

電子レシート API 仕様書	改訂日 : 2022-01-31	頁 1 of 45
----------------	------------------	--------------

電子レシート API 仕様書

V 1.0.0

2022-01-31

目次

1	序章	5
1.1	目的	5
1.2	改版履歴	5
1.3	チーム名	7
1.4	チームの使命	7
1.5	メンバー名簿	7
2	電子レシート API の利用	9
2.1	基本フロー	9
2.2	例外フロー	10
3	アプリケーションの利用申請	12
3.1	アプリケーションの権限	12
3.2	ユースケース	12
	会員権限と企業権限について、それぞれの一般的なユースケースを紹介しま す。	12
3.2.1	会員権限のアプリケーションでレシートを取得する	12
3.2.2	会員権限のアプリケーションがレシートを登録する	13
3.2.3	企業権限のアプリケーションがレシートを登録する	14
3.3	会員権限のアプリケーションの概要	15
3.3.1	アプリケーション情報の取得	15
3.3.2	アクセストークンの取得	15
3.3.3	リクエストの例	15

3.4	企業権限のアプリケーションの概要	17
3.4.1	アプリケーション情報の取得.....	17
3.4.2	アクセストークンの取得.....	17
3.4.3	リクエストの例.....	17
4	API 一覧	19
4.1	電子レシート API の一覧.....	19
5	会員 API	20
5.1	認証API.....	20
5.1.1	リクエスト.....	20
5.1.2	レスポンス.....	20
6	企業 API	23
6.1	認証 API.....	23
6.1.1	リクエスト.....	23
6.1.2	レスポンス.....	23
7	レシート API	26
7.1	レシート一覧取得 API.....	26
7.1.1	リクエスト.....	26
7.1.2	レスポンス.....	26
7.2	レシート詳細取得 API.....	30
7.2.1	リクエスト.....	30
7.2.2	レスポンス.....	30
7.3	レシート印字文字列取得 API.....	35

電子レシート API 仕様書	改訂日 : 2022-01-31	頁 4 of 45
----------------	------------------	--------------

7.3.1	リクエスト	35
7.3.2	レスポンス	35
7.4	レシート PDF 取得 API.....	38
7.4.1	リクエスト	38
7.4.2	レスポンス	38
7.5	レシート登録 API.....	40
7.5.1	リクエスト	40
7.5.2	レスポンス	43
8	参照.....	45
9	未解決の課題.....	45

1 序章

1.1 目的

本書は経済産業省の委託事業として作成した電子レシートの API を、NET 流通システム協議会の電子レシート分科会が標準仕様として策定した、標準電子レシート API に関する仕様書である。

本仕様は、デジタルレシートサーバと POS 端末間や、デジタルレシートサーバとスマートフォン間でのデジタルレシートデータの送受信に関わる API を標準化することで、様々なベンダーの様々な業種業態向けの POS ターミナルを接続可能にして、開発コストやシステムインテグレーションコストを低減させることが目的である。また、デジタルレシートデータを活用するスマートフォンアプリケーションが作りやすくなることで、デジタルレシートデータが活用しやすくなることも目的とする。

1.2 改版履歴

バージョン	イベント	変更	日付
1.0.0	初版		2017-11-16
1.0.1	修正	レシート詳細取得 API のレスポンスの値引きオブジェクトを修正	2017-11-21
1.1.0	変更 追加 変更	ログイン API を認証 API に変更 クライアントの登録について追加 Token の JSON の例に cid を追加	2017-12-1
1.2.0	追加	レシート登録 API を追加	2017-12-4
1.2.1	変更	クライアントの識別方式を変更	2017-12-7
1.2.2	追加 変更	企業認証 API を追加	2017-12-7

	追加	クライアントをアプリケーションに名称変更	
	追加	アプリケーションの利用申請の説明を追記 レシート登録 API の詳細を追記	
1.2.3	変更	エラーの内容を変更	2017-12-18
1.2.4	変更	三章の構成を変更	2017-12-27
	追加	レシート登録 API のヘッダーに Authorization を追加	
1.2.5	変更	正式リリースの為の全体的な文書の見直し	2018-05-14
1.2.6	変更	Token 利用記述の変更 1.1 目的文書の修正 1.4 チーム使命文書の修正 7.1 レシート一覧取得 API のパラメータ修正 7.4 レシート PDF ファイル取得 API の説明文修正	2019-07-16
	追加	5.1, 6.1 任意の Token 技術が利用可能	
	削除	8. Token 詳細情報の削除	
1.0.0	変更	バージョン情報を国際標準提案に合わせ 1.0.0 とした	2022-01-28

1.3 チーム名

主幹

・NET流通システム協議会「電子レシート分科会」

協力団体

一般社団法人オープン・フードサービス・システム・コンソーシアム

「デジタルレシート分科会」

一般社団法人F i n T e c h 協会「電子レシート分科会」

一般社団法人XBRL Japan

1.4 チームの使命

本チームの使命は、販売時点の取引処理に完全なペーパーレス化が求められる現在の小売環境において、電子レシートの標準APIを作成し、JSONとRESTを採用した最新テクノロジー環境への適用を実現して電子レシートの採用を促進することにある。

1.5 メンバー名簿

議長(日本)：

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

貢献者(日本)：

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	LOG NOTE Co.,Ltd.
Soichi Fujii	Microsoft Japan Co., Ltd.
Akio Tajima	NCR Japan, Ltd.
Kazutoshi Ota	OK Systems
Tadashi Furuhata	Seiko Epson Corporation
Ryo Hirukawa	Sharp Business Solutions Corporation
Yusuke Udagawa	SHARP CORPORATION
Takao Tamura	Sorimachi Giken Co.,Ltd
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

