**US\_302468\_Cognizant: Enhance Create Tenant to prevent duplicate for fault tolerance**

**Understanding of the requirement**

Enhance the create tenant call to check if tenant has already been created completely.

**Assumptions**:

* Sandbox account and credentials for RentTrack API are provided and we are able to access
* Any changes made in RentTrack API which is given in the API documentation, will be notified.
* GitLab Apiary access will be provided

**High level Technical design and solution**

* 1. Impacted Modules

bcm-service

bcm-web

renttrack-plugin

* 1. Impacted Workflows
     1. Overview:

On create tenant the BCM module will check to see if a tenant has already been created for the user.

If yes, the system will check to ensure the subscription and the lease was created completely.

If the lease or the subscription is not created, BCM will create the lease and subscription.

If the lease and the subscription are already exists, BCM system will throw an error as “User Already Exists”

* + 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:

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

GET and 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

GET and POST /api/tenant/leases

* + - * + BCM is the consumer
        + Payload : Address, Landlord information and Rental Payment details
        + Error Codes:

201 - Returned when successful

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

409 - User exists.

500 - Internal Server Error

GET and POST /api/tenant/plaid/payment\_accounts

* + - * + BCM is the consumer
        + Payload : public\_token, plaid\_account\_id, pay\_type
        + Error Codes:

201 - Returned when successful

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

409 - User exists.

500 - Internal Server Error

POST and PATCH /api/tenant/leases/{leaseId}/transaction\_finders.{\_format}

BCM is the consumer

Payload : amount\_min, amount\_max, window\_open, window\_close, payment\_account\_url, lease\_url

Error Codes:

201 - Returned when successful

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

404 - Payment Account or Consumer Lease not found

500 - Internal Server Error

GET api/tenant/leases/{leaseId}/transaction\_finders/{id}/search.{\_format}

* + - * + BCM is the consumer
        + Payload : leaseId, transaction finder Id, amount\_min, amount\_max, window\_open and window\_close
        + Error Codes:

200 - Returned when successful

204 - Success with no content

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

404 - Transaction Finder or Consumer Lease not found

500 - Internal Server Error

iii. Messaging Infrastructure: N/A

iv. UI Screens: N/A

* 1. DB Changes:

1. Auditing Tables such as **WebService\_logs, External\_log, Customer\_Usage\_activity**

2. Table Name : **customer\_data**

Table Description :

