

Consumer Mobile SDK

Reference Guide

Version 0.1

Document Configuration Management

Document Identification

Name of Project	
Name of Report	Consumer Mobile SDK Reference Guide
Version	0.1 (draft)
File Name	
Sensitivity Classification	Confidential

Document Change Control

Version	Date Released	Change Notice
0.1		Initial (draft) for review and feedback

Table of Contents

DOCUMENT CONFIGURATION MANAGEMENT2

 DOCUMENT IDENTIFICATION.....2

 DOCUMENT PREPARATION.....ERROR! BOOKMARK NOT DEFINED.

 DOCUMENT CHANGE CONTROL.....2

1 INTRODUCTION12

.....12

2 INIT.....13

 2.1 ECSSERVICEPROVIDER.....13

 PARAMETERS13

 OUTPUT13

 2.2 SAMPLE13

3 SET PROPOSITION ID13

 3.1 ECSSERVICEPROVIDER.....13

 PARAMETERS14

 3.2 SAMPLE14

4 CONFIGURE ECS.....14

 4.1 ECSSERVICEPROVIDER.....14

 PARAMETERS14

 OUTPUT14

 4.2 SAMPLE14

 4.3 ERRORS15

5 CONFIGURE ECS TO FETCH CONFIGURATION.....15

 5.1 ECSSERVICEPROVIDER.....15

 PARAMETERS15

 OUTPUT15

5.2	SAMPLE	15
5.3	ERRORS	15
6	HYBRIS OAUTH AUTHENTICATION	16
6.1	ECSSERVICEPROVIDER.....	16
	PARAMETERS	16
	OUTPUT	16
6.2	SAMPLE	16
6.3	ERRORS	17
7	REFRESH HYBRIS OAUTH	17
7.1	ECSSERVICEPROVIDER.....	17
	PARAMETERS	17
	OUTPUT	17
7.2	SAMPLE	18
7.3	ERRORS	18
8	FETCH PRODUCT LIST	18
8.1	ECSSERVICEPROVIDER.....	18
	PARAMETERS	19
	OUTPUT	19
8.2	SAMPLE	19
8.3	ERRORS	19
9	FETCH PRODUCT FOR CTN	20
9.1	ECSSERVICEPROVIDER.....	20
	PARAMETERS	20
	OUTPUT	20
9.2	SAMPLE	20
9.3	ERRORS	20
10	FETCH PRODUCT LIST FOR SET OF CTNS	22

10.1	ECSSERVICEPROVIDER	22
	PARAMETERS	22
	OUTPUT	22
10.2	SAMPLE	22
10.3	ERRORS.....	22
11	FETCH PRODUCT DETAILS.....	24
11.1	ECSSERVICEPROVIDER	24
	PARAMETERS	24
	OUTPUT	24
11.2	SAMPLE	24
11.3	ERRORS.....	24
12	CREATE SHOPPING CART	25
12.1	ECSSERVICEPROVIDER	25
	PARAMETERS	25
	OUTPUT	25
12.2	SAMPLE	25
12.3	ERRORS.....	26
13	ADD PRODUCT TO SHOPPING CART	26
13.1	ECSSERVICEPROVIDER	26
	PARAMETERS	26
	OUTPUT	26
13.2	SAMPLE	27
13.3	ERRORS.....	27
14	FETCH SHOPPING CART	27
14.1	ECSSERVICEPROVIDER	27
	PARAMETERS	27
	OUTPUT	28

14.2	SAMPLE	28
14.3	ERRORS.....	28
15	UPDATE SHOPPING CART	29
15.1	ECSSERVICEPROVIDER	29
	PARAMETERS	29
	OUTPUT.....	29
15.2	SAMPLE	29
15.3	ERRORS.....	29
16	APPLY VOUCHER.....	30
16.1	ECSSERVICEPROVIDER	30
	PARAMETERS	30
	OUTPUT.....	30
16.2	SAMPLE	30
16.3	ERRORS.....	31
17	FETCH APPLIED VOUCHER.....	31
17.1	ECSSERVICEPROVIDER	31
	PARAMETERS	31
	OUTPUT.....	31
17.2	SAMPLE	32
17.3	ERRORS.....	32
18	REMOVE VOUCHER.....	33
18.1	ECSSERVICEPROVIDER	33
	PARAMETERS	33
	OUTPUT.....	33
18.2	SAMPLE	33
18.3	ERRORS.....	33
19	SET DELIVERY MODE.....	35

19.1	ECSSERVICEPROVIDER	35
	PARAMETERS	35
	OUTPUT	35
19.2	SAMPLE	35
19.3	ERRORS.....	35
20	FETCH DELIVERY MODES.....	36
20.1	ECSSERVICEPROVIDER	36
	PARAMETERS	36
	OUTPUT	36
20.2	SAMPLE	36
20.3	ERRORS.....	37
21	FETCH REGIONS.....	37
21.1	ECSSERVICEPROVIDER	37
	PARAMETERS	37
	OUTPUT	37
21.2	SAMPLE	37
21.3	ERRORS.....	38
22	CREATE ADDRESS	38
22.1	ECSSERVICEPROVIDER	38
	PARAMETERS	38
	OUTPUT	38
22.2	SAMPLE	39
22.3	ERRORS.....	39
23	CREATE AND FETCH ADDRESS	39
23.1	ECSSERVICEPROVIDER	39
	PARAMETERS	40
	OUTPUT	40

23.2	SAMPLE	40
23.3	ERRORS.....	40
24	FETCH SAVED ADDRESS	41
24.1	ECSSERVICEPROVIDER	41
	PARAMETERS	41
	OUTPUT.....	41
24.2	SAMPLE	41
24.3	ERRORS.....	41
25	SET DELIVERY ADDRESS.....	42
25.1	ECSSERVICEPROVIDER	42
	PARAMETERS	42
	OUTPUT.....	42
25.2	SAMPLE	42
25.3	ERRORS.....	42
26	SET AND FETCH DELIVERY ADDRESS	43
26.1	ECSSERVICEPROVIDER	43
	PARAMETERS	43
	OUTPUT.....	43
26.2	SAMPLE	43
26.3	ERRORS.....	44
27	UPDATE ADDRESS	44
27.1	ECSSERVICEPROVIDER	44
	PARAMETERS	44
	OUTPUT.....	44
27.2	SAMPLE	45
27.3	ERRORS.....	45
28	UPDATE AND FETCH ADDRESS.....	46

