

Digital Receipt API specification

Version 1.0.0

OMG Document Number: retail/2022-01-31
Original submission date: Jan. 31, 2022

Standard document URL: <https://www.omg.org/spec/DIGITALRECEIPT/>

USE OF SPECIFICATION - TERMS, CONDITIONS & NOTICES

The material in this document details an Object Management Group specification in accordance with the terms, conditions and notices set forth below. This document does not represent a commitment to implement any portion of this specification in any company's products. The information contained in this document is subject to change without notice.

LICENSES

The companies listed above have granted to the Object Management Group, Inc. (OMG) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version. Each of the copyright holders listed above has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used the specification set forth herein or having conformed any computer software to the specification.

Subject to all of the terms and conditions below, the owners of the copyright in this specification hereby grant you a fully-paid up, non-exclusive, nontransferable, perpetual, worldwide license (without the right to sublicense), to use this specification to create and distribute software and special purpose specifications that are based upon this specification, and to use, copy, and distribute this specification as provided under the Copyright Act; provided that: (1) both the copyright notice identified above and this permission notice appear on any copies of this specification; (2) the use of the specifications is for informational purposes and will not be copied or posted on any network computer or broadcast in any media and will not be otherwise resold or transferred for commercial purposes; and (3) no modifications are made to this specification. This limited permission automatically terminates without notice if you breach any of these terms or conditions. Upon termination, you will destroy immediately any copies of the specifications in your possession or control.

PATENTS

The attention of adopters is directed to the possibility that compliance with or adoption of OMG specifications may require use of an invention covered by patent rights. OMG shall not be responsible for identifying patents for which a license may be required by any OMG specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. OMG specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

IPR Mode

This specification is issued under the **Non-Assert** Mode base on the OMG IPR Policy.
OMG IPR Policy <https://www.omg.org/cgi-bin/doc.cgi?ipr>

GENERAL USE RESTRICTIONS

Any unauthorized use of this specification may violate copyright laws, trademark laws, and communications regulations and statutes. This document contains information which is protected by copyright. All Rights Reserved. No part of this work covered by copyright herein may be reproduced or used in any form or by any means--graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems--without permission of the copyright owner.

DISCLAIMER OF WARRANTY

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE OBJECT MANAGEMENT GROUP AND THE COMPANIES LISTED ABOVE MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE. IN NO EVENT SHALL THE OBJECT MANAGEMENT GROUP OR ANY OF THE COMPANIES LISTED ABOVE BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The entire risk as to the quality and performance of software developed using this specification is borne by you. This disclaimer of warranty constitutes an essential part of the license granted to you to use this specification.

RESTRICTED RIGHTS LEGEND

Use, duplication or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c) (1) (ii) of The Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 or in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clauses at 48 C.F.R. 52.227-19 or as specified in 48 C.F.R. 227-7202-2 of the DoD F.A.R. Supplement and its successors, or as specified in 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors, as applicable. The specification copyright owners are as indicated above and may be contacted through the Object Management Group, 109 Highland Avenue, Needham, MA 02494, U.S.A.

TRADEMARKS

IMM®, MDA®, Model Driven Architecture®, UML®, UML Cube logo®, OMG Logo®, CORBA® and XMI® are registered trademarks of the Object Management Group, Inc., and Object Management Group™, OMG™, Unified Modeling Language™, Model Driven Architecture Logo™, Model Driven Architecture Diagram™, CORBA logos™, XMI Logo™, CWM™, CWM Logo™, IIOPT™, MOF™, OMG Interface Definition Language (IDL)™, and OMG SysML™ are trademarks of the Object Management Group. All other products or company names mentioned are used for identification purposes only, and may be trademarks of their respective owners.

COMPLIANCE

The copyright holders listed above acknowledge that the Object Management Group (acting itself or through its designees) is and shall at all times be the sole entity that may authorize developers, suppliers and sellers of computer software to use certification marks, trademarks or other special designations to indicate compliance with these materials.

Software developed under the terms of this license may claim compliance or conformance with this specification if and only if the software compliance is of a nature fully matching the applicable compliance points as stated in the specification. Software developed only partially matching the applicable compliance points may claim only that the software was based on this specification, but may not claim compliance or conformance with this specification. In the event that testing suites are implemented or approved by Object Management Group, Inc., software developed using this specification may claim compliance or conformance with the specification only if the software satisfactorily completes the testing suites.

OMG's Issue Reporting Procedure

All OMG specifications are subject to continuous review and improvement. As part of this process we encourage readers to report any ambiguities, inconsistencies, or inaccuracies they may find by completing the Issue Reporting Form listed on the main web page <http://www.omg.org>, under Documents, Report a Bug/Issue (http://www.omg.org/report_issue.)

Table of Content

PREFACE	7
1. SCOPE	9
1.1 BACKGROUND	9
1.2 OBJECTIVE OF DIGITAL RECEIPT API SPECIFICATION	9
2. DIRECTION OF THIS SPECIFICATION	12
3. RELATIONSHIP TO OTHER OMG SPECIFICATIONS & ACTIVITIES.....	12
3.1 RELATIONSHIP TO OMG SPECIFICATIONS.....	12
3.2 RELATIONSHIP TO OTHER OMG DOCUMENTS AND WORK IN PROGRESS.....	12
3.3 RELATED NON-OMG ACTIVITIES, DOCUMENTS AND STANDARDS	12
4. ADDITIONAL INFORMATION	13
4.1 ACKNOWLEDGEMENTS.....	14
4.2 MEMBER ROASTER	14
5. DIGITAL RECEIPT API SPECIFICATION	16
5.1. USE OF DIGITAL RECEIPT API	16
5.1.1 BASIC FLOW	16
5.1.2 EXCEPTION FLOW.....	17
5.2. APPLY FOR THE USE OF APPLICATION	19
5.2.1 AUTHORITY OF THE APPLICATION.....	19
5.2.2 USE CASES	19
5.2.2.1 Get a receipt in a membership application.....	19
5.2.2.2 Register Digital Receipt by Membership authority.....	20
5.2.2.3 Register receipt of application with company authority	21
5.2.3 MEMBERSHIP AUTHORITY APPLICATION OVERVIEW.....	21
5.2.3.1 APPLICATION INFORMATION OBTAINING.....	22
5.2.3.2 ACCESS TOKEN OBTAINING	22
5.2.3.3 REQUEST EXAMPLE	22
5.2.4 COMPANY AUTHORITY APPLICATION OVERVIEW	23
5.2.4.1 Application information obtaining	23
5.2.4.2 Access token obtaining	23
5.2.4.3 Request example	23
5.3 API LIST.....	24
5.3.1 DIGITAL RECEIPT API LIST.....	24
5.4 MEMBERSHIP API	25
5.4.1 AUTHENTICATION API	25
5.4.1.1 Request.....	25
5.4.1.2 Response.....	26
5.5 COMPANY API	28
5.5.1 AUTHENTICATION API	28
5.5.1.1 Request.....	28
5.5.1.2 Response.....	29
5.6 RECEIPT API.....	31
5.6.1 API TO GET RECEIPT LIST	31
5.6.1.1 Request.....	31
5.6.1.2 Response.....	32

- 5.6.2.2 RECEIPT DETAIL ACQUISITION API 36
- 5.6.2.1 Request..... 36
- 5.6.2.2 Response..... 37
- 5.6.3 Receipt print character acquisition API 42
- 5.6.3.1 Request..... 42
- 5.6.3.2 Response 43
- 5.6.4 Receipt PDF acquisition API 46
- 5.6.4.1 Request..... 46
- 5.6.4.2 Response 47
- 5.6.5 RECEIPT REGISTRATION API..... 49
- 5.6.5.1 Request..... 49
- 5.6.5.2 Response 53

Preface

OMG

Founded in 1989, the Object Management Group, Inc. (OMG) is an open membership, not-for-profit computer industry standards consortium that produces and maintains computer industry specifications for interoperable, portable, and reusable enterprise applications in distributed, heterogeneous environments. Membership includes Information Technology vendors, end users, government agencies, and academia.

OMG member companies write, adopt, and maintain its specifications following a mature, open process. OMG's specifications implement the Model Driven Architecture® (MDA®), maximizing ROI through a full-lifecycle approach to enterprise integration that covers multiple operating systems, programming languages, middleware and networking infrastructures, and software development environments. OMG's specifications include: UML® (Unified Modeling Language™); CORBA® (Common Object Request Broker Architecture); CWM™ (Common Warehouse Metamodel); and industry-specific standards for dozens of vertical markets.

More information on the OMG is available at <http://www.omg.org/>.

