Genius Mini

API for Android

Public Use

Doc no: EN-PUB-0007 Version 1.0

CAYAN™

The Payment Possibilities Company™

Copyright notice

Contents

1. Overview

With Genius Mini, you can get closer to your customers. Genius Mini works the same way as the Genius Countertop and Handheld devices; you start the sale at your Point of Sale (POS) and your customers use Genius Mini to pay using their credit, debit, or gift card.

You download the Genius application to a mobile device to use with the Genius Mini reader. We push updates, bug fixes, and new features to the Genius application, without you having to update your POS.

Genius Mini exposes the same API as other Genius products, so if you already integrate with Genius, integrating with Genius Mini is straightforward.

Note: Visit https://cayan.com/developers/genius/transactions to find information on the API for Genius Countertop and Genius Handheld.

2. Developer Documentation

This chapter describes how to integrate your POS with the Android version of Genius Mini. When integrating with Android you can use either HTTP, HTTPS, or Android Intents for application to application communication. When using Android Intents, all other schemas, and Data Transfer Objects (DTO) are identical to Genius Countertop and Handheld.

- We support Genius Mini with Android version 4.0.3 upwards.

Note: Future releases of Genius Mini will be compatible Windows mobile devices. These versions will use the same API as Genius Countertop and Handheld.

2.1 Transaction flow

1. Stage Transaction	Generate a unique key (TransportKey) using the createTransaction web service method to stage the transaction parameters.
2. Initiate Transaction	a. Send the TransportKey to the Customer Engagement Device using the GET method and wait for the customer to complete the transaction. b. After the transaction is completed, the device sends the response back to the Point of Sale in XML, JSON or JSONP format.
3. Retrieve Transaction	**Optional** Use the DetailsByTransportKey web service method to retrieve additional payment information for payments like gift and loyalty.

2.1.1 Endpoints

Test endpoints	**Test URL**
Stage Transaction	https://transport.merchantware.net/v4/transportService.asmx
Initiate Transaction	http://[CED-IP-Address]:8080/v2/pos?TransportKey=xxxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx&Format=XML http://[CED-IP-Address]:8080/v2/pos?TransportKey=xxxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx&Format=JSON
Retrieve Details	https://genius.merchantware.net/v1/Reporting.asmx

Live endpoints	**Test URL**
Stage Transaction	https://transport.merchantware.net/v4/transportService.asmx
Initiate Transaction	http://[CED-IP-Address]:8080/v2/pos?TransportKey=xxxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx&Format=XML http://[CED-IP-Address]:8080/v2/pos?TransportKey=xxxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx&Format=JSON
Retrieve Details	https://genius.merchantware.net/v1/Reporting.asmx

2.2 Configuring your POS for Genius Mini

You can bring the application to the foreground using either HTTP, HTTPS, or Android Intents:

2.2.1 HTTP or HTTPS

```
ActivityManager result =
(ActivityManager)context.getSystemService(Context.ACTIVITY_SERVICE);

List<ActivityManager.RunningTaskInfo> rt =
activityManager.getRunningTasks(Integer.MAX_VALUE);

activityManager.moveTaskToFront(getTaskId(), 0);
```

2.2.2 Android Intents

When using Android Intents, you must specify a result activity ID. You should use your POS name when specifying a result activity ID.

```
public static final String RECEIVERID = "android.intent.action.genius.pos.result";
```

Specify a callback in your main application activity:

```
   @Override

public void onReceive(Context context, Intent intent) {

    String result = intent.getStringExtra("result");

    String description = intent.getStringExtra("description");

    Log.d("LogEng", result);

    showResultDialog(result);

    if (Defines.POSAction.valueOf(description) !=
Defines.POSAction.InitiateKeyedEntry) {

        ActivityManager activityManager = (ActivityManager) context

                .getSystemService(Context.ACTIVITY_SERVICE);

        activityManager.moveTaskToFront(getTaskId(), 0);

    }

}
```

2.4 Stage transaction

The CreateTransaction web service method allows you to submit non-sensitive payment information to the payment gateway and returns a unique key (TransportKey) in the response which will be used for all subsequent steps to identify the transaction.

Content-Type: text/xml; charset=utf-8

SOAPAction: http://transport.merchantware.net/v4/CreateTransaction

Test Endpoint	https://transport.merchantware.net/v4/transportService.asmx
Live Endpoint	https://transport.merchantware.net/v4/transportService.asmx

2.4.1 Request parameters

Name	Type	Size	Description
merchantName	String	1-160	The name of the business or organization owning the Merchantware account.
merchantSiteId	String	8-160	The site identifier of a location or storefront owned by the Merchantware account owner.
merchantKey	String	1-160	The software key or password for the site accessing a Merchantware account.
Request	TransportRequest	--	A TransportRequest object containing the transaction data.

2.4.2 TransportRequest object parameters

Name	Type	Size	Description
TransactionType	String	4-14	The transaction type for this transaction. Valid values: • SALE • REFUND • LEVEL2SALE - level 2 transaction for business/corp cards. Requires tax amount, customer code and po number fields. • PREAUTH - preauthorizes card only. A POSTAUTH needs to be performed to capture the transaction. • FORCESALE - used to perform a voice authorization transaction by using the authorization code. Must use the InitiateKeyedEntry method for this function.
Amount	Decimal	0-99,999	The desired amount for a transaction.
ClerkId	String	1-50	A unique ID for the user running the transaction. POS can send the server ID, clerk ID, employee ID, register ID or any other ID that identifies the user or system running the transaction.