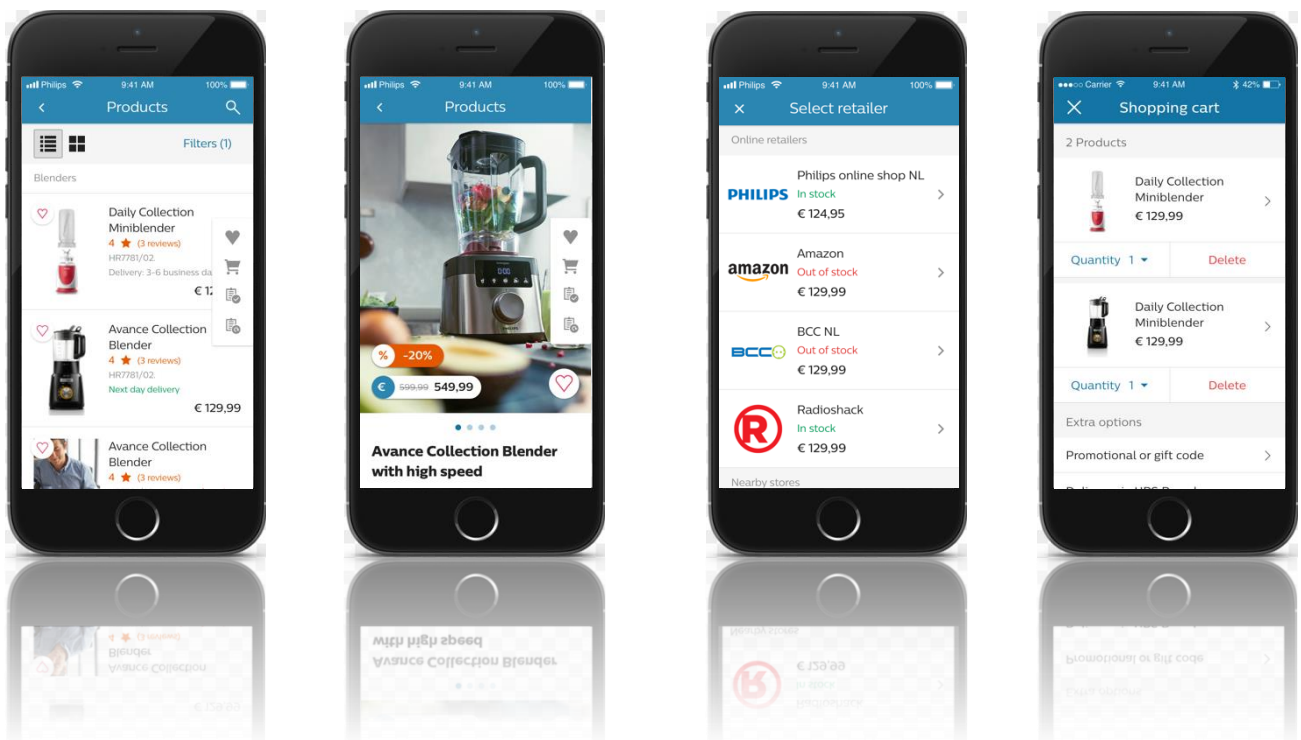
28.1	ECSSERVICEPROVIDER	46
	PARAMETERS	46
	OUTPUT	46
28.2	SAMPLE	46
28.3	ERRORS.....	46
29	DELETE ADDRESS.....	47
29.1	ECSSERVICEPROVIDER	47
	PARAMETERS	47
	OUTPUT	47
29.2	SAMPLE	48
29.3	ERRORS.....	48
30	DELETE AND FETCH ADDRESS.....	48
30.1	ECSSERVICEPROVIDER	48
	PARAMETERS	48
	OUTPUT	49
30.2	SAMPLE	49
30.3	ERRORS.....	49
31	FETCH RETAILERS	49
31.1	ECSSERVICEPROVIDER	49
	PARAMETERS	49
	OUTPUT	50
31.2	SAMPLE	50
31.3	ERRORS.....	50
32	FETCH RETAILERS	50
32.1	ECSSERVICEPROVIDER	50
	PARAMETERS	51
	OUTPUT	51

32.2	SAMPLE	51
32.3	ERRORS.....	51
33	SET PAYMENT DETAILS	52
33.1	ECSSERVICEPROVIDER	52
	PARAMETERS	52
	OUTPUT.....	52
33.2	SAMPLE	52
33.3	ERRORS.....	52
34	FETCH PAYMENT DETAILS	53
34.1	ECSSERVICEPROVIDER	53
	PARAMETERS	53
	OUTPUT.....	53
34.2	SAMPLE	53
34.3	ERRORS.....	53
35	SUBMIT ORDER.....	54
35.1	ECSSERVICEPROVIDER	54
	PARAMETERS	54
	OUTPUT.....	54
35.2	SAMPLE	54
35.3	ERRORS.....	54
36	MAKE PAYMENT	55
36.1	ECSSERVICEPROVIDER	55
	PARAMETERS	55
	OUTPUT.....	55
36.2	SAMPLE	56
36.3	ERRORS.....	56
37	FETCH ORDER HISTORY	56

37.1	ECSSERVICEPROVIDER	56
	PARAMETERS	57
	OUTPUT	57
37.2	SAMPLE	57
37.3	ERRORS.....	57
38	FETCH ORDER DETAIL	58
38.1	ECSSERVICEPROVIDER	58
	PARAMETERS	58
	OUTPUT	58
38.2	SAMPLE	58
38.3	ERRORS.....	58
39	FETCH ORDER DETAIL	59
39.1	ECSSERVICEPROVIDER	59
	PARAMETERS	59
	OUTPUT	59
39.2	SAMPLE	59
39.3	ERRORS.....	59
40	FETCH ORDER DETAIL	60
40.1	ECSSERVICEPROVIDER	60
	PARAMETERS	60
	OUTPUT	60
40.2	SAMPLE	60
40.3	ERRORS.....	61
41	FETCH USER PROFILE.....	61
41.1	ECSSERVICEPROVIDER	61
	PARAMETERS	61
	OUTPUT	61

41.2	SAMPLE	62
41.3	ERRORS.....	62

1 Introduction



The e-commerce software development kit (SDK) defines the In-App purchase experience and how it is delivered to the users. Embedding this into your existing android application enables you to design an integrated solution.