OMG Specifications

As noted, OMG specifications address middleware, modeling and vertical domain frameworks. All OMG Specifications are available from the OMG website at:

<http://www.omg.org/spec>

Specifications are organized by the following categories:

Business Modeling Specifications

Middleware Specifications

- 1 CORBA/IIOP
- 2 Data Distribution Services
- 3 Specialized CORBA

IDL/Language Mapping Specifications

Modeling and Metadata Specifications

- 4 UML, MOF, CWM, XMI
- 5 UML Profile

Modernization Specifications

Platform Independent Model (PIM), Platform Specific Model (PSM), Interface Specifications

- 6 CORBAServices
- 7 CORBAFacilities

CORBA Embedded Intelligence Specifications

CORBA Security Specifications

All of OMG's formal specifications may be downloaded without charge from our website. (Products implementing OMG specifications are available from individual suppliers.) Copies of specifications, available in PostScript and PDF format, may be obtained from the Specifications Catalog cited above or by contacting the Object Management Group, Inc. at:

OMG Headquarters
109 Highland Avenue
Needham, MA 02494
USA
Tel: +1-781-444-0404
Fax: +1-781-444-0320
Email: pubs@omg.org

Certain OMG specifications are also available as ISO standards. Please consult <http://www.iso.org>

Typographical Conventions

The type styles shown below are used in this document to distinguish programming statements from ordinary English. However, these conventions are not used in tables or section headings where no distinction is necessary.

Times/Times New Roman - 10 pt.: Standard body text

NOTE: Terms that appear in italics are defined in the glossary. Italic text also represents the name of a document, specification, or other publication.

Issues

The reader is encouraged to report any technical or editing issues/problems with this specification to http://www.omg.org/report_issue.htm.

1. Scope

This document is a specification document on the standard Digital Receipt API, which has been formulated as a standard specification by the Digital Receipt Subcommittee of the .NET Retail System Council Japan as part of a contract project of the Ministry of Economy, Trade and Industry of Japan.

This specification can connect POS terminals for various business types of business of various vendors by standardizing APIs related to transmission and reception of digital receipt data between digital receipt server and POS terminal, and between digital receipt server and smart devices like smart phones. The purpose is to reduce development costs and system integration costs. Another object is to make it easier to use digital receipt data by making it easier to create a smartphone application that uses digital receipt data.

1.1 Background

Competition between traditional retailers as well as between traditional retailers and online retailers is intensifying. The brick-and-mortar retailer faces challenges that their online competitors can more easily address, such as the need to improve operational efficiency, provide better customer service, and offer new customer experiences.

The development of Point of Sale (POS) systems nowadays includes cashless systems using electronic money, Quick Response (QR) Code settlement, receipt digitization, scanless technology with electronic tags or image recognition, serverless systems leveraging cloud computing, support for smartphones on consumer services, etc.

The digital receipt offering is a standardized format and API to post and retrieve digital receipt data which is essential for customers and retailers to maximize the value of the data.

The current standard today is the [ARTS Digital Receipt standard v3.1.0](#). It is a subset of Point of Sale Log (POSLog)¹ chosen for the specific needs of a receipt. The Digital Receipt was envisioned to be applicable in non-traditional retail channels, so structures were added for catalogue and internet transactions. One key benefit of the Digital Receipt is its potential for channel integration: a common Digital Receipt can be provided in all channels, and a receipt from one channel can include links and promotions for others. The original specification includes the many differences of various tax systems, institutions, business habits etc., which represents a comprehensive amount of transaction data but is difficult for retailers to implement.

1.2 Objective of Digital Receipt API Specification

This document is a specification document on the standard Digital Receipt API, which has been formulated as a standard specification by the Digital Receipt Subcommittee of the .NET Retail System Council Japan as part of a contract project of the Ministry of Economy, Trade and Industry of Japan

¹ https://www.omg.org/retail-depository/arts-odm-73/arts_transaction_concepts.htm

This specification can connect POS terminals for various business types of business of various vendors by standardizing APIs related to transmission and reception of digital receipt data between digital receipt server and POS terminal, and between digital receipt server and smart devices like smart phones. The purpose is to reduce development costs and system integration costs. Another object is to make it easier to use digital receipt data by making it easier to create a smartphone application that uses digital receipt data.

This new standard can specify a streamlined common core of elements to be used as a digital receipt API, which can then be enriched and extended in a standard fashion with local requirements.

The figure 1 shows the necessary relations between digital receipt standards and the digital receipt Application Programming Interface (API) that is specified in this specification.

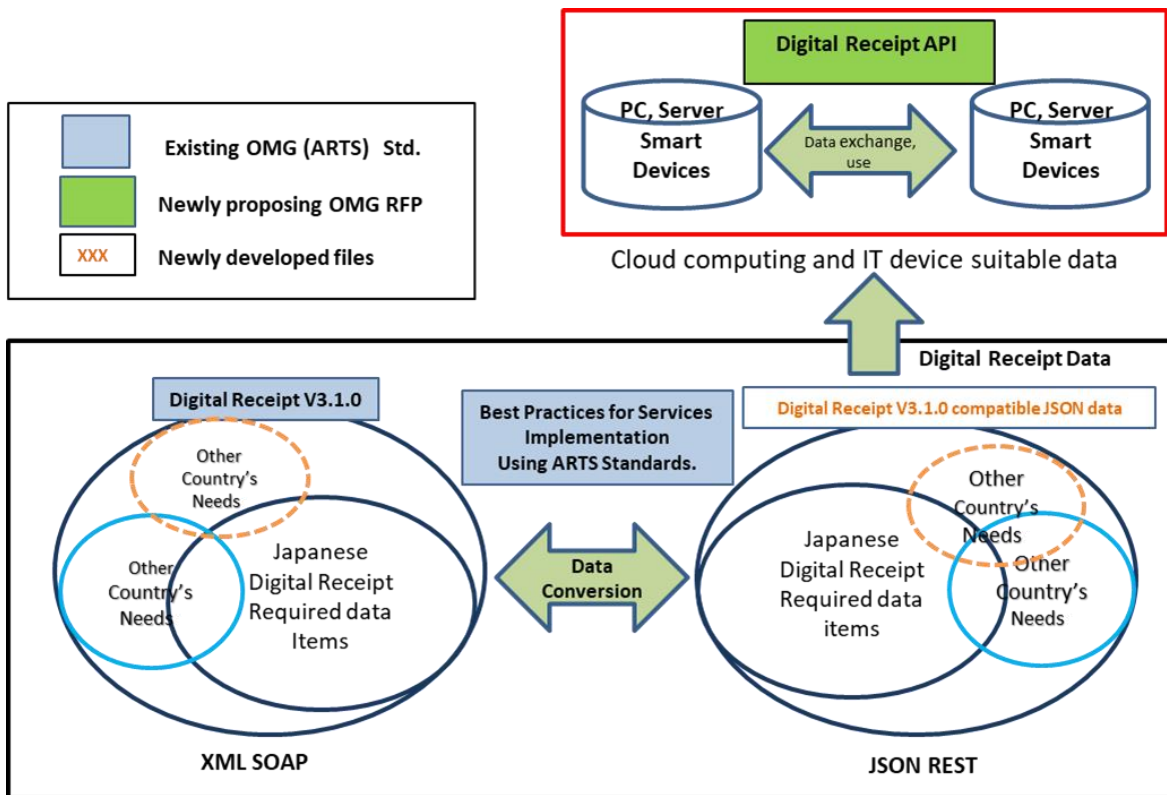


Figure 1 - Schematic of digital receipt standard relationship and API

Digital Receipt standard can play an important role in allowing retailers to offer personalized, relevant communications. For example, a retailer would be able to securely post a receipt to a digital receipt data center; this customer could then grant permission to specific users or groups, such as retailers and other users, to access this repository in return for certain considerations such as offers and improved personalization as shown in Figure 2

Digital Receipt data Information Flow Chart

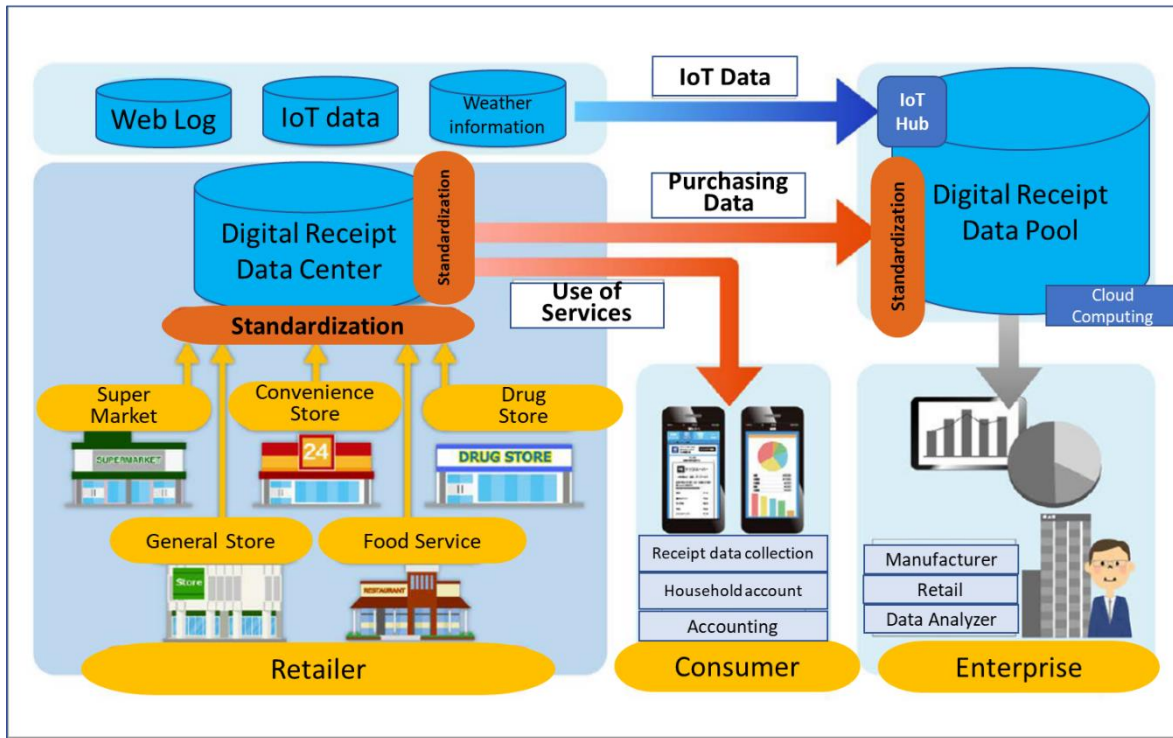


Figure 2 - Digital Receipt data information flow chart

To make those happen, we would like to propose Digital Receipt API specification in here including those.

