**US\_294133\_Cognizant: Establish RentTrack connection with the Build Credit Module**

**US\_294142\_Cognizant: Subscribe user to RentTrack**

**Understanding of the requirement**

First create a connection between RentTrack and Build Credit Module, secondly create tenant user in RentTrack , thirdly create $0 subscription for the created user.

**Assumptions**:

* Sandbox account and credentials are provided and we are able to access.
* Postman collections 2.1 with the related calls are given for development purpose.
* Any changes made in RentTrack API which is given in the API documentation, will be notified.
* DOB will be provided
* GitLab Apiary access will be provided

**High level Technical design and solution**

* 1. Impacted Modules

BCM modules

* 1. Impacted Workflows
     1. Overview:

Invoking sandbox RentTrack API with the given parameters (partner\_username, partner\_password, client\_id and a client\_secret), BCM will get token ID which will be used for further communication with RentTrack call for the given user

These tokens allow BCM to make requests on behalf of the users managed by partner application. Using the *access\_token*, BCM can access the source in the Tenant or Landlord APIs. To generate these tokens we will use the same call as in the above example, however, the username and password parameters will be for a user that BCM has created.

Once the user has been created, using the access\_token, BCM will invoke the PATCH call to update the DOB of the user.

Using the access\_token, BCM will be able to invoke the API to create a subscription for the given user details.

If user is already subscribed, BCM will receive the error code from RentTrack and BCM will send back the corresponding readable response message to the invoker.

If user is subscribed successfully, then the BCM will send back the corresponding readable response message to the invoker.

* + 1. APIs Involved:

POST https://my.sandbox.rt-stg.com/oauth/v2/token

* + - * + BCM is the consumer
        + Payload : partner\_username, partner\_password, client\_id and a client\_secret.

POST https://my.sandbox.rt-stg.com/api/partner/users

* + - * + BCM is the consumer
        + Payload : type, first\_name, last\_name, email, password.
        + Error Codes: TBD

201 - Returned when successful

400 - Error validating data. Please check parameters and retry.

409 - User exists.

500 - Internal Server Error

PATCH https://my.sandbox.rt-stg.com/api/tenant/details

* + - * + BCM is the consumer
        + Payload : first\_name, last\_name, middle\_name, email, phone, date\_of\_birth, ssn, email\_notification, offer\_notification, culture
        + Error Codes:

204 - Returned when successful

400 - Error validating data. Please check parameters and retry.

500 - Internal Server Error

POST /api/tenant/subscriptions/

* + - * + BCM is the consumer
        + Payload : payment\_account\_url, plan\_name, period, uuid, promotion\_code, uuid\_required
        + Error Codes:

201 - Returned when successful

400 - Error validating data. Please check parameters and retry.

409 - Can not create new one subscription

500 - Internal Server Error

* + 1. Messaging Infrastructure: N/A
    2. UI Screens: N/A
  1. DB Changes:

Table Name : **customer\_data**

Once the rent track account has been created, the BCM will store the user token details such as access\_token, refresh\_token and token expiry against ucid which will be used for subsequent renttrack api calls.

Table Fields :

* ucid
* customer\_email
* created\_date
* modified\_date
* access\_token
* refresh\_token
* token\_expiry
* token\_created\_date
  1. Dependencies

