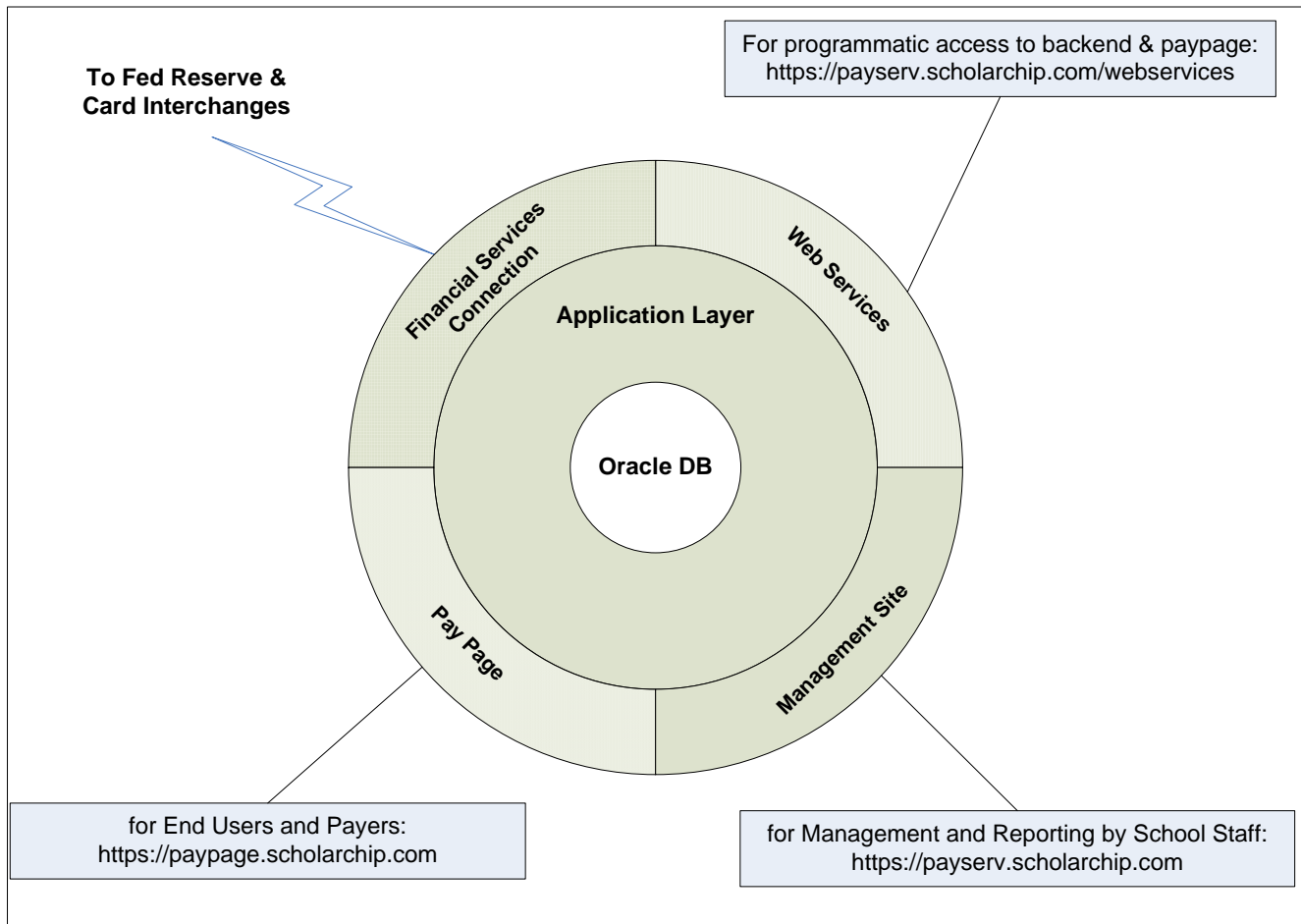




**ScholarChip Card Company, LLC.
PayServ Integration Document
StoreFront Webservice v1.2**



ScholarChip Payment Gateway Overview (PayServ)



AddItemToOrder

1. Purpose:

This function is used to add Inventory items to a pre-existing order.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sSeqidOrder – This is ScholarChip's unique identifier for an order that already exists in the Payserv system. You must call the **DoesOrderExist** function with your unique identifier to retrieve this value.

sSeqidItem – This is ScholarChip's unique identifier for an item that already exists in the Payserv system. You must call the **DoesItemExist** function with your unique identifier to retrieve this value.

sQuantity – This is the quantity of the item(s) that you would like to add to the order. This value must not exceed any order limitations (limit_per_order etc.).

