

Watch integration - API for communication between MPA and Watch

Basic information

Channel allows only to one-way communication without response. There is no response definition. Treat Response as new request without any response.

Communication is based on JSON format. Request object must contains request *type* and *requestId*. Other fields are determined by request type.

Request structure

type	String	Request type. Use API method name
appVersion	String	Current application version. Used to support backward compatibility
sdkVersion	String	Currently used Wearable SDK version
requestId	String	Unique Id of request for matching request with response
chunk	MessageChunk	Optional. Allows to split the request into multiple parts to avoid 1kb limit.

Json structure:

```
{
  type: "EVENT_TYPE" ,
  appVersion: "2.0.0" ,
  requestId: "d516d558-f476-4e7f-b3c0-300261d37d46"
  //other fields reserved for type
}
```

Common models

Token

id	String	Payment Instrument Id
tokenType	String	Enum with value: MC
tokenExpiry	String	Token expiration date in format MM/YY
tokenLastFourDigits	String	Token last four digits
paymentInstrumentExpirationDate	String	Payment instrument expiration date in format MM/YY
paymentInstrumentLastFourDigits	String	Payment instrument last four digits
tokenStatus	String	Payment token status. One of [ACTIVE, SUSPENDED, DELETED]
description	String	Token description
tokenVisual	String	Optional, visualId from DC

PaymentInstrumentStatus

paymentInstrumentId	String	Payment Instrument Id
status	String	Payment token status. One of [ACTIVE, SUSPENDED, DELETED]
credentialsCount	Int	Transaction credentials count

MessageChunk

current	Int	Current chunk. Starts with 1.
total	Int	Number of all chunks.
data	String	Part of the full message.

API for communication to Watch

Method for creating secure session to transfer data between MPA and Watch. During this process SDK on Phone creates unique keys for communication then Watch creates data encryption keys. Method should be used before every AddTokenData and AddTokenCredentials methods	CreateSecureSession(type:String, appVersion:String, requestId: String, secureSessionData:String)	CreateSecureSessionResult(type:String, appVersion:String, sdkVersion:String, requestId: String, secureSessionData:String)	CreateSecureSessionResult(appVersion:String, sdkVersion:String, type:String, requestId:String, error:Error) Errors:
Method for synchronize tokens status in version 1.0 and initialize synchronization from Watch to MPA	SyncStatus(type:String, appVersion:String, requestId: String)	SyncStatusResult(type:String, appVersion:String, sdkVersion:String, requestId:String, pairDeviceData:String, watchPinType:Enum<APP_PIN, WATCH_PIN, NONE>, Deprecated("Not used in v1.1") paymentInstrumentIds : List<String>)	SyncStatusResult(appVersion :String, sdkVersion:String, type:String, requestId:String, error:Error) Errors:
Method for synchronize tokens status between MPA and Watch. Used for update transaction credentials	SyncStatusExtended(type:String, appVersion:String, requestId: String)	SyncStatusExtendedResult(type:String, appVersion:String, sdkVersion:String, requestId:String, chunk: MessageChunk - ChunkedBase64 data (single json to split between next messages): { "paymentInstrumentStatuses":List<PaymentInstrumentStatus> "usedCredentialsInfo":String }	SyncStatusExtendedResult(appVersion :String, sdkVersion:String, type:String, requestId:String, error:Error) Errors: CHUNKED_DATA_PARSE_ERROR
Method for initialize of add new Token on Watch. Method allows to add Token details only, use AddTokenData for add token data.	AddTokenInit(type:String, appVersion::String, requestId: String, token:Token)	AddTokenInitResult(type:String, appVersion:String, sdkVersion:String, requestId:String)	AddTokenInitResult(appVersion:String, sdkVersion:String, type:String, requestId:String, error:Error) Errors: PAYMENT_INSTRUMENT_ALREADY_EXIST, TOKEN_ADD_ERROR

<p>Method for processing of new Token Data</p> <p>When finished MPA should proceed</p> <p>AddTokenCredntialsInit</p>	<p>AddTokenData(type:String, appVersion::String, requestId: String, paymentInstrumentId:String, chunk:MessageChunk)</p> <p>Chunked Base64 data (single json to split between next messages): { a:String,//paymentInstrumentId b:String //encryptedTokenData }</p>	<p>AddTokenDataResult(type:String, appVersion:String, sdkVersion:String, requestId:String)</p>	<p>AddTokenDataResult(appVersion:String, sdkVersion:String, type:String, requestId:String, error:Error)</p> <p>Errors: PAYMENT_INSTRUMENT_ALREADY_EXIST, PAYMENT_INSTRUMENT_NOT_FOUND, CHUNKED_DATA_PARSE_ERROR, TOKEN_ADD_ERROR</p>
<p>Method for initialize of add Token Credentials on Watch.</p>	<p>AddTokenCredentialsInit(type:String, appVersion::String, requestId: String, paymentInstrumentId:String)</p>	<p>AddTokenCredentialsInitResult(type:String, appVersion::String, sdkVersion:String, requestId: String)</p>	<p>AddTokenCredentialsInitResult(type:String, appVersion:String, sdkVersion:String, requestId:String, error:Error)</p> <p>Errors: PAYMENT_INSTRUMENT_NOT_FOUND</p>
<p>Method for processing of add Token Credentials on Watch.</p>	<p>AddTokenCredentialsData(type:String, appVersion::String, requestId: String, chunk:MessageChunk)</p> <p>Chunked Base64 data (single json to split between next messages): { a:String,//paymentInstrumentId b:String //encryptedCredentialsData }</p>	<p>AddTokenCredentialsInitResult(type:String, appVersion:String, sdkVersion:String, requestId: String)</p>	<p>AddTokenCredentialsInitResult(type:String, appVersion:String, sdkVersion:String, requestId:String, error:Error)</p> <p>Errors: PAYMENT_INSTRUMENT_NOT_FOUND, CHUNKED_DATA_PARSE_ERROR, CREDENTIALS_ADD_ERROR</p>
<p>Method for update Token status on Watch</p>	<p>UpdateToken(type:String, requestId:String, paymentInstrumentId:String, status: TokenStatus(DELETED, SUSPENDED, ACTIVE))</p>	<p>UpdateTokenResult(type:String, appVersion:String, sdkVersion:String, requestId:String)</p>	<p>UpdateTokenResult(type:String, appVersion:String, sdkVersion:String, requestId:String, error:Error)</p> <p>Errors: PAYMENT_INSTRUMENT_NOT_FOUND, TOKEN_UPDATE_ERROR</p>

Method for displaying a list of logs on Watch	ShowLogs(type:String, requestId:String)	(no response)	(no response)
Method for collecting logs from Watch	DumpLogs(type:String, requestId:String)	DumpLogsResult(type:String, requestId:String, appVersion:String, sdkVersion:String, logs:List<{ time:String, message:String }>)	DumpLogsResult(type:String, requestId:String, appVersion:String, sdkVersion:String, error:Error) Errors:

Sample

```
//request
{
  "type" : "AddTokenInit" ,
  "appVersion" : "2.0.0" ,
  "requestId" : "d27978a0-5e06-4a05-aef6-b809dcffe53d" ,
  "token" : {Token}
}

//error
{
  "type" : "AddTokenInit" ,
  "appVersion" : "3.0.0" ,
  "requestId" : "d27978a0-5e06-4a05-aef6-b809dcffe53d",
  "error" : "PAYMENT_INSTRUMENT_ALREADY_EXIST"
}
```