The SDK provides the necessary tools to build the application and gives you control over how the application looks and operates. Your own workflow and images can be used to design all aspects of user experience such as how product list appears, how to display shopping cart, displaying address, what details to be shown and what information to collect or not collect during the intake process. This ability to uniquely tailor In-App capabilities offers your application limitless possibilities for using the SDK and gives good user experience.

2 Init

This method is used to initialize the ECSSDKService.

2.1 ECSServiceProvider

```
appInfra = new AppInfra.Builder().build(getApplicationContext());  
public ECSServices(@Nullable String propositionID, @NonNull appInfra)  
{  
  
}
```

Parameters

- propositionID – It is a unique store identification string ,which is specific to business and country
- appInfra – appInfra object created by proposition

Output

- ECSSDKService object

2.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());  
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);
```

3 Set Proposition ID

This method is used to set proposition id.

3.1 ECSServiceProvider

```
void setPropositionID(@NonNull String propositionID);
```

Parameters

- **propositionID** – It is a unique store identification string ,which is specific to business and country

3.2 Sample

```
ECSConfiguration.INSTANCE.setPropositionID("your propositionID");
```

4 Configure ECS

This method is used to configure the ECS.

4.1 ECSServiceProvider

```
void configureECS(ECSCallback<Boolean,Exception> ecsCallback);
```

Parameters

- **ecsCallback** – instance of ECSCallBack of type Boolean and Exception.

Output

- **Success response** – If callback is success, returns true else false
True : represents Hybris flow
False : represents Retailer flow
- **Failure response** – Exception (contains localized error message) and ECSError which contains error code and error type.

4.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.configureECS(new ECSCallback<Boolean, Exception>() {
    @Override
    public void onResponse(Boolean result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

4.3 Errors

Server Error

- ECSSomethingWentWrong (We have encountered technical glitch. Please try after some time) – This error occurs when failed to get response or the proposition id might not be configured in Hybris

5 Configure ECS to Fetch Configuration

This method is used to fetch configuration from Hybris for the specific proposition id.

5.1 ECSServiceProvider

```
void configureECSToGetConfiguration(ECSCallback<ECSConfig, Exception> ecsCallback);
```

Parameters

- ecsCallback – instance of ECSCallback of type ECSConfig and Exception

Output

- Success response – ECSConfig object which contains all the ECS configuration data
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

5.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.configureECSToGetConfiguration(new ECSCallback<ECSConfig, Exception>()
{
    @Override
    public void onResponse(ECSConfig result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

5.3 Errors

Client Errors

- ECSSomethingWentWrong (We have encountered technical glitch. Please try after some time) – This error occurs when failed to get response.

6 Hybris OAuth Authentication

This method is used to make the OAuth authentication with Hybris. User needs to send the object of class `ECOAuthAuthProvider` which contains all the authentication related data as a parameter to the method.

6.1 ECSServiceProvider

```
void hybrisOAuthAuthentication(@NonNull EOAuthAuthProvider ecsoAuthProvider, @NonNull
ECSCallback<ECOAuthAuthData,Exception> ecsListener);
```

Parameters

- `ecsoAuthProvider` – object of the class `ECOAuthAuthProvider` which provide all oauth related data
- `ecsCallback` – instance of `ECSCallback` of type `ECOAuthAuthData` and `Exception`

Output

- Success response – `ECOAuthAuthData` object which contains Janrain id
- Failure response – `Exception` (contains localized error message) and `ECSError` which contains error code and error type.

6.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);
ECOAuthAuthProvider oAuthInput = new EOAuthAuthProvider() {
    @Override
    public String getOAuthID() {
        return "your Jainrain ID";
    }
};

ecsServices. hybrisOAuthAuthentication(oAuthInput, new ECSCallback<ECOAuthAuthData,
Exception>() {
    @Override
    public void onResponse(ECOAuthAuthData result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```


6.3 Errors

Client Errors

- ECSOAuthDetailError (Please provide complete OAuth details) – This error occurs when any OAuth detail is missing, so provide all the required details
- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale
- ECSInvalidTokenError (We have encountered technical glitch (Invalid access token))

Server Errors

- ECSInvalid_grant (We have encountered technical glitch (Invalid Janrain Token). Please re-login) – This error occurs when the Janrain token is invalid
- ECSInvalid_client (We have encountered technical glitch (Bad client credentials). Please re-login)

7 Refresh Hybris OAuth

This method is used to refresh the OAuth authentication with Hybris. User needs to send the object of class ECSOAuthProvider which contains all the authentication related data as a parameter to the method.

7.1 ECSServiceProvider

```
void hybrisRefreshOAuth(ECSOAuthProvider oAuthInput, ECSCallback<ECSOAuthData,
Exception> ecsListener);
```

Parameters

- ecsoAuthProvider – object of the class ECSOAuthProvider which provide all oauth related data
- ecsCallback – instance of ECSCallback of type ECSOAuthData and Exception

Output

- Success response – ECSOAuthData object which contains refresh token
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

7.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);
ECSSOAuthProvider oAuthInput = new ECSSOAuthProvider() {
    @Override
    public String getOAuthID() {
        return "your refresh token";
    }
};

ecsServices.hybrisRefreshOAuth(oAuthInput, new
ECSCallback<ECSSOAuthData,Exception>() {
    @Override
    public void onResponse(ECSSOAuthData result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

7.3 Errors

Client Errors

- ECSSOAuthDetailError (Please provide complete OAuth details) – This error occurs when any OAuth detail is missing, so provide all the required details
- ECSSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSSBaseUrlNotFound (Base URL not found) - occurs when Hybris is not present for the given locale

Server Errors

- ECSInvalid_grant (We have encountered technical glitch (Invalid Janrain Token). Please re-login) – This error occurs when the Janrain token is invalid

8 Fetch Product List

This method is used to fetch list of products for Hybris flow. It returns list of products having summary data fetched from PRX server. This can be called for an anonymous user as well.

8.1 ECSServiceProvider