Name	Type	Size	Description
OrderNumber	String	1-8	The order or invoice number associated with the transaction.
Dba	String	1-50	The business name for the merchant as it should appear to the customer.
SoftwareName	String	1-50	The name of the software application sending the request.
SoftwareVersion	String	1-25	The version number of the software application sending the request.
TransactionId	String	1-25	The merchant-defined identifier for the transaction.
ForceDuplicate	Boolean	--	Override duplicate protection and allow the transaction to process normally. • `true` • `false`
CustomerCode	String	1-50	Required for level 2 transactions. The merchant-defined identifier for the customer involved in the transaction.
PoNumber	String	1-8	Required for level 2 transactions. The customer-defined identifier declaring a purchase order for the transaction.
TaxAmount	Decimal	0-20,000	Required for level 2 transactions. The declared tax amount of the transaction.
TerminalId	String	0-2	Terminal ID to uniquely identify the terminal to the processor. The terminal ID must contain all numeric characters. If no value is supplied, the last 2 digits of the Genius device serial number will be used as a default value. For more information about the TID value that should be used for merchants, please contact our Certifications Team at integrations@cayan.com
HealthCareAmountDetails	Object	--	An optional field used to specify various healthcare amounts for healthcare based cards, such as Flexible Spending Accounts.
TipDetails	Object	--	An optional field used to specify the amount of the staged transaction that is eligible for a tip. The Genius device displays this value to the customer as well as the total value of the transaction. The `EligibleAmount` value is used to calculate the tip when the customer taps the Percentage Tip button.

2.4.3 Response parameters

Name	Type	Description
transportKey	String (GUID)	A GUID that references the staged transaction. Used for subsequest actions like initiating the transaction request on the Genius device.
validationKey	String (GUID)	A GUID that is used to validate the response from the Genius device upon the completion of the transaction.
messages	MessageList	A collection of **message** objects that contain any error messages generated by the request. Each message contains a **field** and **information** field.

2.4.4 Examples

2.4.4.1 Soap request example

```
<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

    <CreateTransaction xmlns="http://transport.merchantware.net/v4/">

    <merchantName>ZERO INC</merchantName>

    <merchantSiteId>00000000</merchantSiteId>

    <merchantKey>00000-00000-00000-00000-00000</merchantKey>

    <request>

        <TransactionType>SALE</TransactionType>

        <Amount>1.23</Amount>

        <ClerkId>ABC123</ClerkId>

        <OrderNumber>ABC123</OrderNumber>

        <Dba>ZERO BRANDS</Dba>

        <SoftwareName>ABC SOFTWARE</SoftwareName>

        <SoftwareVersion>1.0.0.0</SoftwareVersion>

        <Cardholder>VISA TEST CARD</Cardholder>

        <TransactionId>ABC123</TransactionId>

        <ForceDuplicate>FALSE</ForceDuplicate>

        <CustomerCode>ABC123</CustomerCode>

        <PoNumber>ABC123</PoNumber>
```

```
        <TaxAmount>0.09</TaxAmount>

        <TerminalId>01</TerminalId>

        <HealthCareAmountDetails>

            <HealthCareTotalAmount>1.23</HealthCareTotalAmount>

            <ClinicalAmount>0.23</ClinicalAmount>

            <CopayAmount>0.25</CopayAmount>

            <DentalAmount>0.25</DentalAmount>

            <PrescriptionAmount>0.25</PrescriptionAmount>

            <VisionAmount>0.25</VisionAmount>

        </HealthCareAmountDetails>

        <TipDetails>

            <EligibleAmount>1.00</EligibleAmount>

        </TipDetails>

    </request>

</CreateTransaction>

</soap:Body>

</soap:Envelope>
```

2.4.4.2 Soap response example

```
<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

    <CreateTransactionResponse xmlns="http://transport.merchantware.net/v4/">

    <CreateTransactionResult>

        <TransportKey>xxxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx</TransportKey>

        <ValidationKey>xxxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx</ValidationKey>

        <Messages>

            <Message>

                <Field/>

                <Information/>
```

```
            </Message>
            <Message>
                <Field/>
                <Information/>
            </Message>
        </Messages>
    </CreateTransactionResult>
</CreateTransactionResponse>
</soap:Body>
</soap:Envelope>
```

2.5 Initiate Transaction

To initiate a transaction, you can use HTTP, HTTPS, Android Intents, or Javascript:

2.5.1 HTTP or HTTPS

v2 Sample XML request (Default)	`http://[CED-IP-Address]:8080/v2/pos?TransportKey=xxxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx&Format=XML`
v2 Sample JSON request	`http://[CED-IP-Address]:8080/v2/pos?TransportKey=xxxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx&Format=JSON`

2.5.2 Android Intents

2.5.2.1 Android Intents request parameters

Name	Required	Type	Description
ResultReceiverID	Yes	String	Genius returns the result using this ResultReceiverID. You should use your POS name as your ResultReceiverID.
TransportKey	Yes	String	A GUID that references the staged transaction. Used for subsequest actions like initiating the transaction request on the Genius device.
Format	Yes	String	The format that the CED should respond in. Possible values are: • `XML` • `JSON`
URI	Yes	String	/[version]/pos [version] indicates the version of call set in settings