3. Return Values:

This function will return the string representation of “1” on successful execution. In the event of an error, this function will return a human-readable error message containing the word “ERROR.” If you receive an error when calling this function, please use the **GetOrderItems** function to view what was successfully added to the order.



AddLineItemToOrder

1. Purpose:

This function is used to add a line item to an order. A line item is just a way to tell ScholarChip that you would like to take a certain amount for a specific EAGL code (Enterprise Accounting and General Ledger code).

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sSeqidOrder – This is ScholarChip's unique identifier for an order that already exists in the Payserv system. You must call the **DoesOrderExist** function with your unique identifier to retrieve this value.

sLineItemID – This is your identifier for this line item. We allow up to 32 characters in this field. It is recommended that you store this identifier locally for future queries.

sDescription – This is the description of the transaction you would like displayed to the student during the checkout process. We allow up to 64 characters in this field.

sPrice – This is the string representation of the amount you would like the line item to be. It must only contain the digits 0-9 and a decimal point, if needed.

sEAGL – This is a pre-existing EAGL (enterprise account and general ledger codes) that your university has access to. We use this value to report the purchase of this item back to your General Ledger.

3. Return Values:

This function will return the string representation of "1" on successful execution. In the event of an error, this function will return a human-readable error message containing the word "ERROR." If you receive an error when calling this function, please use the **GetOrderItems** function to view what was successfully added to the order.

AddOrderToStudent

1. Purpose:

This function is used to add a pre-existing order to a student in the Payserv system.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sSeqidStudent – This is ScholarChip's unique identifier for a student that already exists in the system. You must call the **DoesStudentExist** function with your unique identifier to retrieve this value.

sSeqidOrder – This is ScholarChip's unique identifier for an order that already exists in the system. You must call the **DoesOrderExist** function with your unique identifier to retrieve this value.

3. Return Values:

This function will return "1" on successful execution. In the event of an error, this function will return a human-readable error message containing the word "ERROR".



AddTransactionToStudent

1. Purpose:

This function is used to create a single transaction attached to a pre-existing student in the Payserv system.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sSeqidStudent – This is ScholarChip's unique identifier for a student that already exists in the system. You must call the **DoesStudentExist** function with your unique identifier to retrieve this value.

sTransactionID – This is your unique identifier. We allow up to 32 characters in this field. It is recommended that you store this identifier locally for future queries.

sAmount – This is the string representation of the amount you would like the transaction to be. It must only contain the digits 0-9 and a decimal point, if needed.

sDescription – This is the description of the transaction you would like displayed to the student during the checkout process. We allow up to 64 characters in this field.

3. Return Values:

This function will return ScholarChip's unique identifier for this transaction. In the event of an error, this function will return a human-readable error message containing the word "ERROR".



AddUnivLineItemToOrder

1. Purpose:

This function is used to add a line item to an order that belongs to a university you have authority over. A line item is just a way to tell ScholarChip that you would like to take a certain amount for a specific EAGL code (Enterprise Accounting and General Ledger code).

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sSeqidOrder – This is ScholarChip's unique identifier for an order that already exists in the Payserv system. You must call the **DoesOrderExist** function with your unique identifier to retrieve this value.

sUnivID – This is a university identifier that you and ScholarChip have agreed upon. This ID must belong to a university you have authority over.

sLineItemID – This is your identifier for this line item. We allow up to 32 characters in this field. It is recommended that you store this identifier locally for future queries.

sDescription – This is the description of the transaction you would like displayed to the student during the checkout process. We allow up to 64 characters in this field.

sPrice – This is the string representation of the amount you would like the line item to be. It must only contain the digits 0-9 and a decimal point, if needed.

sEAGL – This is a pre-existing EAGL (enterprise account and general ledger codes) that your university has access to. We use this value to report the purchase of this item back to your General Ledger.

