Data Services interface spec

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document History** | | | | |
| **Version** | **Date** | **Author** | **Section** | **Changes** |
| 0.1 | 01-10-2016 | Ajay Das | All | Initial draft |
| 1.0 | 07-12-2016 | Ajay Das | All | Release for PI16.5 |
| 1.1 | 20-12-2016 | Ajay Das | All | Added MeasurementGroup |
| 1.2 | 5-3-2018 | Sounavo Dey | 2.4.30 | Add API for GDPR migration |

|  |  |
| --- | --- |
| **Author** | Ajay Das |
| **Approved by** |  |
| **Email Id** | Ajay.das@philips.com |

CONTENTS

1. DEFINITIONS & ABBREVATIONS 3

2. INTRODUCTION 3

2.1 Purpose 3

2.2 Scope 3

2.3 Target Audience 3

2.4 Interfaces 4

2.4.1 Initialize Data Services 4

2.4.2 Initialize Data base Monitors 4

2.4.3 Initialize Synchronisation Monitors 4

2.4.4 Fetch Data from data core since time 5

2.4.5 Fetch All Data from data core 5

2.4.6 Synchronize All Data. 6

2.4.7 Create a moment 6

2.4.8 Create a momentDetail 6

2.4.9 Create a Measurement group 7

2.4.10 Create a Measurement group 7

2.4.11 Create a Measurement group detail 8

2.4.12 Create a Measurement 8

2.4.13 Create a Measurement Detail 9

2.4.14 Save a Moment 9

2.4.15 Update a Moment 9

2.4.16 Fetch a Moment by type 10

2.4.17 Fetch a Moment by id 10

2.4.18 Delete a Moment 11

2.4.19 Delete All Moment 11

2.4.20 Delete All Data 11

2.4.21 Create a Consent 12

2.4.22 Create a Consent Detail 12

2.4.23 Save Consent 13

2.4.24 Update consent 13

2.4.25 Fetch consent 14

2.4.26 Create user characteristics 14

2.4.27 Save user characteristics 14

2.4.28 Update user characteristics 15

2.4.29 Fetch user characteristics 15

2.4.30 Clean Up and Synchronize for GDPR 16

# DEFINITIONS & ABBREVATIONS

|  |  |
| --- | --- |
| API | Application Programming Interface |
| CDP2 | Connected Digital Platforms and Propositions |

# INTRODUCTION

This document provides an overview of public interface APIs for Data Services component in Mobile applications.

## Purpose

The purpose of this document is to elaborate on the public APIs used in Data Services component which helps mobile applications to integrate.

## Scope

This document covers only public APIs used in Data Services. Private APIs and details of classes used is out of scope of this document.

## Target Audience

CDP2 development team.

## Interfaces

### Initialize Data Services

|  |  |
| --- | --- |
| Function definition | public void Initialize(Context context, DataCreator creator, ErrorHandler handler) |
| Class name | DataServices |
| Parameters | Context, DataCreator, ErrorHandler |
| Pre-Condition | Application data base module is initialized. |
| Brief | This the application api to initialize Data services core and backend module. |
| Error scenarios | NA |
| Callback functions | NA |
| Return | NA |
| Type | Synchronous |

### Initialize Data base Monitors

|  |  |
| --- | --- |
| Function definition | public void InitializeMonitors(DBDeletingInterface, DBFetchingInterface, DBSavingInterface, DBUpdatingInterface ) |
| Class name | DataServices |
| Parameters | Data base interface to Delete, fetch, update and save. |
| Pre-Condition | Data services Core and backend modules are initialised. Application data base module is initialized |
| Brief | This is the application api to initialize data base monitor and pass Data base implementation for Add , Delete, Update and Fetch. |
| Error scenarios | NA |
| Callback functions | NA |
| Return | NA |
| Type | Synchronous |

### Initialize Synchronisation Monitors

|  |  |
| --- | --- |
| Function definition | Public void InitializeSyncMonitors(List<DataSender>, List<DataFetcher>) |
| Class name | DataServices |
| Parameters | List of data senders and data fetchers |
| Pre-Condition | Data services Core and backend modules are initialised. |
| Brief | This is the application api to setup the synchronisation monitors for sending and receiving data from data core , Application specific data senders and data fetchers are initialised here. |
| Error scenarios | NA |
| Callback functions | NA |
| Return | NA |
| Type | Synchronous |