RentTrack API

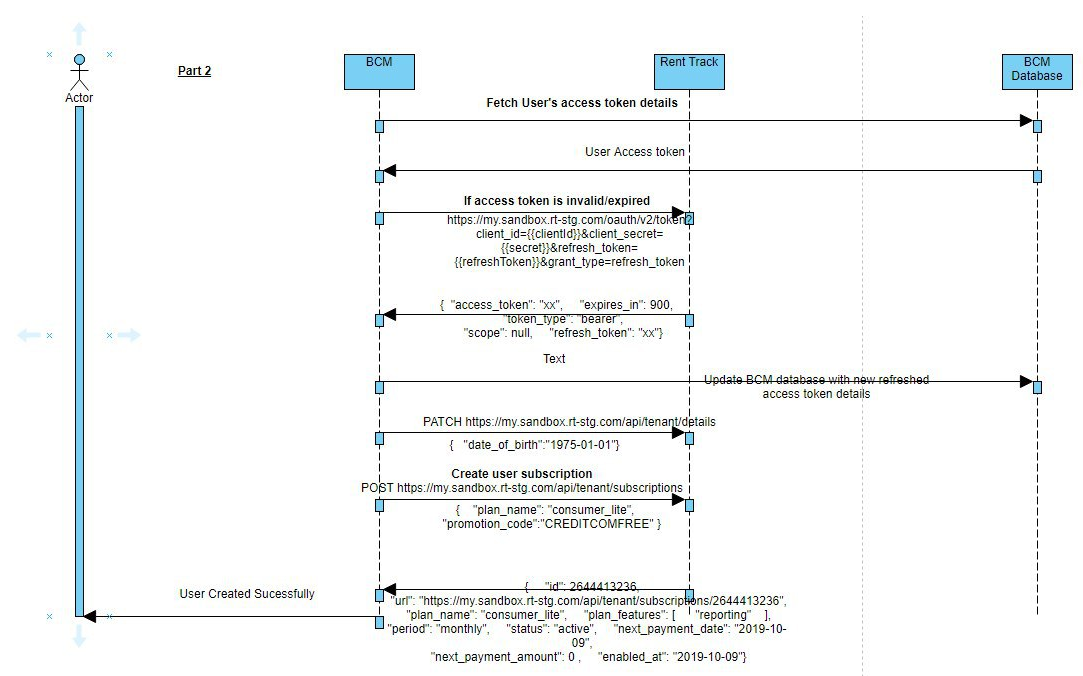
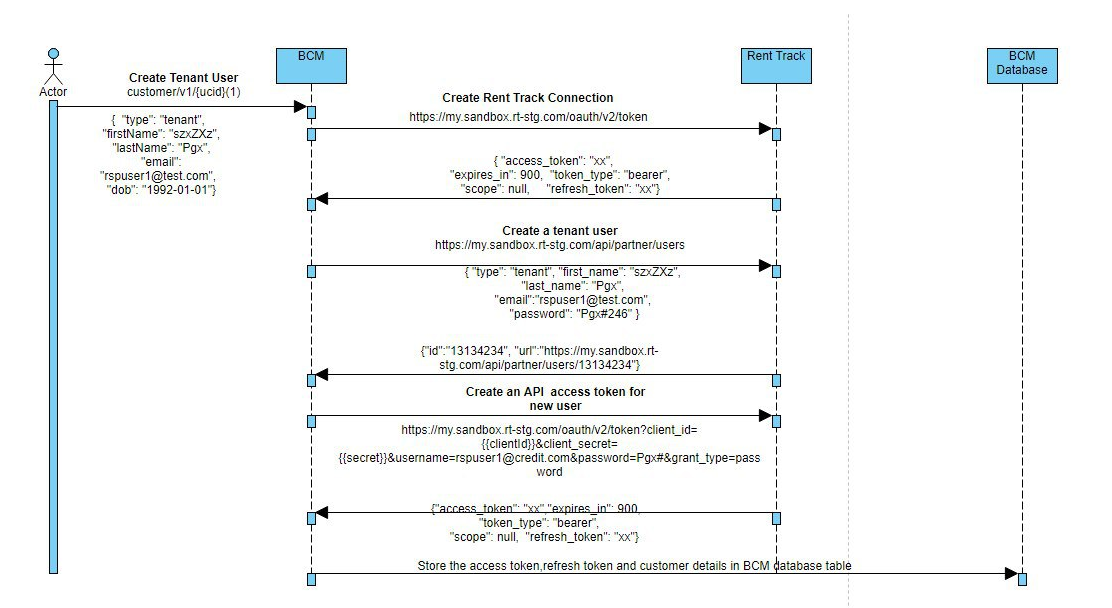
* 1. Blocker Inputs needed

The sandbox access need to be provided

* 1. Design Patterns considered and the chosen one

Service Factory pattern

* 1. sequence diagram



**Business** :

**Step 1** : Invoking sandbox RentTrack API with the required parameters as given below

**Request** :

<https://my.sandbox.rt-stg.com/oauth/v2/token>[?](https://my.sandbox.rt-stg.com/oauth/v2/token?client_id=26_26twn8ycdge8wo4gcc8k8kwwkwwocw00goc4o488s08g00w00w&client_secret=4k7iv14um3k0cwks4gwwg8k80oc4o48cgkog8g0cs48g884cww&username=ibell@progrexion.com&password=cdcR0cks!&grant_type=password)