```
void fetchProducts(int currentPage, int pageSize, ECSCallback<ECSProducts,Exception>
eCSCallback);
```

Parameters

- `currentPage` – refer to the page number for pagination
- `pageSize` – refer to the total number of data per page
- `eCSCallback` – instance of `ECSCallback` of type `ECSProducts` (which contains list of product) and `Exception`

Output

- Success response – List of products having summary data fetched from PRX
- Failure response – `Exception` (contains localized error message) and `ECSError` which contains error code and error type.

8.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.fetchProducts(0, 20, new ECSCallback<ECSProducts, Exception>() {
    @Override
    public void onResponse(ECSProducts result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

8.3 Errors

Client Errors

- `ECSLocaleNotFound` (Please set the locale) – call `configureECS` or `configureECSToGetConfiguration` to set locale
- `ECSSiteIdNotFound` (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- `ECSBaseURLNotFound` (Base URL not found) - occurs when Hybris is not present for the given locale

Server Errors

- `ECSNoSuchElementError` (An error occurred while fetching the data from the server). Please re-login)

9 Fetch Product for CTN

This method is used to fetch product specific to ctN for Hybris flow. It returns a product having detail and summary data fetched from PRX server. This method can be called for an anonymous user as well.

9.1 ECSServiceProvider

```
void fetchProduct(String ctN, ECSCallback<ECSPProduct,Exception> eCSCallback);
```

Parameters

- ctN – refers to the unique number of each product
- eCSCallback – instance of ECSCallback of type ECSPProduct and Exception

Output

- Success response – Product having details and summary data fetched from PRX
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

9.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.fetchProduct(mCTNValue, new ECSCallback <ECSPProduct, Exception>() {
    @Override
    public void onResponse(ECSPProducts result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

9.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale

- `ECSCtnNotProvided` (Please provide the CTN) – occurs when ctn or not provided or invalid ctn is entered

Server Errors

- `ECSNoSuchElementError` (An error occurred while fetching the data from the server). Please re-login)

10 Fetch Product List for Set of CTNs

This method is used to fetch list of products for Hybris flow. It returns list of products having detail and summary data fetched from PRX server. This method can be called for an anonymous user as well.

10.1 ECSServiceProvider

```
void fetchProductSummaries(List<String>
ctns, ECSCallback<List<ECSPRODUCT>, Exception> eCSCallback);
```

Parameters

- ctns – refer to the list of unique numbers of each product
- eCSCallback – instance of ECSCallback containing list of ECSPRODUCT and Exception

Output

- Success response – List of products having summary data fetched from PRX
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

10.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.fetchProductSummaries(productList, new ECSCallback<List<ECSPRODUCT>,
Exception>() {
    @Override
    public void onResponse(List<ECSPRODUCT> result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

10.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – occurs when locale is null, call configureECS or configureECSToGetConfiguration to set locale
- ECSCtnNotProvided (Please provide the CTN) – occurs when ctn or not provided or invalid ctn is entered

Server Errors

- ECSNoSuchElementError (An error occurred while fetching the data from the server). Please re-login)

11 Fetch Product Details

This method is used to fetch product details containing assets and disclaimer for the specific product from PRX server.

11.1 ECSServiceProvider

```
void fetchProductDetails(ECSPProduct product, ECSCallback<ECSPProduct,Exception>
ecsCallback);
```

Parameters

- product – refer to the object of class ECSPProduct for which detail needs to be fetched
- ecsCallback – instance of ECSCallback of type ECSPProduct and Exception

Output

- Success response – ECSPProduct object having assets and disclaimer data fetched from PRX server.
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

11.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.fetchProductDetails(product, new ECSCallback<ECSPProduct, Exception>() {
    @Override
    public void onResponse(ECSPProduct result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

11.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale

- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale
- ECSCtnNotProvided (Please provide the CTN) – occurs when ctn is not provided or invalid ctn is entered

12 Create Shopping Cart

This method is used to create the shopping cart for the user. User needs to oAuth before calling this method.

12.1 ECSServiceProvider

```
void createShoppingCart (ECSCallback<ECSShoppingCart,Exception> ecsCallback);
```

Parameters

- ecsCallback – instance of ECSCallback of type ECSShoppingCart and Exception

Output

- Success response – ECSShoppingCart object
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

12.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.createShoppingCart(new ECSCallback<ECSShoppingCart, Exception>() {
    @Override
    public void onResponse(ECSShoppingCart result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

12.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale

13 Add Product to Shopping Cart

This method is used add product to the shopping cart.

13.1 ECSServiceProvider

```
void addProductToShoppingCart(ECSProduct product, ECSCallback<ECSShoppingCart,  
Exception> ecsCallback);
```

Parameters

- product – an object of type ECSProduct which needs to be added to the shopping cart
- ecsCallback – instance of ECSCallback of type ECSShoppingCart and Exception

Output

- Success response – ECSShoppingCart object having details of the product added to cart.
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

13.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.addProductToShoppingCart(product, new ECSCallback<ECSShoppingCart,
Exception>() {
    @Override
    public void onResponse(ECSShoppingCart result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

13.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale
- ECSCtnNotProvided (Please provide the CTN) – occurs when ctn is not provided or invalid ctn is entered

Server Errors

- ECSCartError (No cart created yet) – This error occurs when cart is not created, to resolve call createShoppingCart
- ECSCartEntryError (The Product is not added in Cart) – occurs when failed to add product to cart

14 Fetch Shopping Cart

This method is used to get the shopping cart details for the user. User needs to oAuth before calling this method.

14.1 ECSServiceProvider

```
void fetchShoppingCart(ECSCallback<ECSShoppingCart,Exception> ecsCallback);
```

Parameters

- ecsCallback – instance of ECSCallback containing ECSShoppingCart and Exception

Output

- Success response – ECSShoppingCart object
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

14.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.fetchShoppingCart(new ECSCallback<ECSShoppingCart, Exception>() {
    @Override
    public void onResponse(ECSShoppingCart result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

14.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale

Server Errors

- ECSNoSuchElementError (An error occurred while fetching the data from the server). Please re-login) – This error occurs when failed to get the response
- ECSCartError (No cart created yet)

15 Update Shopping Cart

This method is used to update the quantity in the shopping cart for the product.

15.1 ECSServiceProvider

```
void updateShoppingCart(int quantity, ECSEntries entriesEntity,
    ECSCallback<ECSShoppingCart, Exception> ecsCallback) ;
```

Parameters

- quantity – an integer value which tells the quantity which needs to be updated
- ecsCallback – instance of ECSCallback of type ECSShoppingCart and Exception

Output

- Success response – ECSEntries object which contains object of type ECSProduct for which the quantity needs to be updated
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

15.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.updateShoppingCart(count, entriesEntity, new
    ECSCallback<ECSShoppingCart, Exception>() {
        @Override
        public void onResponse(ECSShoppingCart result) {

        }

        @Override
        public void onFailure(Exception error, ECSError ecsError) {

        }
    });
```

15.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present

- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale
- ECSCommerceCartModificationError (Quantity must not be less than zero) – occurs when quantity is less than zero

Server Errors

- ECSCommerceCartModificationError (Quantity must not be less than zero) – occurs when quantity is less than zero

16 Apply Voucher

This method is used to apply voucher on the product on cart level or products level.