- 1) Authentication API
Authenticate user/entity and issues an authentication token.
- 2) Get Receipt List API
Get the receipt list information for the authenticated member.
- 3) Get Receipt Detail API
Get the receipt detail information for the authorized member's specified receipt Id.
- 4) Get Receipt Print String API
Get the receipt print string for the authorized member's specified receipt Id.
- 5) Get Receipt PDF API
Get the receipt PDF file for the authorized member's specified receipt Id.
- 6) Receipt Registration API
Register receipt information.

In addition to the standard APIs, the standard should specify an ontology of the common elements in a digital receipt, relationships among them, and axioms that distinguish the concepts from one another.

Use of a logically consistent ontology will ensure unambiguous communications among the services that use the APIs, and provide formal definitions that can be used by business analysts as a component of an enterprise glossary supporting data management and governance processes. This ontology shall be specified in a decidable, standard ontology

language, namely the W3C Web Ontology Language (OWL 2). OWL 2 can be used together with the W3C SPARQL query language, various rule languages, and other applications to support services that leverage the APIs and digital receipt content.

2. Direction of this specification

Direction of this specification is to create a standard API for digital receipts in today's retail environment, where transactions at the point of sale need to be completely paperless, and to facilitate the adoption of digital receipts by enabling them to be applied to modern technology environments using JSON and REST.

3. Relationship to other OMG Specifications & activities

3.1 Relationship to OMG specifications

Unified Modeling Language™ (UML™): <http://www.omg.org/spec/UML/> can be used to develop diagrams for the various digital receipt use cases and provide users with ready-to-use, expressive modeling examples for modeling a system independent of a platform language.

Digital Receipt spec. V3.1.0 <https://www.omg.org/cgi-bin/doc?retail/2018-04-01> provides detailed use-cases across an extensive variety of environments while taking into account many existing schemas providing valuable information to enable integration.

The Ontology Definition Metamodel [ODM] provides a metamodel and profiles supporting RDF 1.1 and OWL 2, enabling representation of the digital receipt ontology that is based on Meta Object Facility Specification [MOF] and is Unified Modeling Language Specification [UML] compliant.

3.2 Relationship to other OMG Documents and work in progress

RDTF Operational Data Model [ODM]: <http://www.omg.org/retail/operational-data-model.htm> is a retail-specific roadmap for database design that allows a company to plan how it organizes and uses its transaction data. Meeting the needs of digital receipt, the ODM enables retailers to describe and understand customer behavior, to track the consumer-to-customer lifecycle.

Best Practices for Services Implementation Using ARTS Standards <https://www.omg.org/cgi-bin/doc?retail/2019-02-05> provides the best guidance for transforming the original XML SOAP use cases into the desirable JSON REST data while maintaining compliance.

3.3 Related non-OMG Activities, Documents and Standards

Point of Sale Log (POSLog):

https://www.omg.org/retail-depository/arts-odm-73/arts_transaction_concepts.htm

Digital Receipt API V1.0.0 specification

Representational State Transfer (REST)

https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

The JavaScript Object Notation (JSON) Data Interchange Format

<https://tools.ietf.org/html/rfc8259>JSON Web Token (JWT) <https://tools.ietf.org/html/rfc7519>Hypertext Transfer Protocol (HTTP): <https://tools.ietf.org/html/rfc2616/>XML Schema Definition (XSD): <https://www.w3.org/standards/xml/schema/>JSON Schema: <http://json-schema.org/>OpenAPI Specification: <https://github.com/OAI/OpenAPI-Specification/>RESTful API Modeling Language (RAML): <https://raml.org/>API Blueprint [APIBP]: <https://apiblueprint.org/>OAuth 2.0 Authorization Framework: <https://tools.ietf.org/html/rfc6749/>Transport Layer Security (TLS): <https://tools.ietf.org/html/rfc5246/>

OWL 2 [OWL2] is a W3C recommendation for ontology specification and is required for implementation of the digital receipt ontology. The OMG's Ontology Definition Metamodel (ODM) specification provides a representation for OWL 2 in UML that may be used by submitters for design and documentation purposes.

OWL-S [OWLS] is a widely used member submission to W3C that might inform the development of the interface aspects of the ontology.

Digital Receipt Data Items Reference List for Japanese Market

Digital Receipt Data Format Specification JSON Version for Japanese Market

Digital Receipt RFP

<https://www>

RFC 7519 - JSON Web Token (JWT)

<https://tools.ietf.org/html/rfc7519>

RFC 6750 - The OAuth 2.0 Authorization Framework: Bearer Token Usage

<https://tools.ietf.org/html/rfc6750>

RFC 7235 - Hypertext Transfer Protocol (HTTP/1.1): Authentication

<https://tools.ietf.org/html/rfc7235>

4. Additional Information

4.1 Acknowledgements

- The following company submitted this specification:
Toshiba GCS
- The following organization contributed this specification
Leader
- NET Retail System Council
"Digital receipt subcommittee"
- Cooperating organization
General Federated Open Food Service Systems Consortium
"Digital receipt subcommittee"
General Association Fintech Association
"Digital receipt subcommittee"
General Association XBRL Japan

4.2 Member Roaster

Chairman

Masanori Sambe	TOSHIBA TEC CORPORATION
----------------	-------------------------

Contributors:

Jin Hashiba	Dai Nippon Printing Co., LTD
Daisaku Kashima	Dai Nippon Printing Co., LTD
Yasuo Sakami	Foresight Co.,Ltd.
Yoshio Yamada	FUJITSU LIMITED
Akira Matsuyoshi	Insight Corporation
Koji Sasaki	LOGNOTE Co.,Ltd.
Soichi Fujii	Microsoft Japan Co., Ltd.
Akio Tajima	NCR Japan, Ltd.
Kazutoshi Ota	OK Systems
Tadashi Furuhata	Seiko Epson Corporation
Takashi Wakatsuki	Sharp Marketing Japan Corporation
Yusuke Udagawa	SHARP CORPORATION
Takao Tamura	Transaction Media Networks, Inc.
Eiki Murakami	Sorimachi Giken Co.,Ltd
Kenichi Nagai	STAR MICRONICS CO., LTD.
Masaaki Morishita	STAR MICRONICS CO., LTD.
Ayako Shimazaki	The Distribution System Research Institute
Takahide Kubota	TOSHIBA TEC CORPORATION
Toyohiro Yasumoto	VINX CORP.

Secretariat

Soichi Fujii	Microsoft Japan Co, Ltd.

