iOS Applications

PDF download from SAP Help Portal:
http://help.sap.com/saphelp_smp305sdk/helpdata/en/4d/67660e6bad457886652b7eb36f2c89/content.htm

Created on March 05, 2015

The documentation may have changed since you downloaded the PDF. You can always find the latest information on SAP Help Portal.

Note

This PDF document contains the selected topic and its subtopics (max. 150) in the selected structure. Subtopics from other structures are not included.
The selected structure has more than 150 subtopics. This download contains only the first 150 subtopics. You can manually download the missing subtopics.

Table of content

1 iOS Applications

Develop an OData online, occasionally offline, or offline application.

In this section:

- Creating a Project and Importing the OData Libraries and Bundles
- Developing iOS Online OData Applications
- Developing iOS Offline OData Applications
- OData APIs for iOS
- Usage Collection
- Developing with MAF Logon for iOS
- Supportability for iOS
- Developing with iOS Reusable UI Components
- Developing with the Extensibility Framework
- Enabling Apple Push Notifications (APNS)
- Running and Testing iOS Applications
- Deploying iOS Applications
- Analyzing Logs and Debugging Errors in iOS Devices

Parent topic: Native OData App Development Using the OData API

1.1 Creating a Project and Importing the OData Libraries and Bundles

Set up the iOS development environment by downloading the required plugins and importing the associated iOS libraries into the iOS development environment.

Prerequisites

Install the OData SDK on the Mac development machine.

Context

For more information on Xcode, refer to the Apple Developer Connection: https://developer.apple.com/technologies/tools/ .

Procedure

1. Start Xcode and select Create a new Xcode project.
   You can execute the OData libraries on architecture armv7 and armv7s.
2. Under iOS, select Application.
3. In the right pane, select Empty Application as the project template and click Next.
4. Enter `<ProjectName>` as the Product Name, `<CorpID>` as the Company Identifier, select Universal as the Device Family product, then click Next.
5. Select a location to save the project and click Create to open it.
6. Navigate to the <SMP_HOME>\MobileSDK<vesrion>\NativeSDK\ODataFramework\iOS\ installation directory. Copy includes and libraries of components you would like to add to your project.
7. Select your project from the Project Navigator, and choose your application's target.
8. Open the Build Phases tab, and within Link Binary with Libraries, add all necessary native frameworks.
9. Choose your project, rather than a specific target, and open the Build Settings tab:
   1. In the Other Linker Flags section, add -ObjC and -all_load. If these settings are not visible, change the filter in the Build Settings bar from Basic to All.
   2. In the header search path section adjust the path to look for public header files. Depending on where you copied the header files in previous steps, the path string looks like this: `$(PROJECT_DIR)/includes/public/**`
   3. In the library search path section adjust the path to look for the binary files. Depending on where you copied the library files in previous steps, the path string looks like this: `$(PROJECT_DIR)/libraries/$(BUILD_STYLE)-universal/*`
10. Build and run your project.

Parent topic: iOS Applications

1.2 Developing iOS Online OData Applications

Understand the basic flow of developing a native iOS application that consumes online OData feeds.

1. Registering Devices and Connecting Users
2. Initializing an Online Store
3. Read Operations
4. Perform Create, Update, Delete Operations
5. Logging Operations

Parent topic: iOS Applications

1.2.1 Registering Devices and Connecting Users

Register a device and log on a user using MAF Logon components.

See Developing with MAF Logon for iOS.

Parent topic: Developing iOS Online OData Applications
Next topic: Initializing an Online Store

1.2.2 Initializing an Online Store

Initialize an online OData store for online OData access.

Prerequisites

Define the back-end OData connections required by the application in SAP Mobile Platform Server using the cockpit.

Context

The ODataOnline library interacts extensively with the OData service. One SODataStore instance interacts with one OData source: If a SAP Mobile Platform Server-hosted application is defined with two or more back-end URLs (whitelisted connections), a separate ODataOnlineStore instance is required for each. Pass in the URL as a NSURL, which must be the baseURL of the OData service. The HttpConversationManager is the network layer, which handles the authentication challenges, and can be configured manually or by using the MAFLogonManager. The SODataOnlineStore hides many of the complexities of interacting with the OData source, and internally uses an OData parser.

Procedure

1. Opening an Online Store requires HttpConversationManager, which you configure with the MAF Logon configurator:
   ```
   self.conversationManager = [[HttpConversationManager alloc] init];
   [self.logonUIViewManager.logonManager.logonConfigurator
   configureManager:self.conversationManager];
   ```
2. Create an online SODataStore. There are two overloaded constructors for the SODataStore:
   ```
   -(id) initWithURL:(NSURL*)url httpConversationManager:
   (HttpConversationManager*)httpConversationManager;

   -(id) initWithURL:(NSURL*)url httpConversationManager:
   (HttpConversationManager*)httpConversationManager options:(SODataOnlineStoreOptions*)options;
   ```
   Using the JSON format greatly reduces network traffic:
   ```
   //Use options to configure the store to send the request payload in JSON format
   SODataOnlineStoreOptions* const storeOptions = [[SODataOnlineStoreOptions alloc] init];
       storeOptions.requestFormat = SODataDataFormatJSON;

   onlineStore = [[SODataOnlineStore alloc] initWithURL:url
   httpConversationManager:manager options:storeOptions];
   ```

Parent topic: Developing iOS Online OData Applications
Previous topic: Registering Devices and Connecting Users
Next topic: Read Operations

1.2.3 Read Operations

Perform Read operations on an ODataEntity or an ODataEntitySet from OData or JSON service connections.

Procedure

An ODataEntity or an ODataEntitySet can be read with the following operation:

```
#pragma mark - SODataRequestDelegate implementation
- (void) requestServerResponse:(id<SODataRequestExecution>)requestExecution {
 id<SODataEntitySet> entitySet = (id<SODataEntitySet>)[(id<SODataResponseSingle>)[ requestExecution response] payl
 //Handle success in the app logic, note that it’s not called on the mainThread, so the UI can’t be manipulated fr
}

-(void)requestFailed:(id<SODataRequestExecution>)requestExecution error:(NSError *)error {
    NSString* msg = [NSString stringWithFormat:@"Request failed: %@", error.description];
    [self showAlertWithTitle:@"Error" message:msg];
}
```

In this section:

- Making HTTP Get Requests from Whitelisted Connections

Parent topic: Developing iOS Online OData Applications
Previous topic: Initializing an Online Store
Next topic: Perform Create, Update, Delete Operations

1.2.3.1 Making HTTP Get Requests from Whitelisted Connections

Make HTTP Get requests from either whitelisted OData or JSON service connections.

Prerequisites

Define a whitelisted back-end connection used by the application in SAP Mobile Platform Server using the cockpit.

Procedure

1. If the whitelisted connection is an OData service:
    1. Configure the HttpConversationManager with the MAFLogonManager’s logonConfigurator to respond to authentication challenges or use a CommonAuthenticationConfiguration and register the providers.
    2. Follow the steps Making HTTP Get Requests.
2. If the whitelisted connection is a JSON service:
    1. Configure the HttpConversationManager with the MAFLogonManager’s logonConfigurator to respond to authentication challenges or use a CommonAuthenticationConfiguration and register the providers.
    2. Use HttpConversationManager to get data:

```
-(void) logonFinishedWithError:(NSError*)anError {
    if (!anError) {

         self.conversationManager = [[HttpConversationManager alloc] init];
[self.logonUIViewManager.logonManager.logonConfigurator configureManager:self.conversationManager];
  [self openOnlineStore];
 }
}
```

Or:

```
HttpConversationManager* manager = [[HttpConversationManager alloc] init];
CommonAuthenticationConfigurator* commonConfig = [[CommonAuthenticationConfigurator alloc] init];
[commonConfig configureManager:manager];
self.conversationManager = manager;
```

Download the data:

```
NSMutableURLRequest* req = [NSMutableURLRequest requestWithURL:url];
    [self.httpConversationManager executeRequest:req
completionHandler:^(NSData *data, NSURLResponse *response, NSError *error_in) {
        //Process the downloaded data
    }];
```

Parent topic: Read Operations

1.2.4 Perform Create, Update, Delete Operations

Perform Create, Update, Delete, and other various operations on the OData source.

See the OData APIs for iOS for more information.

In this section:

- Create Operations
- Update Operations
- Delete Operations
- Miscellaneous Operations
- Function Imports

Parent topic: Developing iOS Online OData Applications
Previous topic: Read Operations
Next topic: Logging Operations

1.2.4.1 Create Operations

Create an SODataEntity using the `scheduleCreateEntity:collectionPath:delegate:options:` method on the SODataStore.

The `scheduleCreateEntity` takes an SODataEntity and the collectionName as the parameter. An SODataEntity is created locally and then passed along as a parameter. The SODataPropMode can be combined using the binary operator.

1. Create a local SODataEntity and set values for the properties.
2. Set the SODataPropMode. Options are All, Keys, Mandatory, Optional.
3. Call the `scheduleCreateEntity` method on the SODataStore. Get the response in the proper delegate method.

```
id<SODataEntity> entity = [[SODataEntityDefault alloc] initWithType:@"EntityType"];
NSError* err = nil;
[self.store allocatePropertiesOfEntity:entity mode:SODataPropModeAll error:&err];
id<SODataProperty> prop = [entity.properties objectForKey:@"PropertyName"];
prop.value = @"Value";
prop = [entity.properties objectForKey:@" PropertyName 2"];
prop.value = @"Value2";
[self.store allocateNavigationProperties:entity error:&err];
self.currentRequestExecution = [self.store scheduleCreateEntity:entity collectionPath:@"CollectionPath" delega
```

Parent topic: Perform Create, Update, Delete Operations

1.2.4.2 Update Operations

Update a SODataEntity using the `scheduleUpdateEntity:delegate:options:` method on the SODataOnlineStore.

Use the `Copy` method to make a copy (as opposed to a reference copy) of an entity. Make the necessary updates on the copy of the SODataEntity. This ensures the original entity remains intact and the app maintains a consistent state, even if the update operation fails (network disconnects, server error, and so on).

```
[store scheduleUpdateEntity:entity delegate:self options:nil];
```

Parent topic: Perform Create, Update, Delete Operations

1.2.4.3 Delete Operations

Delete an SODataEntity using the `scheduleDeleteEntity:delegate:options:` method on the SODataOnlineStore.

```
[store scheduleDeleteEntity:entity delegate:self options:nil];
```

Parent topic: Perform Create, Update, Delete Operations

1.2.4.4 Miscellaneous Operations

There are a number of miscellaneous operations that can be called on an entity. See the OData API for details.

```
- (id<SODataRequestExecution>) scheduleRequest:(id<SODataRequestParam>)request delegate:(id<SODataRequestDelegate>
- (id<SODataRequestExecution>) scheduleReadEntitySet:(NSString*)resourcePath delegate:(id<SODataRequestDelegate>)de
- (id<SODataRequestExecution>) scheduleReadEntityWithResourcePath:(NSString*)resourcePath delegate:(id<SODataReque
- (id<SODataRequestExecution>) scheduleReadEntity:(id<SODataEntity>)entity delegate:(id<SODataRequestDelegate>)dele
- (id<SODataRequestExecution>) scheduleCreateEntity:(id<SODataEntity>)entity collectionPath:(NSString*)collectionPa
- (id<SODataRequestExecution>) scheduleUpdateEntity:(id<SODataEntity>)entity delegate:(id<SODataRequestDelegate>)de
- (id<SODataRequestExecution>) schedulePatchEntity:(id<SODataEntity>)entity delegate:(id<SODataRequestDelegate>)del
- (id<SODataRequestExecution>) scheduleDeleteEntity:(id<SODataEntity>)entity delegate:(id<SODataRequestDelegate>)de
- (id<SODataRequestExecution>) scheduleDeleteEntity:(NSString*)resourcePath etag:(NSString*)etag delegate:(id<SODat
- (id<SODataRequestExecution>) scheduleReadLinkSet:(NSString*)resourcePath delegate:(id<SODataRequestDelegate>)dele
- (id<SODataRequestExecution>) scheduleReadLink:(NSString*)resourcePath delegate:(id<SODataRequestDelegate>)delega
- (id<SODataRequestExecution>) scheduleReadPropertyPrimitive:(NSString*)resourcePath delegate:(id<SODataRequestDele
- (id<SODataRequestExecution>) scheduleReadPropertyComplex:(NSString*)resourcePath delegate:(id<SODataRequestDelega
- (id<SODataRequestExecution>) scheduleReadPropertyRaw:(NSString*)resourcePath delegate:(id<SODataRequestDelegate>
- (id<SODataRequestExecution>) scheduleFunction:(NSString*)resourcePath delegate:(id<SODataRequestDelegate>)delega
```

Parent topic: Perform Create, Update, Delete Operations

1.2.4.5 Function Imports

Function Imports provide custom functionality, and are not tied to any collection. Prior knowledge of the return type of the function import is useful.

```
[store scheduleFunction:@"FunctionImportName?filterCriteria=4" delegate:self options:nil];
```

Parent topic: Perform Create, Update, Delete Operations

1.2.5 Logging Operations

Log various application events, by specifying a log level, then upload the client log to SAP Mobile Platform Server for analysis.