電子レシート API 仕様書	改訂日 : 2022-01-31	頁 8 of 45
----------------	------------------	--------------

Toyohiro Yasumoto	VINX CORP.
-------------------	------------

事務局(日本) :

Soichi Fujii	Microsoft Japan Co, Ltd.
--------------	--------------------------

フードサービス仕様の貢献者(日本) :

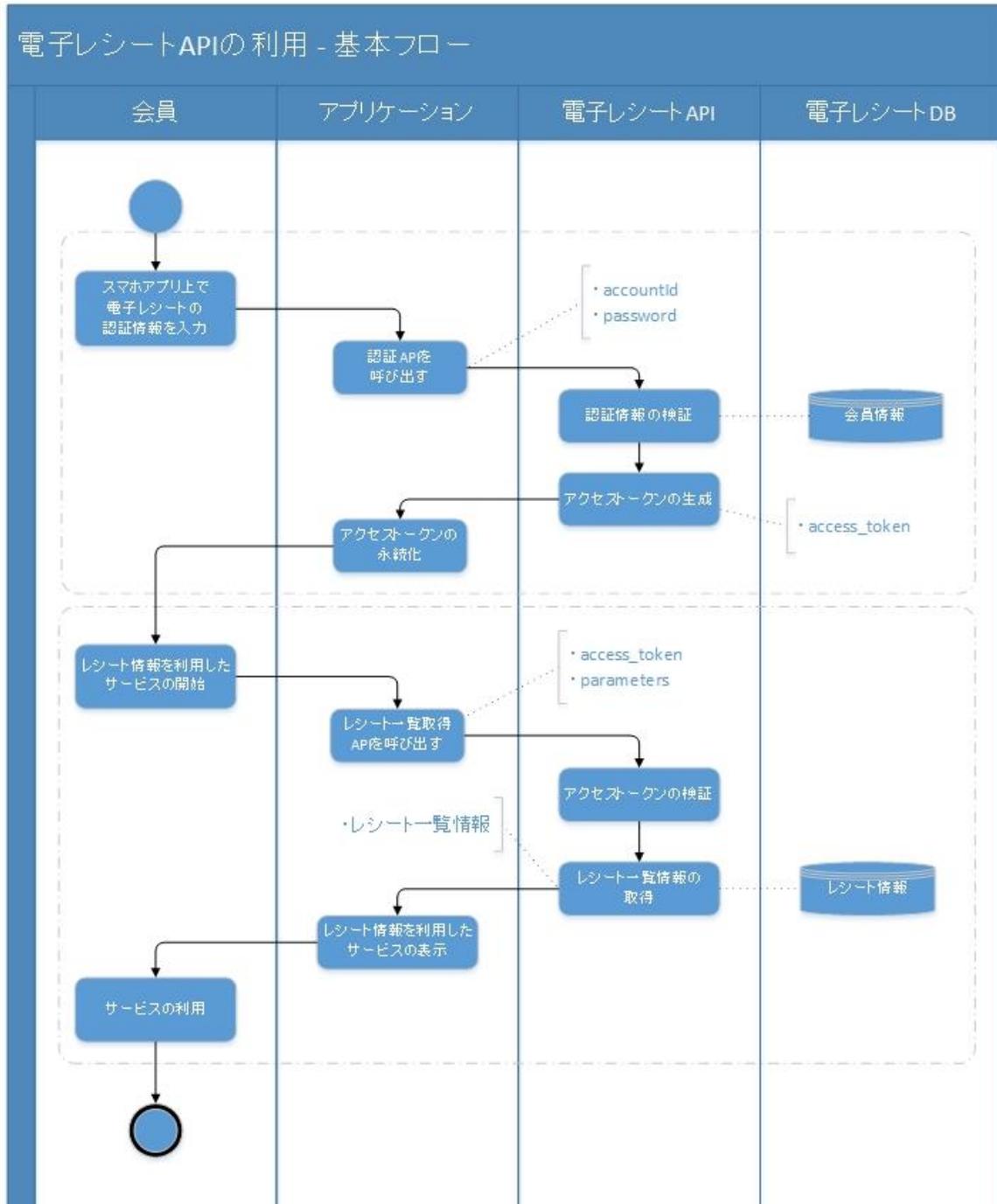
Tadashi Furuhashi	Seiko Epson Corporation
Yoshio Ohba	Seiko Instruments Inc.
Yuu Kusama	Open Foodservice Systems Consortium (OFSC)
Kenji Oohashi	DYNAC CO. , LTD.
Umetsu Noriyuki	Fujitsu Isotec Limited
Shinya Oikawa	Public university corporation Miyagi university
Kazuhiro Kobayashi	Teraoka Seiko Co., 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)