Food Service Specification contributors :

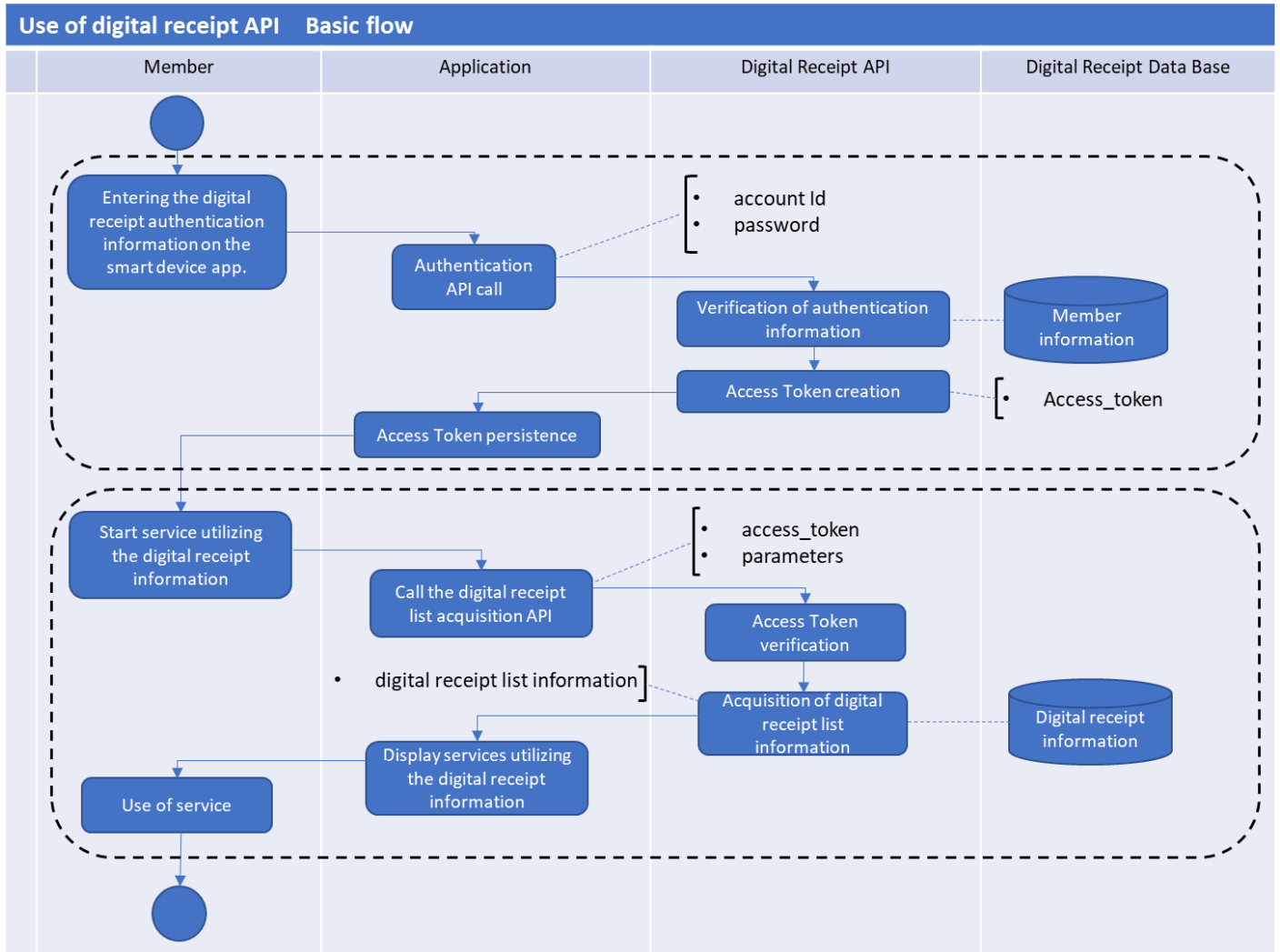
Tadashi Furuhashi	Seiko Epson Corporation
Yoshio Ohba	Seiko Instruments Inc.
Yuu Kusama	Open Foodservice Systems Consortium (OFSC)
Kenji Oohashi	DYNAC CO. , LTD.
Umetsu Noriyuki	Fujitsu Isotec Limited
Shinya Oikawa	Public university corporation Miyagi university
Kazuhiro Kobayashi	Teraoka Seiko Co., Ltd.
Masanori Murai	ASKA-T3 Co., Ltd.
Shigeru Okada	Open Foodservice Systems Consortium(OFSC)
Jun Watanabe	NEC Platforms Limited
Tsuneo Yashima	Just planning Inc.
Kiyoshi Kurokura	CITIZEN SYSTEMS JAPAN CO. LTD.
Koichi Shibata	Seiko Solutions Inc.
Ohko Fujii	MOS FOOD SERVICES, INC.
Kenichi Nagai	STAR MICRONICS CO., LTD.
Makoto Yoshimura	TOSHIBA TEC CORPORATION
Naonori Nagata	TOSHIBA TEC CORPORATION
Yasuo Sakami	Open Foodservice Systems Consortium (OFSC)

5. Digital Receipt API Specification

5.1. Use of Digital Receipt API

5.1.1 Basic Flow

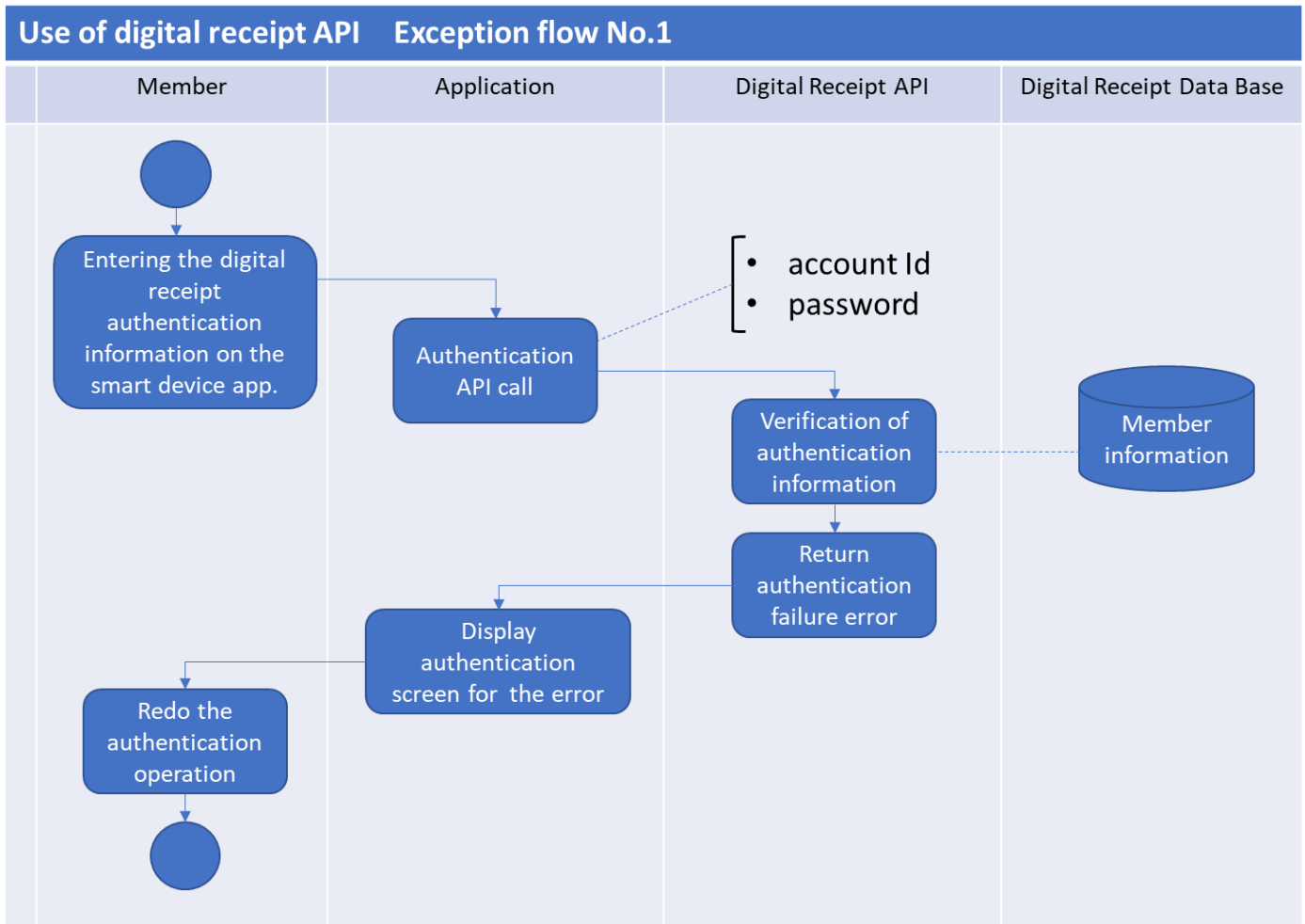
The basic flow is shown in the figure below about DigitalReceipt API. At the beginning of