Prerequisites

Define client log and trace policies in SAP Mobile Platform Server using the cockpit. See Managing Application Logs and Traces.

Context

See Supportability for iOS for detailed information about logging and tracing.

Procedure

1. Initialize the logger, define a log level setting, and log messages:

   ```
   // Initializing logManager
   _logger = [[SAPSupportabilityFacade sharedManager] getClientLogger:@"com.sap.logging.all"];

   // Logging a message
   [_logger logInfo:@"Log something”];
   ```

2. Retrieve logs:

   ```
   NSArray* logData = [[[SAPSupportabilityFacade sharedManager] getClientLogManager]
   getLogEntriesForLogger: @"com.sap.logging.all" withLevel:AllClientLogLevel];
   ```

3. Upload the client log to SAP Mobile Platform Server:

   ```
   SupportabilityUploader* uploader = [[SupportabilityUploader alloc]
   initWithHttpConversationManager:self.conversationManager urlRequest:request];
   [[[SAPSupportabilityFacade sharedManager] getClientLogManager] uploadClientLogs:uploader completion:^(NSError*
           if (error == nil) {
               NSLog(@"upload succeeded");
           } else {
               NSLog(@"upload failed: %@", [error description]);
       }
       }];
   ```

Parent topic: Developing iOS Online OData Applications

Previous topic: Perform Create, Update, Delete Operations

1.3 Developing iOS Offline OData Applications

Understand how to develop a native iOS application that uses an offline OData store.

The basic flow of developing a native iOS application that consumes online OData versus offline OData is the same: registering devices and connecting users, performing create, update, delete operations on the entity uses the same APIs. The main differences involve initializing, populating, and updating an offline OData store versus an online OData store, flushing changes, refreshing the offline store, conflict detection, and so on. This section describes those differences when developing an offline OData application.

1. Defining an Application Configuration File with Defining Requests
2. Querying and Modifying Offline Data
3. Updating and Refreshing Offline Data
4. Designing an Offline Application to Handle Conflicts and Errors
5. Additional Offline OData Features
6. Sample Metadata for Offline OData

Parent topic: iOS Applications

1.3.1 Defining an Application Configuration File with Defining Requests

Define offline application settings in a configuration file and configure SAP Mobile Platform Server based on those settings.

Procedure

1. Create an application configuration initialization (.ini) file that includes the defining requests, and other application-specific information.
2. Send the application configuration initialization file to the SAP Mobile Platform Server Administrator.
   The Administrator imports the .ini file and configures various offline settings to support the offline application. See Defining Applications and Defining Offline Application Settings in the cockpit Online Help.

In this section:

- Defining Requests
- Performance Improvement with Caching and Delta Tracking
- Application Configuration File

Parent topic: Developing iOS Offline OData Applications
Next topic: Querying and Modifying Offline Data
Parent topic: Developing Android Offline OData Applications
Next topic: Querying and Modifying Offline Data

1.3.1.1 Defining Requests

A defining request is an OData read request that targets the OData endpoint associated with the offline store and retrieves a subset of the OData endpoint data. Multiple defining requests can be defined for each OData endpoint.

Defining requests are subset of data from the OData Producer that is sent to the client, either during initialization of the offline store or during a refresh.

Set up defining requests using the ODataOfflineStoreOptions object of the OData Offline API. The application populates the “definingRequests” parameter. You can then reference each request in the application configuration file, as shown below:

```
[defining_request]
name=DefiningRequestName
is_shared_data=N|Y
refresh_interval=NNN
track_deltas=AUTO|ALWAYS|NEVER
delta_token_lifetime=NNN
```

Keep the following in mind when writing defining requests:

- Currently, only defining requests where the resource path identifies an entity set or entity type instance are supported.
- The request name for shared requests must be unique and be the same on both the client and server.
- If you are using a mix of delta-enabled and nondelta-enabled defining requests, and you are using referential constraints to build relationships, the dependent entity in a relationship cannot be in a delta-enabled defining request if the principal entity is in a nondelta-enabled relationship.
- If you are not using deltas, you can use the same entity type in more than one defining request, but you cannot perform a selective refresh of one of those defining requests without the other.
- If you are using a mix of delta-enabled and nondelta-enabled requests, you cannot use the same entity type in both a delta-enabled and a nondelta-enabled defining request, either directly referenced in the resource path or through an $expand.

Using Defining Requests to Build Entity Relationships

To build relationships in your application you can use either $expand queries or referential constraints.

Referential constraints

If you are tracking deltas using the OData Producer and require entity relationships in the offline store, you cannot use $expand queries but instead must have associations with a referential constraint. Ensure that each defining request references a single entity set. The server detects the associations with referential constraints and uses these referential constraints to build the relationships between the types selected in your defining requests.

For example, the sample metadata document has an association between CustomerType and OrderType, called CustomerType_OrderType, which defines a referential constraint. If you specify two defining requests, ~/Customers and ~/Orders, the server uses the referential constraint defined in the association CustomerType_OrderType to build the relationship between the Customers and Orders entity sets in the offline store.

$expand queries

Use $expand queries to build relationships between entity sets only if you are not tracking deltas, or if you are tracking deltas using the server. You cannot use $expand queries if you are tracking deltas using the OData Producer.

Parent topic: Defining an Application Configuration File with Defining Requests

1.3.1.2 Performance Improvement with Caching and Delta Tracking

Use delta query protocol and data caching to significantly improve the performance of your SAP Mobile Platform Server.

Delta Tracking

OData delta change tracking is a method by which a client obtains changes from a data store without downloading the entire data store. Set up your application to track deltas in your application configuration file when you configure your defining requests. If your OData Producer supports delta query protocol to track changes, the provided delta links are automatically used to track changes and you need not configure delta tracking. When possible, use an OData Producer that supports delta query protocol.

When to use delta tracking

Delta tracking sends only changed data to the client, rather than the entire data set. This significantly decreases the load on the back end and is especially useful when performance is critical.

- If your OData Producer does not support delta change tracking, the default delta tracking setting in the application configuration file tracks deltas using SAP Mobile Platform Server. This improves performance on the client, while tracking deltas using the OData Producer improves performance over the entire network.
- If your OData Producer has limited support for delta change tracking, for example, deltas are configured to expire after a set period of time, using the **`always`** setting for delta tracking in the application configuration file helps replace missing delta tracking functionality.

When not to use delta tracking

If all or most of your data is continuously changing so that you expect to be downloading the full payload during every refresh, enabling delta tracking does not improve performance. For example, if you download the temperature value for sensors in 1000 offices, using delta tracking does not decrease the amount of data downloaded during a refresh and turning off delta tracking may increase performance.

Caching

To minimize the load on the OData Producer and possibly improve performance, cache data that is shared by numerous users on SAP Platform Mobile Server. For example, cache data from a query that returns the same results to multiple users. Do not cache user-specific data. For example, do not cache data from a query that returns different results to each user. Set up your application to cache data when you configure your defining requests in your application configuration file.

Parent topic: Defining an Application Configuration File with Defining Requests

1.3.1.3 Application Configuration File

Use the application configuration file to configure OData endpoints and defining requests that determine the data that populates the application database when it is created. Use the cockpit to import the application configuration file.

An application configuration file consists of one or more OData endpoints, each of which can be optionally followed by one or more defining requests.

Each application configuration file represents a single application.All settings have default values, so an application configuration file is only required for non-default behavior.

Endpoints are defined as follows:

```
[endpoint]
name=YourConnectionName
prepopulate_offline_db=Y|N|SHARED-ONLY
prepopulate_offline_db_interval=NNN
indexed_type=<namespace>.<entity type name1>: Property1A [ASC|DESC][, Property1B [ASC|DESC], …]
indexed_type=<namespace>.<entity type name2>: Property2A [ASC|DESC][, Property2B [ASC|DESC], …]
request_format=application/json;q=1,application/atom+xml;q=0.5
delta_request_format=application/atom+xml
```

Parameter	Value	Notes
name	Name	(Required) The name of the endpoint, defined as a **`Connection Name`** in the cockpit.
prepopulate_offline_db	Y|N|SHARED-ONLY	Specifies the type of data that is included in the initial database. Y Shared and unshared data is included in the client database when it is created on the server. The fully populated database, as defined by the defining requests, is deployed to the client. This is the default value. N Creates an empty database with only a schema. To obtain data, the database is refreshed automatically after it is deployed to the client. Shared-only The database includes only shared data and no user-specific data. To obtain client-specific data, the database is refreshed automatically after it is deployed to the client.
prepopulate_offline_db_interval	NNN	If a refresh requires a new database, the shared data is refreshed based on the interval, in minutes, specified. This option applies only when prepopulate_offline_db is set to **`Y`** or **`SHARED-ONLY`** and there is at least one defining request with **`is_shared_data=Y`**.
indexed_type	<namespace>.<entity type name>:	Each indexed_type defined causes the client database to have an index for that type. Defining indexes can improve client query performance. However, defining too many

	PropertyName [ASC|DESC][, AnotherPropertyName [ASC|DESC], …]	indexes can slow the performance of refreshes and updates. Specify multiple indexed_type entries by listing one entry per line. Specify either the primitive property name, or a complex type path expression that ends with a primitive property.
request_format	atom|json|any valid value	Specifies the format to use when communicating with the OData producer. This property applies only when the offline OData service is making initial requests or non-delta request to the OData producer. atom Short for application/atom+xml. json Short for application/json. Any valid value Specify any valid value for an Accept header. The default is: `application/json;q=1,application/atom+xml;q=0.5`
delta_request_format	atom|json|any valid value	Specifies the format to use when communicating with the OData producer. This property applies only when the offline OData service is making delta requests to the OData producer. atom Short for application/atom+xml. json Short for application/json. Any valid value Specify any valid value for an Accept header. The default is: `application/atom+xml`

Create defining requests as follows:

```
[defining_request]
name=DefiningRequestName
is_shared_data=N|Y
refresh_interval=NNN
track_deltas=AUTO|ALWAYS|NEVER
delta_token_lifetime=NNN
```

Currently, only defining requests where the resource path identifies an entity set or entity type instance are supported.

