

Travel: Receipts API v4.0

Travel Service Guide

Last Revised: October 13, 2020

Applies to Concur Travel:

- Professional/Premium edition
 - TMC Partners
 - Direct Customers

- Standard edition
 - TMC Partners
 - Direct Customers

Contents

- Receipts API v4.0 API1**
- Section 1: IMPORTANT – About this Guide1**
- Section 2: Purpose of Scope Document1**
- Section 3: Receipts API v4.0 - Purpose and Description1**
- Section 4: Target Audience.....2**
- Section 5: Professional and Standard Editions.....2**
- Section 6: Prerequisites.....2**
- Section 7: Objectives2**
- Section 8: Development Environment.....3**
- Section 9: Travel Access Control Service5**
- Section 10: Company Level Authentication5**
- Section 11: Obtain Company Level Refresh Token6**
 - Descriptions.....7
- Section 12: Obtain Company Level Access Token8**
 - Descriptions.....9
- Section 13: Application Scopes10**
- Section 14: Geolocations.....11**
- Section 15: User Profile Web Service12**
- Section 16: SAP Concur Unique User ID13**
- Section 17: Receipts API v4.0 – General type14**
- Section 18: Create a Receipt - General.....14**
- Section 19: Embed UUID as a Remark.....17**
- Section 20: Receipt Retrieval17**
- Section 21: FAQ23**
- Section 22: Error Handling24**
 - Development Support 25
- Section 23: Support SAP Concur correlationID.....27**

Revision History

Date	Revision Notes/Comments
January 14, 2022	Updated the copyright year; no other changes; cover date not updated
April 14, 2021	Updated the copyright year; no other changes; cover date not updated
October 13, 2020	Initial publication

Receipts API v4.0 API

Section 1: **IMPORTANT – About this Guide**

Be aware of the following:

- **Infinite variables:** Certain scenarios have an infinite number of variables or extremely unusual circumstances, like rare cancellation or refund situations. So, not all scenarios can be presented in this guide. Also, certain processes may be influenced by third- or fourth-party providers. In some cases, you must contact the provider directly.
- **User interface, fees, rates, schedules:** When other providers change their user interface (for example, web site) or their fees/rates/schedules, they are under no obligation to make SAP Concur aware of those changes. If a screen sample in this guide is outdated because of a change made by a provider, we will update that screen sample when we become aware of the change and at our earliest convenience.
- **Permissions:** A company's admin may or may not have the correct permissions to manage the feature described in this guide. If an admin needs to manage this feature and does not have the proper permissions, they should contact the company's SAP Concur administrator.

Also, the admin should be aware that some of the tasks described in this guide cannot be completed by the company. In this case, the client must contact their TMC (if a TMC provides their support) or SAP Concur (if SAP API

Section 2: **Purpose of Scope Document**

The purpose of this document is to outline the details of our Receipts API v4.0. The document provides samples and descriptions for partner application development teams. This is an ongoing document that will be updated regularly during partner adoption of the Receipts API v4.0 and Authentication APIs.

Section 3: **Receipts API v4.0 - Purpose and Description**

This endpoint provides partner merchants the ability to post a travel invoice on behalf of a traveler using the eReceipts API v4.0. The Receipts API v4.0 API allows RESTful developed applications to use the agency's merchandising system to post fees, service charges, change fees, etc. as an electronic receipt (also referred to as a travel invoice) directly to the user's account in Concur Expense. This endpoint requires OAuth v2.0 and uses UUID as the matching fact for the user.

Section 4: **Target Audience**

Primary audience	This document is intended for SAP Concur TMC Partners who plan to develop a SAP Concur Platform Certified application to support the Receipts API v4.0.
Secondary audience	This document is also intended for SAP Concur customers and third party developers who wish to develop an internal-only application or vendor application to support SAP Concur Platform's Receipts API v4.0 Internally developed applications do not require Application Certification, however, most the specifications in this document should be considered as "best practices" for SAP Concur application development and support. Third party developed applications, branded for usage by Agencies, must be Certified.

Section 5: **Professional and Standard Editions**

This guide discusses both Professional and Standard Concur Travel & Expense. In those cases where a feature applies to one but not the other, that difference is noted.

Section 6: **Prerequisites**

Anyone reading this document should have some background with core SAP Concur applications and services such as Concur Travel and administration. The following list of technical knowledge and skills will aid in the development and certification of your application.

- Business travel industry experience
- Concur Travel and Expense experience
- Concur Travel system administration, agency configuration Administration, Concur Expense administration, user administration
- Traveler profile / GDS Profiles / SAP Concur XML Profile Sync
- RESTful API / XML development, OAuth 2.0, data structures
- Browse <https://developer.concur.com>, especially our [<OAuth>](#), [<Receipts>](#) [<Profile>](#) and [<Get Certified>](#) pages.