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

RentTrack API

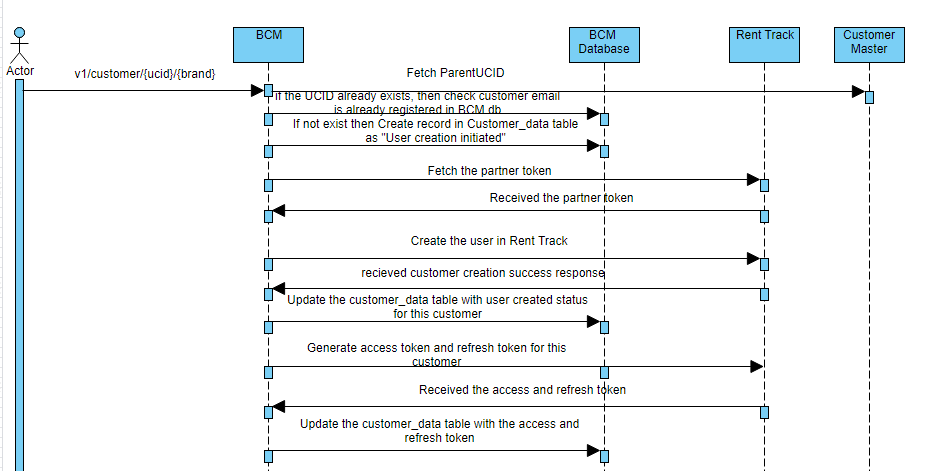
* 1. Blocker Inputs needed

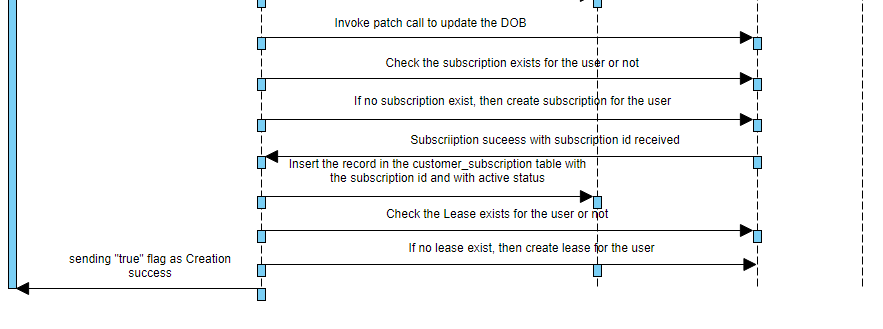
The sandbox access need to be provided

* 1. Design Patterns considered and the chosen one

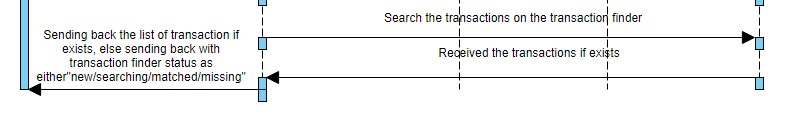
SOA

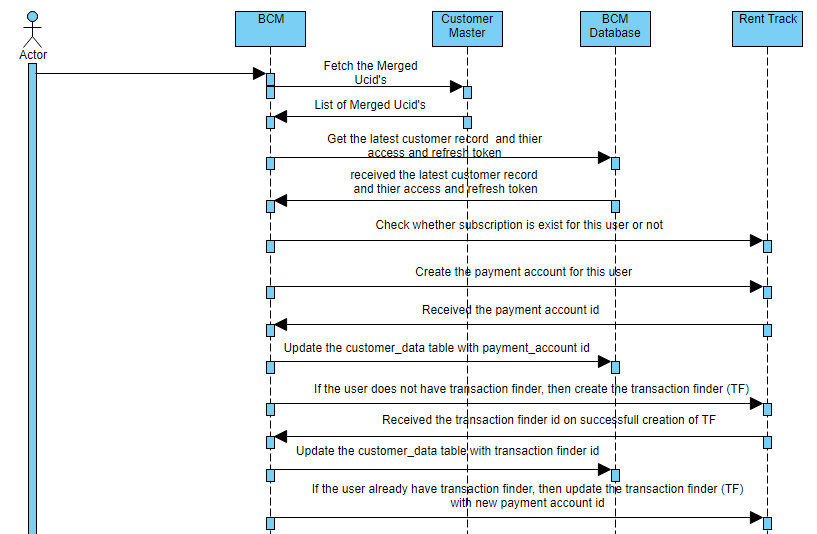
g) Sequence Diagram

**Create Tenant user Account**



**Create Payment Account**

****

**Business** :

**Fault Tolerance of BCM Create Tenant Service**

**Step 1** : BCM will validate the customer data table to check whether the user creation is successful. If yes, BCM will continue from Step 4.

**Step 2:** If the tenant user is not created then the BCM will invoke the 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 3 :** 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 4** : BCM will validate the customer\_data table to check whether the token has been created for the user. If yes, then BCM will skip the access token creation and executes from Step 7. Otherwise BCM will invoke the 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 5 :**

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 6**:

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 7** : 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 8** : Using the access\_token, BCM will be able to call the below API to check whether the subscription has been created for the tenant user.

GET /api/tenant/subscriptions/

**Output :**

| **Parameter** | **Type** | **Versions** | **Description** |
| --- | --- | --- | --- |
| 200 - Returned when successful |  | | |
| [] | array of objects (UserSubscription) | \* |  |
| [][id] | integer | \* |  |
| [][url] | string | \* |  |
| [][payment\_account\_url] | string | \* |  |
| [][plan\_name] | string | \* | {'score\_track'|'credit\_track'|...} |
| [][plan\_features] | array | \* | ['reporting, ‘score’, ...] |
| [][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 | \* |  |

**Step 9:** If the tenant user has active subscription, then BCM will skip the step 10 and executes from Step 11.

**Step 10:**

BCM will invoke the below RentTrack API to create the subscription for the 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 | \* |  |

**Step 11:** For the subscribed user, BCM will be invoke the below API to get a lease to check whether the lease has been already created for the user.

GET /api/tenant/leases.{\_format}

**Output :**

| Parameter | Type | Versions | Description |
| --- | --- | --- | --- |
| 200 - Returned when successful |  | | |
| [] | array of objects (Lease) | \* |  |
| [][id] | integer | \* |  |
| [][url] | string | \* |  |
| [][status] | string | \* |  |
| [][rent] | string | \* |  |
| [][start\_at] | string | \* | Format "YYYY-MM-DD" |
| [][finished\_at] | string | \* | Format "YYYY-MM-DD" |
| [][due\_day] | string | \* |  |
| [][moved\_out] | string | \* | Format "YYYY-MM-DD" |

**Step 12:** If the user has active lease, then the BCM will skip the Step 13 and respond back to the invoker that the User is already Exists.