2.5.2.2 Android Intents code

```
String URI =
PreferenceManager.getDefaultSharedPreferences(getActivity().getApplicationContext()).g
etString("saleVer", getString(R.string.version2)) + "/pos";

Intent intent = new Intent(MainActivity.REQUEST_ACTION);
intent.putExtra("ResultReceiverID",MainActivity.RECEIVERID);
intent.putExtra("TransportKey", params.get("TransportKey"));
intent.putExtra("Format",
PreferenceManager.getDefaultSharedPreferences(getActivity().getApplicationContext()).g
etString("format", getString(R.string.XML)));
intent.putExtra("URI", URI);
getActivity().sendBroadcast(intent);
```

2.5.4 Javascript example

```
var postData = "TransportKey=" + ObtainedTransportKey + "&Format=XML";

var xhttp = new XMLHttpRequest();

xhttp.open("POST", "http://127.0.0.1:8080/v2/pos", false);

xhttp.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded; charset=utf-8" );

xhttp.send(postData);
```

2.5.5 V2 Response parameters

Name	Type	Description
Status	String	The status of the transaction, typically 'APPROVED', 'DECLINED' or 'ERROR'. This value may also have other definitions depending on CardType and context. For cancelled transactions, the status returned will be 'USERCANCELLED' for transactions cancelled by the user or 'POSCANCELLED' for transactions cancelled by the POS system.
AmountApproved	Decimal	The amount of the transaction that was approved.
AuthorizationCode	String	Authorization Code issued by the processor upon receipt of a transaction.
Cardholder	String	The Cardholder associated with the payment card used in a transaction.
AccountNumber	String	A string representing the truncated card number of the payment card used in a transaction.
PaymentType	enum	The issuer type of payment card used in a transaction. • `VISA` - Visa • `MASTERCARD` - Mastercard or Diner's Club • `AMEX` - American Express • `DISCOVER` - Discover • `DEBIT` - PIN Debit • `GIFT` - Gift card • `EBT` - Electronic Balance Transfer card • `LEVELUP` - LevelUp • `VOYAGER` - Voyager fleet card • `WEX` - Wright Express fleet card • `JCB` - Japan Credit Bureau • `CUP` - China Union Pay • `UNKNOWN` - This value is reserved.
EntryMode	enum	The entry mode for this transaction

Name	Type	Description
		• SWIPE - Magnetic stripe • PROXIMITY - Contactless / NFC • BARCODE - Barcode • MANUAL - KeyedSale (**N.B.** Only as result of Initiate Keyed Sale)
ErrorMessage	String	A message indicating why the transaction could not be processed.
Token	String	The identifier received by the Merchantware client when a transaction has been issued. This value may be used to lookup a specific transaction in the history log.
TransactionDate	dateTime	The date and time when a transaction is issued, in UTC time.
TransactionType	enum	The type of transaction, and this value matches up with the TransactionType enumeration. • SALE • AUTHORIZATION • REFUND • ADDVALUE • BALANCEINQUIRY
ResponseType	enum	Possible values: • SINGLE • MULTI • COMPOUND
ValidationKey	String	This ValidationKey should be used to match up with the ValidationKey received in the CreateTransaction call to validate that the response from the CED is authentic and matches the response from the gateway.
AdditionalParameters	Object	Contains extra transaction information when applicable. Possible fields: • <SignatureData>string</SignatureData> - contains the vector signature data if it's collected by the CED. • <AmountDetails><UserTip/><Cashback/><Donation/><Surcharge/><RemainingCardBalance/><Discount/></AmountDetails>. **Note:** For EBT transactions RemaningCardBalance displays the balance remaining on the customer's EBT card for CASH or SNAP. • <Discount> object - there may be more than one discount object inside the response. Take a look at the sample responses for details. • <EbtDetails><EbtType/><FnsId/><Balances><CashAvailableBalance/><SnapAvailableBalance/></Balances></EbtDetails>.<EbtType> can be CASH or SNAP. <FnsId> is shown only when <EbtType> is SNAP. • <KeyedDetails><ExpirationDate>string</ExpirationDate><AvsStreetZipCode>string</AvsStreetZipCode><AvsResponse>char</AvsResponse><CvResponse>char</CvResponse></KeyedDetails>

Name	Type	Description
		• <FsaCard>bool</FsaCard> - Was an FSA card tendered? Will be present only in the response if FSA Healthcare amounts were specified in the create transaciton call. • <Loyalty><AccountNumber>string</AccountNumber><Balances><PointsBalance>int</PointsBalance><AmountBalance>decimal</AmountBalance></Balances></Loyalty> - Will be present only if the gift/loyalty function is active. • <Survey><MerchantMessage>string</MerchantMessage><CustomerMessage>string</CustomerMessage></Survey> - Will be present only for merchants who integrate with TruRating. Warning New fields may be added to Additional Parameters at any time. It is important to ensure you parse all fields and discard those which are not required.
EMV	Object	Contains EMV transaction information when applicable.

2.5.6 EMV object parameters

Name	Type	Description
Aid	String	ID of the EMV application selected by the customer.
ApplicationLabel	String	The application label as displayed to the customer.
PINStatement	String	Statement to signify 'PIN Verified' or 'PIN Locked'.

These EMV fields are the ones required to be used for compliance. For the additional optional fields, see the V2 Examples available above.