2 電子レシート API の利用

2.1 基本フロー

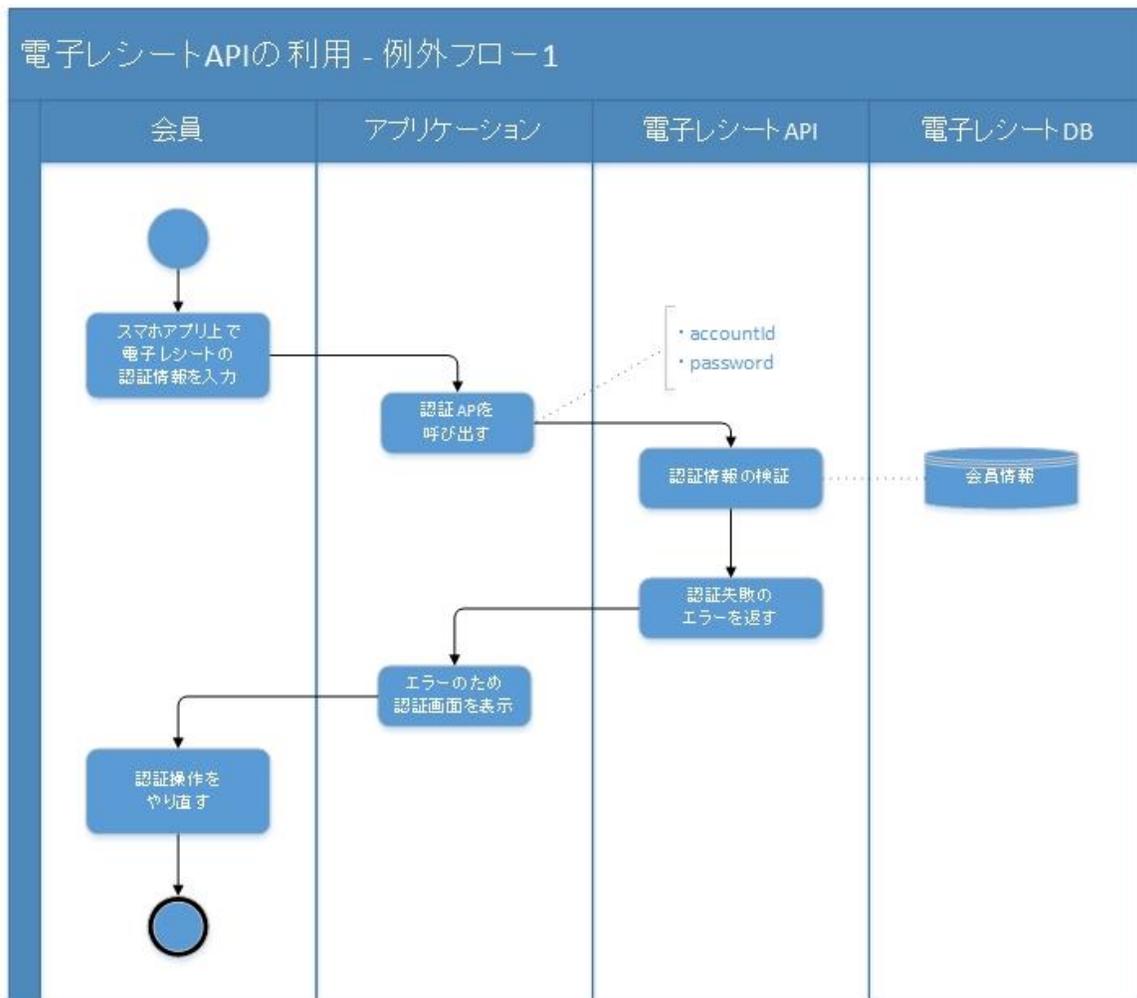
電子レシート API の利用について、基本フローを下図に示します。

基本フローでは、電子レシート会員（消費者）がアプリケーションを通して認証 API を利用し、アクセストークンを取得するフローと、そのアクセストークンを利用してレシート情報を取得するフローを表しています。

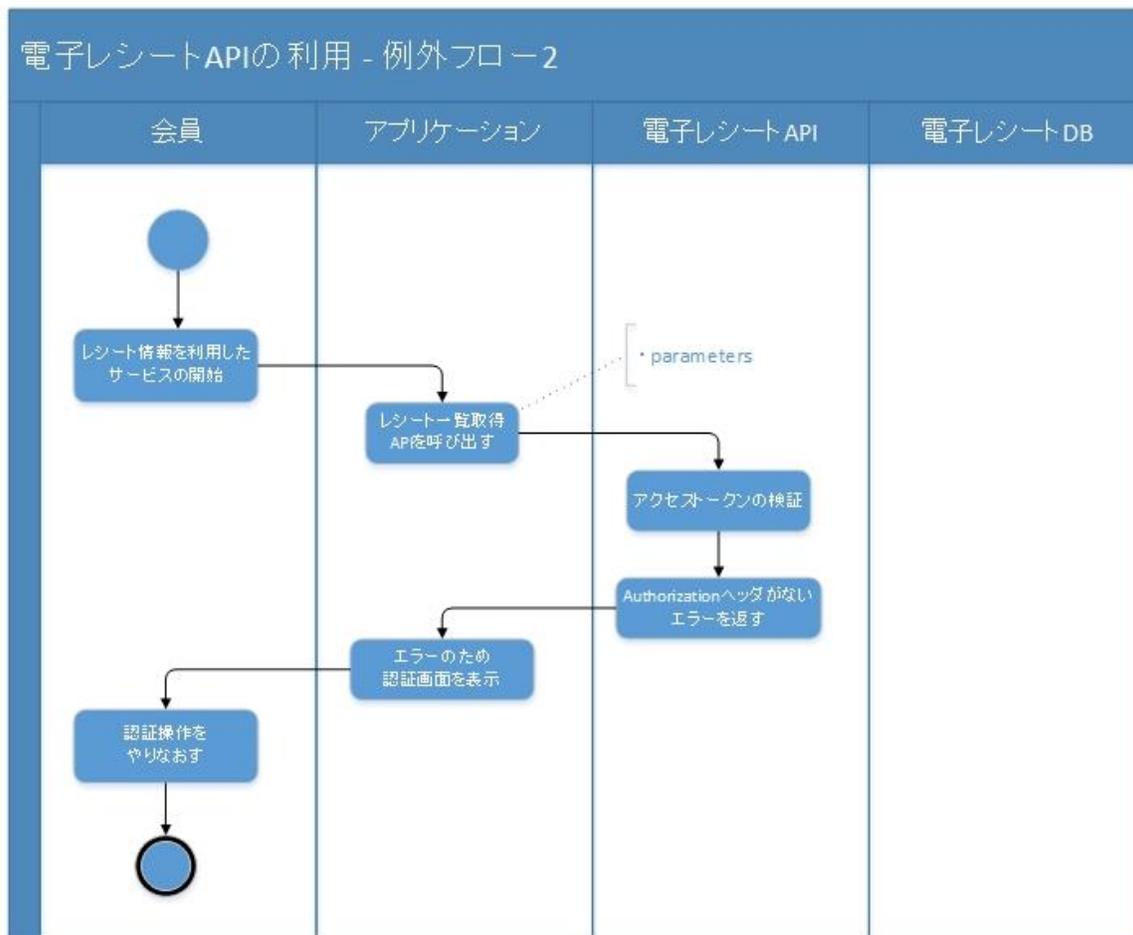


2.2 例外フロー

電子レシート API の利用について、例外フローとして 2 例を以下に示します。
1 つ目の例として、認証 API で認証に失敗した場合のフローを下図に示します。