Parameter	Value	Notes
name	DefiningRequestName	(Required) The name of the defining request. A defining request is an OData read request that targets the OData endpoint associated with the offline store, and retrieves a subset of the OData endpoint data. Defining requests are defined in the ODataOfflineStoreOptions object of the OData Offline API. The application populates the "definingRequests" parameter.
is_shared_data	N|Y	Specifies whether the data is shared among different clients or users. N At least some data is not shared amongst multiple client databases. This is the default value. Y All data in the request is shared among multiple client databases.
refresh_interval	NNN	The interval time, in minutes, between refreshes of the shared data.
track_deltas	AUTO|ALWAYS|NEVER	Specifies how to track OData deltas. Auto If the OData producer supports delta change tracking on the defining request, the delta links returned are used when refreshing the data. If the OData producer does not support delta change tracking on the defining request, the server tracks the changes so that only changed data is exchanged between the client and the server. This is the default value. Never Sets the server to never track delta changes. If the OData producer supports delta change tracking, the delta links returned by the OData producer are used when refreshing data, even if this option is specified. Always Sets the server to always track delta changes. This option is equivalent to Auto if the OData producer does not support delta change tracking on the defining request.
delta_token_lifetime	NNN	The time, in minutes, until the OData delta token expires. While a token is valid, only changed data is refreshed on the client database. When the delta token expires, the entire client database is replaced.

Example

```
[endpoint]
Name=OrderExample
prepopulate_offline_db=SHARED-ONLY
prepopulate_offline_db_interval=1440
indexed_type=SAPODataExample.CustomerType: Surname, GivenName
indexed_type=SAPODataExample.CustomerType: Address/Province, Address/City

[defining_request]
name=Customers
is_shared_data=N

[defining_request]
name=Orders
is_shared_data=N

[defining_request]
name=OrderItems
is_shared_data=N

[defining_request]
name=Products
is_shared_data=Y
refresh_interval=1440
```

Parent topic: Defining an Application Configuration File with Defining Requests

1.3.2 Querying and Modifying Offline Data

Perform offline OData operations using the OData API.

Procedure

Perform Create, Update, Delete, and various other operations on the offline OData store using the same OData API used in an Online application, accept as noted in the Offline Odata API.

Parent topic: Developing iOS Offline OData Applications
Previous topic: Defining an Application Configuration File with Defining Requests
Next topic: Updating and Refreshing Offline Data
Parent topic: Developing Android Offline OData Applications
Previous topic: Defining an Application Configuration File with Defining Requests
Next topic: Updating and Refreshing Offline Data

1.3.3 Updating and Refreshing Offline Data

Refresh the offline store as needed, and flush any changes to the back-end as required.

In this section:

- Offline OData Refreshes and Updates

Parent topic: Developing iOS Offline OData Applications
Previous topic: Querying and Modifying Offline Data
Next topic: Designing an Offline Application to Handle Conflicts and Errors
Parent topic: Developing Android Offline OData Applications
Previous topic: Querying and Modifying Offline Data
Next topic: Designing an Offline Application to Handle Conflicts and Errors

1.3.3.1 Offline OData Refreshes and Updates

Offline OData applications use a local, **offline store** (on-device copy of the enterprise data) to ensure that data is always accessible, even when the mobile device is offline.

Updating the offline store

Applications can make changes to data when the mobile device is offline. Those changes are made to the local offline store, then stored in a request queue. Updates that are queued but have not yet been sent to the back end are reflected in the local offline store. The application developer must ensure that the offline store is periodically updated from the back end when the mobile device is online. This periodic update is called a **refresh**.

Updating the back-end

When the mobile device is online, a request queue **flush** can be performed. Flushing the request queue sends up changes that have been made to the offline store but have not yet been made to the back end. Once a request is successfully received by the back end, the request is deleted from the queue, the change is made to the back end, and a new version of that object is sent to the offline store the next time the application performs a refresh.

Note

Refreshes and request queue flushes are not performed automatically. The offline application must explicitly call the relevant Harmonized API methods to perform a refresh or request queue flush.

Schema upgrades

A schema upgrade of the offline store is performed whenever a new version of the OData service is provided.

Parent topic: Updating and Refreshing Offline Data

1.3.4 Designing an Offline Application to Handle Conflicts and Errors

Use Etags, capture and handle NSError events, view the Error Archive, and use logging and tracing for application debugging when designing and testing your application.

Context

See Supportability for your platform for detailed information about logging and tracing.

In this section:

- Offline OData Conflicts and Errors
- Configuring an Offline Application to Access the ErrorArchive

Parent topic: Developing iOS Offline OData Applications
Previous topic: Updating and Refreshing Offline Data
Next topic: Additional Offline OData Features
Parent topic: Developing Android Offline OData Applications
Previous topic: Updating and Refreshing Offline Data
Next topic: Additional Offline OData Features

1.3.4.1 Offline OData Conflicts and Errors

Design your offline OData application to handle any conflicts and errors. You can provide detailed information about errors via the ErrorArchive.

Offline OData Errors

Offline OData errors fall in the following categories:

Network errors

Occur when communication with the server is interrupted during a request flush or a server refresh.

If you have an Android application, a network error causes an exception; if you have an iOS application, a network error causes an NSError.

You can generally resolve network errors by retrying the operation at a later time. However, if network errors occur while you are configuring your application, ensure that you have specified the correct application configuration settings.

Contract violation errors

Occur when the application attempts to perform an invalid operation, such as sending a malformed request or attempting to call API methods without first calling the openStore method.

If you have an Android application, a contract violation error causes an exception; if you have an iOS application, a contract violation error causes an NSError.

Business logic failures

Occur if business logic at the OData Producer prevents an update that is allowed at the offline store. The error is recorded in the ErrorArchive, which is returned to the client device as an OData feed. The application can obtain the body of the original request and allow the user to re-submit the operation. The application receives notification of these errors through the Request Failure Listener (Android), or the Request Failure Delegate (iOS).

Offline OData Conflicts

Conflicts are specific types of errors that arise when multiple clients simultaneously modify the same data in the back end. Your application should include a process to either avoid conflicts, or to log them when they arise.

OData services use Etags to identify conflicts. However, applications must implement their own conflict resolution logic.

An ETag is an identifier that is assigned to a specific version of an entity. When an entity is modified, a new ETag may be assigned.

To use ETags to detect conflicts in your application:

1. The application retrieves an entity, along with its ETag, from the offline store.
2. When an update request is made, the application sends the request, along with the original ETag, to the offline store.
3. If the ETag that is provided with the update request matches what is currently in the offline store, the update is performed locally and a new ETag is generated. The request and the ETag given back to the application by the offline store are added to the request queue. If the ETag does not match, an error is generated.
4. When a request queue flush is performed, the request and the ETag are sent to the OData Producer. If the ETag matches the ETag in the OData Producer, the update is performed and the updated entity plus a new ETag are retrieved by the application during the next refresh. If the ETag does not match, the error is put in the ErrorArchive and the application then determines how to proceed.

Parent topic: Designing an Offline Application to Handle Conflicts and Errors

1.3.4.2 Configuring an Offline Application to Access the ErrorArchive

The ErrorArchive logs business logic errors and ETag mismatch errors. You can configure an offline application to access the ErrorArchive so that end users can read and delete logged errors.

The ErrorArchive is exposed to the application as an OData feed and is accessible through the OData Offline API in the same way that the application accesses any other information from the offline store.

The following Java example configures the application to access the ErrorArchive:

```
// query the first error
ODataResponseSingle response = store.executeReadEntitySet( “/ErrorArchive”, null );
ODataEntitySet entities = (ODataEntitySet)response.getPayload();
```

```
ODataEntity firstError = entities.getEntities().get(0);
System.out.println( entity.getProperties().get( “Message” );
```

The following Java example configures the application to delete an error:

```
// delete the error
store.executeDeleteEntity( firstError, null );
```

Parent topic: Designing an Offline Application to Handle Conflicts and Errors

1.3.5 Additional Offline OData Features

There are some additional offline OData features.

sap.entityexists

A built-in function that can be used in a $filter system query option to check for the existence of a related entity.

sap.entityexists takes a single parameter that is a member expression that only references navigation properties with an end cardinality of 1 or 0..1.

In the following example, each Order has a relationship to a SalesRep, and each Order either has an assigned SalesRep or it does not.

To get all Orders that have an associated SalesRep:

```
Orders?$filter=sap.entityexists( SalesRep )
```

To get all Orders that do not have an associated SalesRep:

```
Orders?$filter=not sap.entityexists( SalesRep )
```

Prefer header

Only **max page size** can be specified for the Prefer header:

```
odata.maxpagesize=<number greater than 0>
```

Parent topic: Developing iOS Offline OData Applications
Previous topic: Designing an Offline Application to Handle Conflicts and Errors
Next topic: Sample Metadata for Offline OData
Parent topic: Developing Android Offline OData Applications
Previous topic: Designing an Offline Application to Handle Conflicts and Errors
Next topic: Sample Metadata for Offline OData