3. Return Values:

This function will return ScholarChip's unique identifier for this line item. In the event of an error, this function will return a human-readable error message containing the word "ERROR." If you receive an error when calling this function, please use the **GetOrderItems** function to view what was successfully added to the order.



CreateItem

1. Purpose:

This function is used to create inventory items in the Payserv system. These items can then be added to orders.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sEAGL – This is a pre-existing EAGL (enterprise account and general ledger codes) that your university has access to. We use this value to report the purchase of this item back to your General Ledger.

sItemID – This is your unique identifier. We allow up to 32 alphanumeric characters in this field. Is it recommended that you store this identifier locally for future queries.

sItemName – This is how you would like the item represented on the Payserv system. This value will also be displayed to the user during the checkout process. We allow up to 64 characters in this field.

sPrice – This is the string representation of the amount you would like the item to be. It must only contain the digits 0-9 and a decimal point, if needed.

sItemDescription – This is a longer description of the item. This value will be displayed to the user during the checkout process. We allow up to 512 characters in this field.

sShippingRequired – This must be either “Y” or “”. If an item is marked for shipping, we will collect shipping information during the checkout process.

sLimitPerOrder – This must be a positive integer. If a limit per order is set, the checkout process will limit the number that can be added to the cart.

sLocked – This must be either “Y” or “”. If an item is marked as locked, users will be unable to change the quantity in their cart during the checkout process.

sQuantity – This must be a positive integer. This is used to set the initial quantity of the item in the system. A value of -1 in this field will mark the item as having an unlimited quantity.

sLowMark – This must be a positive integer. This is used during the checkout process to stop the sale of the item if the quantity left of that item reaches the value provided. If **sQuantity** is set to -1, this value must also be set to -1.

3. Return Values:

This function will return ScholarChip's unique identifier for this item. In the event of an error, this function will return a human-readable error message containing the word “ERROR”.



CreateOrder

1. Purpose:

This function is used to create an order in the Payserv system. Once an order is created, you can use **AddItemToOrder** to start building your order.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sOrderID – This is your unique identifier. We allow up to 32 characters in this field. Is it recommended that you store this identifier locally for future queries.

3. Return Values:

This function will return ScholarChip's unique identifier for this order. In the event of an error, this function will return a human-readable error message containing the word "ERROR".



CreateStudent

1. Purpose:

This function is used to create a student in the Payserv system.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sStudentID – This is your unique identifier. We allow up to 32 characters in this field. Is it recommended that you store this identifier locally for future queries.

sFirstName – This is the first name of the student being created. We allow up to 32 characters in this field.

sLastName – This is the last name of the student being created. We allow up to 32 characters in this field.

sEmailAddress = This is the email address of the student. We allow up to 64 characters in this field.

3. Return Values:

This function will return ScholarChip's unique identifier for this student. In the event of an error, this function will return a human-readable error message containing the word "ERROR".



CreateUnivItem

1. Purpose:

This function is used to create inventory items on a university you have authority over in the Payserv system. These items can then be added to orders.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sUnivID – This is a university identifier that you and ScholarChip have agreed upon. This ID must belong to a university you have authority over.

sEAGL – This is a pre-existing EAGL (enterprise account and general ledger codes) that your university has access to. We use this value to report the purchase of this item back to your General Ledger.

sItemID – This is your unique identifier. We allow up to 32 alphanumeric characters in this field. It is recommended that you store this identifier locally for future queries.

sItemName – This is how you would like the item represented on the Payserv system. This value will also be displayed to the user during the checkout process. We allow up to 64 characters in this field.

sPrice – This is the string representation of the amount you would like the item to be. It must only contain the digits 0-9 and a decimal point, if needed.

sItemDescription – This is a longer description of the item. This value will be displayed to the user during the checkout process. We allow up to 512 characters in this field.

sShippingRequired – This must be either “Y” or “”. If an item is marked for shipping, we will collect shipping information during the checkout process.

sLimitPerOrder – This must be a positive integer. If a limit per order is set, the checkout process will limit the number that can be added to the cart.

sLocked – This must be either “Y” or “”. If an item is marked as locked, users will be unable to change the quantity in their cart during the checkout process.

sQuantity – This must be a positive integer. This is used to set the initial quantity of the item in the system. A value of -1 in this field will mark the item as having an unlimited quantity.

sLowMark – This must be a positive integer. This is used during the checkout process to stop the sale of the item if the quantity left of that item reaches the value provided. If **sQuantity** is set to -1, this value must also be set to -1.