### Fetch Data from data core since time

|  |  |
| --- | --- |
| Function definition | public void fetchDataSince(@Nullable final DateTime sinceTimestamp) |
| Class name | DataServices |
| Parameters | DateTime |
| Pre-Condition | Data services core and backend module are initialised.Sync monitors are setup |
| Brief | This is the application api to fetch all data from data core since a specified data and time. |
| Error scenarios | NA |
| Callback functions | onSuccess(ArrayList< Object> data)  onFailure(Exception exception) |
| Return | Void |
| Type | Asynchronous |

### Fetch All Data from data core

|  |  |
| --- | --- |
| Function definition | public void fetchAllData() |
| Class name | DataServices |
| Parameters | NA |
| Pre-Condition | Data services core and backend module are initialised.sync monitor is setup |
| Brief | This is the application api to fetch all the data from data core. |
| Error scenarios | NA |
| Callback functions | onSuccess(ArrayList< Object> data)  onFailure(Exception exception) |
| Return | Void |
|  |  |
| Type | Asynchronous |

### Synchronize All Data.

|  |  |
| --- | --- |
| Function definition | public void Synchronize() |
| Class name | DataServices |
| Parameters | NA |
| Pre-Condition | Data services core and backend module are initialised.sync monitor is setup |
| Brief | This is the application api to synchronize the local data base with data core and viceversa. |
| Error scenarios | NA |
| Callback functions | onSuccess(ArrayList< Object> data)  onFailure(Exception exception) |
| Return | void |
| Type | Asynchronous |

### Create a moment

|  |  |
| --- | --- |
| Function definition | public Moment createMoment(@NonNull final string type) |
| Class name | DataServices |
| Parameters | String - MomentType |
| Pre-Condition | Data services core and backend module are initialised. Monitors are setup, Application data base module is setup. |
| Brief | This the application api to create a moment data. |
| Error scenarios | NA |
| Callback functions | NA |
| Return | Moment |
| Type | Synchronous |

### Create a momentDetail

|  |  |
| --- | --- |
| Function definition | public MomentDetail createMomentDetail(@NonNull final string MomentDetailtype, @NonNull final Moment moment) |
| Class name | DataServices |
| Parameters | String - MomentDetailType , Moment |
| Pre-Condition | Data services core and backend module are initialised. And a moment is created. Application data base module is setup |
| Brief | This is the application api to create a MomentDetail and the moment detail is attached to the moment. |
| Error scenarios | NA |
| Callback functions | NA |
| Return | MomentDetail |
| Type | Synchronous |

### Create a Measurement group

|  |  |
| --- | --- |
| Function definition | public MeasurementGroup createMeasurementGroup(@NonNull final Moment moment) |
| Class name | DataServices |
| Parameters | Moment |
| Pre-Condition | Data services core and backend module are initialised. Monitors are setup, Application data base module is setup. |
| Brief | This is the application api to create a measurement group that is within a moment. |
| Error scenarios | NA |
| Callback functions | NA |
| Return | Measurement group |
| Type | Synchronous |

### Create a Measurement group

|  |  |
| --- | --- |
| Function definition | public MeasurementGroup createMeasurementGroup(@NonNull final MeasurementGroup measurementgroup) |
| Class name | DataServices |
| Parameters | MeasurementGroup |
| Pre-Condition | Data services core and backend module are initialised. Monitors are setup, Application data base module is setup. |
| Brief | This is the application api to create a measurement group that is within a measurement group. |
| Error scenarios | NA |
| Callback functions | NA |
| Return | Measurement group |
| Type | Synchronous |

### Create a Measurement group detail

|  |  |
| --- | --- |
| Function definition | public MeasurementGroupDetail create Measurement group Detail(@NonNull final string MeasurementGroupDetailType, @NonNull final MeasurementGroup measurementgroup) |
| Class name | DataServices |
| Parameters | String - MeasurementGroupDetailType , MeasurementGroup |
| Pre-Condition | Data services core and backend module are initialised. And a measurement group is created. Application data base module is setup |
| Brief | This is the application api to create a measurement group detail and the measurement group detail is attached to the measurement group. |
| Error scenarios | NA |
| Callback functions | NA |
| Return | MomentDetail |
| Type | Synchronous |

### Create a Measurement

|  |  |
| --- | --- |
| Function definition | public Measurement createMeasurement(@NonNull final string MeasurementType, @NonNull final MeasurementGroup measurementgroup) |
| Class name | DataServices |
| Parameters | String - MeasurementType , MeasurementGroup |
| Pre-Condition | Data services core and backend module are initialised. And a measurement group is created. Application data base module is setup |
| Brief | This is the application api to create a Measurement and the measurement is attached to the measurement group. |
| Error scenarios | NA |
| Callback functions | NA |
| Return | Measurement |
| Type | Synchronous |