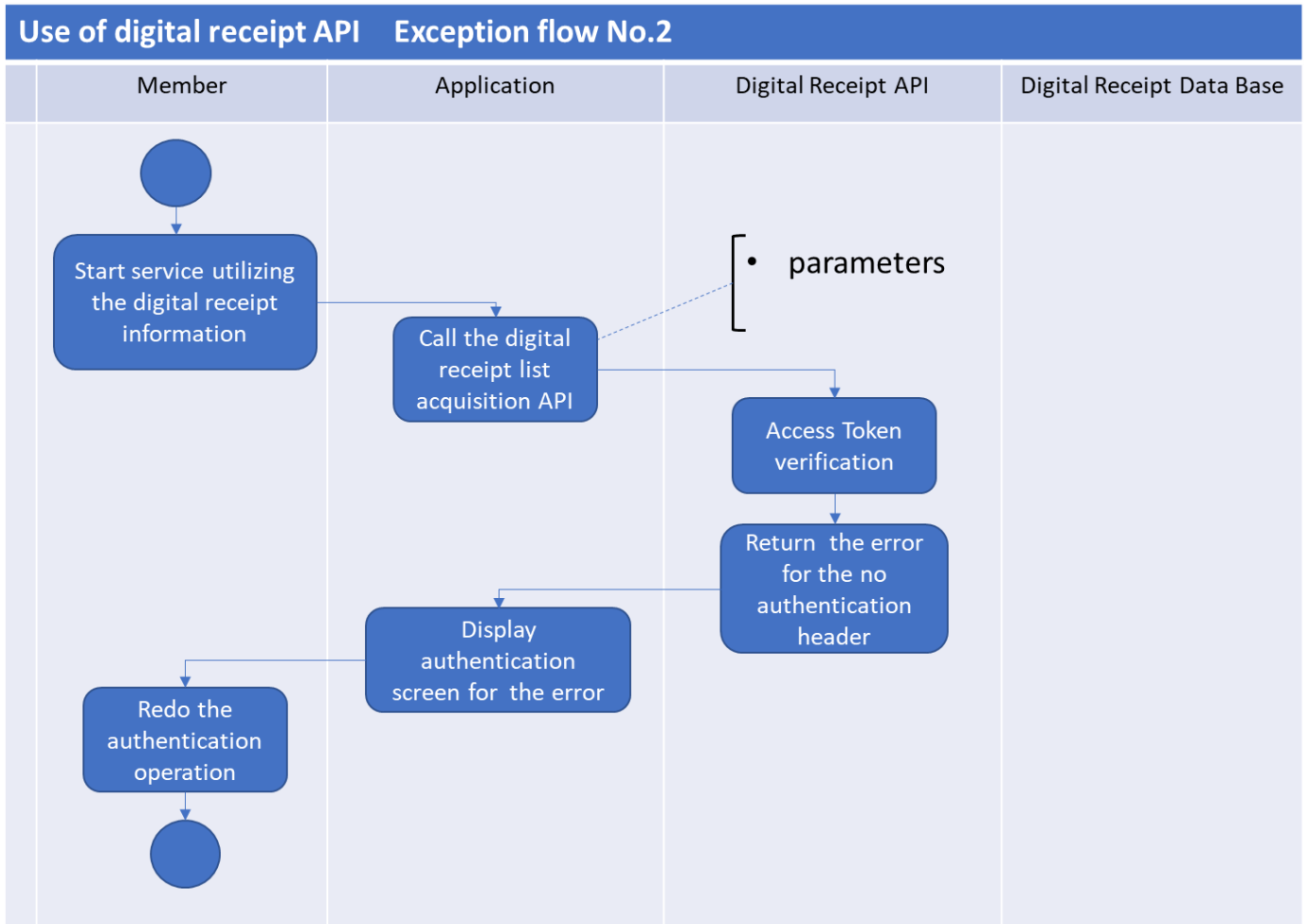
the basic flow, there is a flow that DigitalReceipt Member (Consumers) use the Authentication API through the applications to obtain Access Token. Next, there is a flow to get the Digital Receipt using the Access Token.

5.1.2 Exception flow

Two examples are shown below as an exception flow about Digital Receipt API.s a first example, the flow when authentication fails with Authentication API is shown in the figure below.



- As a second example, the flow when no Access Token is set with Get Receipt List API is shown in the figure below.



5.2. Apply for the use of application

In order to ensure security, Provide a mechanism for applications (Bookkeeping software, accounting software, health management software, etc.) to apply for Digital Receipt API usage in advance.

5.2.1 Authority of the Application

Applications authority are two types as below.

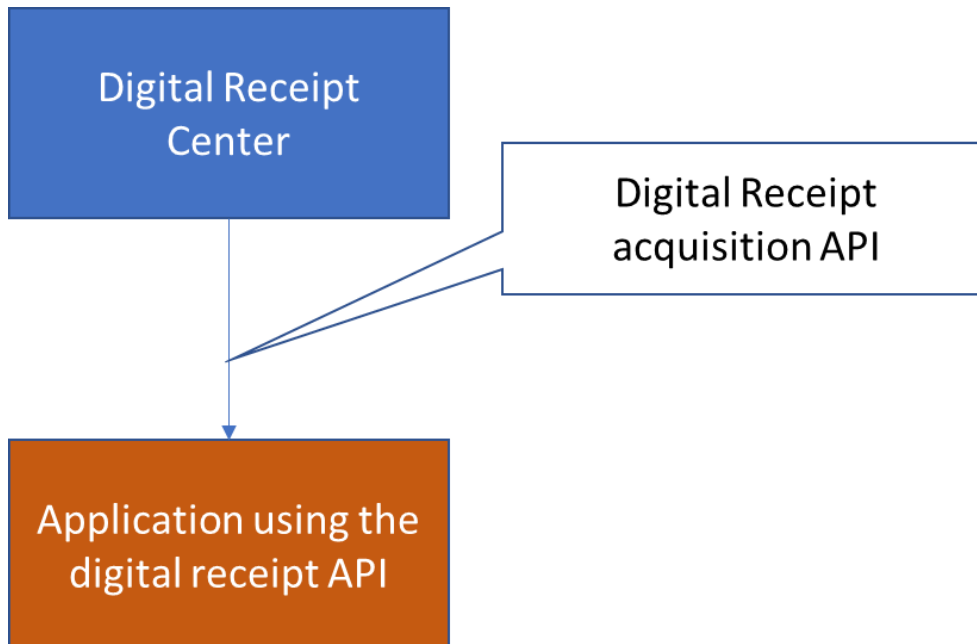
- Membership authority: access permission as applications of the consumer.
- Company authority: access permission as applications of the retailers etc.

5.2.2 Use cases

- A general use case for the Membership and Company authority introduce will.

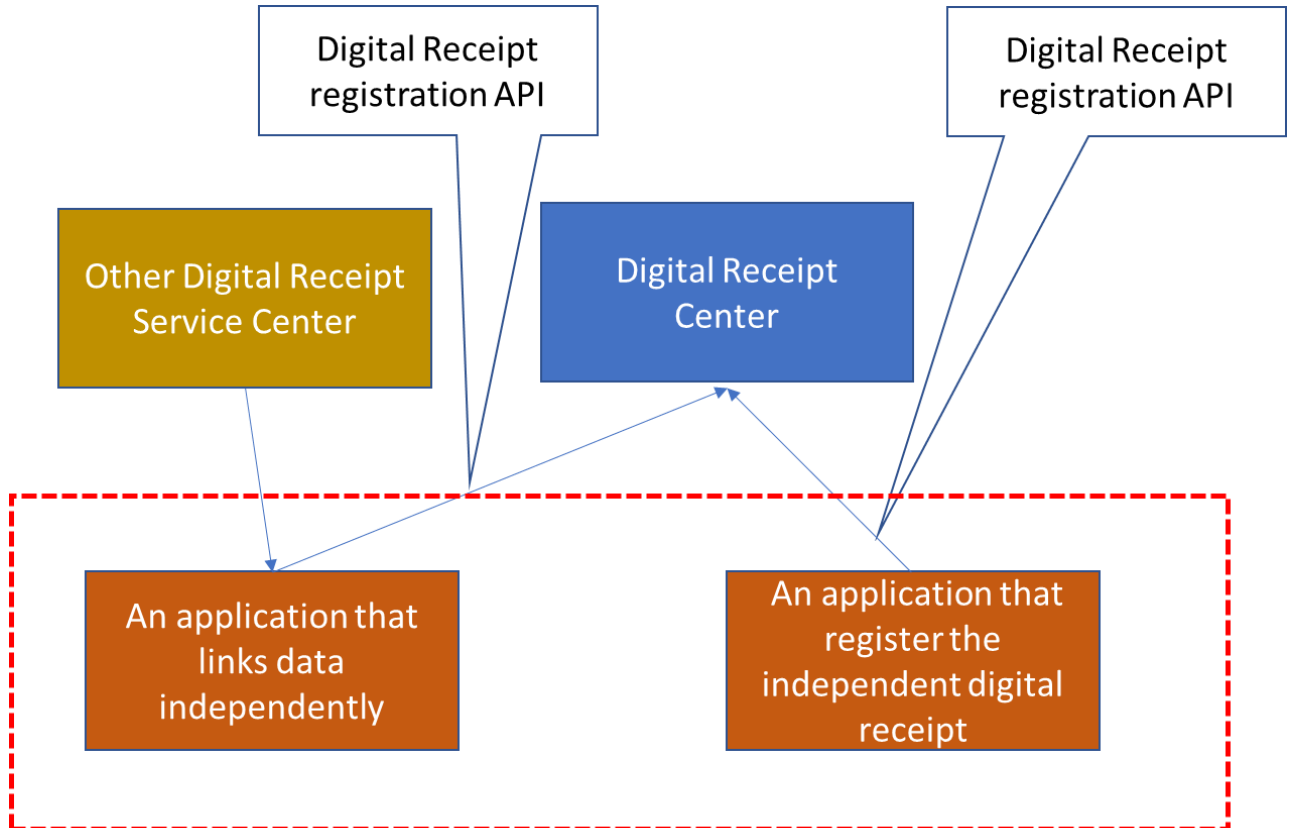
5.2.2.1 Get a receipt in a membership application

- Membership authority can obtain Digital Receipt.
- Both the Electronic Receipts registered with the membership authority and the company authority can be obtained (pictured below).
- However, when using the Receipt PDF acquisition API, only Electronic Receipts registered with Company authority can be obtained.