2つ目の例として、レシート一覧取得 API でアクセストークンを設定しなかった場合のフローを下図に示します。



3 アプリケーションの利用申請

セキュリティ確保の為、電子レシート API を利用するアプリケーション（家計簿ソフトや会計ソフト、健康管理ソフト等）が予め利用申請を要する仕組みを提供します。

3.1 アプリケーションの権限

アプリケーションの権限は下記の 2 種類があります。

1. 会員権限 …… 消費者のアプリケーションとしてのアクセス権限
2. 企業権限 …… 小売事業者の POS アプリケーション等のアクセス権限

3.2 ユースケース

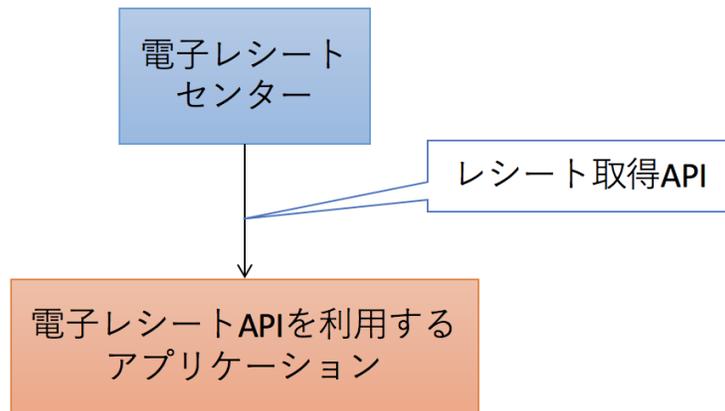
会員権限と企業権限について、それぞれの一般的なユースケースを紹介します。

3.2.1 会員権限のアプリケーションでレシートを取得する

会員権限では、レシートの取得が可能です。

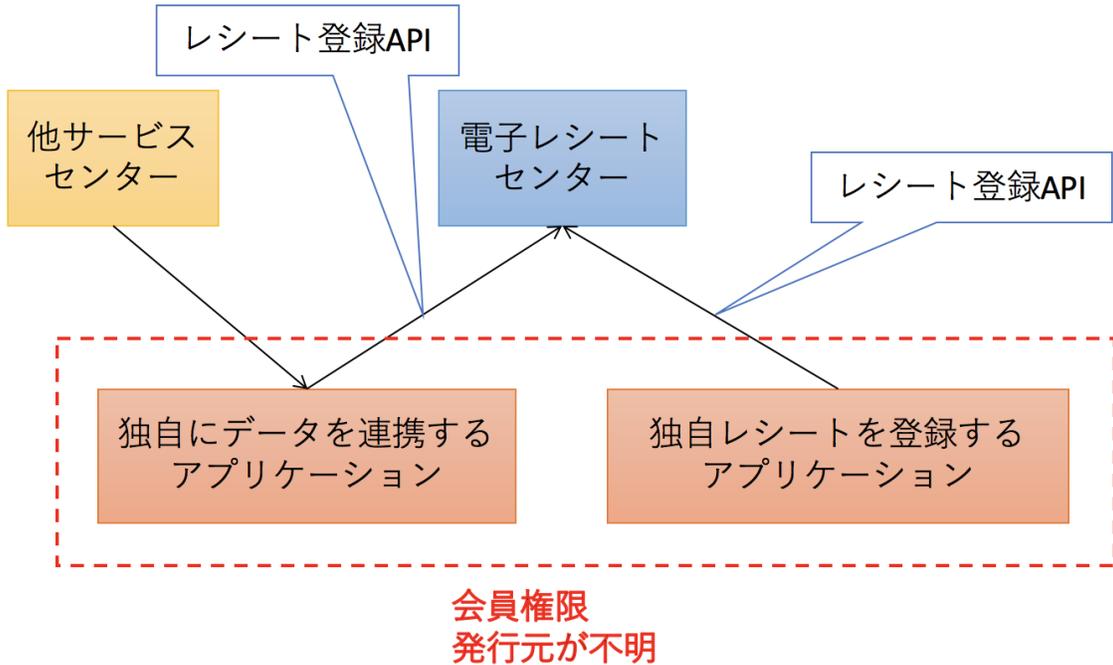
会員権限で登録されたレシートと企業権限で登録されたレシートの両方が取得できます（下図）。

ただし、レシート PDF 取得 API は、企業権限で登録されたレシートのみ取得できます。



3.2.2 会員権限のアプリケーションがレシートを登録する

会員権限のアプリケーションはレシートの登録が可能です。
会員権限のアプリケーションから登録されたレシートは、そのアプリケーション独自のレシートとして登録されるため、電子レシートセンターでは発行元が不明なレシートとして登録されます。
会員権限でレシートを登録するユースケースとしては、電子レシートセンターと連携していない他サービスセンターのデータ連携や会員が独自に作成したレシートの登録となります（下図）。