Section 7: **Objectives**

After reviewing this document, readers should have a better understanding of the following:

- How to develop SAP Concur integrations (applications) through the SAP Concur Platform

- Support structures and elements of our Receipts API v4.0
- How to obtain a company level OAuth refresh token
- How to obtain a company level OAuth access token
- How to generate general receipt type data (JSON) at point of sale
- How to capture or manage the traveler's UUID during transaction process
- POST the eReceipt using UUID as matching fact
- How to view the receipt data and rendered image in Concur Expense
- How to record the SAP Concur Correlation_ID
- Support Receipts API v4.0 error conditions
- How to prepare for application certification
- How to prepare for application enablement & deployment

Section 8: **Development Environment**

The following is a general list of items provided by SAP Concur that are necessary for application development and testing:

- A Professional Edition instance of Concur Travel or Concur Travel & Expense.
- A second instance of Concur Travel hosted in a data center hosted in a second data center.
- Default travel configuration.
- Default agency configuration with SAP Concur-Sabre PCC.
- Sandboxes may not have Concur Travel booking capabilities. It is available upon request.
- To POST a Receipts API v4.0 and Form of Payment within a company, that company's instance must be "activated".
- To access corporate discounts the application must be registered in our Travel Supplier system.
- A default "development" app with a unique Client ID, Client Secret and applicable scopes. Log a case to request additional applications or scopes.
- Sandbox Admin credentials with Demo / Dev user credentials. Log a case to request additional permissions or users.
- Access to developer.concur.com for the latest and greatest information on the following:
 - [<Receipts API v4.0>](#) resource
 - [<Authentication>](#) resource
 - [<Travel Profile>](#) resource
- Access to RESTful development tool such as POSTMAN or SOAPUI to send JSON/XML requests and receive XML responses

NOTE: To retrieve a traveler's itinerary and profile with program loyalty information and form of payment information, your application must be registered in our travel supplier system. This is an SAP Concur managed procedure and must be requested.

NOTE: To retrieve a traveler's itinerary, profile, and form of payment within a specific agency and agency configuration, you must register your agency Company UUID with your Application ID. This is an SAP Concur managed procedure and will be required before proceeding with development.

Section 9: **Travel Access Control Service**

A new travel access control service has been built into the Travel Profile and Travel Itinerary API. This service, managed internally by SAP Concur, can be used when specified by settings defined in agency applications. Agency applications that are deployed to enterprises that are serviced by multiple agencies are required to integrate with this service.

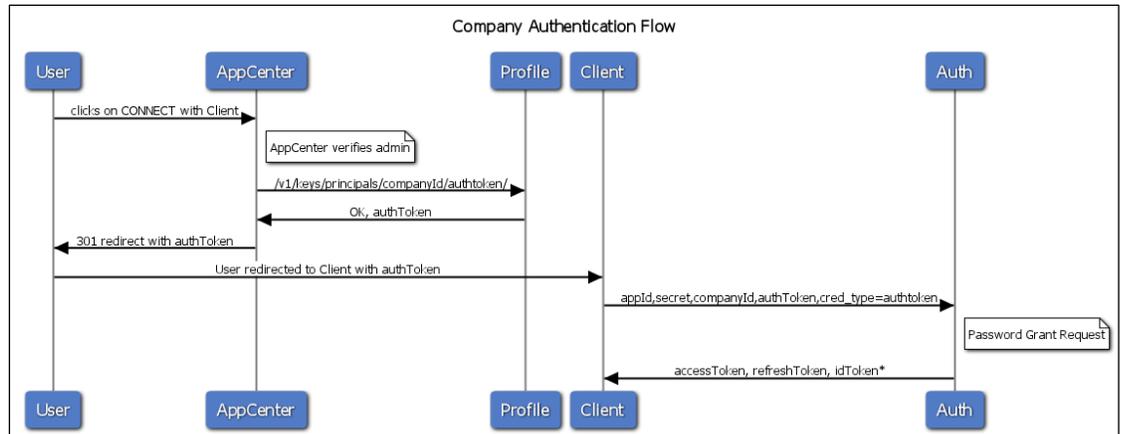
App Center Partners, Triplink Suppliers, and SAP Concur clients can use the Travel Profile and Itinerary APIs as is. Their applications will not be affected, and their endpoint destinations remain the same. Partner applications will be allowed to use their user level tokens to retrieve profiles for an individual or company-level tokens to retrieve profiles from an entire enterprise organization regardless of agency service affiliation or number of agency configurations.

For TMCs using the Travel Profile API v2.0, the access control service has been built into the existing Profile and Form of Payment endpoints. Data accepted and returned by the API is now controlled by a service that monitors all active Agency IDs and their associative active Travel Configuration IDs. Agency applications expecting profile and form of payment data from travelers under their own agencies can now be received or returned by Agency only. Profile and form of payment data associated with a different agency will NOT be received or returned. To use this security feature, an agency UUID **must** be specified in the properties (profile) of the agency's application ID (Client_ID).

During the booking process, the traveler's UUID, a unique identifier, must be passed to the agency's point-of-sale systems. When the booking or servicing process is complete, the transaction must contain the traveler's UUID before it can be successfully posted back to Concur Expense.

Section 10: **Company Level Authentication**

Your application will obtain and store *one* company-level OAuth refresh token for *each* company that opts to use your application. You will obtain that authenticated refresh token via an authorized 24-hour single-use request token from SAP Concur. Your 24-hour request token allows you to obtain a six month refresh token which is keyed off a unique application identifier, application secret, and the company's UUID. The refresh token will be used repeatedly to obtain 60-minute access (bearer) tokens that your application will use to conduct transaction API calls to and from SAP Concur. If you plan to deploy your application across specific agencies or travel configurations within a single company, SAP Concur can provide that capability via the Travel Access Control Service.