5.2.2.2 Register Digital Receipt by Membership authority

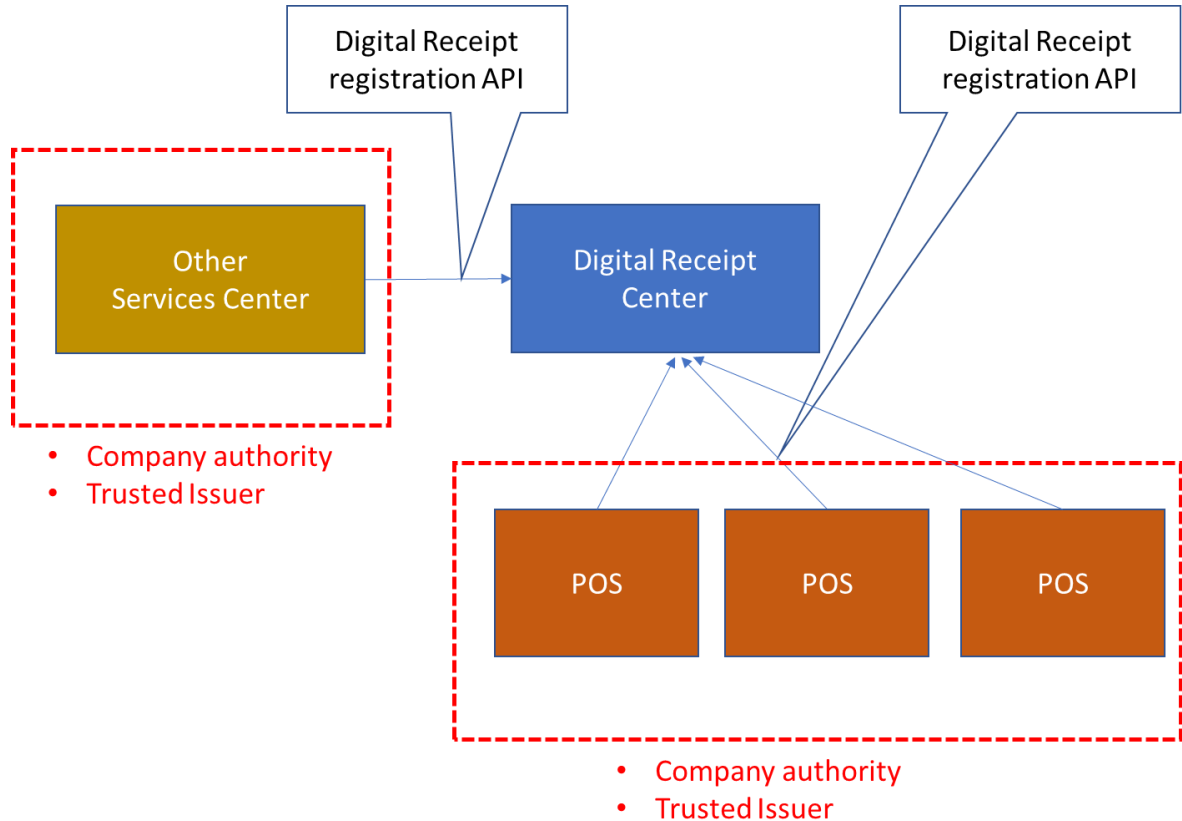
- Membership authority can register Electronic Receipts.
- Digital Receipt registered from Membership authority will be recorded as receipts with unknown issuer sources in the Digital Receipt Center. Because it does not have the Company authority of the receipt issuing company.
- As a use case for registering Digital Receipt with Membership authority, Registration from other services not cooperating with the Electronic Receipt Center, registration of the receipt independently created by the household account book software, etc. (below)



- Membership authority
- Issuer is confidential

5.2.2.3 Register receipt of application with company authority

- Company authority applications can only register receipts.
- The receipt will be registered as a receipt of the trusted publisher with the Digital Receipt Center.
- As a use case to use company authority, register receipt information with the Digital Receipt Center from other services or POS described as below.
- In addition, company authority application is not allowed to obtain receipts.



- When both of "Membership Authority" and "Company Authority" application register the receipt that have a same ID, the receipt of the "Company Authority" is valid.
- In addition, when register the receipt multiple times with the same ID and same authority, the last receipt is valid.

5.2.3 Membership authority application overview

- The application of the membership authority can obtain the receipt of the authorized member and issue the receipt of unknown publisher.
- The receipt registered in the application of the membership authority cannot be used as public documents like receipt PDF file with time stamp.

5.2.3.1 Application information obtaining

- In order to ensure security, provide ID (application ID) and application authentication function.
- Use Application ID to identify application uniquely.
- Application authentication function uses secret strings to identify the application.

5.2.3.2 Access token obtaining

- When membership authority application use API, need to retrieve access token using authentication API of membership authority then use API with the access token.

5.2.3.3 Request example

- Authentication API example is shown below.

Method	POST			
URI	/members/auth			
Headers	Content-Type: application/json			
Body				
Property Name	Type	Size	Required	Remarks
applicationId	String	20	○	Uniquely identify an application ID.
applicationSecret	String	40	○	A secret string to identify the application.
accountId	String	1-256	○	The phone number or email address to log in to the Digital Receipt system.
password	String	1-256	○	The password used to log in to the Digital Receipt system.
Body Example				
<pre>{ "applicationId": {APPLICATION_ID}, "applicationSecret": {APPLICATION_SECRET}, "accountId": "name@example.jp", "password": "password" }</pre>				

Need to set application ID and application secret that is published / managed by System Management Company to "APPLICATION_ID" and "APPLICATION_SECRET".

5.2.4 Company authority application overview

- A company authority application can register a receipt for an authorized company.
- The receipt registered in the application of the company authority can be used as public documents like receipt PDF file with time stamp.

5.2.4.1 Application information obtaining

- Application ID is used for uniquely identifies the application. You cannot change the ID once it is registered.
- The application secret is a secret string that you use to identify your application.

5.2.4.2 Access token obtaining

- When company authority application use API, need to retrieve access token using authentication API of company authority then use API with the access token.

5.2.4.3 Request example

- Authentication API an example is shown below.

Method	POST			
URI	/companies/auth			
Headers	Content-Type: application/json			
Body				
Property Name	Type	Size	Required	Remarks
applicationId	String	20	○	Uniquely identify an application ID.
applicationSecret	String	40	○	A secret string to identify the application.
companyCode	String	13	○	Company code.
password	String	1-256	○	The password that is issued by the Digital Receipt system.
Body Example				
<pre>{ "applicationId": {APPLICATION_ID}, "applicationSecret": {APPLICATION_SECRET}, "companyCode": "0000000000001", "password": "password" }</pre>				

Need to set application ID and application secret that is published / managed by System Management Company to "APPLICATION_ID" and "APPLICATION_SECRET"

5.3 API list

5.3.1 Digital Receipt API list

- A list of Digital Receipt APIs is shown below.

API Name	URI	Method	Overview
Membership Authentication API *1	/members/auth	POST	It authenticates the member and issues an authentication token.
Company Authentication API *1	/companies/auth	POST	It authenticates the company and issues an authentication token.
Get Receipt List API *1	/receipts	GET	Get the receipt list information for the authenticated member.
Get Receipt Detail API *1	/receipts/{receiptId}	GET	Get the receipt detail information for the authorized member's specified receiptId.
Get Receipt Print String API *1	/receipts/{receiptId}/stringArray	GET	Get the receipt print string for the authorized member's specified receiptId.
Get Receipt PDF API *1	/receipts/{receiptId}/pdf	GET	Get the receipt of PDF file for the authorized member's specified receiptId
Receipt Registration API *2	/receipts	POST	Register receipt information.

*1 URI example:<https://exp-openapi.sampledomain.jp/v1> Write it so as to follow after.

*2 URI example:<https://exp-public-receiver.sampledomain.jp/srr> Write it so as to follow after.

Special Notes

None

5.4 Membership API

- It is an API on the premise that it is used by consumer used Digital Receipt application.

5.4.1 Authentication API

- Digital Receipt member certification.
- If the authentication succeeds, issue the access token of the receipt API.
- Arbitrary Token technology is available for access token.

