returned. The access token owner (OAuth Consumer) must have the Web Services Admin role to use this parameter. query string

- You can now see the list of reports for this **LoginID**

Try it out! Hide Response ⓘ

Request URL

https://sandbox.api.sap.com/concur/api/v3.0/expense/reports?user=Mohan.kumar1@Bestrun.com

Response Body

```
{
  "Items": [1],
  "NextPage": null
}
```

Response Code

200