The following is a summary of steps necessary to obtain a company-level refresh token

- Obtain company name and company ID (also known as CliqID)
- Authorized SAP Concur representatives will generate 24-hour request tokens and issue the tokens using secured and encrypted communication methods
- Use the 24-hour request token to generate a refresh token. The request token can only be used up five times.
- Store and reuse the refresh_token – a UUID4 identifier that allows your application to obtain fresh access_tokens
- Use the 60 minute access_tokens to make transaction API calls
- Develop a process that updates refresh_tokens before their 6 month expiration period
- Develop a process that obtains access tokens before the 60 minute expiration period.
- Avoid generating new access tokens for repeated transactions within a 60 minute time period.

Section 11: Obtain Company Level Refresh Token

REQUEST

POST <https://us.api.Concursolutions.com/oauth2/v0/token>

HEADER

Content-Type:application/x-www-form-urlencoded

BODY

client_id:c14f2547#####e94f235c39
 client_secret:1bfb####-#####245ce

```
grant_type:password
username:3b#####ecd
password:fd44#####5d331
credtype:authtoken
```

RESPONSE:

```
{
  "expires_in": 3600,
  "scope": "openid TRVPRF PASSV COMPD EMERG TSAI TMCSF MEDIC UNUTX
USER COMPANY NOTIF FOP user_read user.read company.read",
  "token_type": "Bearer",
  "access_token": "BYxVfClZpo-zw...",
  "refresh_token": "a95caa0f-249a...",
  "refresh_expires_in": 1517039403,
  "id_token": "4n7i2-e9zP9qrpc4B...",
  "geolocation": "https://us.api.Concursolutions.com"
}
```

Descriptions

- The **expires_in** value is returned in seconds. Your **access_token** is valid for sixty minutes
- The **scopes** returned in the response are what your application has been registered to work with. Scopes are selectively enabled based on the functionality required. The values returned in the response should never change. If new scopes are added, your application will require re-certification.
- The **token_type** – SAP Concur returns the value of “Bearer”, an industry standard. The value of “Bearer” can be interpreted as “allow access to the bearer of this token”
- The **access_token**, a JWT, informs SAP Concur that the bearer of the returned token has been authorized to access the SAP Concur API and perform specific actions as specified by the scopes that have been granted. The **access_token** is valid for sixty minutes from the time of the response. If necessary, you may architect your application to scale up and use multiple access tokens to spawn multiple threads.
- The **refresh_token**, also a JWT, is the unique token that contains the information required to obtain a new **access_token** or **id_token**. Refresh tokens are good for a minimum of six months and are subject to strict storage requirements to ensure they are not compromised. Refresh tokens can also be revoked. Your application is expected to overwrite or replace your stored refresh tokens in case the response returns a different **refresh_token** value.
- The value in **refresh_expires_in** is returned in epoch time. Use a library that provides conversion capability. For a UI version, navigate to <https://www.epochconverter.com>

- The **id_token**, also a JWT, is returned. For company-level or enterprise-level applications, there is little need to retrieve details of the company-level authenticated user unless it is for auditing purposes. The information stored in the ID token JWT is necessary for user-level mobile-based applications such as Uber or Triplink supplier applications like Avis or Marriot.
- The value for **geolocation** should be stored as your application's base URI. Since SAP Concur has multiple data centers, it may be required to obtain and store data from customers who are hosted in EMEA as well as the US. If your application receives **error code 16, "invalid request", user lives elsewhere**, your application must be able to submit a second request to "us.api.Concursolutions.com" or "emea.api.concursolutions.com" and store that geolocation. The geolocation also identifies for SAP Concur geographically where the user is stored.
- Store the refresh token, access tokens, expiration date and instance URL (geolocation) along with your internal information about the company's profile.

! Do not append application keys and values to the request URL.

! Requests containing confidential keys and values will be rejected.

! Ensure your web servers are not configured with TLS V1.1.
https://assets.concur.com/concurtraining/cte/en-us/FAQ_TLS_1.1_End_of_Support.pdf

NOTE: Remove `charset=utf-8` from content-type header in your request. Our implementation of OAuth2, according to the IETF standards <https://tools.ietf.org/html/draft-ietf-oauth-v2-31#appendix-B>, requires all clients to remove the `charset=utf-8` from the Content-Type header.

Section 12: Obtain Company Level Access Token

REQUEST

POST <https://us.api.Concursolutions.com/oauth2/v0/token>

HEADER

Content-Type:application/x-www-form-urlencoded

BODY

```
client_id:c14f2547#####e94f235c39
client_secret:1bfb####-#####245ce
grant_type:refresh_token
refresh_token: {refresh_token}
```

RESPONSE:

```
{
  "expires_in": 3600,
  "scope": "openid TRVPRF PASSV COMPD EMERG TSAI TMCSP MEDIC UNUTX
USER COMPANY NOTIF FOP user_read user.read company.read",
  "token_type": "Bearer",
  "access_token": "BYxVfC1Zpo-zw...",
  "refresh_token": "a95caa0f-249a...",
  "refresh_expires_in": 1517039403,
  "id_token": "4n7i2-e9zP9qrpc4B...",
  "geolocation": "https://us.api.Concursolutions.com"
}
```