2.5.7 V2 examples non-EMV

2.5.7.1 XML response example

```
<?xml version="1.0" encoding="utf-8"?>

<TransactionResult>

    <Status>APPROVED</Status>

    <AmountApproved>6.01</AmountApproved>

    <AuthorizationCode>OK0601</AuthorizationCode>

    <Cardholder>TEST CARD/GENIUS</Cardholder>

    <AccountNumber>**********0026</AccountNumber>

    <PaymentType>VISA</PaymentType>

    <EntryMode>SWIPE</EntryMode>

    <ErrorMessage/>

    <Token>601601601</Token>
```

```
    <TransactionDate>3/29/2016 2:58:41 PM</TransactionDate>
    <TransactionType>SALE</TransactionType>
    <ResponseType>SINGLE</ResponseType>
    <ValidationKey>bd68008f-d4fc-45c2-8eb9-2656fd56fd29</ValidationKey>
    <AdditionalParameters>
        <SignatureData/>
        <AmountDetails>
            <UserTip>0.00</UserTip>
            <Cashback>0.00</Cashback>
            <Donation>0.00</Donation>
            <Surcharge>0.00</Surcharge>
            <Discount>
                <Total>0.00</Total>
            </Discount>
        </AmountDetails>
        <Loyalty>
            <AccountNumber>string</AccountNumber>
            <Balances>
                <PointsBalance>int</PointsBalance>
                <AmountBalance>decimal</AmountBalance>
            </Balances>
        </Loyalty>
    </AdditionalParameters>
</TransactionResult>
```

2.5.7.2 JSON response example

```
{
  "Status": "APPROVED",
  "AmountApproved": "6.01",
  "AuthorizationCode": "OK0601",
```

```
  "Cardholder": "TEST CARD\/GENIUS",
  "AccountNumber": "************0026",
  "PaymentType": "VISA",
  "EntryMode": "SWIPE",
  "ErrorMessage": "",
  "Token": "601601601",
  "TransactionDate": "3\/29\/2016 3:01:09 PM",
  "TransactionType": "SALE",
  "ResponseType": "SINGLE",
  "ValidationKey": "65ef9de0-bd3f-40f8-9186-8687971c856b",
  "AdditionalParameters": {
    "SignatureData": "",
    "AmountDetails": {
      "UserTip": "0.00",
      "Cashback": "0.00",
      "Donation": "0.00",
      "Surcharge": "0.00",
      "Discount": {
        "Total": "0.00"
      }
    }
  }
}
```


2.5.8 V2 examples

2.5.8.1 XML response example


```
<?xml version="1.0" encoding="utf-8"?>
<TransactionResult>
    <Status>APPROVED</Status>
    <AmountApproved>6.01</AmountApproved>
```

```
<AuthorizationCode>OK0601</AuthorizationCode>
<Cardholder>TEST CARD/GENIUS</Cardholder>
<AccountNumber>**********0026</AccountNumber>
<PaymentType>VISA</PaymentType>
<EntryMode>SWIPE</EntryMode>
<ErrorMessage/>
<Token>601601601</Token>
<TransactionDate>3/29/2016 2:58:41 PM</TransactionDate>
<TransactionType>SALE</TransactionType>
<ResponseType>SINGLE</ResponseType>
<ValidationKey>bd68008f-d4fc-45c2-8eb9-2656fd56fd29</ValidationKey>
<AdditionalParameters>
    <SignatureData/>
    <AmountDetails>
        <UserTip>0.00</UserTip>
        <Cashback>0.00</Cashback>
        <Donation>0.00</Donation>
        <Surcharge>0.00</Surcharge>
        <Discount>
            <Total>0.00</Total>
        </Discount>
    </AmountDetails>
    <EMV>
        <ApplicationInformation>
            <Aid>A0000000032010</Aid>
            <ApplicationLabel>Visa Electron</ApplicationLabel>
            <ApplicationExpiryDate>1/1/2017</ApplicationExpiryDate>
            <ApplicationEffectiveDate>1/1/2014</ApplicationEffectiveDate>
            <ApplicationInterchangeProfile>1359</ApplicationInterchangeProfile>
            <ApplicationVersionNumber>1</ApplicationVersionNumber>
```

```
    <ApplicationTransactionCounter>826</ApplicationTransactionCounter>
    <ApplicationUsageControl>1234</ApplicationUsageControl>
    <ApplicationPreferredName>Visa Electron</ApplicationPreferredName>
    <ApplicationDisplayName>Visa Electron</ApplicationDisplayName>
</ApplicationInformation>
<CardInformation>
    <MaskedPan>************0026</MaskedPan>
    <PanSequenceNumber>1</PanSequenceNumber>
    <CardExpiryDate>01/19</CardExpiryDate>
</CardInformation>
<ApplicationCryptogram>
    <CryptogramType>TC</CryptogramType>
    <Cryptogram>9F424AA68EB17A2</Cryptogram>
</ApplicationCryptogram>
<CvmResults>010001</CvmResults>
<IssuerApplicationData>123AB123456123</IssuerApplicationData>
<TerminalVerificationResults>123AB12345</TerminalVerificationResults>
<UnpredictableNumber>2147483647</UnpredictableNumber>
<Amount>
    <AmountAuthorised>6.01</AmountAuthorised>
    <AmountOther>0.00</AmountOther>
</Amount>
<PosEntryMode>05</PosEntryMode>
<TerminalInformation>
    <TerminalType>22</TerminalType>
    <IfdSerialNumber>1211-121-1212</IfdSerialNumber>
    <TerminalCountryCode>840</TerminalCountryCode>
    <TerminalID>12345678</TerminalID>
    <TerminalActionCodeDefault>DC404F800</TerminalActionCodeDefault>
    <TerminalActionCodeDenial>DC404F800</TerminalActionCodeDenial>
```

```
                <TerminalActionCodeOnline>DC404F800</TerminalActionCodeOnline>
            </TerminalInformation>
            <TransactionInformation>
                <TransactionType>GOODS</TransactionType>
                <TransactionCurrencyCode>840</TransactionCurrencyCode>
                <TransactionStatusInformation>E800</TransactionStatusInformation>
            </TransactionInformation>
            <CryptogramInformationData>AB</CryptogramInformationData>
            <PINStatement>PIN Verified</PINStatement>
            <CvmMethod>ONLINE PIN</CvmMethod>
            <IssuerActionCodeDefault>1234567890</IssuerActionCodeDefault>
            <IssuerActionCodeDenial>1234567890</IssuerActionCodeDenial>
            <IssuerActionCodeOnline>1234567890</IssuerActionCodeOnline>
            <AuthorizationResponseCode>00</AuthorizationResponseCode>
        </EMV>
        <Loyalty>
            <AccountNumber>string</AccountNumber>
            <Balances>
                <PointsBalance>int</PointsBalance>
                <AmountBalance>decimal</AmountBalance>
            </Balances>
        </Loyalty>
    </AdditionalParameters>
</TransactionResult>
```