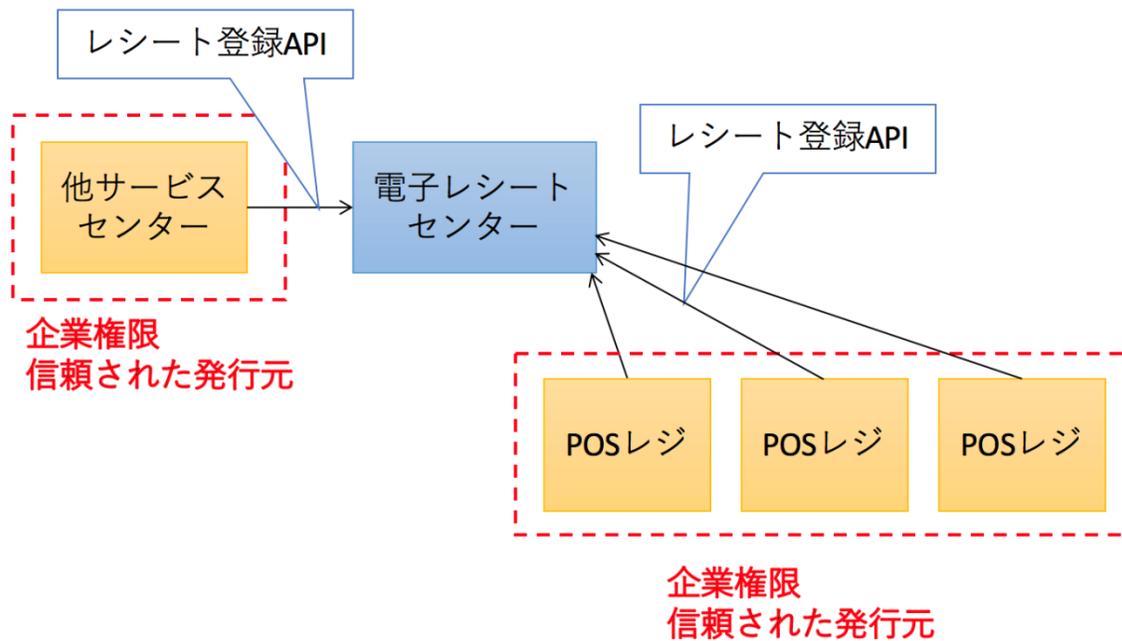


3.2.3 企業権限のアプリケーションがレシートを登録する

企業権限のアプリケーションはレシートの登録のみ可能です。

登録されるレシートは、電子レシートセンターに信頼された発行元のレシートとして登録されます。

企業権限を利用するユースケースとしては、他サービスセンターから電子レシートセンターへのレシート情報の連携や POS レジからの登録となります（下図）。また、企業権限ではレシートの取得はできません。



また、同じ ID のレシートが、会員権限と企業権限の両方から登録された場合、企業権限のレシートが優先されます。

同じ ID のレシートが同じ権限で複数回登録された場合、最後に登録したレシートが優先されます。

3.3 会員権限のアプリケーションの概要

会員権限のアプリケーションは、認証された会員のレシートの取得と発行元の不明なレシートの登録を行うことが可能です。

会員権限のアプリケーションで登録されたレシートは、公的な書類（タイムスタンプ付きレシート PDF ファイル等）として利用することはできません。

3.3.1 アプリケーション情報の取得

セキュリティ確保の為に、アプリケーションを一意に識別する為の ID (applicationId) と、アプリケーションを識別するために利用する秘密の文字列 (applicationSecret) を使用したアプリケーション認証機能を提供します。

3.3.2 アクセストークンの取得

会員権限のアプリケーションから API を利用するには、会員権限向けの認証 API でアクセストークンを取得し、アクセストークンを利用し各種 API を利用します。

3.3.3 リクエストの例

認証 API の例を下記に示します。

Method	POST			
URI	/members/auth			
Headers	Content-Type: application/json			
Body				
プロパティ名	型	サイズ	必須	備考
applicationId	String	20	○	アプリケーションを一意に識別する ID です。
applicationSecret	String	40	○	アプリケーションを識別するための秘密の文字列です。
accountId	String	1-256	○	電子レシートシステムのログインに利用する電話番号または E メールアドレスです。
password	String	1-256	○	電子レシートシステムのログインに利用するパスワードです。
Body の例				
<pre>{ "applicationId": {APPLICATION_ID}, "applicationSecret": {APPLICATION_SECRET}, "accountId": "name@example.jp",</pre>				

電子レシート API 仕様書	改訂日: 2022-01-31	頁 16 of 45
----------------	-----------------	---------------

```
"password": "password"  
}
```

APPLICATION_ID と APPLICATION_SECRET には、システム運営会社が発行／管理するアプリケーション ID とアプリケーションシークレットを設定します。

3.4 企業権限のアプリケーションの概要

企業権限のアプリケーションは、認証された企業のレシートの登録を行うことが可能です。

企業権限のアプリケーションで登録されたレシートは、公的な書類（タイムスタンプ付きレシート PDF ファイル等）として利用することができます。

3.4.1 アプリケーション情報の取得

アプリケーション ID はアプリケーションを一意に識別する ID で、一度登録されたものは変更することはできません。

アプリケーションシークレットは、アプリケーションを識別するために利用する秘密の文字列です。

3.4.2 アクセストークンの取得

企業権限のアプリケーションから API を利用するには、企業権限向けの認証 API でアクセストークンを取得し、アクセストークンを利用し各種 API を利用します。