Descriptions

- The **expires_in** value is returned in seconds. Your **access_token** is valid for sixty minutes.
- The **scopes** returned in the response are what your application has been registered to work with. Scopes are selectively enabled based on the functionality required. The values returned in the response should never change. If new scopes are added, your application will require re-certification.
- The **token_type** – SAP Concur returns the value of “Bearer”, an industry standard. The value of “Bearer” can be interpreted as “allow access to the bearer of this token”.
- The **access_token**, a JWT, informs SAP Concur that the bearer of the returned token has been authorized to access the SAP Concur API and perform specific actions as specified by the scopes that have been granted. The **access_token** is valid for sixty minutes from the time of the response. If necessary, you may architect your application to scale up and use multiple access tokens to spawn multiple threads.
- The **refresh_token**, also a JWT, is the unique token that contains the information required to obtain a new **access_token** or **id_token**. Refresh tokens are good for a minimum of six months and are subject to strict storage requirements to ensure they are not compromised. Refresh tokens can also be revoked. Your application is expected to overwrite or replace your stored refresh tokens in case the response returns a different **refresh_token** value.
- The value in **refresh_expires_in** is returned in epoch time. Use a library that provides conversion capability. For a UI version, navigate to <https://www.epochconverter.com>
- The **id_token**, also a JWT, is returned. For company-level or enterprise-level applications like Receipts API v4.0, there is no need to retrieve details of the company-level authenticated user. The information stored in the ID token JWT is necessary for User-level authentication-based applications such as Uber or Triplink supplier applications like Avis or Marriot.

- The value for **geolocation** should be stored as your application's base URI. Since SAP Concur has multiple data centers, it may be required to obtain and store data from customers who are hosted in EMEA as well as the US. If your application receives **error code 16, "invalid request", user lives elsewhere**, your application must be able to submit a second request to "us.api.Concursolutions.com" or "eema.api.Concursolutions.com" and store that geolocation. The geolocation also identifies for SAP Concur geographically where the user is stored.
- Store the refresh token, access tokens, expiration date and instance URL (geolocation) along with your internal information about the company's profile.

! Do not append application keys and values to the request URL.

! Requests containing confidential keys and values will be rejected.

! Ensure your web servers are not configured with TLS V1.1.
https://assets.concur.com/concurtraining/cte/en-us/FAQ_TLS_1.1_End_of_Support.pdf

NOTE: Remove `charset=utf-8` from content-type header in your request. Our implementation of OAuth2, according to the IETF standards <https://tools.ietf.org/html/draft-ietf-oauth-v2-31#appendix-B>, requires all clients to remove the `charset=utf-8` from the Content-Type header.

Section 13: **Application Scopes**

Development receipts API v4.0 applications by default will have the following scopes registered:

- openid
- user.read
- company.read
- FOP
- GHOST
- TRVPRF
- PASSV
- EMERG
- TSAI
- TMCSP
- MEDIC
- UNUTX

- NOTIF
- COMPD
- ITINER
- Receipts.read
- Receipts.write

For security purposes, **production** receipts API v4.0 applications will have limited scopes registered. The designated scopes are:

- openid
- user.read
- company.read
- receipts.writeonly

Explanations for scopes are documented here:

<https://developer.concur.com/api-reference/authentication/scopes.html>

If you wish to have your production application(s) include additional scopes or endpoints, you must submit a request to the technical enablement team submit you must have your application re-certified.

Section 14: Geolocations

SAP Concur has multiple datacenters:

<https://developer.concur.com/platform/base-uris.html>

- United States: <https://us.api.concursolutions.com>
- EMEA: <https://emea.api.concursolutions.com>
- Public Sector: <https://usg.api.concursolutions.com>
- China: <https://www-cn.api.concurcdc.cn>

Host Location	Geolocation / URL from refresh token based on host location of Company GUID	The URL to use for Receipts API v4.0
EMEA	https://emea.api.Concursolutions.com/oauth2/v0/token	https://emea.api.Concursolutions.com/receipts/v4/users/
US	https://us.api.Concursolutions.com/oauth2/v0/token	https://us.api.Concursolutions.com/receipts/v4/users/
PSCC	https://usg.api.Concursolutions.com/oauth2/v0/token	https://usg.api.Concursolutions.com/receipts/v4/users/

Section 15: User Profile Web Service

Get User Information

<https://developer.concur.com/api-explorer/v3-0/Users.html>

UUID is required for the Receipts v4.0 API. UUID is available via the Travel Profile v2.0 API. It may be necessary to use the User API to retrieve the user's UUID if you are storing login ID and email ID but not yet transitioned to the Travel Profile API.

Use the following request parameters to filter your searches:

Retrieve UUID for all active users

GET

```
https://us.api.concursolutions.com/users?offset=100&limit=1000&isactive=true
Authorization: Bearer {access_token}
```

Retrieve UUID by Login ID

GET

```
https://us.api.concursolutions.com/users/?loginid=logind_id@domain
Authorization: Bearer {access_token}
```

Retrieve UUID by primary Email ID

GET

```
https://us.api.concursolutions.com/users/?primaryemail=email@domain.com
Authorization: Bearer {access_token}
```

RESPONSE:

```
{
  "total": "1",
  "offset": 0,
  "limit": 100,
  "companyinfo": {
    "name": "Company Name, Inc",
    "address": "601 108th Ave NE \ Suite 1000",
    "city": "Bellevue",
    "state": "WA",
    "zip": "98004",
    "country": "US"
  },
  "Items": [
    {
      "Active": true,
      "CountryCode": "US",
      "CellPhoneNumber": "425-590-5000",
      "PrimaryEmail": "Email1@domain.com",
      "EmployeeID": 106,
      "ID": "### UUID ###",
      "Emails": {
        "PrimaryEmail": "Email1@domain.com",
```

```

        "VerifiedEmail": null,
        "email2": "Email2@domain.com",
        "email3": "",
        "email4": "",
        "email5": ""
    },
    "OrganizationUnit": null,
    "MiddleName": "Middle",
    "LastName": "Last
    "FirstName": "first",
    "LoginID": "login_id@domain.com"
}
]
}

```

Section 16: **SAP Concur Unique User ID**

NOTE: RECOMMENDATION - ADOPT UUID as traveler’s unique identifier.