2.5.8.2 JSON response example

```
{
  "Status": "APPROVED",
  "AmountApproved": "6.01",
  "AuthorizationCode": "OK0601",
```

```
"Cardholder": "TEST CARD\/GENIUS",
"AccountNumber": "************0026",
"PaymentType": "VISA",
"EntryMode": "SWIPE",
"ErrorMessage": "",
"Token": "601601601",
"TransactionDate": "3\/29\/2016 3:01:09 PM",
"TransactionType": "SALE",
"ResponseType": "SINGLE",
"ValidationKey": "65ef9de0-bd3f-40f8-9186-8687971c856b",
"AdditionalParameters": {
  "SignatureData": "",
  "AmountDetails": {
    "UserTip": "0.00",
    "Cashback": "0.00",
    "Donation": "0.00",
    "Surcharge": "0.00",
    "Discount": {
      "Total": "0.00"
    }
  },
  "EMV": {
    "ApplicationInformation": {
      "Aid": "A0000000032010",
      "ApplicationLabel": "Visa Electron",
      "ApplicationExpiryDate": "1\/1\/2017",
      "ApplicationEffectiveDate": "1\/1\/2014",
      "ApplicationInterchangeProfile": "1359",
      "ApplicationVersionNumber": "1",
      "ApplicationTransactionCounter": "826",
```

```
    "ApplicationUsageControl": "1234",
    "ApplicationPreferredName": "Visa Electron",
    "ApplicationDisplayName": "Visa Electron"
  },
  "CardInformation": {
    "MaskedPan": "************0026",
    "PanSequenceNumber": "1",
    "CardExpiryDate": "01\/19"
  },
  "ApplicationCryptogram": {
    "CryptogramType": "TC",
    "Cryptogram": "9F424AA68EB17A2"
  },
  "CvmResults": "010001",
  "IssuerApplicationData": "123AB123456123",
  "TerminalVerificationResults": "123AB12345",
  "UnpredictableNumber": "2147483647",
  "Amount": {
    "AmountAuthorised": "6.01",
    "AmountOther": "0.00"
  },
  "PosEntryMode": "05",
  "TerminalInformation": {
    "TerminalType": "22",
    "IfdSerialNumber": "1211-121-1212",
    "TerminalCountryCode": "840",
    "TerminalID": "12345678",
    "TerminalActionCodeDefault": "DC404F800",
    "TerminalActionCodeDenial": "DC404F800",
    "TerminalActionCodeOnline": "DC404F800"
```

```
      },
      "TransactionInformation": {
        "TransactionType": "GOODS",
        "TransactionCurrencyCode": "840",
        "TransactionStatusInformation": "E800"
      },
      "CryptogramInformationData": "AB",
      "PINStatement": "PIN Verified",
      "CvmMethod": "ONLINE PIN",
      "IssuerActionCodeDefault": "1234567890",
      "IssuerActionCodeDenial": "1234567890",
      "IssuerActionCodeOnline": "1234567890",
      "AuthorizationResponseCode": "00",
      "UnexpectedEMVField": null,
      "UnexpectedEmptyField1": null,
      "UnexpectedEmptyField2": null,
      "UnexpectedEmptyField3": null
    }
  }
}
```

2.6 Retrieve transaction details

The DetailsByTransportKey web service method allows the Point of Sale developers to send in the original TransportKey and request additional payment information at a later time.

Content-Type: text/xml; charset=utf-8

SOAPAction: http://schemas.merchantwarehouse.com/genius/10/Reporting/DetailsBy TransportKey

Test Endpoint	https://genius.merchantware.net/v1/Reporting.asmx
Live Endpoint	https://genius.merchantware.net/v1/Reporting.asmx

2.6.1 Request parameters

Name	Type	Size	Description
Name	String	1-160	The name of the business or organization owning the Merchantware account.
SiteID	String	8-160	The site identifier of a location or storefront owned by the Merchantware account owner.
Key	String	1-160	The software key or password for the site accessing a Merchantware account.
TransportKey	String	1-160	The unique key generated using the createTransaction method.

2.6.2 Response parameters

Name	Type	Description
Status	String	The Status of the transaction, whether approved or declined. This value may also have other definitions depending on CardType and context.
ErrorMessage	String	A message indicating why the transaction could not be processed.
TotalAmountApproved	String	The amount of the transaction that was approved. If partial authorizations are enabled, this will be the authorized amount and may be different than the requested amount.
RequestedAmount	String	The requested amount of the transaction.
ResponseType	String	SINGLE, MULTI or Compound. Typical credit, debit and EBT payment types will return 'SINGLE' as the ResponseType. MULTI or COMPOUND response types are returned for additional payment types such as gift and loyalty.
Payment Details	PaymentDetail	Collection of detail objects for the transaction processed. If more than one payment type was processed to complete the transaction, each sub-transaction will be in the collection. For

Name	Type	Description
		example, if a $100 transaction is processed using a $30 gift card and $70 credit card transaction, there will be a payment detail for each one within the payment details object.