1.3.6 Sample Metadata for Offline OData

The sample offline OData metadata is referred to throughout the documentation.

```
<?xml version="1.0" encoding="utf-8" standalone="no" ?>
<edmx:Edmx xmlns:edmx="http://schemas.microsoft.com/ado/2007/06/edmx" Version="1.0">
  <edmx:DataServices xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" m:DataServiceVersion=
    <Schema xmlns="http://schemas.microsoft.com/ado/2008/09/edm" Namespace="SAPODataExample" xmlns:d="http://schema
      <EntityType Name="CustomerType">
        <Key>
          <PropertyRef Name="ID"/>
        </Key>
        <Property Name="ID" Nullable="false" Type="Edm.Int32"/>
        <Property Name="Surname" MaxLength="20" Nullable="false" Type="Edm.String"/>
        <Property Name="GivenName" MaxLength="20" Nullable="false" Type="Edm.String"/>
        <Property Name="ShippingAddress" Type="SAPOData.AddressType" Nullable="true"/>
        <Property Name="BillingAddress" Type="SAPOData.AddressType" Nullable="true"/>
        <Property Name="CompanyName" MaxLength="32" Nullable="true" Type="Edm.String"/>
        <NavigationProperty Name="Orders" FromRole="CustomerRole" Relationship="SAPOData.CustomerType_OrderType" To
      </EntityType>
      <EntityType Name="OrderType">
        <Key>
          <PropertyRef Name="ID"/>
        </Key>
        <Property Name="ID" Nullable="false" Type="Edm.Int32"/>
        <Property Name="CustomerID" Nullable="false" Type="Edm.Int32"/>
        <Property Name="OrderDate" Nullable="false" Precision="6" Type="Edm.DateTime"/>
        <Property Name="FinancialCode" MaxLength="2" Nullable="true" Type="Edm.String"/>
        <Property Name="Region" MaxLength="7" Nullable="true" Type="Edm.String"/>
        <Property Name="SalesRepresentative" Nullable="false" Type="Edm.Int32"/>
        <NavigationProperty Name="Customer" FromRole="OrderRole" Relationship="SAPOData.CustomerType_OrderType" To
        <NavigationProperty Name="OrderItems" FromRole="OrderRole" Relationship="SAPOData.OrderType_OrderItemType"
      </EntityType>
      <EntityType Name="OrderItemType">
        <Key>
          <PropertyRef Name="ID"/>
          <PropertyRef Name="LineID"/>
        </Key>
        <Property Name="ID" Nullable="false" Type="Edm.Int32"/>
        <Property Name="LineID" Nullable="false" Type="Edm.Int16"/>
        <Property Name="ProductID" Nullable="false" Type="Edm.Int32"/>
        <Property Name="Quantity" Nullable="false" Type="Edm.Int32"/>
        <Property Name="ShipDate" Nullable="false" Precision="6" Type="Edm.DateTime"/>
        <NavigationProperty Name="Order" FromRole="OrderItemRole" Relationship="SAPOData.OrderType_OrderItemType"
        <NavigationProperty Name="Product" FromRole="OrderItemRole" Relationship="SAPOData.ProductType_OrderItemTyp
      </EntityType>
      <EntityType Name="ProductType">
        <Key>
          <PropertyRef Name="ID"/>
        </Key>
        <Property Name="ID" Nullable="false" Type="Edm.Int32"/>
        <Property Name="Name" MaxLength="15" Nullable="false" Type="Edm.String"/>
        <Property Name="Description" MaxLength="30" Nullable="false" Type="Edm.String"/>
        <Property Name="Size" MaxLength="18" Nullable="false" Type="Edm.String"/>
        <Property Name="Color" MaxLength="18" Nullable="false" Type="Edm.String"/>
        <Property Name="Quantity" Nullable="false" Type="Edm.Int32"/>
        <Property Name="UnitPrice" Nullable="false" Precision="15" Scale="2" Type="Edm.Decimal"/>
        <Property Name="Photo" MaxLength="2147483647" Nullable="true" Type="Edm.Binary"/>
        <NavigationProperty Name="OrderItems" FromRole="ProductRole" Relationship="SAPOData.ProductType_OrderItemTy
      </EntityType>
      <ComplexType Name="AddressType">
        <Property Name="Unit" Type="Edm.Int32" Nullable="true"/>
        <Property Name="Line1" Type="Edm.String" Nullable="true" MaxLength="128" />
        <Property Name="Line2" Type="Edm.String" Nullable="true" MaxLength="128" />
        <Property Name="City" Type="Edm.String" Nullable="true" MaxLength="128" />
        <Property Name="Province" Type="Edm.String" Nullable="true" MaxLength="32" />
        <Property Name="PostalCode" Type="Edm.String" Nullable="true" MaxLength="6" />
      </ComplexType>
      <Association Name="CustomerType_OrderType">
        <End Multiplicity="*" Role="OrderRole" Type="SAPOData.OrderType"/>
        <End Multiplicity="1" Role="CustomerRole" Type="SAPOData.CustomerType"/>
        <ReferentialConstraint>
          <Principal Role="CustomerRole">
            <PropertyRef Name="ID"/>
          </Principal>
          <Dependent Role="OrderRole">
            <PropertyRef Name="CustomerID"/>
          </Dependent>
        </ReferentialConstraint>
      </Association>
      <Association Name="ProductType_OrderItemType">
        <End Multiplicity="*" Role="OrderItemRole" Type="SAPOData.OrderItemType"/>
        <End Multiplicity="1" Role="ProductRole" Type="SAPOData.ProductType"/>
        <ReferentialConstraint>
          <Principal Role="ProductRole">
            <PropertyRef Name="ID"/>
          </Principal>
          <Dependent Role="OrderItemRole">
            <PropertyRef Name="ProductID"/>
          </Dependent>
        </ReferentialConstraint>
      </Association>
      <Association Name="OrderType_OrderItemType">
        <End Multiplicity="*" Role="OrderItemRole" Type="SAPOData.OrderItemType"/>
        <End Multiplicity="1" Role="OrderRole" Type="SAPOData.OrderType">
          <OnDelete Action="Cascade"/>
        </End>
        <ReferentialConstraint>
          <Principal Role="OrderRole">
            <PropertyRef Name="ID"/>
          </Principal>
          <Dependent Role="OrderItemRole">
            <PropertyRef Name="ID"/>
          </Dependent>
        </ReferentialConstraint>
      </Association>
      <EntityContainer Name="SAPOData_Container" m:IsDefaultEntityContainer="true">
        <EntitySet Name="Customers" EntityType="SAPOData.CustomerType"/>
        <EntitySet Name="Orders" EntityType="SAPOData.OrderType"/>
        <EntitySet Name="OrderItems" EntityType="SAPOData.OrderItemType"/>
        <EntitySet Name="Products" EntityType="SAPOData.ProductType"/>
        <AssociationSet Name="Customers_Orders" Association="SAPOData.CustomerType_OrderType">
          <End EntitySet="Orders" Role="OrderRole"/>
          <End EntitySet="Customers" Role="CustomerRole"/>
        </AssociationSet>
        <AssociationSet Name="Products_OrderItems" Association="SAPOData.ProductType_OrderItemType">
          <End EntitySet="OrderItems" Role="OrderItemRole"/>
          <End EntitySet="Products" Role="ProductRole"/>
        </AssociationSet>
        <AssociationSet Name="Orders_OrderItems" Association="SAPOData.OrderType_OrderItemType">
          <End EntitySet="OrderItems" Role="OrderItemRole"/>
          <End EntitySet="Orders" Role="OrderRole"/>
        </AssociationSet>
      </EntityContainer>
    </Schema>
  </edmx:DataServices>
</edmx:Edmx>
```

Parent topic: Developing iOS Offline OData Applications
Previous topic: Additional Offline OData Features
Parent topic: Developing Android Offline OData Applications
Previous topic: Additional Offline OData Features

1.4 OData APIs for iOS

The SAP Mobile Platform SDK for iOS provides three APIs for accessing and manipulating an OData source.

In this section:

- OData Online API for iOS
- OData Offline API for iOS
- OData API for iOS

Parent topic: iOS Applications

1.4.1 OData Online API for iOS

Use the OData Online Store API to manage an Online OData store (open, close, refresh, and so on).

In this section:

- SAP Mobile Platform 3.0 OData SDK for iOS
- Deprecated API List

Parent topic: OData APIs for iOS

1.4.1.1 SAP Mobile Platform 3.0 OData SDK for iOS

Use the OData SDK API reference as the primary source for API usage and error code information.

Refer to the OData SDK API reference for each available package.

In this section:

- SODataCertificateCredentials class
- SODataConstants class
- SODataConverter class
- SODataCredentials protocol
- SODataOnlineStore class
- SODataOnlineStoreDelegate protocol
- SODataOnlineStoreOptions class
- SODataRequestCredentialsDelegate protocol
- SODataRequestExecutedDelegate protocol
- SODataRequestExecutionBatchDefault class
- SODataRequestExecutionDefault class
- SODataRequestExecutionSingleDefault class
- SODataUserCredentials class
- SODataUtility class
- XS_DataContextDefault class
- XS_DeferredValueDefault class

Parent topic: OData Online API for iOS

1.4.1.1.1 SODataCertificateCredentials class

Delegate for certificate credentials.

Syntax

@interface `SODataCertificateCredentials` : `NSObject` <`SODataCredentials`>

Parent topic: SAP Mobile Platform 3.0 OData SDK for iOS

Related Information

SODataCredentials protocol

1.4.1.1.2 SODataConstants class

Syntax

@interface `SODataConstants` : `NSObject`

Parent topic: SAP Mobile Platform 3.0 OData SDK for iOS

1.4.1.1.3 SODataConverter class

Syntax

@interface `SODataConverter` : `NSObject`

Members

All members of SODataConverter, including inherited members.

Methods

Method	Description
+ (NSDictionary *) annotationsDictFromXSAnnotationMap: (XS_AnnotationMap *) annotationMap	
+ (NSMutableDictionary *) complexPropertiesDictFromXSComplexTypeMap: (XS_ComplexTypeMap *) xsComplexTypes	
- (BOOL) convertPropertiesOfSODataEntity: (SODataPropertyDefault *) property xsEntity: (XS_EntityValue *) xsEntity error: (NSError **) error	
- (BOOL) convertPropertiesOfXSEntity: (XS_EntityValue *) xsEntity toEntityProperties: (NSMutableDictionary *) entityProperties error: (NSError **) error	
- (id< SODataProperty >) convertXSPropertyDataValue: (XS_DataValue *) propValue withPropertyInfo: (XS_PropertyInfo *) propInfo error: (NSError **) error	
- (NSDateComponents *) dateComponentsFromXSGlobalDateTime: (XS_GlobalDateTime *) globalDateTime	
- (NSDate *) dateFromXSDateTime: (XS_LocalDateTime *) xsDateTime	
- (id< SODataEntity >) entityFromString: (NSString *) entityStr entitySet: (XS_EntitySet *) entitySet error: (NSError **) error	
- (id< SODataEntity >) entityFromXSEntity: (XS_EntityValue *) xsEntity context: (XS_DataContext *) dataContext error: (NSError **) error	
- (XS_EntitySet *) entitySetFromResourcePath: (NSString *) resourcePath propertyInfo: (XS_PropertyInfo **) propInfo_in_out functionImport: (XS_DataMethod **) functionImport_in_out requestMethod: (NSString *) requestMethod_in responseType: (NSNumber **) responseType_in_out	
- (id< SODataEntitySet >) entitySetFromString: (NSString *) entitySetStr entitySet: (XS_EntitySet *) entitySet error: (NSError **) error	
- (id< SODataEntitySet >) entitySetFromXSEntityList: (XS_EntityValueList *) xsEntityList context: (XS_DataContext *) dataContext error: (NSError **) error	
- (id< SODataError >) errorFromString: (NSString *) str error: (NSError **) error	
+ (NSMutableDictionary *) facetDictFromXSParam: (XS_Parameter *) xsParam	
+ (NSMutableDictionary *) facetDictFromXSPropertyInfo: (XS_PropertyInfo *) xsPropInfo	
- (id) initWithUrl: (NSString *) baseURL_in csdl: (XS_CsdlDocument *) csdl_in	
- (id< SODataLink >) linkFromString: (NSString *) str error: (NSError **) error	
- (id< SODataPayload >) linksFromString: (NSString *) str error: (NSError **) error	
+ (SODataEDMTypes) mapXSTypeToEDMType: (int32_t) xs_type_code typeName: (NSString **) typeName_in_out	
+ (NSMutableDictionary *) navigationPropertiesDictFromXSNavigationPropertiesList: (XS_PropertyInfoList *) xsNavigationPropertiesList	
+ (SODataMetaEntitySetDefault *) oDataEntitySetFromXSEntitySet: (XS_EntitySet *) xs_EntitySet	
+ (SODataMetaEntityContainerDefault *) oDataMetaEntityContainerFromXSEntityContainer: (XS_EntityContainer *) xs_entityContainer	
+ (NSMutableDictionary *) propertiesDictFromXSPropInfoList: (XS_PropertyInfoList *) propInfoList	
- (id< SODataProperty >) propertyFromString: (NSString *) responseString entitySet: (XS_EntitySet *) entitySet propertyInfo: (XS_PropertyInfo *) propInfo error: (NSError **) error	Deprecated.
- (id< SODataRawValue >) rawValueFromString: (NSString *) responseString	
- (NSString *) relationIdentifierFromRel: (NSString *) rel	
- (NSData *) serializedDataFromEntity: (id< SODataEntity >) entity buildStyle: (SODataBuildStyles) buildStyle entitySet: (XS_EntitySet *) xsEntitySet error: (NSError **) error	
- (NSData *) serializedDataFromEntitySet: (id< SODataEntitySet >) entitySet buildStyle: (SODataBuildStyles) buildStyle entitySet: (XS_EntitySet *) xsEntitySet error: (NSError **) error	
- (NSData *) serializedDataFromEntitySet: (id< SODataEntitySet >) entitySet buildStyle: (SODataBuildStyles) buildStyle entitySet: (XS_EntitySet *) xsEntitySet error: (NSError *__autoreleasing *) error	
- (NSData *) serializedDataFromLink: (id< SODataLink >) link buildStyle: (SODataBuildStyles) buildStyle error: (NSError **) error	
- (NSData *) serializedDataFromLink: (id< SODataLink >) link buildStyle: (SODataBuildStyles) buildStyle error: (NSError *__autoreleasing *) error	
- (SODataDuration *) soDataDurationFromXSDayTimeDuration: (XS_DayTimeDuration *) xsDuration	
+ (id< SODataMetadata >) SODataMetadataFromXS_CsdlDocument: (XS_CsdlDocument *) csdl metadataXML: (NSString *) metadataXML error: (NSError	