[client\_id=26\_26twn8ycdge8wo4gcc8k8kwwkwwocw00goc4o488s08g00w00w&client\_secret=4k7iv14um3k0cwks4gwwg8k80oc4o48cgkog8g0cs48g884cww&username=ibell@progrexion.com&password=cdcR0cks!&grant\_type=password](https://my.sandbox.rt-stg.com/oauth/v2/token?client_id=26_26twn8ycdge8wo4gcc8k8kwwkwwocw00goc4o488s08g00w00w&client_secret=4k7iv14um3k0cwks4gwwg8k80oc4o48cgkog8g0cs48g884cww&username=ibell@progrexion.com&password=cdcR0cks!&grant_type=password)

**Response :**

{"access\_token":"OGI4NWM4NDY3NTE3ZDE3MmYwYThmNGI3ZDk0ODJhYmMwMDIzOTU5YzBlYzViYmU2YzNkNDBhOTMwMmNlMjUwZA","expires\_in":900,"token\_type":"bearer","scope":null,"refresh\_token":"ODk5Y2FmMDhlYWZhMWZjMTAwNzI1OGIxMjExNGEyMzBiYmY1MWNhZTE4Zjk4Y2NhOTc5ODZlZDc4ZmZjYjY4Zg"}

**Step 2 :** BCM will use the below rentTrack API, to create account in RentTrack for extra credit users

POST /api/partner/users.{\_format}

**Input** :

| Parameter | Type | Required? | Description |
| --- | --- | --- | --- |
| type |  | false | Only user type 'tenant' is supported. |
| first\_name |  | false | User firstname. |
| last\_name |  | false | User lastname. |
| email |  | false | User email. |
| password |  | false | User password. |
| holding\_id |  | false | Holding id in RentTrack system. |
| resident\_id |  | false | Accounting system resident id. |

**Output** :

| Parameter | Type | Versions | Description |
| --- | --- | --- | --- |
| 201 - Returned when successful |  | | |
| id | integer | \* |  |
| url | string | \* |  |

**Step 3** : Invoking sandbox RentTrack API with the required parameters to get the access token for the created user as given below

**Request** :

<https://my.sandbox.rt-stg.com/oauth/v2/token>[?](https://my.sandbox.rt-stg.com/oauth/v2/token?client_id=26_26twn8ycdge8wo4gcc8k8kwwkwwocw00goc4o488s08g00w00w&client_secret=4k7iv14um3k0cwks4gwwg8k80oc4o48cgkog8g0cs48g884cww&username=ibell@progrexion.com&password=cdcR0cks!&grant_type=password)

[client\_id=26\_26twn8ycdge8wo4gcc8k8kwwkwwocw00goc4o488s08g00w00w&client\_secret=4k7iv14um3k0cwks4gwwg8k80oc4o48cgkog8g0cs48g884cww&username={email}&password={password}&grant\_type=password](https://my.sandbox.rt-stg.com/oauth/v2/token?client_id=26_26twn8ycdge8wo4gcc8k8kwwkwwocw00goc4o488s08g00w00w&client_secret=4k7iv14um3k0cwks4gwwg8k80oc4o48cgkog8g0cs48g884cww&username=ibell@progrexion.com&password=cdcR0cks!&grant_type=password)

**Response :**

{"access\_token":"OGI4NWM4NDY3NTE3ZDE3MmYwYThmNGI3ZDk0ODJhYmMwMDIzOTU5YzBlYzViYmU2YzNkNDBhOTMwMmNlMjUwZA","expires\_in":900,"token\_type":"bearer","scope":null,"refresh\_token":"ODk5Y2FmMDhlYWZhMWZjMTAwNzI1OGIxMjExNGEyMzBiYmY1MWNhZTE4Zjk4Y2NhOTc5ODZlZDc4ZmZjYjY4Zg"}

**Step 4 :**

Using the *access\_token*, BCM can access any resource for the particular tenant in the RentTrack system. BCM will store the access\_token in customer\_data table against the ucid.

**Step 5** :

If the access\_token is expired (identified by expires\_in parameter value), BCM will invoke the refresh call using the refresh\_token to get a fresh access\_token.