3.4.3 リクエストの例

認証 API の例を下記に示します。

Method	POST			
URI	/companies/auth			
Headers	Content-Type: application/json			
Body				
プロパティ名	型	サイズ	必須	備考
applicationId	String	20	○	アプリケーションを一意に識別する ID です。
applicationSecret	String	40	○	アプリケーションを識別するための秘密の文字列です。
companyCode	String	13	○	企業コードです。
Password	String	1-256	○	電子レシートシステムにより発行されるパスワードです。
Body の例				
<pre>{ "applicationId": {APPLICATION_ID}, "applicationSecret": {APPLICATION_SECRET}, "companyCode": "0000000000001", "password": "password" }</pre>				

電子レシート API 仕様書	改訂日: 2022-01-31	頁 18 of 45
----------------	-----------------	---------------

}

APPLICATION_ID と APPLICATION_SECRET には、システム運営会社が発行／管理するアプリケーション ID とアプリケーションシークレットを設定します。

4 API 一覧

4.1 電子レシート API の一覧

電子レシート API の一覧を以下に示します。

API 名	URI	メソッド	概要
会員認証 API *1	/members/auth	POST	会員の認証を行い、認証トークンを発行します。
企業認証 API *1	/companies/auth	POST	企業の認証を行い、認証トークンを発行します。
レシート一覧取得 API *1	/receipts	GET	認証された会員のレシート一覧情報を取得します。
レシート詳細取得 API *1	/receipts/{receiptId}	GET	認証された会員の指定された receiptId のレシート詳細情報を取得します。
レシート印字文字列取得 API *1	/receipts/{receiptId}/stringArray	GET	認証された会員の指定された receiptId のレシート印字文字列を取得します。
レシート PDF 取得 API *1	/receipts/{receiptId}/pdf	GET	認証された会員の指定された receiptId のレシート PDF ファイルを取得します。
レシート登録 API *2	/receipts	POST	レシート情報を登録します。

*1 URI 例 : <https://exp-openapi.sampledomain.jp/v1> の後続く様に記述します。

*2 URI 例 : <https://exp-public-receiver.sampledomain.jp/srr> の後続く様に記述します。

特記事項

5 会員 API

電子レシート会員（消費者）のアプリケーション等が利用することを前提にした API です。

5.1 認証 API

電子レシート会員の認証を行います。

認証に成功した場合、レシート API へのアクセストークンを発行します。

アクセストークンには任意の Token 技術が利用可能です。

5.1.1 リクエスト

Method	POST			
URI	/members/auth			
Headers	Content-Type: application/json			
Body				
プロパティ名	型	サイズ	必須	備考
applicationId	String	20	○	アプリケーションを一意に識別する ID です。
applicationSecret	String	40	○	アプリケーションを識別するための秘密の文字列です。
accountId	String	1-256	○	電子レシートシステムのログインに利用する電話番号または E メールアドレスです。
password	String	1-256	○	電子レシートシステムのログインに利用するパスワードです。
Body の例				
<pre>{ "applicationId": {APPLICATION_ID}, "applicationSecret": {APPLICATION_SECRET}, "accountId": "name@example.jp", "password": "password" }</pre>				

5.1.2 レスポンス

正常

Status Code	200
Headers	Content-Type: application/json
Body	

電子レシート API 仕様書	改訂日: 2022-01-31	頁 21 of 45
----------------	-----------------	---------------

プロパティ名	型	サイズ	必須	備考
access_token	String	-	○	アクセストークンです。
Body の例				
<pre>{ "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMhRHDcEfxjoYZgeFONFh7HgQ" }</pre>				

エラー

Status Code	4xx			
Headers	Content-Type: application/json			
Body				
プロパティ名	型	サイズ	必須	備考
error	Object	-	○	エラーのオブジェクトです。
message	String	-	○	エラーメッセージです。
type	String	-	○	エラータイプです。
Body の例				
<pre>{ "error": { "message": "Error message.", "type": "ErrorType" } }</pre>				

エラーの内容

Status Code	Type	Message	備考
400	ApiVersionError	Application is not supported for this API version.	APIバージョンが不正
401	MemberAuthError	Account ID or password is incorrect.	会員認証情報の誤り
403	AccountLockoutError	Account is locked out.	会員認証の連続失敗によるアカウントのロックアウト
401	ApplicationAuthError	Application ID or application secret is incorrect.	アプリケーション認証情報の誤り

500	DataAccessError	Data Access Error	アプリケーションサービス内でデータにアクセス不可
500	InternalServerError	Internal Service Error	アプリケーションサービス内部で何らかのエラーが発生

特記事項

なし

6 企業 API

企業のアプリケーション（POS等）が利用することを前提にした API です。

6.1 認証 API

企業の認証を行います。

認証に成功した場合、レシート API へのアクセストークンを発行します。

アクセストークンには任意の Token 技術が利用可能です。

6.1.1 リクエスト

Method	POST			
URI	/companies/auth			
Headers	Content-Type: application/json			
Body				
プロパティ名	型	サイズ	必須	備考
applicationId	String	20	○	アプリケーションを一意に識別する ID です。
applicationSecret	String	40	○	アプリケーションを識別するための秘密の文字列です。
companyCode	String	13	○	企業コードです。
password	String	1-256	○	電子レシートシステムにより発行されるパスワードです。
Body の例				
<pre>{ "applicationId": {APPLICATION_ID}, "applicationSecret": {APPLICATION_SECRET}, "companyCode": "0000000000001", "password": "password" }</pre>				

6.1.2 レスポンス

正常

Status Code	200			
Headers	Content-Type: application/json			
Body				
プロパティ名	型	サイズ	必須	備考

access_token	String	-	○	JSON Web Token です。
Body の例				
<pre>{ "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ" }</pre>				

エラー

Status Code	4xx			
Headers	Content-Type: application/json			
Body				
プロパティ名	型	サイズ	必須	備考
error	Object	-	○	エラーのオブジェクトです。
message	String	-	○	エラーメッセージです。
type	String	-	○	エラータイプです。
Body の例				
<pre>{ "error": { "message": "Error message.", "type": "ErrorType" } }</pre>				