The UUID format is being used throughout the SAP Concur platform and new OAuth methodology. Your application’s client (company) identifier, client secret, and authentication tokens are all in UUID format. Even error IDs are returned in UUID format.

UUID v4 information can be found here

https://en.wikipedia.org/wiki/Universally_unique_identifier .

This tool is also useful - <https://www.uuidgenerator.net/>

UUID is now available with every user record stored at SAP Concur. UUIDs are automatically generated whenever an employee or traveler record is created - even during 350 record type imports. And as the standard indicates, the UUID value (which is randomly generated) is guaranteed unique at SAP Concur. Because of this uniqueness, UUID can be used as a matching-fact and the unique identifier at SAP Concur. You should no longer use the XML Sync ID or Employee ID which are often not unique and is responsible for numerous user record errors and support queries. Login ID, still unique at SAP Concur, is widely adopted by SAP Concur Platform partner applications but has not be widely adopted with XML Profile Sync.

We ask Agencies to migrate off the old Sync ID associated with the deprecated XML Profile Sync. We highly recommend transitioning to support SAP Concur UUID.

In 2017, UUID was exposed in the Receipts API v4.0 API and it was also made available in the PNR editing tool. The label and data in the Receipts API v4.0 response will look like this:

```
<UUID>e58#####-###0a</UUID>
```

SAP Concur UUID is required for the [eReceipts v4 API](#). It is the only element supported as its matching-fact. eReceipts v4.0 does not support Login ID, Email ID, or any other attribute.

Section 17: **Receipts API v4.0 – General type**

Version 4.0 of the Receipts API offers features like more receipt types, automatic e-receipt generation in end user's preferred language and ability for partners to provide detailed tax information. Unlike version 3.0, we are discontinuing the use of matching facts; instead the partner will have to create a receipt for a specific end user. Receipts 4.0 works only with the new [Authentication API](#) and uses JSON as its data interchange format.

The Receipts API supports several receipt types – documented here:

<https://developer.concur.com/api-reference/receipts/supported-receipt-types.html>

The available schemas for specific receipt types are reserved for Concur Travel partners, also known as Triplink Suppliers, for their specific types. Triplink Supplier partners and App Center Partners must have SAP Concur Triplink *and* App Center Partnership agreements before their applications can be Certified. App Center partnership agreements are also required for ground transportation and ride share receipt providers.

The General receipt type is available for Agency POS transactions as they do not fall under any of the Supplier specific receipt types. The definition is documented here:

<https://developer.concur.com/api-reference/receipts/supported-receipt-types.html#general-receipt>

The Receipts v4.0 API accepts three different formats for posting a receipt:

1. **Receipt Data** - Your receipt data is stored along with a system generated receipt image file.
2. **Receipt Data & Receipt Image** - Your receipt data and receipt image file are stored.
3. **Receipt Image w/o Data** - Your receipt image file is stored along with some accompanying metadata.

All of the above are valid receipt resources, but the service draws a distinction between resources with data versus resources that are standalone images.

Resources of standalone images are referred to as Image-Only Receipts.

Resources with data are schema-enforced and are referred to as e-receipts. This is the recommended method.

Section 18: **Create a Receipt - General**

Create a basic general receipt type

REQUEST

POST <https://us.api.concursolutions.com/receipts/v4/users/{uuid}>

Content-type:application/json
Authorization: Bearer {access_token}

BODY:

```
"reference": "Invoice #100539586",
"dateTime": "2020-07-20T00:00:00-0800",
"total": "20.00",
"subtotal": "18.00",
"taxesTotal": "2.00",
"currencyCode": "USD",
"seller": {
  "name": "Travel Agency, Inc",
  "location": {
    "name": "Booking Fee",
    "number": "100539586",
    "latitude": 37.80062,
    "longitude": -122.40027,
    "internetAddress": "https://www.concursolutions.com/",
    "emailAddress": "https://www.concursolutions.com/",
    "telephoneNumber": "(888)555-1212",
    "faxNumber": "",
    "address": {
      "streetAddress": "601 108th Ave NE #1000",
      "addressLocality": "Bellevue ",
      "addressRegion": "WA",
      "addressCountry": "US",
      "postalCode": "98004"
    }
  }
},
"taxes": [
  {
    "authority": {
      "addressCountry": "US",
      "addressRegion": "WA"
    },
    "name": "Local Sales Tax",
    "rate": 10,
    "amount": "2.00"
  }
],
"payments": [
  {
    "amount": "20.00",
    "cardDetail": {
      "cardType": "American Express",
      "creditCardId": "1009",
      "authorizationCode": "AB987654321"
    }
  }
],
"lineItems": [
  {
    "sequenceNumber": 1,
```

```

"reference": "",
"description": "Invoice# 100539586 ",
"quantity": 1,
"unitCost": "19.00",
"subtotal": "19.00",
"taxesTotal": "2.00",
"total": "20.00",
"taxes": [
  {
    "authority": {
      "addressCountry": "US",
      "addressRegion": "WA"
    },
    "name": "Discount",
    "rate": 5,
    "amount": "-1.00"
  }
]
}