### Create a Measurement Detail

|  |  |
| --- | --- |
| Function definition | Public  MeasurementDetail createMeasurementDetail(@NonNull final string MeasurementDetailType,  @NonNull final Measurement measurement) |
| Class name | DataServices |
| Parameters | String MeasurementDetailType, Moment |
| Pre-Condition | Data services core and backend module are initialised. And a moment is created. Application data base module is setup. |
| Brief | This is the application api to create a Measurement Detail and the measurement detail is attached to the moment. |
| Error scenarios | NA |
| Callback functions | NA |
| Return | MeasurementDetail |
| Type | Synchronous |

### Save a Moment

|  |  |
| --- | --- |
| Function definition | public void save(@NonNull final Moment moment) |
| Class name | DataServices |
| Parameters | Moment |
| Pre-Condition | Data services core and backend module are initialised. And a moment is created. Application data base module is setup. |
| Brief | This is the application api to save a moment to data base and sync to data core. |
| Error scenarios | NA |
| Callback functions | onSuccess(Object data)  onFailure(Exception exception) |
| Return | void |
| Type | Asynchronous |

### Update a Moment

|  |  |
| --- | --- |
| Function definition | public void update (@NonNull final Moment moment) |
| Class name | DataServices |
| Parameters | Moment |
| Pre-Condition | Data services core and backend module are initialised. And a moment is created. Application data base module is setup. |
| Brief | This is the application api to update a moment in data base and data sync. |
| Error scenarios | NA |
| Callback functions | onSuccess(Object data)  onFailure(Exception exception) |
| Return | void |
| Type | Asynchronous |

### Fetch a Moment by type

|  |  |
| --- | --- |
| Function definition | public void FetchMoment(@NonNull final string MomentType) |
| Class name | DataServices |
| Parameters | String - MomentType |
| Pre-Condition | Data services core and backend module are initialised. Application data base module is setup.And a moment is created in both db and data core |
| Brief | This is the application api to fetch moment data by moment type from data base. |
| Error scenarios | NA |
| Callback functions | onSuccess(ArrayList< Object> data)  onFailure(Exception exception) |
| Return | void |
| Type | Asynchronous |

### Fetch a Moment by id

|  |  |
| --- | --- |
| Function definition | public void fetchMomentById(final int id) |
| Class name | DataServices |
| Parameters | Id |
| Pre-Condition | Data services core and backend module are initialised. Application data base module is setup.And a moment is created in both db and data core |
| Brief | This is the application api to fetch a moment data by id from data base. |
| Error scenarios | NA |
| Callback functions | onSuccess(ArrayList< Object> data)  onFailure(Exception exception) |
| Return | Void |
| Type | Asynchronous |

### Delete a Moment

|  |  |
| --- | --- |
| Function definition | public void deleteMoment(@NonNull final Moment moment) |
| Class name | DataServices |
| Parameters | Moment |
| Pre-Condition | Data services core and backend module are initialised. Application data base module is setup.And a moment is created in both db and data core |
| Brief | This is the application api to delete a moment from data base and data sync. |
| Error scenarios | NA |
| Callback functions | onSuccess(Object data)  onFailure(Exception exception) |
| Return | Void |
| Type | Asynchronous |

### Delete All Moment

|  |  |
| --- | --- |
| Function definition | public void deleteAllMoment(@NonNull final string MomentType) |
| Class name | DataServices |
| Parameters | None |
| Pre-Condition | Data services core and backend module are initialised. And a moment is created in both db and data core |
| Brief | This is the application api to delete all moment data from data base and data sync. |
| Error scenarios | NA |
| Callback functions | onSuccess(Object data)  onFailure(Exception exception) |
| Return | void |
| Type | Asynchronous |

### Delete All Data

|  |  |
| --- | --- |
| Function definition | public void deleteAll () |
| Class name | DataServices |
| Parameters | none |
| Pre-Condition | Data services core and backend module are initialised. And a moment is created in both db and data core |
| Brief | This is the application api to delete all data from data base. |
| Error scenarios | NA |
| Callback functions | onSuccess(Object data)  onFailure(Exception exception) |
| Return | void |
| Type | Asynchronous |

### Create a Consent

|  |  |
| --- | --- |
| Function definition | public Consent createConsent() |
| Class name | DataServices |
| Parameters | NULL |
| Pre-Condition | Data services core and backend module are initialised. Monitors are setup, Application data base module is setup. |
| Brief | This the application api to create a consent data. |
| Error scenarios | NA |
| Callback functions | NA |
| Return | Consent |
| Type | Synchronous |