2.6.3 Examples

2.6.3.1 Soap request example

```
<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

  <soap:Body>

    <DetailsByTransportKey
xmlns="http://schemas.merchantwarehouse.com/genius/10/Reporting">

      <Name>string</Name>

      <SiteID>string</SiteID>

      <Key>string</Key>

      <TransportKey>string</TransportKey>

    </DetailsByTransportKey>

  </soap:Body>

</soap:Envelope>
```

2.6.3.2 Soap response example

```
<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

  <soap:Body>

    <DetailsByTransportKeyResponse xmlns="http://tempuri.org/">

      <DetailsByTransportKeyResult>

        <Status>UNKNOWN or APPROVED or FAILED or DECLINED or DECLINED_DUPLICATE or
REFERRAL</Status>

        <ErrorMessage>string</ErrorMessage>

        <TotalAmountApproved>decimal</TotalAmountApproved>

        <RequestedAmount>decimal</RequestedAmount>
```

```
        <ResponseType>UNKNOWN or SINGLE or MULTI or COMPOUND</ResponseType>

        <PaymentDetails>

          <PaymentDetail>

            <PaymentType>UNKNOWN or AMEX or DISCOVER or MASTERCARD or VISA or DEBIT or
EBT or EGC or WEX or VOYAGER or JCB or CUP or LU</PaymentType>

            <Status>UNKNOWN or APPROVED or FAILED or DECLINED or DECLINED_DUPLICATE or
REFERRAL</Status>

            <ErrorMessage>string</ErrorMessage>

            <TransactionType>UNKNOWN or SALE or REFUND or
AUTHORIZATION</TransactionType>

            <Token>string</Token>

            <AuthorizationCode>string</AuthorizationCode>

            <Customer>string</Customer>

            <Email>string</Email>

            <PhoneNumber>string</PhoneNumber>

            <AccountNumber>string</AccountNumber>

            <ExpirationDate>string</ExpirationDate>

            <EntryMode>UNKNOWN or MANUAL or SWIPE or AUTHORIZATION or PROXIMITY or
BARCODE</EntryMode>

            <TransactionDate>dateTime</TransactionDate>

            <AmountDetail>

                <AmountApproved>30.00</AmountApproved>

                <AmountCharged>0</AmountCharged>

                <TaxAmount>1.88</TaxAmount>

                <TipAmount>2.50</TipAmount>

                <UserTipAmount>2.50</UserTipAmount>

                <DiscountAmount>4.00</DiscountAmount>

                <VoucherAmount>1.00</VoucherAmount>

                <RemainingCardBalance>100.00</RemainingCardBalance>

            </AmountDetail>

            <SignatureDetail>

                <SignatureType>VECTOR</SignatureType>
```

```
                <Signature>10,10^110,110^0,65535^10,110^110,10^0,65535^~</Signature>
            </SignatureDetail>
            <GiftDetail>
                <Balance>25.35</Balance>
            </GiftDetail>
            <LoyaltyDetail>
                <Visits>25</Visits>
                <LastVisit>2011-07-02T00:00:00</LastVisit>
                <LifetimeSpend>560.33</LifetimeSpend>
                <Balance>144</Balance>
            </LoyaltyDetail>
            <AdditionalResponseParameters>
                <FsaCard>false</FsaCard>
            </AdditionalResponseParameters>
          </PaymentDetail>
        </PaymentDetails>
        <AdditionalResponseParameters />
      </DetailsByTransportKeyResult>
    </DetailsByTransportKeyResponse>
  </soap:Body>
</soap:Envelope>
```

2.7 Additional terminal functions

2.7.1 Cancel transaction

`Action=Cancel` Sends a request to the Genius device to cancel the current transaction. You can send a `Cancel` request using HTTP, HTTPs, Android Intents, or Javascript.

2.7.1.1 HTTP or HTTPS

Sample request string	`http://[CED-IP-Address]:8080/v1/pos?Action=Cancel&Format=XML`

2.7.1.2 Android Intents request parameters

Name	Required	Type	Description
ResultReceiverID	Yes	String	Genius returns the result using this ResultReceiverID. You should use your POS name as your ResultReceiverID.
Action	Yes	String	`Cancel` – Method to initiate request
Format	Yes	String	The format that the CED should respond in. Possible values are: • `XML` • `JSON`
URI	Yes	String	/[version]/pos [version] indicates the version of call set in settings

2.7.1.3 Android Intents code

```
String URI =
PreferenceManager.getDefaultSharedPreferences(getActivity().getApplicationContext()).g
etString("cancelVer", getString(R.string.version1)) + "/pos";

Intent intent = new Intent(MainActivity.REQUEST_ACTION);
intent.putExtra("ResultReceiverID",MainActivity.RECEIVERID);
intent.putExtra("Action", "Cancel");
intent.putExtra("Format",
PreferenceManager.getDefaultSharedPreferences(getActivity().getApplicationContext()).g
etString("format", getString(R.string.XML)));
intent.putExtra("URI", URI);
getActivity().sendBroadcast(intent);
```

2.7.1.5 Response parameters

Name	Type	Description
Status	String	The status of the cancel request. Possible values are Cancelled, TransactionApproved_NoSignatureCollected, Denied, Error.
ResponseMessage	String	Message for Denied or Error transactions. (I.e. Can't connect, can't cancel while processing, terminal at idle screen).
AdditionalParameters	Object	Placeholder for any necessary future fields.