**Request**:

[https://my.sandbox.rt-stg.com/oauth/v2/token?client\_id=26\_26twn8ycdge8wo4gcc8k8kwwkwwocw00goc4o488s08g00w00w&client\_secret=4k7iv14um3k0cwks4gwwg8k80oc4o48cgkog8g0cs48g884cww](https://my.sandbox.rt-stg.com/oauth/v2/token?client_id=26_26twn8ycdge8wo4gcc8k8kwwkwwocw00goc4o488s08g00w00w&client_secret=4k7iv14um3k0cwks4gwwg8k80oc4o48cgkog8g0cs48g884cww&username=ibell@progrexion.com&password=cdcR0cks!&grant_type=password)&refresh\_token=ODk5Y2FmMDhlYWZhMWZjMTAwNzI1OGIxMjExNGEyMzBiYmY1MWNhZTE4Zjk4Y2NhOTc5ODZlZDc4ZmZjYjY4Zg&grant\_type=refresh\_token Host: my.renttrack.com

**Response:**

{ "access\_token": "ZjYwZDEzOTgzMjJmYjc4NWRlYjdkOWZjMTMzYjg3MTAyYWU2Mzc3NWVkYjRhZjBiOGY3ZjFmNzY4OTkzNTcxNA", "expires\_in": 3600, "token\_type": "bearer", "scope": null, "refresh\_token": "MGMxODdmZjkxZmI1Mzk5ZDkzMTFhY2NmZDdiMDQ4Y2Y0MmE1ZDFiNWRiNjhmNzc5MWQzMWE1MTJhMTkwZjg0MA" }

**Step 6** : BCM will use the below rentTrack API, to update DOB for the created user in RentTrack.

PATCH /api/tenant/details.{\_format}

**Input** :

| Parameter | Type | Required? | Format | Description |
| --- | --- | --- | --- | --- |
| first\_name | string | false | {length: min: 1, max: 65}, {match: /^[a-zA-Z \-'\s]{1,65}$/}, {not blank} |  |
| last\_name | string | false | {length: min: 1, max: 65}, {match: /^[a-zA-Z \-'\s]{1,65}$/}, {not blank} |  |
| middle\_name | string | false |  |  |
| email | string | false | {not blank}, {email address} |  |
| phone | string | false | {match: /^\d{10}$/} |  |
| date\_of\_birth | DateTime | false |  |  |
| ssn | string | false | {match: /^\d{3}-\d{2}-\d{4}$/}, {match: /^\d{3}-\d{3}-\d{3}$/} | Tenant social security number. |
| email\_notification | boolean | false |  |  |
| offer\_notification | boolean | false |  |  |
| culture | string | false | {not blank} |  |

**Output**:

204 - Success

**Step 7** : Usign the access\_token, BCM will be able to call the below API to create a subscription for the created user.

POST /api/tenant/subscriptions.{\_format}

**Input** :

| Parameter | Type | Required? | Format | Description |
| --- | --- | --- | --- | --- |
| plan\_name | string | true | {not blank}, [score\_track|score\_track\_lite|credit\_track|consumer\_premium|consumer\_lite|consumer\_levelcredit\_report] | consumer\_lite is the default value |
| promotion\_code | string | false |  | CREDITCOMFREE is the default value |

**Output** :

| Parameter | Type | Versions | Description |
| --- | --- | --- | --- |
| 201 - Returned when successful |  | | |
| id | integer | \* |  |
| url | string | \* |  |
| payment\_account\_url | string | \* |  |
| plan\_name | string | \* | {'score\_track'|'credit\_track'} |
| period | string | \* | {'monthly'|'yearly'} |
| status | string | \* | {'active'|'cancelled'|'delinquent'} |
| last\_payment\_date | DateTime | \* | Format "YYYY-MM-DD" |
| next\_payment\_date | DateTime | \* | Format "YYYY-MM-DD" |
| next\_payment\_amount | float | \* |  |
| discounted\_until | DateTime | \* | Format "YYYY-MM-DD" |
| free\_until | DateTime | \* | Format "YYYY-MM-DD" |
| enabled\_at | DateTime | \* | Format "YYYY-MM-DD" |
| cancelled\_at | DateTime | \* | Format "YYYY-MM-DD" |
| external\_property\_id | string | \* |  |

**Clarifications**:

* In the single sign on document, it is mentioned that the kick off call will be set up to get the setup information? Will it be arranged before development or it is production setup ? RentTrack call setup done for our clarification.
* What is the unique value to identify the rentTrack user to do the PATCH request call ?

Access\_token is the unique value for all the calls to identify the rentTrack user.

**Code / Framework References**

**Misc:**