エラーの内容

Status Code	Type	Message	備考
400	ApiVersionError	Application is not supported for this API version.	APIバージョンが不正
401	CompanyAuthError	Company code or password is incorrect.	企業認証情報の誤り
401	ApplicationAuthError	Application ID or application secret is incorrect.	アプリケーション認証情報の誤り
500	DataAccessError	Data Access Error	アプリケーションサービス内でデータにアクセス不可

電子レシート API 仕様書	改訂日 : 2022-01-31	頁 25 of 45
----------------	------------------	---------------

500	InternalServiceError	Internal Service Error	アプリケーションサービス内部で何らかのエラーが発生
-----	----------------------	------------------------	---------------------------

7 レシート API

電子レシート情報の取得等に関する API です。

7.1 レシート一覧取得 API

Token により認証された会員の電子レシート一覧情報を取得します。
一覧情報の絞り込み条件として、クエリパラメータを利用することが出来ます。

7.1.1 リクエスト

Method	GET			
URI	/receipts			
Headers	Authorization: Bearer {Token}			
Query Parameters				
パラメータ名	型	サイズ	必須	備考
from	String	14	×	from 以降の取引日時を指定します。 標準電子レシートフォーマットの ReceiptDateTime に対応します。
to	String	14	×	to 以前の取引日時を指定します。 標準電子レシートフォーマットの ReceiptDateTime に対応します。
Query Parameters の例				
/receipts?from=20180201000000&to=20180228235959				

7.1.2 レスポンス

正常

Status Code	200
Headers	Content-Type: application/json
Body	
標準電子レシートフォーマットのレシートデータの配列を持つオブジェクトです。詳細に関しては ARTS Digital Receipt Specification v3.1.0 を参照ください。	
Body の例	
<pre>{ "DigitalReceiptIndex": [{ "DigitalReceipt": { "@@MajorVersion": "3", "ReceiptID": "201801011000000000000100000100000000010001", "Transaction": [{</pre>	

```
"@@TypeCode": "SaleTransaction",
"BusinessUnit": [
  {
    "UnitID": {
      "#value": "拠点 (店舗) 名"
    }
  }
],
"WorkstationID": {
  "#value": "122-1"
},
"ReceiptDateTime": {
  "#value": "2018-09-22T09:23:00"
},
"ReceiptNumber": {
  "#value": "123456789999"
},
"InvoiceNumber": [
  {
    "#value": "4565"
  },
  {
    "#value": "3545"
  }
],
"RetailTransaction": [
  {
    "Total": [
      {
        "@@TotalType": "TransactionGrossAmount",
        "#value": 2376
      },
      {
        "@@TotalType": "TransactionNetAmount",
        "#value": 2376
      },
      {
        "@@TotalType": "TransactionGrandAmount",
        "#value": 2376
      },
      {
        "@@TotalType": "TransactionTaxIncluded",
```


			Bearer スキームがない
401	TokenIncorrectError	Token is incorrect.	Token が不正
401	TokenExpiredError	Token is expired.	Token が有効期限切れ
500	DataAccessError	Data Access Error	アプリケーションサービス内でデータにアクセス不可
500	InternalServerError	Internal Service Error	アプリケーションサービス内部で何らかのエラーが発生

特記事項
なし

7.2 レシート詳細取得 API

Token により認証された会員の receiptId により指定された電子レシートの詳細情報を取得します。

receiptId とは、レシート一覧取得 API のレスポンスに含まれる DigitalReceipt.ReceiptID を指します。この ID はレシートを一意に特定する為の ID で、システムにより自動的に発行されることを想定しています。

詳細情報には、企業情報、店舗情報、レジ番号、レシート番号、取引日時、購買情報、店舗ロゴ画像、販促ビットマップ画像が含まれます。

7.2.1 リクエスト

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

7.2.2 レスポンス

正常

Status Code	200
Headers	Content-Type: application/json
Body	ARTS フォーマットのレシート情報です。 項目については下記の例と ARTS Digital Receipt Specification v3.1.0 を参照ください。
Body の例	<pre>{ "DigitalReceipt": { "@@MajorVersion": "3", "Transaction": [{ "BusinessUnit": [{ "UnitID": { "#value": "拠点 (店舗) 名" } }] }, { "WorkstationID": { "#value": "122-1" } }], "ReceiptDateTime": {</pre>

```
"#value": "2018-09-22T09:23:00"
},
"ReceiptNumber": {
  "#value": "123456789999"
},
"InvoiceNumber": [
  {
    "#value": "4565"
  },
  {
    "#value": "3545"
  }
],
"RetailTransaction": [
  {
    "LineItem": [
      {
        "Sale": {
          "ItemID": [
            {
              "@@Name": "A ランチセット",
              "#value": "123"
            }
          ],
          "ExtendedAmount": {
            "#value": 1080
          }
        },
        "SequenceNumber": [
          1
        ]
      },
      {
        "Sale": {
          "ItemID": [
            {
              "@@Name": "B ランチセット",
              "#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
  },
],
```


400	ApiVersionError	Application is not supported for this API version.	APIバージョンが不正
401	TokenMissingError	Token is required.	Authorizationヘッダがない Bearer スキームがない
401	TokenIncorrectError	Token is incorrect.	Tokenが不正
401	TokenExpiredError	Token is expired.	Tokenが有効期限切れ
404	ReceiptNotExistError	Receipt not exists.	指定されたレシートが存在しない
500	DataAccessError	Data Access Error	アプリケーションサービス内でデータにアクセス不可
500	InternalServerError	Internal Service Error	アプリケーションサービス内部で何らかのエラーが発生