2.7.1.6 Examples

2.7.1.6.1 XML response example

```
<?xml version="1.0" encoding="utf-8"?>

<CancelResult>

    <Status>Cancelled</Status>

    <ResponseMessage></ResponseMessage>

    <AdditionalParameters></AdditionalParameters>

</CancelResult>
```

2.7.1.6.2 JSON response example

```
{

  "Status": "Cancelled",

  "ResponseMessage": "",

  "AdditionalParameters": {


  }

}
```

2.7.2 Initiate keyed entry

`Action=InitiateKeyedEntry` can be used for credit/debit cards and gift/loyalty cards. It sends a request to the CED to process manual entry of payment data, which may include PAN, Expiration Date, CVV, and/or Zip. This function can be used only when a transaction has been staged and the Genius device displays the selection screen, swipe screen, or gift card capture screen. Using this function without an existing transaction on the device will result in a failed response. If you intend to use this feature, you must contact the Certification team.

Additionally, you can select the payment method by supplying the **PaymentType** parameter. Currently, we support only the **GIFT** PaymentType.

You can send an `InitiateKeyedEntry` request using HTTP, HTTPS, Android Intents, or Javascript:

2.7.2.1 HTTP or HTTPS

Sample request string	`http://[CED-IP-Address]:8080/v2/pos?Action=InitiateKeyedEntry&Format=XML`
	
	`http://[CED-IP-Address]:8080/v2/pos?Action=InitiateKeyedEntry&PaymentType=GIFT&Format=XML`

2.7.2.2 Android Intents request parameters

Name	Required	Type	Description
ResultReceiverID	Yes	String	Genius returns the result using this ResultReceiverID. You should use your POS name as your ResultReceiverID.
Action	Yes	String	`InitiateKeyedEntry` – Method to initiate request
PaymentType	Optional	String	• `GIFT`
Format	Yes	String	The format that the CED should respond in. Possible values are: • `XML` • `JSON`
URI	Yes	String	/[version]/pos [version] indicates the version of call set in settings

2.7.2.3 Android Intents code

```
String URI =
PreferenceManager.getDefaultSharedPreferences(getActivity().getApplicationContext()).g
etString("keyedEntryVer", getString(R.string.version1)) + "/pos";

Intent intent = new Intent(MainActivity.REQUEST_ACTION);
intent.putExtra("ResultReceiverID",MainActivity.RECEIVERID);
intent.putExtra("Action", "InitiateKeyedEntry");
intent.putExtra("PaymentType", paymentType.name());
intent.putExtra("Format",
PreferenceManager.getDefaultSharedPreferences(getActivity().getApplicationContext()).g
etString("format", getString(R.string.XML)));
intent.putExtra("URI", URI);
getActivity().sendBroadcast(intent);
```

2.7.2.4 Request parameters

Name	Type	Size	Description
PaymentType	String	1-5	(Optional) The payment type that will be used for the keyed entry. This has the same result as pressing the payment method button. Currently, we support only the **GIFT** PaymentType.

2.7.2.6 Response parameters

Name	Type	Description
Status	String	The status of the transaction. Values can be Success, Failure or Declined.

2.7.2.7 Examples

2.7.2.7.1 XML response example

```
<?xml version="1.0" encoding="utf-8"?>

<InitiateKeyedEntryResult>

    <Status>Success</Status>

</InitiateKeyedEntryResult>
```

2.7.2.7.2 JSON response example

```
{

  "Status": "Success"

}
```

2.8 Extended functionality

Using Genius's extended functionality you can implement the following post-sale functions:

- Void
- Refund
- Post-authorization
- Repeat sale transaction
- Settle batch (close day)
- Stored value – activate
- Stored value – add value
- Stored value – balance inquiry

Note: Visit https://cayan.com/developers/genius/transactions#extended to find information about Genius's extended functionality.

2.9 Genius Your Way

2.9.1 Start get agreement

The `GetAgreement` request prompts the Genius device to display a multi-line agreement with Accept and Decline buttons to capture the customer's choice. You can send a `GetAgreement` request using HTTP, HTTPS, Android Intents, or Javascript:

2.9.1.1 HTTP or HTTPS

GetAgreement
POST Method
	`http://[CED-IP-Address]:8080/v1/pos`
	
	POST data, content type x-www-form-urlencoded:
	
	`Action=GetAgreement&RequestID=xxx&Title=xxx&AgreementText=xxx&AcceptLabel=xxx&DeclineLabel=xxx&Format=xxx`

2.9.1.2 Android Intents request parameters

Name	Type	Required	Description
ResultReceiverID	Yes	String	Genius returns the result using this ResultReceiverID. You should use your POS name as your ResultReceiverID.
Action	String	Yes	`GetAgreement` - Method to initiate an agreement.
RequestID	String	Yes	An ID supplied by the PoS that is used to identify the GetAgreement request.
Title	String	Optional	A title that is displayed on the Genius device.
AgreementText	String	Optional	The text of the agreement. Tabs are replaced with four spaces prior to being displayed.
AcceptLabel	String	Optional	The label applied to the **Accept** button. Default = "Accept"
DeclineLabel	String	Optional	The label applied to the **Decline** button. Default = "Decline"
Format	String	Yes	The format that the CED should respond in. Possible values are: • `XML`
URI	Yes	String	/[version]/pos [version] indicates the version of call set in settings