**) error	
- (XS_DayTimeDuration *) xSDayTimeDurationFromSODataDuration: (SODataDuration *) sODataDuration	
- (XS_EntityValueList *) xsEntityListFromEntitySet: (id< SODataEntitySet >) entitySet entitySet: (XS_EntitySet *) xsEntitySet error: (NSError **) error	
- (XS_EntityValue *) xsEntryFromEntity: (id< SODataEntity >) entity entitySet: (XS_EntitySet *) entitySet error: (NSError **) error	
- (XS_GlobalDateTime *) xSGlobalDateTimeFromDate: (NSDate *) date	
- (XS_LocalDateTime *) xSLocalDateTimeFromDate: (NSDate *) date	

Parent topic: SAP Mobile Platform 3.0 OData SDK for iOS

1.4.1.1.3.1 annotationsDictFromXSAnnotationMap: method

Syntax

+ (NSDictionary *) `annotationsDictFromXSAnnotationMap` : (XS_AnnotationMap *) annotationMap

Parent topic: SODataConverter class

1.4.1.1.3.2 complexPropertiesDictFromXSComplexTypeMap: method

Syntax

+ (NSMutableDictionary *) `complexPropertiesDictFromXSComplexTypeMap` : (XS_ComplexTypeMap *) xsCompleyTypes

Parent topic: SODataConverter class

1.4.1.1.3.3 convertPropertiesOfSODataEntity:xsEntity:error: method

Syntax

- (BOOL) `convertPropertiesOfSODataEntity` : (SODataPropertyDefault *) property `xsEntity` : (XS_EntityValue *) xsEntity `error` : (NSError **) error

Parent topic: SODataConverter class

1.4.1.1.3.4 convertPropertiesOfXSEntity:toEntityProperties:error: method

Syntax

- (BOOL) `convertPropertiesOfXSEntity` : (XS_EntityValue *) xsEntity `toEntityProperties` : (NSMutableDictionary *) entityProperties `error` : (NSError **) error

Parent topic: SODataConverter class

1.4.1.1.3.5 convertXSPropertyDataValue:withPropertyInfo:error: method

Syntax

- (id< SODataProperty >) `convertXSPropertyDataValue` : (XS_DataValue *) propValue `withPropertyInfo` : (XS_PropertyInfo *) propInfo `error` : (NSError **) error

Parent topic: SODataConverter class

1.4.1.1.3.6 dateComponentsFromXSGlobalDateTime: method

Syntax

- (NSDateComponents *) `dateComponentsFromXSGlobalDateTime` : (XS_GlobalDateTime *) globalDateTime

Parent topic: SODataConverter class

1.4.1.1.3.7 dateFromXSDateTime: method

Syntax

- (NSDate *) `dateFromXSDateTime` : (XS_LocalDateTime *) xsDateTime

Parent topic: SODataConverter class

1.4.1.1.3.8 entityFromString:entitySet:error: method

Syntax

- (id< SODataEntity >) `entityFromString` : (NSString *) entityStr `entitySet` : (XS_EntitySet *) entitySet `error` : (NSError **) error

Parent topic: SODataConverter class

1.4.1.1.3.9 entityFromXSEntity:context:error: method

Syntax

- (id< SODataEntity >) entityFromXSEntity : (XS_EntityValue *) xsEntity context : (XS_DataContext *) dataContext error : (NSError **) error

Parent topic: SODataConverter class

1.4.1.1.3.10 entitySetFromResourcePath:propertyInfo:functionImport:requestMethod:responseType: method

Syntax

- (XS_EntitySet *) entitySetFromResourcePath : (NSString *) resourcePath propertyInfo : (XS_PropertyInfo **) propInfo_in_out functionImport : (XS_DataMethod **) functionImport_in_out requestMethod : (NSString *) requestMethod_in responseType : (NSNumber **) responseType_in_out

Parent topic: SODataConverter class

1.4.1.1.3.11 entitySetFromString:entitySet:error: method

Syntax

- (id< SODataEntitySet >) entitySetFromString : (NSString *) entitySetStr entitySet : (XS_EntitySet *) entitySet error : (NSError **) error

Parent topic: SODataConverter class

1.4.1.1.3.12 entitySetFromXSEntityList:context:error: method

Syntax

- (id< SODataEntitySet >) entitySetFromXSEntityList : (XS_EntityValueList *) xsEntityList context : (XS_DataContext *) dataContext error : (NSError **) error

Parent topic: SODataConverter class

1.4.1.1.3.13 errorFromString:error: method

Syntax

- (id< SODataError >) errorFromString : (NSString *) str error : (NSError **) error

Parent topic: SODataConverter class

1.4.1.1.3.14 facetDictFromXSParam: method

Syntax

+ (NSMutableDictionary *) facetDictFromXSParam : (XS_Parameter *) xsParam

Parent topic: SODataConverter class

1.4.1.1.3.15 facetDictFromXSPropertyInfo: method

Syntax

+ (NSMutableDictionary *) facetDictFromXSPropertyInfo : (XS_PropertyInfo *) xsPropInfo

Parent topic: SODataConverter class

1.4.1.1.3.16 initWithUrl:csdl: method

Syntax

- (id) initWithUrl : (NSString *) baseURL_in csdl : (XS_CsdlDocument *) csdl_in

Parent topic: SODataConverter class

1.4.1.1.3.17 linkFromString:error: method

Syntax

- (id< SODataLink >) linkFromString : (NSString *) str error : (NSError **) error

Parent topic: SODataConverter class

1.4.1.1.3.18 linksFromString:error: method

Syntax

- (id< SODataPayload >) linksFromString : (NSString *) str error : (NSError **) error

Parent topic: SODataConverter class

1.4.1.1.3.19 mapXSTypeToEDMType:typeName: method

Syntax

```
+ ( SODataEDMTypes ) mapXSTypeToEDMType : ( int32_t ) xs_type_code typeName : ( NSString ** ) typeName_in_out
```

Parent topic: SODataConverter class

1.4.1.1.3.20 navigationPropertiesDictFromXSNavigationPropertiesList: method

Syntax

```
+ ( NSMutableDictionary * ) navigationPropertiesDictFromXSNavigationPropertiesList : ( XS_PropertyInfoList * )
xsNavigationPropertiesList
```

Parent topic: SODataConverter class

1.4.1.1.3.21 oDataEntitySetFromXSEntitySet: method

Syntax

```
+ ( SODataMetaEntitySetDefault * ) oDataEntitySetFromXSEntitySet : ( XS_EntitySet * ) xs_EntitySet
```

Parent topic: SODataConverter class

1.4.1.1.3.22 oDataMetaEntityContainerFromXSEntityContainer: method

Syntax

```
+ ( SODataMetaEntityContainerDefault * ) oDataMetaEntityContainerFromXSEntityContainer : ( XS_EntityContainer * ) xs_entityContainer
```

Parent topic: SODataConverter class

1.4.1.1.3.23 propertiesDictFromXSPropInfoList: method

Syntax

```
+ ( NSMutableDictionary * ) propertiesDictFromXSPropInfoList : ( XS_PropertyInfoList * ) propInfoList
```

Parent topic: SODataConverter class

1.4.1.1.3.24 propertyFromString:entitySet:propertyInfo:error: method [deprecated]

Deprecated.

Syntax

```
- ( id< SODataProperty > ) propertyFromString : ( NSString * ) responseString entitySet : ( XS_EntitySet * ) entitySet propertyInfo : (
XS_PropertyInfo * ) propInfo error : ( NSError ** ) error
```

Remarks

Deprecated

Parent topic: SODataConverter class

1.4.1.1.3.25 rawValueFromString: method

Syntax

```
- ( id< SODataRawValue > ) rawValueFromString : ( NSString * ) responseString
```

Parent topic: SODataConverter class

1.4.1.1.3.26 relationIdentifierFromRel: method

Syntax

```
- ( NSString * ) relationIdentifierFromRel : ( NSString * ) rel
```

Parent topic: SODataConverter class

1.4.1.1.3.27 serializedDataFromEntity:buildStyle:entitySet:error: method

Syntax

```
- ( NSData * ) serializedDataFromEntity : ( id< SODataEntity > ) entity buildStyle : ( SODataBuildStyles ) buildStyle entitySet : (
XS_EntitySet * ) xsEntitySet error : ( NSError ** ) error
```

Parent topic: SODataConverter class

1.4.1.1.3.29 serializedDataFromEntitySet:buildStyle:entitySet:error: method

Syntax

- (NSData *) serializedDataFromEntitySet : (id< SODataEntitySet >) entitySet buildStyle : (SODataBuildStyles) buildStyle entitySet : (XS_EntitySet *) xsEntitySet error : (NSError **) error

Parent topic: SODataConverter class

1.4.1.1.3.29 serializedDataFromEntitySet:buildStyle:entitySet:error: method

Syntax

- (NSData *) serializedDataFromEntitySet : (id< SODataEntitySet >) entitySet buildStyle : (SODataBuildStyles) buildStyle entitySet : (XS_EntitySet *) xsEntitySet error : (NSError *__autoreleasing *) error

Parent topic: SODataConverter class

1.4.1.1.3.31 serializedDataFromLink:buildStyle:error: method

Syntax

- (NSData *) serializedDataFromLink : (id< SODataLink >) link buildStyle : (SODataBuildStyles) buildStyle error : (NSError **) error

Parent topic: SODataConverter class

1.4.1.1.3.31 serializedDataFromLink:buildStyle:error: method

Syntax

- (NSData *) serializedDataFromLink : (id< SODataLink >) link buildStyle : (SODataBuildStyles) buildStyle error : (NSError *__autoreleasing *) error

Parent topic: SODataConverter class

1.4.1.1.3.32 soDataDurationFromXSDayTimeDuration: method

Syntax

- (SODataDuration *) soDataDurationFromXSDayTimeDuration : (XS_DayTimeDuration *) xsDuration

Parent topic: SODataConverter class

1.4.1.1.3.33 SODataMetadataFromXS_CsdlDocument:metadataXML:error: method

Syntax

+ (id< SODataMetadata >) SODataMetadataFromXS_CsdlDocument : (XS_CsdlDocument *) csdl metadataXML : (NSString *) metadataXML error : (NSError **) error

Parent topic: SODataConverter class

1.4.1.1.3.34 xSDayTimeDurationFromSODataDuration: method

Syntax

- (XS_DayTimeDuration *) xSDayTimeDurationFromSODataDuration : (SODataDuration *) sODataDuration

Parent topic: SODataConverter class

1.4.1.1.3.35 xsEntityListFromEntitySet:entitySet:error: method

Syntax

- (XS_EntityValueList *) xsEntityListFromEntitySet : (id< SODataEntitySet >) entitySet entitySet : (XS_EntitySet *) xsEntitySet error : (NSError **) error

Parent topic: SODataConverter class

1.4.1.1.3.36 xsEntryFromEntity:entitySet:error: method

Syntax

- (XS_EntityValue *) xsEntryFromEntity : (id< SODataEntity >) entity entitySet : (XS_EntitySet *) entitySet error : (NSError **) error

Parent topic: SODataConverter class

1.4.1.1.3.37 xSGlobalDateTimeFromDate: method

Syntax

- (XS_GlobalDateTime *) xSGlobalDateTimeFromDate : (NSDate *) date

Parent topic: SODataConverter class

1.4.1.1.3.38 xSLocalDateTimeFromDate: method

Syntax

`- ( XS_LocalDateTime * ) xSLocalDateTimeFromDate : ( NSDate * ) date`

Parent topic: SODataConverter class

1.4.1.1.4 SODataCredentials protocol

Protocol for SOData credentials handling.