16.1 ECSServiceProvider

```
void applyVoucher(String voucherCode, ECSCallback<List<ECSVoucher>, Exception>
ecsCallback);
```

Parameters

- voucherCode – a string value which has to be applied on cart or product
- ecsCallback – instance of ECSCallback of type ECSVoucher and Exception

Output

- Success response – list of ECSVoucher object with updated data
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

16.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.applyVoucher(voucherId, new ECSCallback<List<ECSVoucher>, Exception>()
{
    @Override
    public void onResponse(List<ECSVoucher> result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }

});
```

16.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale

Server Errors

- ECSUnsupportedVoucherError (Sorry, this voucher can't be redeemed) - occurs when invalid voucher code is entered
- ECSVoucherOperationError (Sorry, this voucher can't be redeemed)

17 Fetch Applied Voucher

This method is used to fetch the list of applied vouchers.

17.1 ECSServiceProvider

```
void fetchAppliedVouchers (ECSCallback<List<ECSVoucher>, Exception> ecsCallback);
```

Parameters

- ecsCallback – instance of ECSCallback containing list of ECSVoucher and Exception

Output

- Success response – List of ECSVoucher object containing list of applied vouchers
- Failure response – Exception (contains localized error message) and ECSException which contains error code and error type.

17.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.fetchAppliedVouchers(new ECSCallback<List<ECSVoucher>, Exception>() {
    @Override
    public void onResponse(List<ECSVoucher> result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

17.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale

Server Errors

- ECSVoucherOperationError (Sorry, this voucher can't be redeemed)
- ECSUnsupportedVoucherError (Sorry, this voucher can't be redeemed) - occurs when invalid voucher code is entered

18 Remove Voucher

This method is used to remove the applied voucher.

18.1 ECSServiceProvider

```
void removeVoucher(String voucherCode, ECSCallback<List<ECSVoucher>, Exception>
ecsCallback);
```

Parameters

- voucherCode – string value which is a unique number of which voucher has to be removed
- ecsCallback – instance of ECSCallback containing list of ECSVoucher and Exception

Output

- Success response – list of ECSVoucher object
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

18.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.removeVoucher(voucherCode, new ECSCallback<List<ECSVoucher>,
Exception>() {
    @Override
    public void onResponse(List<ECSVoucher> result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

18.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale
- ECSUnsupportedVoucherError (Sorry, this voucher can't be redeemed) - occurs when invalid voucher code is entered

Server Errors

- `ECSVoucherOperationError` (Sorry, this voucher can't be redeemed)
- `ECSUnsupportedVoucherError` (Sorry, this voucher can't be redeemed) - occurs when invalid voucher code is entered

19 Set Delivery Mode

This method is used to set delivery mode for cart.

19.1 ECSServiceProvider

```
void setDeliveryMode(ECSDeliveryMode deliveryModes, ECSCallback<Boolean, Exception> ecsCallback);
```

Parameters

- deliveryModes – ECSDeliveryMode object to be set as delivery mode
- ecsCallback – instance of ECSCallback of type Boolean and Exception

Output

- Success response – returns true if delivery mode is set successfully
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

19.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.setDeliveryMode(deliveryMode, new ECSCallback<Boolean, Exception>(){
    @Override
    public void onResponse(Boolean result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

19.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present

- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale
- ECSUnsupportedDeliveryModeError (Selected delivery mode is not supported)

Server Errors

- ECSUnsupportedDeliveryModeError (Selected delivery mode is not supported)

20 Fetch Delivery Modes

This method is used to fetch the list of delivery modes.

20.1 ECSServiceProvider

```
void fetchDeliveryModes(ECSCallback<List<ECSDeliveryMode>, Exception> ecsCallback);
```

Parameters

- ecsCallback – instance of ECSCallback containing list of ECSDeliveryMode and Exception

Output

- Success response – list of ECSDeliveryMode object which contains list of delivery modes
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

20.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.fetchDeliveryModes(new ECSCallback<List<ECSDeliveryMode>, Exception>()
{
    @Override
    public void onResponse(ECSDeliveryMode result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

20.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale

21 Fetch Regions

This method is used to fetch the list of regions for the selected country.

21.1 ECSServiceProvider

```
void fetchRegions(ECSCallback<List<ECSRegion>, Exception> ecsCallback);
```

Parameters

- ecsCallback – instance of ECSCallback containing list of ECSRegion and Exception

Output

- Success response – list of ECSRegion object which contains list of regions
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

21.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.fetchRegions(new ECSCallback<List<ECSRegion>, Exception>() {
    @Override
    public void onResponse(List<ECSRegion> result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

21.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale

Server Errors

- ECSCountryCodeNotGiven (Please provide valid country code)
- ECSRegionisocode (Region selected is invalid)

22 Create Address

This method is used to create new address for the user.

22.1 ECSServiceProvider

```
void createAddress(ECSAddress address, ECSCallback<ECSAddress, Exception>
ecsCallback);
```

Parameters

- address – an object of class ECSAddress which needs to be added
- ecsCallback – instance of ECSCallback of type ECSAddress and Exception

Output

- Success response – ECSAddress object which contains address details
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

22.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.createAddress(address, new ECSCallback< ECSAddress, Exception>() {
    @Override
    public void onResponse(ECSAddress result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

22.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale

Server Errors

- ECSCountryCodeNotGiven (Please provide valid country code)
- ECSRegionisocode (Region selected is invalid) –
- ECSPostalCode (ZIP code selected is invalid)
- ECSfirstName (Please notice that the following characters are not allowed: \") (* ^ % \$ # @ ! + , . ? / : ; { } [] ` ~ =)
- ECSlastName (Please notice that the following characters are not allowed: \") (* ^ % \$ # @ ! + , . ? / : ; { } [] ` ~ =)
- ECSphone1 (Phone number selected is invalid)
- ECSaddressId (Given address doesn't exist or belong to another user)
- ECSValidationError (Data validation failed)

23 Create and Fetch Address

This method is used to create new address for the user and fetch the list of all saved addresses.

23.1 ECSServiceProvider