2.9.1.4 Android Intents code

```
String URI =
PreferenceManager.getDefaultSharedPreferences(getActivity().getApplicationContext()).g
etString("getAgreementVer", getString(R.string.version1)) + "/pos";

Intent intent = new Intent(MainActivity.REQUEST_ACTION);
intent.putExtra("ResultReceiverID",MainActivity.RECEIVERID);
intent.putExtra("Action", "GetAgreement");
intent.putExtra("RequestID", RequestID);
if (!Title.isEmpty())
    intent.putExtra("Title", Title);
if (!AgreementText.isEmpty())
    intent.putExtra("AgreementText", AgreementText);
if (!AcceptLabel.isEmpty())
    intent.putExtra("AcceptLabel", AcceptLabel);
if (!DeclineLabel.isEmpty())
    intent.putExtra("DeclineLabel", DeclineLabel);
intent.putExtra("Format",
PreferenceManager.getDefaultSharedPreferences(getActivity().getApplicationContext()).g
etString("format", getString(R.string.XML)));
intent.putExtra("URI", URI);
getActivity().sendBroadcast(intent);
```

2.9.1.5 Response parameters

Name	Type	Description
Status	String	Status of the GetAgreement response. Possible values: • `Success` • `Timeout` • `POSCancelled`
RequestID	String	The original identifier that was sent by the PoS with the GetAgreement request.

2.9.1.6 Examples

2.9.1.6.1 XML response example

```
<?xml version="1.0" encoding="utf-8"?>

<AgreementTextResponse>

    <RequestID>1234</RequestID>

    <Status>Accepted/Status>

</AgreementTextResponse>
```

2.9.1.6.2 JSON response example

```
{

    "RequestID": "1234",

    "Status": "Accepted"

}
```

2.9.2 Start signature capture

You can send a `GetSignature` request using HTTP, HTTPS, or Android Intents:

2.9.2.1 HTTP or HTTPS

GetSignature GET Method	`http://[CED-IP-Address]:8080/v1/pos?Action=GetSignature&RequestID=xxx&Title=xxx&Format=xxx`

2.9.2.2 HTTP or HTTPS request parameters

Key	Type	Size	Description
Action	String	`1-16`	`GetSignature` - Method to initiate a signature capture.
RequestID	String	1-40	An ID supplied by the PoS that is used to identify the GetSignature request.
Title	String	1-36	A title that is displayed on the Genius device.
Format	String	1-5	The format that the CED should respond in. Possible values are: • `XML` • `JSON`

2.9.2.3 Android Intents request parameters

Name	Required	Type	Description
ResultReceiverID	Yes	String	Genius returns the result using this ResultReceiverID. You should use your POS name as your ResultReceiverID.
Action	Yes	String	`GetSignature` - Method to initiate an agreement.
RequestID	Yes	String	An ID supplied by the PoS that is used to identify the GetSignature request.
Title	Optional	String	A title that is displayed on the Genius device.
Format	Yes	String	The format that the CED should respond in. Possible values are: • `XML` • `JSON`
URI	Yes	String	/[version]/pos [version] indicates the version of call set in settings

2.9.2.4 Android Intents code

```
String URI =
PreferenceManager.getDefaultSharedPreferences(getActivity().getApplicationContext()).g
etString("getSignatureVer", getString(R.string.version1)) + "/pos";

Intent intent = new Intent(MainActivity.REQUEST_ACTION);
intent.putExtra("ResultReceiverID",MainActivity.RECEIVERID);
intent.putExtra("Action", "GetSignature");
intent.putExtra("RequestID", RequestID);
if (!Title.isEmpty())
    intent.putExtra("Title", Title);
intent.putExtra("Format",
PreferenceManager.getDefaultSharedPreferences(getActivity().getApplicationContext()).g
etString("format", getString(R.string.XML)));
intent.putExtra("URI", URI);
getActivity().sendBroadcast(intent);
```

2.9.2.5 Important clarification

2.9.2.5.1 Cancelling incomplete signatures

The Cancel request should be used if the signature capture is not completed. For example, the consumer chooses not to sign. This cancels only the signature request, send a second Cancel request to cancel the full transaction.

2.9.2.6 Response parameters

Name	Type	Description
Status	String	Status of the GetSignature response. Possible values: • `Success` • `Timeout` • `POSCancelled`
SignatureData	String	The customer's signature as vector data.
RequestID	MessageList	The original identifier that was sent by the PoS with the GetSignature request.

2.9.2.7 Examples

2.9.2.7.1 XML response example

```
<?xml version="1.0" encoding="utf-8"?>

<GetSignatureResult>

    <Status>Success</Status>

    <SignatureData>306,48^306,49^307,49^308,50^309,50^310,51^311,52^313,53^315,55^318,
56^322,59^327,61^332,64^338,67^344,70^352,74^360,77^369,81^377,84^386,87^0,65535^~</Si
gnatureData>

    <RequestID>1234</RequestID>

</GetSignatureResult>
```

2.9.2.7.2 JSON response example


```
{

  "Status": "Success",

  "SignatureData":
"189,45^190,45^190,44^191,43^192,42^192,41^193,40^194,39^195,39^196,37^197,36^199,35^2
03,33^206,33^209,32^213,32^217,32^223,33^229,34^236,35^243,36^0,65535^~",

  "RequestID": "1234"

}
```


CAYAN™
The Payment Possibilities Company™