5.4.1.1 Request

Method	POST			
URI	/members/auth			
Headers	Content-Type: application/json			
Body				
Property Name	Type	Size	Required	Remarks
applicationId	String	20	○	The ID that uniquely identifies the application.
applicationSecret	String	40	○	A secret string to identify the application.
accountId	String	1-256	○	This is the phone number or e-mail address you use to log in to the Digital Receipt system.
password	String	1-256	○	This is the password used to log in to the Digital Receipt system.
Body Example				
<pre>{ "applicationId": {APPLICATION_ID}, "applicationSecret": {APPLICATION_SECRET}, "accountId": "name@example.jp", "password": "password" }</pre>				

5.4.1.2 Response

Success

Status Code	200			
Headers	Content-Type: application/json			
Body				
Property Name	Type	Size	Required	Remarks
access_token	String	-	○	This is token.
Body Example				
<pre>{ "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoiYWRtaW4iOnRydWV9.TjVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ" }</pre>				

Error

Status Code	4xx			
Headers	Content-Type: application/json			
Body				
Property Name	Type	Size	Required	Remarks
error	Object	-	○	Error object.
message	String	-	○	Error message.
type	String	-	○	Error type.
Body Example				
<pre>{ "error": { "message": "Error message.", "type": "ErrorType" } }</pre>				

Error content

Status Code	Type	Message	Remarks
400	ApiVersionError	Application is not supported for this API version.	Incorrect API version
401	MemberAuthError	Account ID or password is incorrect.	Error in member authentication information.
403	AccountLockOutError	Account is locked out.	Account lockout due to continuous failure of membership authentication.
401	ApplicationAuthError	Application ID or application secret is incorrect.	Error in application authentication information.
500	DataAccessError	Data Access Error.	Data cannot be accessed within the application service.
500	InternalServiceError	Internal Service Error.	Some error occurred inside the application service.

Special Notes

None

5.5 Company API

- It is an API based on the assumption that it is used by company applications (POS etc.).

5.5.1 Authentication API

- The company authenticates.
- If the authentication succeeds, issue an authentication token of the receipt API. Arbitrary Token technology is available for access token.

5.5.1.1 Request

Method	POST			
URI	/companies/auth			
Headers	Content-Type: application/json			
Body				
Property Name	Type	Size	Required	Remarks
applicationId	String	20	○	Uniquely identify an application ID.
applicationSecret	String	40	○	A secret string to identify the application.
companyCode	String	13	○	The company code.
password	String	1-256	○	The password that is issued by the Digital Receipt system.
Body Example				
<pre>{ "applicationId": {APPLICATION_ID}, "applicationSecret": {APPLICATION_SECRET}, "companyCode": "0000000000001", "password": "password" }</pre>				

5.5.1.2 Response

Success

Status Code	200			
Headers	Content-Type: application/json			
Body				
Property Name	Type	Size	Required	Remarks
access_token	String	-	○	This is token.
Body Example				
<pre>{ "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibm90IjoiZSI6IkpvaG4gRG9lIiwiaWF0IjoiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMhRHDcEfxjoYZgeFONFh7HgQ" }</pre>				

Error

Status Code	4xx			
Headers	Content-Type: application/json			
Body				
Property Name	Type	Size	Required	Remarks
error	Object	-	○	Error object.
	String	-	○	Error message.
type	String	-	○	Error type.
Body Example				
<pre>{ "error": { "message": "Error message.", "type": "ErrorType" } }</pre>				

Error content

Status Code	Type	Message	Remarks
400	ApiVersionError	Application is not supported for this API version.	Incorrect API version
401	CompanyAuthError	Company code or password is incorrect.	Error in company authentication information
401	ApplicationAuthError	Application ID or application secret is incorrect.	Error in application authentication information
500	DataAccessError	Data Access Error.	Data cannot be accessed within the application service.
500	InternalServiceError	Internal Service Error.	Some error occurred inside the application service.

Special Notes

None

5.6 Receipt API

- It is the API concerning acquisition of Digital Receipt information.

5.6.1 API to get receipt list

- We will get electronic receipt list information of members who have been authenticated by Token.
- Query parameters can be used as a narrowing condition of list information.

5.6.1.1 Request

Method	GET			
URI	/receipts			
Headers	Authorization: Bearer {Token}			
Query Parameters				
Parameter name	Type	Size	Required	Remarks
from	String	14	×	Specify the transaction date and time after the specified date and time. Corresponds to "ReceiptDateTime" in standard Digital Receipt format.
to	String	14	×	Specify the transaction date before the specified date and time. Corresponds to "ReceiptDateTime" in standard Digital Receipt format.
Query Parameters Example				
/receipts?from=20180201000000&to=20180228235959				

5.6.1.2 Response

Success

Status Code	200
Headers	Content-Type: application/json
Body	
It is an object with an array of receipt information in standard format. For the items, see the example below and the ARTS Digital Receipt Specification v3.1.0.	
Body Example	
<pre>{ "DigitalReceiptIndex": [{ "DigitalReceipt": { "@@MajorVersion": "3", "ReceiptID": "201801011000000000000100000100000000010001", "Transaction": [{ "@@TypeCode": "SaleTransaction", "BusinessUnit": [{ "UnitID": { "#value": "Branch Name" } }], "WorkstationID": { "#value": "122-1" }, "ReceiptDateTime": { "#value": "2018-09-22T09:23:00" }, "ReceiptNumber": { "#value": "123456789999" }, "InvoiceNumber": [{ "#value": "4565" }, { "#value": "3545" }], "RetailTransaction": [{</pre>	

}

Error content

Status Code	Type	Message	Remarks
400	ApiVersionError	Application is not supported for this API version.	Incorrect API version
400	RequestParametersInvalidError	Request parameters are invalid or missing.	Invalid request parameter
401	TokenMissingError	Token is required.	There is no Authorization header or Bearer scheme
401	TokenIncorrectError	Token is incorrect.	Token is illegal
401	TokenExpiredError	Token is expired.	Token Expired
500	DataAccessError	Data Access Error.	Data cannot be accessed within the application service.
500	InternalServiceError	Internal Service Error.	Some error occurred inside the application service.

Special Notes

None

5.6.2.2 Receipt Detail Acquisition API

- This allows the application to get receipt details specified by "receiptId" which is certified by Token.
- "receiptId" means "DigitalReceipt.ReceiptID" of Receipt list acquisition API. This ID can identify a receipt. It should be issued by the system automatically. Receipt details include company information, store information, cashier number, receipt number, transaction date and time, purchase information, store logo image, and promotional bitmap image.

5.6.2.1 Request

Method	GET
Sites	/receipts/{receiptId}
Headers	Authorization: Bearer {Token}

5.6.2.2 Response

Normal

Status Code	200
Headers	Content-Type: application/json
Body	
Receipt information as a standard format. For items, refer to the following examples and ARTS Digital Receipt Specification v3.1.0.	
Example for Body	
<pre>{ "DigitalReceipt": { "@@MajorVersion": "3", "Transaction": [{ "BusinessUnit": [{ "UnitID": { "#value": "Branch Name" } }], "WorkstationID": { "#value": "122-1" }, "ReceiptDateTime": { "#value": "2018-09-22T09:23:00" }, "ReceiptNumber": { "#value": "123456789999" }, "InvoiceNumber": [{ "#value": "4565" }, { "#value": "3545" }], "RetailTransaction": [{ "LineItem": [{ "Sale": { "ItemID": [</pre>	

```
{
  "@@Name": "Lunch Set A",
  "#value": "123"
},
"ExtendedAmount": {
  "#value": 1080
},
"SequenceNumber": [
  1
],
},
{
  "Sale": {
    "ItemID": [
      {
        "@@Name": "Lunch Set B",
        "#value": "124"
      }
    ],
    "ExtendedAmount": {
      "#value": 1296
    },
    "SequenceNumber": [
      2
    ],
  },
  "Tender": {
    "Amount": {
      "#value": 3000
    },
    "TenderChange": [
      {
        "Amount": {
          "#value": 624
        }
      }
    ]
  },
  "SequenceNumber": [
    3
  ],
},
{
```

```
"Advertising": {
  "AdvertisingID": "444",
  "ImageData": ["8999"],
  "ImageNumber": ["999"],
  "ImageURI": ["https://example.jp/image.jpg" ],
  "Text": ["888"],
  "Code": ["99"],
  "Barcode":
["\\u0034\\u0039\\u0030\\u0031\\u0032\\u0033\\u0034\\u0035\\u0036\\u0037\\u0038\\u0039\\u0034" ],
  "URI": ["https://example.jp/"]
}
},
"Total": [
{
"@@TotalType": "TransactionGrossAmount",
"#value": 2376
},
{
"@@TotalType": "TransactionNetAmount",
"#value": 2376
},
{
"@@TotalType": "TransactionGrandAmount",
"#value": 2376
},
{
"@@TotalType": "TransactionTaxIncluded",
"#value": 176
}
]
}
]
}
]
```

Error

Status Code	4xx
Headers	Content-Type: application/json
Body	

Property name	Type	Size	Have to	Remarks
error	Object	-	○	Error object.
message	String	-	○	Error message.
type	String	-	○	Error type.
Body Example of				
<pre>{ "error": { "message": "Error message.", "type": "ErrorType" } }</pre>				

Error content

Status Code	Type	Message	Remarks
400	ApiVersionError	Application is not supported for this API version.	Incorrect API version
401	TokenMissingError	Token is required.	Authorization header does not exist. Bearer scheme does not exist.
401	TokenIncorrectError	Token is incorrect.	Token is illegal
401	TokenExpiredError	Token is expired.	Token Expired
404	ReceiptNotExistError	Receipt not exists.	The specified receipt does not exist
500	DataAccessError	Data Access Error.	Data cannot be accessed within the application service.
500	InternalServiceError	Internal Service Error.	Some error occurred inside the application service.