```

RESPONSE:

201 Created

The rendered receipt in Concur Expense as an available expense will look like this :

Travel Agency, Inc 601 108th Ave NE #1000 Bellevue WA US 98004 https://www.concursolutions.com/ (888)555-1212		\$20.00 American Express - 1009 10/03/2020 12:00 AM Receipt: Invoice #100539586	
Description	Quantity	Unit Price	Amount
Invoice# 100539586	1	\$19.00	\$19.00
Discount			\$-1.00
			Subtotal: \$18.00
			Tax: \$2.00
			Total: \$20.00

NOTE: A "Seller Name" must be defined and submitted to your partner enablement representative as your identifier. A minimum 100x100 pixel image file of your organization's logo should be submitted as well. The logo image will automatically be rendered onto your eReceipt upon viewing.

NOTE: Store the **concur-correlationid** value returned in the header response for logging, troubleshooting, or case escalation purposes

NOTE: Make note of the geolocation where the company user exists to POST the eReceipt correctly. If the user's geolocation is unknown or incorrect then default the request to <https://us.api.concursolutions.com>. The error message will return the correct geolocation for the user.

```
Http: 400
{
  "error": "invalid_request",
  "error_description": "user lives elsewhere",
  "concur-correlationid": "557de40c-79a4-41d4-a1d2-73fd56341842",
  "geolocation": "https://emea.api.concursolutions.com/",
  "code": 16
}
```

Section 19: **Embed UUID as a Remark**

It may be necessary to capture the UUID of the traveler during the booking process, especially if LoginID or other unique identifiers are being used.

One method currently available is to include UUID as an element in General Remarks - Ticketing. Below is an example from the Template Editor:

GeneralRemark(Ticketing)
ElementPrefix: B*CTUUID-
User/UUID

Upon finishing, parse the UUID and use that value to post the receipt upon completion of the sales transaction or purchasing process.

Refer to the PNR Finishing / Template Editor documentation here:

https://www.concurtraining.com/customers/tech_pubs/TravelDocs/TSGs/PNRFinishingEditor.pdf

NOTE: Only authorized and certified Agency or Concur Travel configuration specialists should consider making the suggested changes.

Section 20: **Receipt Retrieval**

The Receipts V4 API provides GET access to single user's receipts that have been submitted through the API via UUID. The response will return a comprehensive list of receipts associated to the user.

Images for receipts directly associated to expense reports, entries or requests should be obtained via the [Image v1 API](#). Only the rendered receipt image file will be returned. Examples of applications that should use the Image v1 API versus the Receipts v4 API include ERP integrations for financial journal entry postings, VAT reclaim integrations that obtain expense transactions to calculate VAT reclaim, project billing integrations used to substantiate expenses billed back, or business intelligence tools to track actual expenses vs budgets.

Retrieve Receipts for the User by UUID

REQUEST

```
GET
https://us.api.concursolutions.com/receipts/v4/users/{uuid}
Content-type:application/json
Authorization: Bearer {access_token}
```

RESPONSE:

200

```
{
  "receipts": [
    {
      "stateModifiedAt": "2020-10-05T15:13:14.426Z",
      "modifiedAt": "2020-10-05T15:13:14.426Z",
      "companyId": "#####",
      "dateTimeReceived": "2020-10-05T15:13:14.426Z",
      "entityId": "#####",
      "validationSchema":
"http://schema.concursolutions.com/general-receipt.schema.json",
      "receipt": {
        "merchant": {
          "name": "Travel Agency, Inc",
          "location": {
            "number": "100539586",
            "emailAddress":
"http://www.concursolutions.com/",
            "telephoneNumber": "(888)555-1212",
            "address": {
              "addressCountry": "US",
              "addressLocality": "Bellevue ",
              "addressRegion": "WA",
              "streetAddress": "601 108th Ave NE #1000",
              "postalCode": "98004"
            },
            "latitude": 37.80062,
            "name": "Booking Fee",
            "faxNumber": null,
            "internetAddress":
"http://www.concursolutions.com/",
            "longitude": -122.40027
          },
          "reference": "Invoice #100539586",

```

```

    "dateTime": "2020-10-03T00:00:00-08:00",
    "seller": {
      "name": "Travel Agency, Inc",
      "location": {
        "number": "100539586",
        "emailAddress":
"https://www.concursolutions.com/",
        "telephoneNumber": "(888)555-1212",
        "address": {
          "addressCountry": "US",
          "addressLocality": "Bellevue ",
          "addressRegion": "WA",
          "streetAddress": "601 108th Ave NE #1000",
          "postalCode": "98004"
        },
        "latitude": 37.80062,
        "name": "Booking Fee",
        "faxNumber": null,
        "internetAddress":
"https://www.concursolutions.com/",
        "longitude": -122.40027
      }
    },
    "lineItems": [
      {
        "reference": null,
        "sequenceNumber": 1,
        "total": "20.00",
        "quantity": 1,
        "taxesTotal": "2.00",
        "subtotal": "19.00",
        "unitCost": "19.00",
        "description": "Invoice# 100539586 ",
        "taxes": [
          {
            "name": "Discount",
            "amount": "-1.00",
            "rate": 5,
            "authority": {
              "addressCountry": "US",
              "addressRegion": "WA"
            }
          }
        ]
      }
    ],
    "app":
"https://us.api.concursolutions.com/profile/v1/apps/dafc536c-8cf7-42be-8d40-d8f9a5a79d96",
    "total": "20.00",
    "taxesTotal": "2.00",
    "subtotal": "18.00",
    "payments": [
      {
        "cardDetail": {