**Step 13 :** For the subscribed user, If there is no active lease, then BCM will be invoke the below API to create a lease for the given lease details and respond back to the invoker with 200 response (User created successfully)

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

**Input :**

| Parameter | Type | Required? | Format | Description |
| --- | --- | --- | --- | --- |
| address | object (LeaseAddressDTO) | true |  |  |
| address[address1] | string | true | {not blank} |  |
| address[address2] | string | false |  |  |
| address[state] | string | true | {not blank} |  |
| address[city] | string | true | {not blank} |  |
| address[zip] | string | true | {not blank} |  |
| address[country] | string | false |  |  |
| rent | float | false |  | Rent amount. Include decimal. |
| due\_day | integer | false |  | Day of the month. |
| landlord | object (LeaseLandlordDTO) | false |  |  |
| landlord[type] | string | false |  |  |
| landlord[name] | string | false |  |  |
| landlord[phone] | string | false |  |  |
| landlord[email] | string | false |  |  |
| start\_at | DateTime | false |  | Lease start date. Format YYYY-mm-dd. |

**Output** :

| Parameter | Type | Versions | Description |
| --- | --- | --- | --- |
| 201 - Returned when successful |  | | |
| id | integer | \* |  |
| url | string | \* |  |
| address | object (LeaseAddress) | \* |  |
| status | string | \* |  |
| rent | string | \* |  |
| start\_at | string | \* | Format "YYYY-MM-DD" |
| finished\_at | string | \* | Format "YYYY-MM-DD" |
| due\_day | string | \* |  |
| landlord | object (LeaseLandlord) | \* |  |
| back\_report\_start\_at | string | \* | Format "YYYY-MM-DD" |

**Fault Tolerance of BCM Create Payment Account Service**

**Step 1:** BCM will receive the public token and plaid account id from the RSP(UI).

Request Payload:

{

"public\_token":

"public-sandbox-b0e2c4ee-a763-4df5-bfe9-46a46bce993d”

“plaid\_account\_id”:”i544dsd561g13”

}

**Step 2:** BCM will validate the active subscription for the user by using the below RentTrack API call. If the subscription is active, then BCM Module proceed with the create payment Account Service. Otherwise, BCM will respond back to the invoker with the message User is not subscribed.

GET /api/tenant/subscriptions/

**Response :**

| Parameter | Type | Versions | Description |
| --- | --- | --- | --- |
| 200 - Returned when successful |  | | |
| [] | array of objects (UserSubscription) | \* |  |
| [][id] | integer | \* |  |
| [][url] | string | \* |  |
| [][payment\_account\_url] | string | \* |  |
| [][plan\_name] | string | \* | {'score\_track'|'credit\_track'|...} |
| [][plan\_features] | array | \* | ['reporting, ‘score’, ...] |
| [][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 | \* |  |

**Step 3:**

BCM will invoke the below rentTrack API to check whether the payment Account has been exists for the tenant user. If the payment Account exists, then BCM will skip the

step 4

**RentTrack sandbox : GET /api/tenant/plaid/payment\_accounts**

**Response:**

| Parameter | Type | Versions | Description |
| --- | --- | --- | --- |
| 200 - Returned when successful |  | | |
| id | integer | \* |  |
| url | string | \* |  |
| nickname | string | \* |  |
| account\_name | string | \* |  |
| type | string | \* |  |
| last\_four | string | \* |  |
| expiration | string | \* | Format: "YYYY-mm" |
| billing\_address\_url | string | \* |  |
| institution\_id | string | \* |  |
| plaid\_account | boolean | \* |  |
| needs\_reconnect | boolean | \* |  |
| historical\_update\_complete | boolean | \* |  |
| card\_brand | string | \* |  |
| created\_at | DateTime | \* |  |

**Step 4:**

If the payment Account is not created, then using the public token and plaid account id, BCM will invoke below RentTrack API to create a new payment account

**RentTrack sandbox : POST /api/tenant/plaid/payment\_accounts**

**Input** :

| Parameter | Type | Required? | Format | Description |
| --- | --- | --- | --- | --- |
| public\_token | string | true | {not blank} | Plaid public token |
| plaid\_account\_id | string | false |  | Selected Id of Plaid Payment Account (optional) |
| pay\_type | string | false | [general|subscription|billpay|nopay] |  |

**Output** :

| Parameter | Type | Versions | Description |
| --- | --- | --- | --- |
| 201 - Returned when successful |  | | |
| [] | array of objects (PaymentAccount) | \* |  |
| [][id] | integer | \* |  |
| [][url] | string | \* |  |
| [][nickname] | string | \* |  |
| [][type] | string | \* |  |
| [][last\_four] | string | \* |  |
| [][expiration] | string | \* | Format: "YYYY-mm" |
| [][institution\_id] | string | \* |  |
| [][plaid\_account] | boolean | \* |  |
| [][needs\_reconnect] | boolean | \* |  |
| [][card\_brand] | string | \* |  |