Special Notes
None

5.6.3 Receipt print character acquisition API

- This API get the printing character string of the Digital Receipt, which is specified by the receiptId of the members, authenticated by Token. receiptId indicates DigitalReceipt.ReceiptID, which is included in the response of receipt list acquisition API.
- This ID is an ID to specify the receipt uniquely, and it is assumed to be issued automatically by the system. It does not include a standard print command which is sent to the printer from a standard POS (for example, Code which is called Escape sequence etc.).

5.6.3.1 Request

Method	GET
URI	/receipts/{receiptId}/stringArray
Headers	Authorization: Bearer {Token}

5.6.3.2 Response

Success

Status Code	200
Headers	Content-Type: application/json
Body	<p>Receipt printing character string, store logo, promotional images.</p> <p>Please refer ARTS Digital Receipt Specification v3.1.0 and the example below for more information.</p>
Body Example	<pre>{ "DigitalReceipt": { "@@MajorVersion": 3, "@@MinorVersion": 1, "@@FixVersion": 0, "Transaction": [{ "ReceiptImage": [{ "ReceiptLine": [{ "#value": " RECEIPT_TEXT " }, { "#value": " RECEIPT_TEXT " }, { "#value": " RECEIPT_TEXT " }, { "#value": " RECEIPT_TEXT " }, { "#value": " RECEIPT_TEXT " }, { "#value": " RECEIPT_TEXT " }, { "#value": " RECEIPT_TEXT " }] }], "Logo": { "@@LogoFormat": "JPG", "FileName": "https://example.jp/logo.jpg" }, "RetailTransaction": [{ "LineItem": [{ "Advertising": { "AdvertisingID": "444", "ImageData": ["8999"], "ImageNumber": ["999"], "ImageURI": ["https://example.jp/image.jpg"], </pre>

```
    "Text": ["888"],
    "Code": ["99"],
    "Barcode": ["\\u0034\\u0039\\u0030\\u0031\\u0032\\u0033\\u0034\\u0035\\u
0036\\u0037\\u0038\\u0039\\u0034" ],
    "URI": ["https://example.jp/"]
  }
}
]
}
]
}
]
}
]
```

Error

Status Code	4xx			
Headers	Content-Type: application/json			
Body				
Property Name	Type	Size	Required	Remarks
error	Object	-	○	Error object.
message	String	-	○	Error message.
type	String	-	○	Error type.
Body Example				
<pre>{ "error": { "message": "Error message.", "type": "ErrorType" } }</pre>				

Error content

Status Code	Type	Message	Remarks
400	ApiVersionError	Application is not supported for this API version.	Incorrect API version
401	TokenMissingError	Token is required.	Authorization Header is not there

			Bearer scheme is not there
401	TokenIncorrectError	Token is incorrect.	Token is illegal
401	TokenExpiredError	Token is expired.	Token Expired
404	ReceiptNotExistError	Receipt not exists.	The specified receipt does not exist

Special Notes

None

5.6.4 Receipt PDF acquisition API

- This API gets the time stamp and the PDF file with e-signature of the Digital Receipt, which is specified by the receiptId of the members, authenticated by Token.
- "receiptId" indicates "DigitalReceipt.ReceiptID" which is included in the response of Receipt list acquisition API. This ID is an ID to specify the receipt uniquely, and it is assumed to be issued automatically by the system.

5.6.4.1 Request

Method	GET
URI	/receipts/{receiptId}/pdf
Headers	Authorization: Bearer {Token}

5.6.4.2 Response

Success

Status Code	200
Headers	Content-Type: application/pdf
Body	
PDF file will be downloaded.	

Error

Status Code	4xx			
Headers	Content-Type: application/json			
Body				
Property Name	Type	Size	Required	Remarks
error	Object	-	○	Error object.
message	String	-	○	Error message.
type	String	-	○	Error type.
Body Example				
<pre>{ "error": { "message": "Error message.", "type": "ErrorType" } }</pre>				

Error content

Status Code	Type	Message	Remarks
400	ApiVersionError	Application is not supported for this API version.	Incorrect API version
401	TokenMissingError	Token is required.	No Authorization header Bearer scheme is not there
401	TokenIncorrectError	Token is incorrect.	Token is illegal
401	TokenExpiredError	Token is expired.	Token Expired
404	ReceiptNotExistError	Receipt not exists.	The specified receipt does not exist
404	ReceiptUnauthenticationError	Receipt is not trusted.	The publisher of the specified receipt is unknown
500	DataAccessError	Data Access Error.	Data cannot be accessed within the application service.
500	InternalServiceError	Internal Service Error.	Some error occurred inside the application service.

Special Notes

None

5.6.5 Receipt registration API

- Register the Digital Receipt.

5.6.5.1 Request

Method	POST
URI	/receipts
Headers	Content-Type: application/json Authorization: Bearer {Token}
Body	Standard Digital Receipt format in JSON.
Body Example	<pre>{ "DigitalReceipt": { "@@MajorVersion": "3", "Transaction": [{ "ReceiptImage": [{ "ReceiptLine": [{ "#value": " RECEIPT_TEXT " }, { "#value": " RECEIPT_TEXT " }, { "#value": " RECEIPT_TEXT " }, { "#value": " RECEIPT_TEXT " }, { "#value": " RECEIPT_TEXT " }, { "#value": " RECEIPT_TEXT " }, { "#value": " RECEIPT_TEXT " }] }], "Logo": { "@@LogoFormat": "JPG", "FileName": "https://example.jp/logo.jpg" }, "BusinessUnit": [{ "UnitID": { "#value": "Branch Name" } }], "WorkstationID": { "#value": "122-1" }, "ReceiptDateTime": { "#value": "2018-09-22T09:23:00" } }] } }</pre>

```
"ReceiptNumber": {
  "#value": "123456789999"
},
"InvoiceNumber": [
  {
    "#value": "4565"
  },
  {
    "#value": "3545"
  }
],
"RetailTransaction": [
  {
    "LineItem": [
      {
        "Sale": {
          "ItemID": [
            {
              "@@Name": "Lunch Set A",
              "#value": "123"
            }
          ],
          "ExtendedAmount": {
            "#value": 1080
          }
        },
        "SequenceNumber": [
          1
        ]
      },
      {
        "Sale": {
          "ItemID": [
            {
              "@@Name": "Lunch Set B",
              "#value": "124"
            }
          ],
          "ExtendedAmount": {
            "#value": 1296
          }
        },
        "SequenceNumber": [
          2
        ]
      }
    ]
  }
]
```

```

    "Tender": {
      "Amount": {
        "#value": 3000
      },
      "TenderChange": [
        {
          "Amount": {
            "#value": 624
          }
        }
      ]
    },
    "SequenceNumber": [
      3
    ]
  },
  {
    "Advertising": {
      "AdvertisingID": "444",
      "ImageData": ["8999"],
      "ImageNumber": ["999"],
      "ImageURI": ["https://example.jp/image.jpg"],
      "Text": ["888"],
      "Code": ["99"],
      "Barcode":
["\u0034\u0039\u0030\u0031\u0032\u0033\u0034\u0035\u0036\u0037\u0038\u0039\u0034"],
      "URI": ["https://example.jp/"]
    }
  },
  "Total": [
    {
      "@@TotalType": "TransactionGrossAmount",
      "#value": 2376
    },
    {
      "@@TotalType": "TransactionNetAmount",
      "#value": 2376
    },
    {
      "@@TotalType": "TransactionGrandAmount",
      "#value": 2376
    },
    {
      "@@TotalType": "TransactionTaxIncluded",
      "#value": 176
    }
  ]
}

```

```
}  
  ]  
  }  
  ]  
  }  
  ]  
  }  
}
```

5.6.5.2 Response

Success

Status Code	200
Headers	Content-Type: text/plain

Error

Status Code	400 / 401			
Headers	Content-Type: application/json			
Body				
Property Name	Type	Size	Required	Remarks
error	Object	-	○	Error object.
message	String	-	○	Error message.
type	String	-	○	Error type.
Body Example				
<pre>{ "error": { "message": "Error message.", "type": "ErrorType" } }</pre>				

Error content

Status Code	Type	Message	Remarks
400	ApiVersionError	Application is not supported for this API version.	Incorrect API version
400	ReceiptDataError	Digital Receipt is invalid data.	The data in the Digital Receipt is incorrect
401	TokenMissingError	Token is required.	Authorization header not included. Bearer scheme not included.
401	TokenIncorrectError	Token is incorrect.	Token is incorrect
401	TokenExpiredError	Token is expired.	Token is Expired
401	MemberAuthError	Account ID or password is incorrect.	Incorrect membership credentials
401	CompanyAuthError	Company code or password is incorrect.	Incorrect company credentials
500	DataAccessError	Data Access Error.	Data cannot be accessed within the application service.
500	InternalServiceError	Internal Service Error.	Some error occurred inside the application service.

Special Notes

None