```
void createAndFetchAddress(ECSAddress address, ECSCallback<List<ECSAddress>,
Exception> ecsCallback);
```

Parameters

- address – an object of class ECSAddress which needs to be added
- ecsCallback – instance of ECSCallback containing list of ECSAddress and Exception

Output

- Success response – ECSAddress object which contains list of addresses
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

23.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.createAndFetchAddress(address, new ECSCallback<List<ECSAddress>,
Exception>() {
    @Override
    public void onResponse(<List<ECSAddress> result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

23.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseUrlNotFound (Base URL not found) - occurs when Hybris is not present for the given locale

Server Errors

- ECSCountryCodeNotGiven (Please provide valid country code)
- ECSRegionisocode (Region selected is invalid) –
- ECSPostalCode (ZIP code selected is invalid)
- ECSfirstName (Please notice that the following characters are not allowed: \") (* ^ % \$ # @ ! + , . ? / : ; { } [] ` ~ =)
- ECSlastName (Please notice that the following characters are not allowed: \") (* ^ % \$ # @ ! + , . ? / : ; { } [] ` ~ =)
- ECSphone1 (Phone number selected is invalid)
- ECSaddressId (Given address doesn't exist or belong to another user)
- ECSValidationError (Data validation failed)

24 Fetch Saved Address

This method is used to fetch the list of all saved addresses for the user.

24.1 ECSServiceProvider

```
void fetchSavedAddresses (ECSCallback<List<ECSAddress>, Exception> ecsCallback);
```

Parameters

- `ecsCallback` – instance of `ECSCallback` containing list of `ECSAddress` and `Exception`

Output

- Success response – list of `ECSAddress` object which contains list of saved addresses
- Failure response – `Exception` (contains localized error message) and `ECSError` which contains error code and error type.

24.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.fetchSavedAddresses(new ECSCallback<List<ECSAddress>, Exception>() {
    @Override
    public void onResponse(List<ECSAddress> result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

24.3 Errors

Client Errors

- `ECSLocaleNotFound` (Please set the locale) – call `configureECS` or `configureECSToGetConfiguration` to set locale
- `ECSSiteIdNotFound` (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- `ECSBaseURLNotFound` (Base URL not found) - occurs when Hybris is not present for the given locale

Server Errors

- ECSNoSuchElementError (An error occurred while fetching the data from the server). Please re-login) – This error occurs when failed to get the response

25 Set Delivery Address

This method is used to set the delivery address of the user.

25.1 ECSServiceProvider

```
void setDeliveryAddress(ECSAddress address, ECSCallback<Boolean, Exception>
ecsCallback);
```

Parameters

- address – an object of class ECSAddress which has to be set as delivery address
- ecsCallback – instance of ECSCallback of type Boolean and Exception

Output

- Success response – returns true if selected address is set as delivery address successfully
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

25.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.setDeliveryAddress(addresses, new ECSCallback<Boolean, Exception>() {
    @Override
    public void onResponse(Boolean result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

25.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale

- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale

Server Errors

- ECSUnsupportedDeliveryModeError (Selected delivery mode is not supported)
- ECSaddressId (Given address doesn't exist or belong to another user)

26 Set and Fetch Delivery Address

This method is used to set the delivery address of the user and fetch the address.

26.1 ECSServiceProvider

```
void setAndFetchDeliveryAddress(ECSAddress address, ECSCallback<List<ECSAddress>,
Exception> ecsCallback);
```

Parameters

- address – an object of class ECSAddress which has to be set as delivery address
- ecsCallback – instance of ECSCallback containing list of ECSAddress and Exception

Output

- Success response – returns the list of addresses
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

26.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.setAndFetchDeliveryAddress(address,new ECSCallback<List<ECSAddress>,
Exception>() {
    @Override
    public void onResponse(List<ECSAddress> result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

26.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale

Server Errors

- ECSUnsupportedDeliveryModeError (Selected delivery mode is not supported)
- ECSAddressId (Given address doesn't exist or belong to another user)

27 Update Address

This method is used to update the address of the user. This is used whenever the user edits the address.

27.1 ECSServiceProvider

```
void updateAddress(boolean isDefaultAddress, ECSAddress address,
    ECSCallback<Boolean, Exception> ecsCallback);
```

Parameters

- isDefaultAddress – Boolean value, if it is true the updated address is set as default address
- address – ECSAddress object for which the address is to be updated
- ecsCallback – instance of ECSCallback of type Boolean and Exception

Output

- Success response – returns true if the address is updated successfully
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

27.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.updateAddress(false, addresses, new ECSCallback<Boolean, Exception>() {

    @Override
    public void onResponse(Boolean result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }

});
```

27.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale

Server Errors

- ECSUnsupportedDeliveryModeError (Selected delivery mode is not supported)
- ECSCountryCodeNotGiven (Please provide valid country code)
- ECSRegionisocode (Region selected is invalid) –
- ECSPostalCode (ZIP code selected is invalid)
- ECSfirstName (Please notice that the following characters are not allowed: \") (* ^ % \$ # @ ! + , . ? / : ; { } [] ` ~ =)
- ECSlastName (Please notice that the following characters are not allowed: \") (* ^ % \$ # @ ! + , . ? / : ; { } [] ` ~ =)
- ECSphone1 (Phone number selected is invalid)
- ECSaddressId (Given address doesn't exist or belong to another user)
- ECSValidationError (Data validation failed)

28 Update and Fetch Address

This method is used to update the address of the user and fetch the list of addresses.