**Step 5 :** BCM will validate the Customer Data table to check whether the transaction finder has been already created for the tenant user. If the transaction finder is already exists, then the BCM will invoke the below RentTrack API to update the transaction finder and executes from Step 7

PATCH api/tenant/transaction\_finders/{id}.{\_format}

Request Body:

| **Parameter** | **Type** | **Required?** | **Format** | **Description** |
| --- | --- | --- | --- | --- |
| amount\_min | float | false | {not blank} |  |
| amount\_max | float | false | {not blank} |  |
| window\_open | integer | false | {not blank} | Day of the month to open transactions window 1-31 |
| window\_close | integer | false | {not blank} | Day of the month to close transactions window 1-31 |
| payment\_account\_url | custom handler result for (ResourceUrlEncodedRelation) | false |  | Resource url for Payment Account. Can be included on url. |

**Step 6 :** If there is no transaction finder for the tenant user, then the BCM Module Get the active leave for the user by using the below RentTrack API

GET /api/tenant/leases

**Response Body:**

| Parameter | Type | Versions | Description |
| --- | --- | --- | --- |
| 200 - Returned when successful |  | | |
| [] | array of objects (Lease) | \* |  |
| [][id] | integer | \* |  |
| [][url] | string | \* |  |
| [][status] | string | \* |  |
| [][rent] | string | \* |  |
| [][start\_at] | string | \* | Format "YYYY-MM-DD" |
| [][finished\_at] | string | \* | Format "YYYY-MM-DD" |
| [][due\_day] | string | \* |  |
| [][moved\_out] | string | \* | Format "YYYY-MM-DD" |

**Step 7:** If there is no active Lease, then the BCM Module respond back to the invoker with the error message has “Lease Not Found”.

**Step 8:** if there is an active lease, then BCM will invoke the below RentTrack API call to create the transaction finder

POST /api/tenant/leases/{leaseId}/transaction\_finders.{\_format}

**Request Body :**

| **Parameter** | **Type** | **Required?** | **Format** | **Description** |
| --- | --- | --- | --- | --- |
| amount\_min | float | true | {not blank} |  |
| amount\_max | float | true | {not blank} |  |
| window\_open | integer | true | {not blank} | Day of the month to open transactions window 1-31 |
| window\_close | integer | true | {not blank} | Day of the month to close transactions window 1-31 |
| payment\_account\_url | custom handler result for (ResourceUrlEncodedRelation) | true |  | Resource url for Payment Account. Can be included on url. |
| lease\_url | custom handler result for (ResourceUrlEncodedRelation) | true |  | Resource url for Consumer Lease. Can be included on url. |

**Response :**

| Parameter | Type | Versions | Description |
| --- | --- | --- | --- |
| 201 - Returned when successful |  | | |
| id | integer | \* |  |
| url | string | \* |  |
| status | string | \* | Status of transaction finder: {'new'|'matched'|'searching'|'multiple'|'missing'|'closed'} |

**Step 7:** If the transaction finder has been created successfully, then BCM will invoke the below RentTrack API call to search the transactions, Otherwise respond back to the invoker with the error message.

GET /api/tenant/leases/{leaseId}/transaction\_finders/{id}/search.{\_format}

Filters

|  |
| --- |
| **Name** |
| **amount\_min**   |  |  | | --- | --- | | Datatype | float | | Description | Minimum amount of transaction. | |
| **amount\_max**   |  |  | | --- | --- | | Datatype | float | | Description | Maximum amount of transaction. | |
| **window\_open**   |  |  | | --- | --- | | Datatype | integer | | Description | Day of the month. | |
| **window\_close**   |  |  | | --- | --- | | Datatype | integer | | Description | Day of the month. | |

##### **Response:**

| **Parameter** | **Type** | **Versions** | **Description** |
| --- | --- | --- | --- |
| 200 - Returned when successful |  | | |
| [] | array of objects (PlaidTransaction) | \* |  |
| [][transaction\_id] | string | \* |  |
| [][amount] | string | \* |  |
| [][name] | string | \* |  |
| [][category\_id] | string | \* |  |
| [][date] | DateTime | \* | Format: "YYYY-MM-DD" |

**Step 8:** BCM will respond back to the invoker with the search transaction response

**Clarifications**:

BCM will save the transaction finder id and the payment account id against the BCM customer Id in customer\_data table in the BCM database.

**Code / Framework References**

**Misc:**