3. Return Values:

This function will return ScholarChip's unique identifier for this item. In the event of an error, this function will return a human-readable error message containing the word “ERROR”.

CreateUnivOrder

1. Purpose:

This function is used to create an order on a university you have authority over in the Payserv system. Once an order is created, you can use **AddItemToOrder** to start building your order.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sUnivID – This is a university identifier that you and ScholarChip have agreed upon. This ID must belong to a university you have authority over.

sOrderID – This is your unique identifier. We allow up to 32 characters in this field. Is it recommended that you store this identifier locally for future queries.

3. Return Values:

This function will return ScholarChip's unique identifier for this order. In the event of an error, this function will return a human-readable error message containing the word "ERROR".



DeleteItemFromOrder

1. Purpose:

This function is used to remove an item which was previously added to an order.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sSeqidOrder – This is ScholarChip's unique identifier for the order you are trying to delete items from. You must call the **DoesOrderExist** function with your unique identifier to retrieve this value.

sSeqidItem – This is ScholarChip's unique identifier for the item you are trying to delete. You must call the **DoesItemExist** function with your unique identifier to retrieve this value.

sQuantity – This must be a positive integer. The value represents the number of the item you are trying to remove from the order. If you try to delete more than exist of the item on the order, you will receive an error.

3. Return Values:

This function will return “1” on successful execution. In the event of an error, this function will return a human-readable error message containing the word “ERROR”.



DoesItemExist

1. Purpose:

This function is used to check if an item already exists in the Payserv system.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sItemID – This is your unique identifier. We allow up to 32 characters in this field. Is it recommended that you store this identifier locally for future queries.

3. Return Values:

This function will return ScholarChip's unique identifier for this item. If the item does not exist, this function will return the string "ERROR: Item does not exist". In the event of an error, this function will return a human-readable error message containing the word "ERROR".



DoesOrderExist

1. Purpose:

This function is used to check if an order already exists in the Payserv system.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sOrderID – This is your unique identifier. We allow up to 32 characters in this field. Is it recommended that you store this identifier locally for future queries.

3. Return Values:

This function will return ScholarChip's unique identifier for this order. If the order does not exist, this function will return the string "ERROR: Order does not exist". In the event of an error, this function will return a human-readable error message containing the word "ERROR".

DoesStudentExist

1. Purpose:

This function is used to check if a student already exists in the Payserv system.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sStudentID – This is your unique identifier. We allow up to 32 characters in this field. Is it recommended that you store this identifier locally for future queries.

3. Return Values:

This function will return ScholarChip's unique identifier for this student. If the student does not exist, this function will return the string "ERROR: Student does not exist". In the event of an error, this function will return a human-readable error message containing the word "ERROR".



DoesTransactionExist

1. Purpose:

This function is used to check if a transaction already exists in the Payserv system.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sTransactionID – This is your unique identifier. We allow up to 32 characters in this field. Is it recommended that you store this identifier locally for future queries.

3. Return Values:

This function will return ScholarChip's unique identifier for this transaction. If the transaction does not exist, this function will return the string "ERROR: Transaction does not exist". In the event of an error, this function will return a human-readable error message containing the word "ERROR".



GetInventoryItems

1. Purpose:

This function is used to get a list of all inventory items that exist in the Payserv system. The results are filtered based on what you have access to.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

3. Return Values:

This function returns XML. Please reference the XML example below for a list of field values. Inside of the container there will be a list of Item fields. In the event of an error, this function will return a human-readable error message containing the word "ERROR".