特記事項
なし

7.3 レシート印字文字列取得 API

Token により認証された会員の receiptId により指定された電子レシートの印字文字列を取得します。

receiptId とは、レシート一覧取得 API のレスポンスに含まれる DigitalReceipt.ReceiptID を指します。この ID はレシートを一意に特定する為の ID で、システムにより自動的に発行されることを想定しています。

通常の POS からプリンターに送付される一般的な印字命令、例えばエスケープシーケンスと呼称される Code 等は含まれません。

7.3.1 リクエスト

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

7.3.2 レスポンス

正常

Status Code	200
Headers	Content-Type: application/json
Body	レシート印字文字列、店舗ロゴ、販促イメージです。 項目については下記の例と ARTS Digital Receipt Specification v3.1.0 を参照ください。
Body の例	<pre> {{ "DigitalReceipt": { "@@MajorVersion": 3, "@@MinorVersion": 1, "@@FixVersion": 0, "Transaction": [{ "ReceiptImage": [{ "ReceiptLine": [{ "#value": " RECEIPT_TEXT " }, { "#value": " RECEIPT_TEXT " },] }] }] } } </pre>


```
{
  "error": {
    "message": "Error message.",
    "type": "ErrorType"
  }
}
```

エラーの内容

Status Code	Type	Message	備考
400	ApiVersionError	Application is not supported for this API version.	APIバージョンが不正
401	TokenMissingError	Token is required.	Authorizationヘッダーがない Bearer スキームがない
401	TokenIncorrectError	Token is incorrect.	Token が不正
401	TokenExpiredError	Token is expired.	Token が有効期限切れ
404	ReceiptNotExistError	Receipt not exists.	指定されたレシートが存在しない
500	DataAccessError	Data Access Error	アプリケーションサービス内でデータにアクセス不可
500	InternalServerError	Internal Service Error	アプリケーションサービス内部で何らかのエラーが発生

特記事項
なし

電子レシート API 仕様書	改訂日: 2022-01-31	頁 38 of 45
----------------	-----------------	---------------

7.4 レシート PDF 取得 API

Token により認証された会員の receiptId により指定された電子レシートの PDF ファイルを取得します。

receiptId とは、レシート一覧取得 API のレスポンスに含まれる DigitalReceipt.ReceiptID を指します。この ID はレシートを一意に特定する為の ID で、システムにより自動的に発行されることを想定しています。

7.4.1 リクエスト

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

7.4.2 レスポンス

正常

Status Code	200
Headers	Content-Type: application/pdf
Body	PDF file がダウンロードされます。

エラー

Status Code	4xx			
Headers	Content-Type: application/json			
Body				
プロパティ名	型	サイズ	必須	備考
error	Object	-	○	エラーのオブジェクトです。
message	String	-	○	エラーメッセージです。
type	String	-	○	エラータイプです。
Body の例	<pre>{ "error": { "message": "Error message.", "type": "ErrorType" } }</pre>			

エラーの内容

Status Code	Type	Message	備考
-------------	------	---------	----

400	ApiVersionError	Application is not supported for this API version.	APIバージョンが不正
401	TokenMissingError	Token is required.	Authorizationヘッダがない Bearer スキームがない
401	TokenIncorrectError	Token is incorrect.	Tokenが不正
401	TokenExpiredError	Token is expired.	Tokenが有効期限切れ
404	ReceiptNotExistError	Receipt not exists.	指定されたレシートが存在しない
404	ReceiptUnauthenticationError	Receipt is not trusted.	指定されたレシートの発行元が不明
500	DataAccessError	Data Access Error	アプリケーションサービス内でデータにアクセス不可
500	InternalServiceError	Internal Service Error	アプリケーションサービス内部で何らかのエラーが発生

特記事項
なし

7.5 レシート登録 API

レシートの登録を行います。

7.5.1 リクエスト

Method	POST
URI	/receipts
Headers	Content-Type: application/json Authorization: Bearer {Token}
Body	標準電子レシートフォーマットの JSON です。
Body の例	<pre>{ "DigitalReceipt": { "@@MajorVersion": "3", "Transaction": [{ "ReceiptImage": [{ "ReceiptLine": [{ "#value": " RECEIPT_TEXT " }, { "#value": " RECEIPT_TEXT " }] }] }], "BusinessUnit": [{ "UnitID": { "#value": "拠点（店舗）名" } }], "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": "A ランチセット",
              "#value": "123"
            }
          ],
          "ExtendedAmount": {
            "#value": 1080
          }
        },
        "SequenceNumber": [
          1
        ]
      },
      {
        "Sale": {
          "ItemID": [
            {
              "@@Name": "B ランチセット",
              "#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
  },
  {

```


エラーの内容

Status Code	Type	Message	備考
400	ApiVersionError	Application is not supported for this API version.	APIバージョンが不正
400	ReceiptDataError	Digital receipt is invalid data.	電子レシートのデータが不正
401	TokenMissingError	Token is required.	Authorizationヘッダがない Bearer スキームがない
401	TokenIncorrectError	Token is incorrect.	Token が不正
401	TokenExpiredError	Token is expired.	Tokenが有効期限切れ
401	MemberAuthError	Account ID or password is incorrect.	会員認証情報の誤り
401	CompanyAuthError	Company code or password is incorrect.	企業認証情報の誤り
500	DataAccessError	Data Access Error	アプリケーションサービス内でデータにアクセス不可
500	InternalServiceError	Internal Service Error	アプリケーションサービス内部で何らかのエラーが発生

特記事項
なし

電子レシート API 仕様書	改訂日 : 2022-01-31	頁 45 of 45
----------------	------------------	---------------

8 参照

ARTS Digital Receipt Specification v3.1.0

RFC 7519 - JSON Web Token (Token)

<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>

9 未解決の課題