Syntax

`@protocol SODataCredentials`

Parent topic: SAP Mobile Platform 3.0 OData SDK for iOS

1.4.1.1.5 SODataOnlineStore class

Syntax

`@interface SODataOnlineStore : NSObject < SODataStoreAsync , SODataStoreSync , SODataRequestExecutedDelegate >`

Members

All members of SODataOnlineStore, including inherited members.

Methods

Method	Description
- (id< SODataEntity >) allocateNavigationProperties: (id< SODataEntity >) entity error: (NSError **) error	
- (id< SODataEntity >) allocatePropertiesOfEntity: (id< SODataEntity >) entity mode: (SODataPropModeFlags) mode error: (NSError **) error	
- (NSMutableDictionary *) asyncRequests	
- (void) dealloc	
- (NSString *) determineEntitySetFromResourcePath: (NSString *) resourcePath error: (NSError **) error	
- (NSString *) determineEntityTypeFromResourcePath: (NSString *) resourcePath error: (NSError **) error	
- (SODataTypes) determineODataTypeFromResourcePath: (NSString *) resourcePath mode: (SODataRequestModes) mode error: (NSError **) error	
- (void) downloadMetaData	
- (id< SODataResponseSingle >) executeCreateEntity: (id< SODataEntity >) entity collectionPath: (NSString *) collectionPath options: (NSDictionary *) options error: (NSError **) error	
- (id< SODataResponseSingle >) executeDeleteEntity: (NSString *) resourcePath etag: (NSString *) etag options: (NSDictionary *) options error: (NSError **) error	
- (id< SODataResponseSingle >) executeDeleteEntity: (id< SODataEntity >) entity options: (NSDictionary *) options error: (NSError **) error	
- (id< SODataResponseSingle >) executeFunction: (NSString *) resourcePath options: (NSDictionary *) options error: (NSError **) error	
- (id< SODataResponseSingle >) executePatchEntity: (id< SODataEntity >) entity options: (NSDictionary *) options error: (NSError **) error	
- (id< SODataResponseSingle >) executeReadEntity: (id< SODataEntity >) entity options: (NSDictionary *) options error: (NSError **) error	
- (id< SODataResponseSingle >) executeReadEntitySet: (NSString *) resourcePath options: (NSDictionary *) options error: (NSError **) error	
- (id< SODataResponseSingle >) executeReadEntityWithResourcePath: (NSString *) resourcePath options: (NSDictionary *) options error: (NSError **) error	
- (id< SODataResponseSingle >) executeReadLink: (NSString *) resourcePath options: (NSDictionary *) options error: (NSError **) error	
- (id< SODataResponseSingle >) executeReadLinkSet: (NSString *) resourcePath options: (NSDictionary *) options error: (NSError **) error	
- (id< SODataResponseSingle >) executeReadPropertyComplex: (NSString *) resourcePath options: (NSDictionary *) options error: (NSError **) error	
- (id< SODataResponseSingle >) executeReadPropertyPrimitive: (NSString *) resourcePath options: (NSDictionary *) options error: (NSError **) error	
- (id< SODataResponseSingle >) executeReadPropertyRaw: (NSString *) resourcePath options: (NSDictionary *) options error: (NSError **) error	
- (id< SODataResponse >) executeRequest: (id< SODataRequestParam >) request error: (NSError **) error	

- (id< SODataResponseSingle >) executeUpdateEntity: (id< SODataEntity >) entity options: (NSDictionary *) options error: (NSError **) error	
- (id) initWithURL: (NSURL *) url httpConversationManager: (HttpConversationManager *) httpConversationManager	Create a new store instance asynchronously with with given service url, requestHeaders and delegates.
- (id) initWithURL: (NSURL *) url httpConversationManager: (HttpConversationManager *) httpConversationManager options: (SODataOnlineStoreOptions *) options	Create a new store instance asynchronously with with given service url, requestHeaders and delegates.
- (BOOL) openStoreWithError: (NSError **) error	Opens the store with the URL passed during init.
- (void) openWithURL: (NSString *) serviceUrl	
- (void) requestExecutedWithId: (NSString *) uniqueId	
- (id< SODataRequestExecution >) requestExecutionFromBatchRequest: (id< SODataRequestParamBatch >) batchRequest error: (NSError **) error delegate: (id) delegate	
- (id< SODataRequestExecution >) requestExecutionFromSingleRequest: (id< SODataRequestParamSingle >) request error: (NSError **) error delegate: (id) delegate	
- (id< SODataResponseSingle >) responseSingleFromResponse: (id< SODataResponse >) response error: (NSError **) error	
- (id< SODataRequestExecution >) scheduleCreateEntity: (id< SODataEntity >) entity collectionPath: (NSString *) collectionPath delegate: (id< SODataRequestDelegate >) delegate options: (NSDictionary *) options	
- (id< SODataRequestExecution >) scheduleDeleteEntity: (id< SODataEntity >) entity delegate: (id< SODataRequestDelegate >) delegate options: (NSDictionary *) options	
- (id< SODataRequestExecution >) scheduleDeleteEntity: (NSString *) resourcePath etag: (NSString *) etag delegate: (id< SODataRequestDelegate >) delegate options: (NSDictionary *) options	
- (id< SODataRequestExecution >) scheduleFunction: (NSString *) resourcePath delegate: (id< SODataRequestDelegate >) delegate options: (NSDictionary *) options	
- (id< SODataRequestExecution >) schedulePatchEntity: (id< SODataEntity >) entity delegate: (id< SODataRequestDelegate >) delegate options: (NSDictionary *) options	
- (id< SODataRequestExecution >) scheduleReadEntity: (id< SODataEntity >) entity delegate: (id< SODataRequestDelegate >) delegate options: (NSDictionary *) options	
- (id< SODataRequestExecution >) scheduleReadEntitySet: (NSString *) resourcePath delegate: (id< SODataRequestDelegate >) delegate options: (NSDictionary *) options	
- (id< SODataRequestExecution >) scheduleReadEntityWithResourcePath: (NSString *) resourcePath delegate: (id< SODataRequestDelegate >) delegate options: (NSDictionary *) options	
- (id< SODataRequestExecution >) scheduleReadLink: (NSString *) resourcePath delegate: (id< SODataRequestDelegate >) delegate options: (NSDictionary *) options	
- (id< SODataRequestExecution >) scheduleReadLinkSet: (NSString *) resourcePath delegate: (id< SODataRequestDelegate >) delegate options: (NSDictionary *) options	
- (id< SODataRequestExecution >) scheduleReadPropertyComplex: (NSString *) resourcePath delegate: (id< SODataRequestDelegate >) delegate options: (NSDictionary *) options	
- (id< SODataRequestExecution >) scheduleReadPropertyPrimitive: (NSString *) resourcePath delegate: (id< SODataRequestDelegate >) delegate options: (NSDictionary *) options	
- (id< SODataRequestExecution >) scheduleReadPropertyRaw: (NSString *) resourcePath delegate: (id< SODataRequestDelegate >) delegate options: (NSDictionary *) options	
- (id< SODataRequestExecution >) scheduleRequest: (id< SODataRequestParam >) request delegate: (id< SODataRequestDelegate >) delegate	
- (id< SODataRequestExecution >) scheduleUpdateEntity: (id< SODataEntity >) entity delegate: (id< SODataRequestDelegate >) delegate options: (NSDictionary *) options	
- (void) setDefaultValueForProperty: (SODataPropertyDefault *) property propertyInfo: (XS_PropertyInfo *) propInfo	
- (SODataOnlineStore *) syncOpenStoreWithError: (NSError **) anError	
- (BOOL) validateStoreWithError: (NSError **) error	

Parent topic: SAP Mobile Platform 3.0 OData SDK for iOS

1.4.1.1.5.1 allocateNavigationProperties:error: method

Syntax

- (id< SODataEntity >) `allocateNavigationProperties` : (id< SODataEntity >) entity `error` : (NSError **) error

Parent topic: SODataOnlineStore class

1.4.1.1.5.2 allocatePropertiesOfEntity:mode:error: method

Syntax

- (id< SODataEntity >) allocatePropertiesOfEntity : (id< SODataEntity >) entity mode : (SODataPropModeFlags) mode error : (NSError **) error

Parent topic: SODataOnlineStore class

1.4.1.1.5.3 asyncRequests method

Syntax

- (NSMutableDictionary *) asyncRequests

Parent topic: SODataOnlineStore class

1.4.1.1.5.4 dealloc method

Syntax

- (void) dealloc

Parent topic: SODataOnlineStore class

1.4.1.1.5.5 determineEntitySetFromResourcePath:error: method

Syntax

- (NSString *) determineEntitySetFromResourcePath : (NSString *) resourcePath error : (NSError **) error

Parent topic: SODataOnlineStore class

1.4.1.1.5.6 determineEntityTypeFromResourcePath:error: method

Syntax

- (NSString *) determineEntityTypeFromResourcePath : (NSString *) resourcePath error : (NSError **) error

Parent topic: SODataOnlineStore class

1.4.1.1.5.7 determineODataTypeFromResourcePath:mode:error: method

Syntax

- (SODataTypes) determineODataTypeFromResourcePath : (NSString *) resourcePath mode : (SODataRequestModes) mode error : (NSError **) error

Parent topic: SODataOnlineStore class

1.4.1.1.5.8 downloadMetaData method

Syntax

- (void) downloadMetaData

Parent topic: SODataOnlineStore class

1.4.1.1.5.9 executeCreateEntity:collectionPath:options:error: method

Syntax

- (id< SODataResponseSingle >) executeCreateEntity : (id< SODataEntity >) entity collectionPath : (NSString *) collectionPath options : (NSDictionary *) options error : (NSError **) error

Parent topic: SODataOnlineStore class

1.4.1.1.5.10 executeDeleteEntity:etag:options:error: method

Syntax

- (id< SODataResponseSingle >) executeDeleteEntity : (NSString *) resourcePath etag : (NSString *) etag options : (NSDictionary *) options error : (NSError **) error

Parent topic: SODataOnlineStore class

1.4.1.1.5.11 executeDeleteEntity:options:error: method

Syntax

- (id< SODataResponseSingle >) executeDeleteEntity : (id< SODataEntity >) entity options : (NSDictionary *) options error : (NSError **) error

Parent topic: SODataOnlineStore class

1.4.1.1.5.12 executeFunction:options:error: method

Syntax

- (id< SODataResponseSingle >) executeFunction : (NSString *) resourcePath options : (NSDictionary *) options error : (NSError **) error