```

```

        "cardType": "American Express",
        "authorizationCode": "AB987654321",
        "creditCardId": "1009"
    },
    "amount": "20.00"
  }
],
"taxes": [
  {
    "name": "Local Sales Tax",
    "amount": "2.00",
    "rate": 10,
    "authority": {
      "addressCountry": "US",
      "addressRegion": "WA"
    }
  }
],
"currencyCode": "USD",
"user":
"https://us.api.concursolutions.com/profile/v1/principals/#####
###"
},
"image":
"https://us.api.concursolutions.com/receipts/v4/#####/image
",
  "userId": "#####",
  "id": "#####",
  "digitizationStatus": "NO_PROCESSING_REQUIRED",
  "expense": {
    "paymentType": "CREDIT_CARD"
  },
  "self":

```

Retrieve Receipt by Receipt ID

Individual receipts can also be retrieved by receipt ID.

REQUEST

GET <https://us.api.concursolutions.com/receipts/v4/{receiptID}>

Content-type:application/xml

Authorization: Bearer {access_token}

RESPONSE

200

```

{
  "stateModifiedAt": "2020-10-05T15:29:15.082Z",
  "modifiedAt": "2020-10-05T15:29:15.082Z",
  "companyId": "ef#####fd",
  "dateTimeReceived": "2020-10-05T15:13:14.426Z",

```

```

"entityId": "p#####n",
"validationSchema": "http://schema.concursolutions.com/general-
receipt.schema.json",
"receipt": {
  "merchant": {
    "name": "Travel Agency, Inc",
    "location": {
      "number": "100539586",
      "emailAddress": "https://www.concursolutions.com/",
      "telephoneNumber": "(888)555-1212",
      "address": {
        "addressCountry": "US",
        "addressLocality": "Bellevue ",
        "addressRegion": "WA",
        "streetAddress": "601 108th Ave NE #1000",
        "postalCode": "98004"
      },
      "latitude": 37.80062,
      "name": "Booking Fee",
      "faxNumber": null,
      "internetAddress": "https://www.concursolutions.com/",
      "longitude": -122.40027
    }
  },
  "reference": "Invoice #100539586",
  "dateTime": "2020-10-03T00:00:00-08:00",
  "seller": {
    "name": "Travel Agency, Inc",
    "location": {
      "number": "100539586",
      "emailAddress": "https://www.concursolutions.com/",
      "telephoneNumber": "(888)555-1212",
      "address": {
        "addressCountry": "US",
        "addressLocality": "Bellevue ",
        "addressRegion": "WA",
        "streetAddress": "601 108th Ave NE #1000",
        "postalCode": "98004"
      },
      "latitude": 37.80062,
      "name": "Booking Fee",
      "faxNumber": null,
      "internetAddress": "https://www.concursolutions.com/",
      "longitude": -122.40027
    }
  },
  "lineItems": [
    {
      "reference": null,
      "sequenceNumber": 1,
      "total": "20.00",
      "quantity": 1,
      "taxesTotal": "2.00",
      "subtotal": "19.00",
      "unitCost": "19.00",

```

```

        "description": "Invoice# 100539586 ",
        "taxes": [
            {
                "name": "Discount",
                "amount": "-1.00",
                "rate": 5,
                "authority": {
                    "addressCountry": "US",
                    "addressRegion": "WA"
                }
            }
        ]
    },
    "app":
    "https://us.api.concursolutions.com/profile/v1/apps/#####",
    "total": "20.00",
    "taxesTotal": "2.00",
    "subtotal": "18.00",
    "payments": [
        {
            "cardDetail": {
                "cardType": "American Express",
                "authorizationCode": "AB987654321",
                "creditCardId": "1009"
            },
            "amount": "20.00"
        }
    ],
    "taxes": [
        {
            "name": "Local Sales Tax",
            "amount": "2.00",
            "rate": 10,
            "authority": {
                "addressCountry": "US",
                "addressRegion": "WA"
            }
        }
    ],
    "currencyCode": "USD",
    "user":
    "https://us.api.concursolutions.com/profile/v1/principals/#####",
    "image":
    "https://us.api.concursolutions.com/receipts/v4/d3c45f8443544e40a9289ee4ff09056b/image",
    "userId": "#####",
    "id": "#####",
    "digitizationStatus": "NO_PROCESSING_REQUIRED",
    "expense": {
        "paymentType": "CREDIT_CARD"
    },

```

```

    "self":
    "https://us.api.concursolutions.com/receipts/v4/d3c45f8443544e40a9289ee4ff0
    9056b",
    "template":
    "https://us.api.concursolutions.com/receipts/v4/{receiptId}"
  }

```

Retrieve Image-Only Receipt by Receipt ID

From the list of receipts returned in the response for the user, URIs for receipt images are available for each of the receipts returned. Using the company level OAuth token, you may retrieve the image-only receipt from our Imaging service in an imaging file format.