### Create a Consent Detail

|  |  |
| --- | --- |
| Function definition | public ConsentDetail createConsentDetail((@NonNull final string ConsentDetailType, @NonNull final Consent consent) |
| Class name | DataServices |
| Parameters | String ConsentDetailType , Consent |
| Pre-Condition | Data services core and backend module are initialised. Monitors are setup, Application data base module is setup. |
| Brief | This the application api to create a consent detail. |
| Error scenarios | NA |
| Callback functions | NA |
| Return | ConsentDetail |
| Type | Synchronous |

### Save Consent

|  |  |
| --- | --- |
| Function definition | public void Save(@NonNull final Consent consent) |
| Class name | DataServices |
| Parameters | Consent |
| Pre-Condition | Data services core and backend module are initialised. Monitors are setup, Application data base module is setup. |
| Brief | This the application api to save a consent. |
| Error scenarios | NA |
| Callback functions | NA |
| Return | void |
| Type | Synchronous |

### Update consent

|  |  |
| --- | --- |
| Function definition | public void Update(@NonNull final Consent consent) |
| Class name | DataServices |
| Parameters | Consent |
| Pre-Condition | Data services core and backend module are initialised. Monitors are setup, Application data base module is setup. |
| Brief | This the application api to update consent. |
| Error scenarios | NA |
| Callback functions | NA |
| Return | void |
| Type | Synchronous |

### Fetch consent

|  |  |
| --- | --- |
| Function definition | public void FetchConsent() |
| Class name | DataServices |
| Parameters | NULL |
| Pre-Condition | Data services core and backend module are initialised. Monitors are setup, Application data base module is setup. |
| Brief | This the application api to fetch consent. |
| Error scenarios | NA |
| Callback functions | onSuccess(ArrayList< Object> data)  onFailure(Exception exception) |
| Return | void |
| Type | Asynchronous |

### Create user characteristics

|  |  |
| --- | --- |
| Function definition | public usercharacteristics createUserCharacteristics(@NonNull final usercharacteristics UserCharacteristics) |
| Class name | DataServices |
| Parameters | userCharacteristics |
| Pre-Condition | Data services core and backend module are initialised. Monitors are setup, Application data base module is setup. |
| Brief | This the application api to create a user characteristics. |
| Error scenarios | NA |
| Callback functions | NA |
| Return | usercharacteristics |
| Type | Synchronous |

### Save user characteristics

|  |  |
| --- | --- |
| Function definition | public void Save(@NonNull final usercharacteristics UserCharacteristics) |
| Class name | DataServices |
| Parameters | usercharacteristics |
| Pre-Condition | Data services core and backend module are initialised. Monitors are setup, Application data base module is setup. |
| Brief | This the application api to save a user characteristics. |
| Error scenarios | NA |
| Callback functions | NA |
| Return | void |
| Type | Synchronous |

### Update user characteristics

|  |  |
| --- | --- |
| Function definition | public void Update(@NonNull final usercharacteristics UserCharacteristics) |
| Class name | DataServices |
| Parameters | usercharacteristics |
| Pre-Condition | Data services core and backend module are initialised. Monitors are setup, Application data base module is setup. |
| Brief | This the application api to update usercharacteristics. |
| Error scenarios | NA |
| Callback functions | NA |
| Return | void |
| Type | Synchronous |

### Fetch user characteristics

|  |  |
| --- | --- |
| Function definition | public void FetchUserCharacteristics() |
| Class name | DataServices |
| Parameters | NULL |
| Pre-Condition | Data services core and backend module are initialised. Monitors are setup, Application data base module is setup. |
| Brief | This the application api to fetch user characteristics. |
| Error scenarios | NA |
| Callback functions | onSuccess(ArrayList< Object> data)  onFailure(Exception exception) |
| Return | void |
| Type | Asynchronous |

### Clean Up and Synchronize for GDPR

|  |  |
| --- | --- |
| Function definition | public void migrateGDPR() |
| Class name | DataServices |
| Parameters | NULL |
| Pre-Condition | Data services core and backend module are initialised. Monitors are setup, Application data base module is setup. |
| Brief | This is the application API to do one time clean up and sync to migrate to a GDPR compatible data model. If GDPRMigrationFlag is not set then all synced moments and insights are deleted and GDPRMigration is set. |
| Error scenarios | NA |
| Callback functions | onSuccess(ArrayList< Object> data)  onFailure(Exception exception) |
| Return | void |
| Type | Asynchronous |