Parent topic: SODataOnlineStore class

1.4.1.1.5.13 executePatchEntity:options:error: method

Syntax

- (id< SODataResponseSingle >) executePatchEntity : (id< SODataEntity >) entity options : (NSDictionary *) options error : (NSError **) error

Parent topic: SODataOnlineStore class

1.4.1.1.5.14 executeReadEntity:options:error: method

Syntax

- (id< SODataResponseSingle >) executeReadEntity : (id< SODataEntity >) entity options : (NSDictionary *) options error : (NSError **) error

Parent topic: SODataOnlineStore class

1.4.1.1.5.15 executeReadEntitySet:options:error: method

Syntax

- (id< SODataResponseSingle >) executeReadEntitySet : (NSString *) resourcePath options : (NSDictionary *) options error : (NSError **) error

Parent topic: SODataOnlineStore class

1.4.1.1.5.16 executeReadEntityWithResourcePath:options:error: method

Syntax

- (id< SODataResponseSingle >) executeReadEntityWithResourcePath : (NSString *) resourcePath options : (NSDictionary *) options error : (NSError **) error

Parent topic: SODataOnlineStore class

1.4.1.1.5.17 executeReadLink:options:error: method

Syntax

- (id< SODataResponseSingle >) executeReadLink : (NSString *) resourcePath options : (NSDictionary *) options error : (NSError **) error

Parent topic: SODataOnlineStore class

1.4.1.1.5.18 executeReadLinkSet:options:error: method

Syntax

- (id< SODataResponseSingle >) executeReadLinkSet : (NSString *) resourcePath options : (NSDictionary *) options error : (NSError **) error

Parent topic: SODataOnlineStore class

1.4.1.1.5.19 executeReadPropertyComplex:options:error: method

Syntax

- (id< SODataResponseSingle >) executeReadPropertyComplex : (NSString *) resourcePath options : (NSDictionary *) options error : (NSError **) error

Parent topic: SODataOnlineStore class

1.4.1.1.5.20 executeReadPropertyPrimitive:options:error: method

Syntax

- (id< SODataResponseSingle >) executeReadPropertyPrimitive : (NSString *) resourcePath options : (NSDictionary *) options error : (NSError **) error

Parent topic: SODataOnlineStore class

1.4.1.1.5.21 executeReadPropertyRaw:options:error: method

Syntax

- (id< SODataResponseSingle >) executeReadPropertyRaw : (NSString *) resourcePath options : (NSDictionary *) options error : (NSError **) error

Parent topic: SODataOnlineStore class

1.4.1.1.5.22 executeRequest:error: method

Syntax

- (id< SODataResponse >) executeRequest : (id< SODataRequestParam >) request error : (NSError **) error

Parent topic: SODataOnlineStore class

1.4.1.1.5.23 executeUpdateEntity:options:error: method

Syntax

- (id< SODataResponseSingle >) executeUpdateEntity : (id< SODataEntity >) entity options : (NSDictionary *) options error : (NSError **) error

Parent topic: SODataOnlineStore class

1.4.1.1.5.24 initWithURL:httpConversationManager: method

Create a new store instance asynchronously with with given service url, requestHeaders and delegates.

Syntax

- (id) initWithURL : (NSURL *) url httpConversationManager : (HttpConversationManager *) httpConversationManager

Parameters

url

of the service

httpConversationManager

HttpConversationManager object which contains the logon data and runs the requests

Parent topic: SODataOnlineStore class

1.4.1.1.5.25 initWithURL:httpConversationManager:options: method

Create a new store instance asynchronously with with given service url, requestHeaders and delegates.

Syntax

- (id) initWithURL : (NSURL *) url httpConversationManager : (HttpConversationManager *) httpConversationManager options : (SODataOnlineStoreOptions *) options

Parameters

url

of the service

httpConversationManager

HttpConversationManager object which contains the logon data and runs the requests

options

of the store

Parent topic: SODataOnlineStore class

1.4.1.1.5.26 openStoreWithError: method

Opens the store with the URL passed during init.

Syntax

- (BOOL) openStoreWithError : (NSError **) error

Parent topic: SODataOnlineStore class

1.4.1.1.5.27 openWithURL: method

Syntax

- (void) openWithURL : (NSString *) serviceUrl

Parent topic: SODataOnlineStore class

1.4.1.1.9.1 requestExecutedWithId: method

Syntax

- (void) requestExecutedWithId : (NSString *) uniqueId

Parent topic: SODataOnlineStore class

1.4.1.1.5.29 requestExecutionFromBatchRequest:error:delegate: method

Syntax

- (id< SODataRequestExecution >) requestExecutionFromBatchRequest :(id< SODataRequestParamBatch >) batchRequest error :(NSError **) error delegate :(id) delegate

Parent topic: SODataOnlineStore class

1.4.1.1.5.30 requestExecutionFromSingleRequest:error:delegate: method

Syntax

- (id< SODataRequestExecution >) requestExecutionFromSingleRequest :(id< SODataRequestParamSingle >) request error :(NSError **) error delegate :(id) delegate

Parent topic: SODataOnlineStore class

1.4.1.1.5.31 responseSingleFromResponse:error: method

Syntax

- (id< SODataResponseSingle >) responseSingleFromResponse :(id< SODataResponse >) response error :(NSError **) error

Parent topic: SODataOnlineStore class

1.4.1.1.5.32 scheduleCreateEntity:collectionPath:delegate:options: method

Syntax

- (id< SODataRequestExecution >) scheduleCreateEntity :(id< SODataEntity >) entity collectionPath :(NSString *) collectionPath delegate :(id< SODataRequestDelegate >) delegate options :(NSDictionary *) options

Parent topic: SODataOnlineStore class

1.4.1.1.5.33 scheduleDeleteEntity:delegate:options: method

Syntax

- (id< SODataRequestExecution >) scheduleDeleteEntity :(id< SODataEntity >) entity delegate :(id< SODataRequestDelegate >) delegate options :(NSDictionary *) options

Parent topic: SODataOnlineStore class

1.4.1.1.5.34 scheduleDeleteEntity:etag:delegate:options: method

Syntax

- (id< SODataRequestExecution >) scheduleDeleteEntity :(NSString *) resourcePath etag :(NSString *) etag delegate :(id< SODataRequestDelegate >) delegate options :(NSDictionary *) options

Parent topic: SODataOnlineStore class

1.4.1.1.5.35 scheduleFunction:delegate:options: method

Syntax

- (id< SODataRequestExecution >) scheduleFunction :(NSString *) resourcePath delegate :(id< SODataRequestDelegate >) delegate options :(NSDictionary *) options

Parent topic: SODataOnlineStore class

1.4.1.1.5.36 schedulePatchEntity:delegate:options: method

Syntax

- (id< SODataRequestExecution >) schedulePatchEntity :(id< SODataEntity >) entity delegate :(id< SODataRequestDelegate >) delegate options :(NSDictionary *) options

Parent topic: SODataOnlineStore class

1.4.1.1.5.37 scheduleReadEntity:delegate:options: method

Syntax

- (id< SODataRequestExecution >) scheduleReadEntity :(id< SODataEntity >) entity delegate :(id< SODataRequestDelegate >) delegate options :(NSDictionary *) options

Parent topic: SODataOnlineStore class

1.4.1.1.5.38 scheduleReadEntitySet:delegate:options: method

Syntax

- (id< SODataRequestExecution >) scheduleReadEntitySet :(NSString *) resourcePath delegate :(id< SODataRequestDelegate >) delegate options :(NSDictionary *) options

Parent topic: SODataOnlineStore class

1.4.1.1.5.39 scheduleReadEntityWithResourcePath:delegate:options: method

Syntax

- (id< SODataRequestExecution >) scheduleReadEntityWithResourcePath : (NSString *) resourcePath delegate : (id< SODataRequestDelegate >) delegate options : (NSDictionary *) options

Parent topic: SODataOnlineStore class

1.4.1.1.5.40 scheduleReadLink:delegate:options: method

Syntax

- (id< SODataRequestExecution >) scheduleReadLink : (NSString *) resourcePath delegate : (id< SODataRequestDelegate >) delegate options : (NSDictionary *) options

Parent topic: SODataOnlineStore class

1.4.1.1.5.41 scheduleReadLinkSet:delegate:options: method

Syntax

- (id< SODataRequestExecution >) scheduleReadLinkSet : (NSString *) resourcePath delegate : (id< SODataRequestDelegate >) delegate options : (NSDictionary *) options

Parent topic: SODataOnlineStore class

1.4.1.1.5.42 scheduleReadPropertyComplex:delegate:options: method

Syntax

- (id< SODataRequestExecution >) scheduleReadPropertyComplex : (NSString *) resourcePath delegate : (id< SODataRequestDelegate >) delegate options : (NSDictionary *) options

Parent topic: SODataOnlineStore class

1.4.1.1.5.43 scheduleReadPropertyPrimitive:delegate:options: method

Syntax

- (id< SODataRequestExecution >) scheduleReadPropertyPrimitive : (NSString *) resourcePath delegate : (id< SODataRequestDelegate >) delegate options : (NSDictionary *) options

Parent topic: SODataOnlineStore class

1.4.1.1.5.44 scheduleReadPropertyRaw:delegate:options: method

Syntax

- (id< SODataRequestExecution >) scheduleReadPropertyRaw : (NSString *) resourcePath delegate : (id< SODataRequestDelegate >) delegate options : (NSDictionary *) options

Parent topic: SODataOnlineStore class

1.4.1.1.5.45 scheduleRequest:delegate: method

Syntax

- (id< SODataRequestExecution >) scheduleRequest : (id< SODataRequestParam >) request delegate : (id< SODataRequestDelegate >) delegate

Parent topic: SODataOnlineStore class

1.4.1.1.5.46 scheduleUpdateEntity:delegate:options: method

Syntax

- (id< SODataRequestExecution >) scheduleUpdateEntity : (id< SODataEntity >) entity delegate : (id< SODataRequestDelegate >) delegate options : (NSDictionary *) options

Parent topic: SODataOnlineStore class

1.4.1.1.5.47 setDefaultValueForProperty:propertyInfo: method

Syntax

- (void) setDefaultValueForProperty : (SODataPropertyDefault *) property propertyInfo : (XS_PropertyInfo *) propInfo

Parent topic: SODataOnlineStore class

1.4.1.1.5.48 syncOpenStoreWithError: method

Syntax

- (SODataOnlineStore *) syncOpenStoreWithError : (NSError **) anError

Parent topic: SODataOnlineStore class

1.4.1.1.5.49 validateStoreWithError: method

Syntax

- (BOOL) validateStoreWithError : (NSError **) error

Parent topic: SODataOnlineStore class

1.4.1.1.6 SODataOnlineStoreDelegate protocol

Protocol for SOData open store.

Syntax

@protocol SODataOnlineStoreDelegate

Members

All members of SODataOnlineStoreDelegate, including inherited members.

Methods

Method	Description
- (void) onlineStoreOpenFailed: (SODataOnlineStore *) store error: (NSError *) error	Error delegate if store open failed.
- (void) onlineStoreOpenFinished: (SODataOnlineStore *) store	Succesful delegate call after store opened.

Parent topic: SAP Mobile Platform 3.0 OData SDK for iOS

1.4.1.1.6.1 onlineStoreOpenFailed:error: method

Error delegate if store open failed.

Syntax

- (void) onlineStoreOpenFailed : (SODataOnlineStore *) store error : (NSError *) error

Parent topic: SODataOnlineStoreDelegate protocol

1.4.1.1.6.2 onlineStoreOpenFinished: method

Succesful delegate call after store opened.