28.1 ECSServiceProvider

```
void updateAndFetchAddress(boolean isDefaultAddress, ECSAddress address,
    ECSCallback<List<ECSAddress>, Exception> ecsCallback);
```

Parameters

- isDefaultAddress – Boolean value, if it is true the updated address is set as default address
- address – ECSAddress object for which the address is to be updated
- ecsCallback – instance of ECSCallback containing list of ECSAddress and Exception

Output

- Success response – returns the list of addresses
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

28.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.updateAndFetchAddress(true, address,new ECSCallback<List<ECSAddress>,
Exception>() {
    @Override
    public void onResponse(List<ECSAddress> result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

28.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale

Server Errors

- ECSUnsupportedDeliveryModeError (Selected delivery mode is not supported)
- ECSCountryCodeNotGiven (Please provide valid country code)
- ECSRegionisocode (Region selected is invalid) –
- ECSPostalCode (ZIP code selected is invalid)
- ECSfirstName (Please notice that the following characters are not allowed: \") (* ^ % \$ # @ ! + , . ? / : ; { } [] ` ~ =)
- ECSlastName (Please notice that the following characters are not allowed: \") (* ^ % \$ # @ ! + , . ? / : ; { } [] ` ~ =)
- ECSphone1 (Phone number selected is invalid)
- ECSaddressId (Given address doesn't exist or belong to another user)
- ECSValidationError (Data validation failed)

29 Delete Address

This method is used to delete the address of the user.

29.1 ECSServiceProvider

```
void deleteAddress(ECSAddress address, ECSCallback<Boolean, Exception>
ecsCallback);
```

Parameters

- address – ECSAddress object for which the address is to be deleted
- ecsCallback – instance of ECSCallback of type Boolean and Exception

Output

- Success response – returns true if the address is deleted successfully
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

29.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.deleteAddress(address, new ECSCallback<Boolean, Exception>() {
    @Override
    public void onResponse(Boolean result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

29.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale

Server Errors

- ECSAddressId (Given address doesn't exist or belong to another user)

30 Delete and Fetch Address

This method is used to delete the address of the user and fetch the list of addresses.

30.1 ECSServiceProvider

```
void deleteAndFetchAddress(@NonNull ECSAddress address, @NonNull
ECSCallback<List<ECSAddress>, Exception> ecsCallback);
```

Parameters

- address – ECSAddress object for which the address is to be deleted
- ecsCallback – instance of ECSCallback containing list of ECSAddress and Exception

Output

- Success response – returns list of addresses
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

30.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.deleteAndFetchAddress(address, new ECSCallback<List<ECSAddress>,
Exception>() {

    @Override
    public void onResponse(List<ECSAddress> result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }

});
```

30.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale

Server Errors

- ECSAddressId (Given address doesn't exist or belong to another user)

31 Fetch Retailers

This method is used to fetch the list of available retailers for the selected product.

31.1 ECSServiceProvider

```
void fetchRetailers(String ctn, ECSCallback<ECSRetailerList,Exception>ecsCallback);
```

Parameters

- ctn – ctn of the selected product

- `ecsCallback` – instance of `ECSCallback` of type `ECSRetailerList` and `Exception`

Output

- Success response – list of `ECSRetailerList` object which contains list of retailers
- Failure response – `Exception` (contains localized error message) and `ECSError` which contains error code and error type.

31.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.fetchRetailers(ctn, new ECSCallback<ECSRetailerList, Exception>()
    @Override
    public void onResponse(ECSRetailerList result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

31.3 Errors

Client Errors

- `ECSLocaleNotFound` (Please set the locale) – call `configureECS` or `configureECSToGetConfiguration` to set locale
- `ECSCtnNotProvided` (Please provide the CTN) – occurs when `ctn` or not provided or invalid `ctn` is entered

Server Errors

- `ECSCtnNotProvided` (Please provide the CTN) – occurs when `ctn` is not provided or invalid `ctn` is entered

32 Fetch Retailers

This method is used to fetch the list of available retailers for the selected product.

32.1 ECSServiceProvider

```
void fetchRetailers(ECSProduct product, ECSCallback<ECSRetailerList,Exception>
ecsCallback);
```

Parameters

- product – ECSProduct object
- ecsCallback – instance of ECSCallback of type ECSRetailerList and Exception

Output

- Success response – list of ECSRetailerList object which contains list of retailers
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

32.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.fetchRetailers(product, new ECSCallback<ECSRetailerList, Exception>() {
    @Override
    public void onResponse(ECSRetailerList result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

32.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSCtnNotProvided (Please provide the CTN) – occurs when ctn or not provided or invalid ctn is entered

Server Errors

- ECSCtnNotProvided (Please provide the CTN) – occurs when ctn is not provided or invalid ctn is entered

33 Set Payment Details

This method is used to set payment details for the purchase.

33.1 ECSServiceProvider

```
void setPaymentDetails(String paymentDetailsId, ECSCallback<Boolean, Exception>
ecsCallback);
```

Parameters

- paymentDetailsId – a string id representing the payment method
- ecsCallback – instance of ECSCallback of type Boolean and Exception

Output

- Success response – returns true, if payment details is set successfully
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

33.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.setPaymentDetails(paymentId, new ECSCallback<Boolean, Exception>(){
    @Override
    public void onResponse(Boolean result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

33.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale

- ECSInvalidPaymentInfoError (Please provide valid payment details)

Server Errors

- ECSInvalidPaymentInfoError (Please provide valid payment details)

34 Fetch Payment Details

This method is used to fetch list of existing payment methods.

34.1 ECSServiceProvider

```
void fetchPaymentsDetails(ECSCallback<List<ECSPayment>, Exception> ecsCallback);
```

Parameters

- ecsCallback – instance of ECSCallback containing list of ECSPayment and Exception

Output

- Success response – ECSPayment object
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

34.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.fetchPaymentsDetails(new ECSCallback<List<ECSPayment>, Exception>() {
    @Override
    public void onResponse(List<ECSPayment> result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

34.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale

- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale

35 Submit Order

This method is used to place the order by entering cvv, after selecting the payment method.

35.1 ECSServiceProvider

```
void submitOrder(String cvv, ECSCallback<ECSOrderDetail, Exception> ecsCallback);
```

Parameters

- cvv – string value to be entered for placing the order
- ecsCallback – instance of ECSCallback of type ECSOrderDetail and Exception

Output

- Success response – ECSOrderDetail object which contains details of order placed
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

35.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.submitOrder(pSecurityCode, new ECSCallback<ECSOrderDetail, Exception>()
{
    @Override
    public void onResponse(ECSOrderDetail result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

35.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale

- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale

Server Errors

- ECSSessionCart (Please select your delivery address, delivery mode and payment method before submitting an order)

36 Make Payment

This method is used to make the payment after entering all the payment details.

36.1 ECSServiceProvider