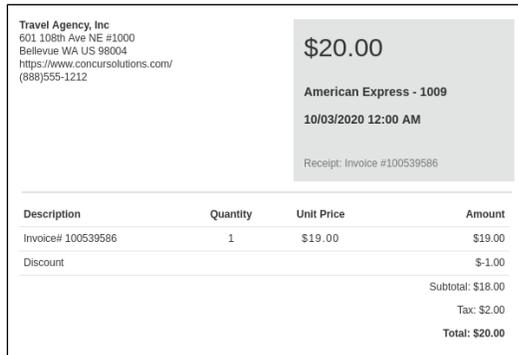
REQUEST

GET <https://us.api.concursolutions.com/receipts/v4/{receiptID}/image>

Content-type:application/xml
 Authorization: Bearer {access_token}

RESPONSE

200



Save the response.png file that is returned using the unique Receipt ID as the identifier.

Section 21: **FAQ**

Q. Can anyone create receipts for traveler's and expense reports?

A. No. Only authorized partners and certified partner applications are granted permissions and secured scopes by the Concur App Center, Platform Enablement team, and Core Services team.

Q. Can anyone connect to an agency receipts application?

A. No. Company connections to certified partner applications can only be made by Administrators.

Q. Can I disconnect my company from an agency receipts application?

A. Yes. Company Administrators can disconnect their organization to any connected application upon request. Also, the partner receipt application can be excluded by any organization upon request. Applications in the Concur App Center can be disabled or hidden by the App Center Administrator or Concur upon request. The Concur App Center can also be disabled upon request.

Section 22: **Error Handling**

REQUIREMENT: Retain the GUID returned in all SAP Concur responses under the header "SAP Concur-correlationid". It is a value that will be referenced to locate transactions in our logging system

REQUIRED: Support the following HTTP error conditions / codes:
<https://developer.concur.com/tools-support/reference/http-codes.html>

REQUIRED: Provide retry logic for timeouts and 500/503 errors.

RECOMMENDATION: Provide your own unique "SAP Concur-correlationid" header and randomly generated value – inserted with a company_id for example. If you provide the following in your request header, SAP Concur will return the same value in the response header

SAP Concur-correlationid: 2997-e17fb88b-5b9a-41b9-b285-6da70eeba98a

RECOMMENDATION: Error logging - retain logs for a minimum of 7 days for troubleshooting and support.

Development Support

- For travel configuration questions/support, contact Travel Solutions or your Alliance Manager.
- Review Release Notes regularly – API updates are delivered with monthly releases:
 - ◆ **Error! Hyperlink reference not valid.**
 - ◆ http://www.SAPConcurtraining.com/customers/tech_pubs/TravelDocs/ReleaseNotes/CCC_RN.htm
- Review content on developer.concur.com regularly – updates are frequent.
- If you need access or permissions to Preview content on developer.concur.com, please contact your Travel Solutions or Travel Support representative.
- For reproducible issues, log a case via case management system to the SAP Support team. Support will escalate to respective Platform Partner Services and R&D teams.
 - Visit <https://developer.oncur.com/tools-support/requesting-partner-support.html>
 - Partner Support Case Management System - https://sapSAPConcur.my.salesforce.com/secur/login_portal.jsp?orgId=00D600000007Dq3&portalId=06060000000PrEi
- Support engineers attempt to reproduce your application's issues using a RestUI based request/response application such as POSTMAN or SoapUI. Our application Analysts and Engineers are not developers so they do not require .C, .JS or any other source language files.
- Please have the following data and identifiers available:
 - ◆ Application Identifiers
 - Application Name (Your agency may have more than one certified application)
 - Brief application description
 - Last 6 characters of your application's client_ID.
 - DO NOT submit / paste full application IDs (keys), secrets, or OAuth tokens
 - The UUID formatted SAP Concur Correlation ID from your response headers
 - ◆ Company Identifiers:
 - Company Name or Customer Name (the name that SAP Concur recognizes)
 - The SAP Concur Company ID (also known as CliqID)
 - The Travel Config ID

- The expense entity code or entity ID
- The company GUID if available
- ◆ Identifiers of affected areas:
 - Affected user(s) if applicable
 - The employee name or employee ID
 - The UUID if possible
 - LoginID if necessary
 - Expense report name / ID
 - Invoice/Payment Request name/ID
 - Trip Itinerary name / Travel Request name/ ID
 - Booking Segment name/source
 - Trip ID
 - Receipt ID
 - Locator/PNR
 - Date range
- ◆ Reproduction Information:
 - Provide contextual information related to the issue so engineers have an understanding of what your application was attempting to do at the time
 - Function –retrieving data or creating/updating data
 - Parameter(s) – by creation date, last modified date, by ID, etc.
 - Affected product areas – Imaging, expense types, dates, amounts, city codes, workflow
- ◆ API-transaction related information:
 - Token (never supply this via email)
 - Request URL
 - Method (GET, POST, PUT)
 - Request Headers
 - Request Body
 - Response Headers
 - Response Body
 - Any additional log or response information relevant to the issue

! **NOTE: DO NOT SEND CLIENT OR PARTNER SENSITIVE DATA** such as OAuth refresh tokens, application_ids, application_secrets through Email. Use secured channels ONLY!

Section 23: Support SAP Concur correlationID

This is a screen shot of a typical "Body" response for retrieving a profile in POSTMAN.



Select "Headers" for the response headers. Note "SAP Concur-correlationid"

We require that you store the response GUID when an error occurs. When you log a case, please include the SAP Concur-correlationid GUID so we can retrieve the error(s) from our internal tools / logging system. This is what response headers look like in POSTMAN:



Here's what it would look like when we search for your application's errors using the SAP Concur-correlationID GUID. It is vital for troubleshooting efficiency.