Syntax

- (void) onlineStoreOpenFinished : (SODataOnlineStore *) store

Parent topic: SODataOnlineStoreDelegate protocol

1.4.1.1.7 SODataOnlineStoreOptions class

Syntax

@interface SODataOnlineStoreOptions : NSObject

Parent topic: SAP Mobile Platform 3.0 OData SDK for iOS

1.4.1.1.8 SODataRequestCredentialsDelegate protocol

Protocol for SOData request credential handling.

Syntax

@protocol SODataRequestCredentialsDelegate

Members

All members of SODataRequestCredentialsDelegate, including inherited members.

Methods

Method	Description
- (id< SODataCredentials >) credentialsForRequest	Credentials for use with the request.

Parent topic: SAP Mobile Platform 3.0 OData SDK for iOS

1.4.1.1.8.1 credentialsForRequest method

Credentials for use with the request.

Syntax

- (id< SODataCredentials >) `credentialsForRequest`

Parent topic: SODataRequestCredentialsDelegate protocol

Related Information

SODataCredentials protocol

1.4.1.1.9 SODataRequestExecutedDelegate protocol

Syntax

@protocol `SODataRequestExecutedDelegate`

Members

All members of SODataRequestExecutedDelegate, including inherited members.

Methods

Method	Description
- (void) requestExecutedWithId: (NSString *) uniqueId	

Parent topic: SAP Mobile Platform 3.0 OData SDK for iOS

1.4.1.1.9.1 requestExecutedWithId: method

Syntax

- (void) `requestExecutedWithId` : (NSString *) uniqueId

Parent topic: SODataRequestExecutedDelegate protocol

1.4.1.1.10 SODataRequestExecutionBatchDefault class

Syntax

@interface `SODataRequestExecutionBatchDefault` : `SODataRequestExecutionDefault` < `SODataRequestDelegate` >

Members

All members of SODataRequestExecutionBatchDefault, including inherited members.

Methods

Method	Description
- (BOOL) addRequestExecutionFromRequestParam: (id< SODataRequestParamSingle >) batchItem toRequestArray: (NSMutableArray *) requestArray_in batchRequests: (NSArray *) batchRequests error: (NSError **) error	
- (NSString *) buildURLString	
- (id< SODataResponseChangeset >) createResponseChangesetFromRequest: (id< SODataRequestChangeset >) requestParamChangeset urlRequest: (ChangeSet *) changeSet httpResponses: (NSArray *) httpResponses error: (NSError **) error	
- (id) initWithRequest: (id< SODataRequestParamBatch >) request_in	
- (NSMutableURLRequest *) requestWithURL: (NSString *) url_in	
- (id< SODataResponse >) responseFromRequest: (NSMutableURLRequest *) urlRequest httpResponse: (NSHTTPURLResponse *) httpResponse responseData: (NSData *) responseData error: (NSError **) error_in	
- (BOOL) setupPostBodyWithError: (NSError **) error	
- (void) setupRequestMethod	
- (BOOL) setupRequestParamsForRequest: (NSMutableURLRequest *) batchRequest requestParam: (id< SODataRequestParamBatch >) requestParamBatch error: (NSError **) error	

Parent topic: SAP Mobile Platform 3.0 OData SDK for iOS

1.4.1.1.10.1 addRequestExecutionFromRequestParam:toRequestArray:batchRequests:error: method

Syntax

- (BOOL) `addRequestExecutionFromRequestParam` : (id< SODataRequestParamSingle >) batchItem `toRequestArray` : (NSMutableArray *) requestArray_in `batchRequests` : (NSArray *) batchRequests `error` : (NSError **) error

Parent topic: SODataRequestExecutionBatchDefault class

1.4.1.1.11.2 buildURLString method

Syntax

- (NSString *) buildURLString

Parent topic: SODataRequestExecutionBatchDefault class

1.4.1.1.10.3 createResponseChangesetFromRequest:urlRequest:httpResponses:error: method

Syntax

- (id< SODataResponseChangeset >) createResponseChangesetFromRequest : (id< SODataRequestChangeset >) requestParamChangeset urlRequest : (ChangeSet *) changeSet httpResponses : (NSArray *) httpResponses error : (NSError **) error

Parent topic: SODataRequestExecutionBatchDefault class

1.4.1.1.11.7 initWithRequest: method

Syntax

- (id) initWithRequest : (id< SODataRequestParamBatch >) request_in

Parent topic: SODataRequestExecutionBatchDefault class

1.4.1.1.10.5 requestWithURL: method

Syntax

- (NSMutableURLRequest *) requestWithURL : (NSString *) url_in

Parent topic: SODataRequestExecutionBatchDefault class

1.4.1.1.10.6 responseFromRequest:httpResponse:responseData:error: method

Syntax

- (id< SODataResponse >) responseFromRequest : (NSMutableURLRequest *) urlRequest httpResponse : (NSHTTPURLResponse *) httpResponse responseData : (NSData *) responseData error : (NSError **) error_in

Parent topic: SODataRequestExecutionBatchDefault class

1.4.1.1.10.7 setupPostBodyWithError: method

Syntax

- (BOOL) setupPostBodyWithError : (NSError **) error

Parent topic: SODataRequestExecutionBatchDefault class

1.4.1.1.10.8 setupRequestMethod method

Syntax

- (void) setupRequestMethod

Parent topic: SODataRequestExecutionBatchDefault class

1.4.1.1.10.9 setupRequestParamsForRequest:requestParam:error: method

Syntax

- (BOOL) setupRequestParamsForRequest : (NSMutableURLRequest *) batchRequest requestParam : (id< SODataRequestParamBatch >) requestParamBatch error : (NSError **) error

Parent topic: SODataRequestExecutionBatchDefault class

1.4.1.1.11 SODataRequestExecutionDefault class

Syntax

@interface SODataRequestExecutionDefault : NSObject < SODataRequestExecution >

Members

All members of SODataRequestExecutionDefault, including inherited members.

Methods

Method	Description
- (void) addLocationAndCodeToResponse: (id< SODataResponseSingle >) responseSingle httpResponse: (NSHTTPURLResponse *) httpResponse	Calls to the request delegates if request finished.

- (NSString *) buildURLString	Builds the string to send the request to.
- (void) cancelExecution	
- (BOOL) checkODataError: (NSString *) responseString	Check for ODataError, returns YES if the response is an ODataError.
- (NSString *) fullResourcePathFromResourcePathWithContentID: (NSString *) resourcePathWithContentID usingRequestParams: (NSArray *) requestParams	resolve resourcePath from request referring to contentID in a changeset
- (NSString *) generateUUID	Generates a unique string.
- (id) initWithRequest: (id< SODataRequestParam >) request_in	
- (id< SODataPayload >) payloadFromRequest: (NSMutableURLRequest *) request httpResponse: (NSHTTPURLResponse *) httpResponse responseData: (NSData *) responseData requestParam: (id< SODataRequestParam >) requestParam resolvedResourcePath: (NSString *) resolvedResourcePath error: (NSError **) error	Create a payload using the request response and the requestParam.
- (void) processHttpResponse: (NSURLResponse *) httpResponse responseData: (NSData *) data error: (NSError *) error	
- (NSMutableURLRequest *) requestWithURL: (NSString *) url	Creates the URL for networkRequest.
- (id< SODataResponse >) responseFromFailedRequest: (NSMutableURLRequest *) urlRequest requestParam: (id< SODataRequestParam >) requestParam requestError: (NSError *) requestError httpResponse: (NSHTTPURLResponse *) httpResponse responseData: (NSData *) responseData error: (NSError **) error_in	
- (id< SODataResponse >) responseFromFailedRequestWithRequestParam: (id< SODataRequestParam >) requestParam error: (NSError **) error_in	Create a response using the requestParam.
- (id< SODataResponse >) responseFromRequest: (NSMutableURLRequest *) urlRequest httpResponse: (NSHTTPURLResponse *) httpResponse responseData: (NSData *) responseData error: (NSError **) error_in	Method for creating the response from the networkRequest.
- (id< SODataResponse >) responseFromSyncRequest: (NSMutableURLRequest *) urlRequest_in httpResponse: (NSHTTPURLResponse *) httpResponse responseData: (NSData *) responseData error: (NSError **) error	Creates the reponse object from request-response.
- (void) runSyncRequest	
- (void) setBaseURL: (NSString *) baseURL_in csdl: (XS_CsdlDocument *) csdl	Set url and csdl.
- (BOOL) setupPostBodyWithError: (NSError **) error	Sets the post body.
- (void) setupRequestHeaders	Sets the headers for the request.
- (void) setupRequestMethod	———— TO BE IMPLEMENTED IN SUBCLASSES ————
- (BOOL) setupRequestWithError: (NSError **) error	Sets the request parameters.
- (void) startAsyncNetworkRequest	Handles execution state and calls to the delegate.
- (void) startAsyncRequest	Starts the execution asynchronously.
- (id< SODataResponse >) startSyncRequestWithError: (NSError **) error	Starts the execution synchronously.
- (void) waitForSyncFinish	

Parent topic: SAP Mobile Platform 3.0 OData SDK for iOS

1.4.1.1.11.1 addLocationAndCodeToResponse:httpResponse: method

Calls to the request delegates if request finished.

Syntax

- (void) `addLocationAndCodeToResponse` : (id< SODataResponseSingle >) responseSingle `httpResponse` : (NSHTTPURLResponse *) httpResponse

Remarks

Called back by request if the request failed Helper method for adding location and code to Response

Parent topic: SODataRequestExecutionDefault class

1.4.1.1.11.2 buildURLString method

Builds the string to send the request to.

Syntax

- (NSString *) `buildURLString`

Parent topic: SODataRequestExecutionDefault class

1.4.1.1.11.3 cancelExecution method

Syntax

- (void) `cancelExecution`

Parent topic: SODataRequestExecutionDefault class

1.4.1.1.11.4 checkODataError: method

Check for ODataError, returns YES if the response is an ODataError.

Syntax

- (BOOL) `checkODataError` : (NSString *) responseString

Parent topic: SODataRequestExecutionDefault class

1.4.1.1.11.5 fullResourcePathFromResourcePathWithContentID: usingRequestParams: method

resolve resourcePath from request referring to contentID in a changeset

Syntax

- (NSString *) `fullResourcePathFromResourcePathWithContentID` : (NSString *) resourcePathWithContentID `usingRequestParams` : (NSArray *) requestParams

Parent topic: SODataRequestExecutionDefault class

1.4.1.1.11.6 generateUUID method

Generates a unique string.

Syntax

- (NSString *) `generateUUID`

Parent topic: SODataRequestExecutionDefault class

1.4.1.1.11.7 initWithRequest: method

Syntax

- (id) `initWithRequest` : (id< SODataRequestParam >) request_in

Parent topic: SODataRequestExecutionDefault class

1.4.1.1.11.8 payloadFromRequest:httpResponse:responseData:requestParam:resolvedResourcePath:error: method

Create a payload using the request response and the requestParam.

Syntax

- (id< SODataPayload >) `payloadFromRequest` : (NSMutableURLRequest *) request `httpResponse` : (NSHTTPURLResponse *) httpResponse `responseData` : (NSData *) responseData `requestParam` : (id< SODataRequestParam >) requestParam `resolvedResourcePath` : (NSString *) resolvedResourcePath `error` : (NSError **) error

Parent topic: SODataRequestExecutionDefault class

1.4.1.1.11.9 processHttpResponse:responseData:error: method

Syntax

- (void) `processHttpResponse` : (NSURLResponse *) httpResponse `responseData` : (NSData *) data `error` : (NSError *) error

Parent topic: SODataRequestExecutionDefault class