```
void makePayment(ECSOrderDetail orderDetail, ECSAddress billingAddress,
ECSCallback<ECSPaymentProvider, Exception> ecsCallback);
```

Parameters

- orderDetail – object of class ECSOrderDetail which contains placed order details
- billingAddress – object of class ECSAddress which contains billing address
- ecsCallback – instance of ECSCallback of type ECSPaymentProvider and Exception

Output

- Success response – ECSPaymentProvider object which contains world pay url
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

36.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.makePayment(orderDetail, billingAddress, new
ECSCallback<ECSPaymentProvider, Exception>() {
    @Override
    public void onResponse(ECSPaymentProvider result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

36.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale
- ECSorderIdNil (Please provide valid order) – occurs for invalid order id

Server Errors

- ECSorderIdNil (Please provide valid order) – occurs for invalid order id
- ECSPaymentAuthorizationError (Payment authorization was not successful)

37 Fetch Order History

This method is used to fetch the order history of the submitted orders.

37.1 ECSServiceProvider

```
void fetchOrderHistory(int pageNumber, int pageSize, ECSCallback<ECSOrderHistory,
Exception> ecsCallback);
```


Parameters

- `pageNumber` – refer to the page number containing data
- `pageSize` – refer to the total number of pages
- `ecsCallback` – instance of `ECSCallback` of type `ECSOrderHistory` and `Exception`

Output

- Success response – `ECSOrderHistory` object which contains details of placed order
- Failure response – `Exception` (contains localized error message) and `ECSError` which contains error code and error type.

37.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.fetchOrderHistory(pageNo, 0, new ECSCallback<ECSOrderHistory,
Exception>() {
    @Override
    public void onResponse(ECSOrderHistory result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

37.3 Errors

Client Errors

- `ECSLocaleNotFound` (Please set the locale) – call `configureECS` or `configureECSToGetConfiguration` to set locale
- `ECSSiteIdNotFound` (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- `ECSBaseURLNotFound` (Base URL not found) - occurs when Hybris is not present for the given locale

38 Fetch Order Detail

This method is used to fetch the details of given orders.

38.1 ECSServiceProvider

```
void fetchOrderDetail(String orderId, ECSCallback<ECSOrderDetail,Exception>
ecsCallback);
```

Parameters

- `orderId` – string value which represents unique id for each order
- `ecsCallback` – instance of `ECSCallback` of type `ECSOrderDetail` and `Exception`

Output

- Success response – `ECSOrderDetail` object which contains details of given order
- Failure response – `Exception` (contains localized error message) and `ECSError` which contains error code and error type.

38.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.fetchOrderDetail(orderNumber, new ECSCallback<ECSOrderDetail,
Exception>() {
    @Override
    public void onResponse(ECSOrderDetail result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

38.3 Errors

Client Errors

- `ECSLocaleNotFound` (Please set the locale) – call `configureECS` or `configureECSToGetConfiguration` to set locale
- `ECSSiteIdNotFound` (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- `ECSBaseURLNotFound` (Base URL not found) - occurs when Hybris is not present for the given locale
- `ECSorderIdNil` (Please provide valid order) – occurs for invalid order id

Server Errors

- ECSOrderIdNil (Please provide valid order) – occurs for invalid order id

39 Fetch Order Detail

This method is used to fetch the details of given orders.

39.1 ECSServiceProvider

```
void fetchOrderDetail(ECSOrderDetail
orderDetail, ECSCallback<ECSOrderDetail,Exception> ecsCallback);
```

Parameters

- orderDetail – object of class ECSOrderDetail
- ecsCallback – instance of ECSCallback of type ECSOrderDetail and Exception

Output

- Success response – ECSOrderDetail object which contains details of given order
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

39.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.fetchOrderDetail(orderDetail, new ECSCallback<ECSOrderDetail,
Exception>() {
    @Override
    public void onResponse(ECSOrderDetail result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

39.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale

- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale
- ECSorderIdNil (Please provide valid order) – occurs for invalid order id

Server Errors

- ECSorderIdNil (Please provide valid order) – occurs for invalid order id

40 Fetch Order Detail

This method is used to fetch the details of given orders.

40.1 ECSServiceProvider

```
void fetchOrderDetail(ECSOrders orders, ECSCallback<ECSOrders, Exception>
ecsCallback);
```

Parameters

- orders – object of class ECSOrders
- ecsCallback – instance of ECSCallback of type ECSOrders and Exception

Output

- Success response – ECSOrders object which contains details of given order
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

40.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.fetchOrderDetail(orders, new ECSCallback<ECSOrders, Exception>() {
    @Override
    public void onResponse(ECSOrders result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

40.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale
- ECSOrderIdNil (Please provide valid order) – occurs for invalid order id

Server Errors

- ECSOrderIdNil (Please provide valid order) – occurs for invalid order id

41 Fetch User Profile

This method is used to fetch the user details.

41.1 ECSServiceProvider

```
void fetchUserProfile (ECSCallback<ECSUserProfile,Exception> ecsCallback);
```

Parameters

- ecsCallback – instance of ECSCallback of type ECSUserProfile and Exception

Output

- Success response – ECSUserProfile object which contains details of user
- Failure response – Exception (contains localized error message) and ECSError which contains error code and error type.

41.2 Sample

```
appInfra = new AppInfra.Builder().build(getApplicationContext());
ECSServices ecsServices = new ECSServices( "your propositionID", appInfra);

ecsServices.fetchUserProfile(new ECSCallback<ECSUserProfile, Exception>() {
    @Override
    public void onResponse(ECSUserProfile result) {

    }

    @Override
    public void onFailure(Exception error, ECSError ecsError) {

    }
});
```

41.3 Errors

Client Errors

- ECSLocaleNotFound (Please set the locale) – call configureECS or configureECSToGetConfiguration to set locale
- ECSSiteIdNotFound (Philips shop is not available for the selected country, only retailer mode is available) – occurs when failed to get config or Hybris not present
- ECSBaseURLNotFound (Base URL not found) - occurs when Hybris is not present for the given locale

Common Errors

- ECSInvalid_grant (We have encountered technical glitch (Invalid Janrain Token). Please re-login) – This error occurs when the Janrain token is invalid
 - ECSInvalid_client (We have encountered technical glitch (Bad client credentials). Please re-login)
 - ECSUnsupported_grant_type (We have encountered technical glitch (Unsupported grant type). Please re-login)
 - ECSOAuthNotCalled (User authorization is required for this action)
- Whenever these error are encountered call hybrisRefreshOAuth or re-login.