<InventoryItems>

<Item>

<SEQID>460000000139</SEQID>

<DESCRIPTION>Description of test item 2</DESCRIPTION>

<DATE_CREATED>2010-03-05T11:33:20-05:00</DATE_CREATED>

<CREATED_BY>test</CREATED_BY>

<LAST_MODIFIED>2010-03-05T11:33:20-05:00</LAST_MODIFIED>

<MODIFIED_BY>test</MODIFIED_BY>

<INHERITABLE>N</INHERITABLE>

<SEQID_ENTITY>100000001062</SEQID_ENTITY>

<ENTITY_TYPE>UniversityCampus</ENTITY_TYPE>

<SHIPPING>Y</SHIPPING>

<QUANTITY_LOCKED>Y</QUANTITY_LOCKED>

<QUANTITY_LIMIT>1</QUANTITY_LIMIT>

<NAME>Test Item 2</NAME>

<CLIENT_ID>99393939393</CLIENT_ID>

<SEQID_UNIVERSITY>100000001062</SEQID_UNIVERSITY>

<PRICE>5.99</PRICE>

</Item>

</InventoryItems>



GetOrderCallbackURL

1. Purpose:

This function is used to get the callback URL of the order. This callback URL is the site where the user will be redirected after completing the checkout process. To set the callback URL please see **SetOrderCallbackURL**.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sSeqidOrder – This is ScholarChip's unique identifier for the order. You must call the **DoesOrderExist** function with your unique identifier to retrieve this value.

3. Return Values:

This function will return the callback URL on successful execution. In the event of an error, this function will return a human-readable error message containing the word "ERROR".



GetOrderItems

1. Purpose:

This function is used to get a list of all inventory items that have been previously added to an order.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sSeqidOrder – This is ScholarChip's unique identifier for the order. You must call the **DoesOrderExist** function with your unique identifier to retrieve this value.

3. Return Values:

This function returns XML. Please reference the XML example below for a list of field values. Inside of the container there will be a list of Item fields. In the event of an error, this function will return a human-readable error message containing the word "ERROR".

<OrderItems>

<Item>

<SEQID>460000000139</SEQID>

<DESCRIPTION>Description of test item 2</DESCRIPTION>

<DATE_CREATED>2010-03-05T11:33:20-05:00</DATE_CREATED>

<CREATED_BY>test</CREATED_BY>

<LAST_MODIFIED>2010-03-05T11:33:20-05:00</LAST_MODIFIED>

<MODIFIED_BY>test</MODIFIED_BY>

<INHERITABLE>N</INHERITABLE>

<SEQID_ENTITY>100000001062</SEQID_ENTITY>

<ENTITY_TYPE>UniversityCampus</ENTITY_TYPE>

<SHIPPING>Y</SHIPPING>

<QUANTITY_LOCKED>Y</QUANTITY_LOCKED>

<QUANTITY_LIMIT>1</QUANTITY_LIMIT>

<NAME>Test Item 2</NAME>

<CLIENT_ID>99393939393</CLIENT_ID>

<SEQID_UNIVERSITY>100000001062</SEQID_UNIVERSITY>

<PRICE>5.99</PRICE>

</Item>

</OrderItems>



GetOrderStatus

1. Purpose:

This function is used to get the status of a previously created order.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sSeqidOrder – This is ScholarChip's unique identifier for the order. You must call the **DoesOrderExist** function with your unique identifier to retrieve this value.

3. Return Values:

This function will return the status of the order. Please see the list of status codes below for a full list of return values. In the event of an error, this function will return a human-readable error message containing the word "ERROR".

GetOrderStatus Return Codes:

1. *Cancelled* – The order has been cancelled.
2. *Expired* – The order was created, however, the user failed to complete the checkout process within 30 minutes of order creation.
3. *Initialized* – The order was created, however, the user has not yet completed the checkout process.
4. *Invalid* – The order was created, however, it does not contain any inventory items to be displayed during the checkout process. A valid order has at least one inventory item in its cart.
5. *Pending* – The user has completed the checkout process and all associated transaction have been processed. This is used in the event that an order requires a manual step like shipping. Once you have completed the manual step, you will need to call the **MarkOrderProcessed** function to complete the order.
6. *Processed* – The user has completed the checkout process, all associated transactions have been processed, and all manual steps (shipping) have been completed.
7. *Failed* – The user has completed the checkout process, however an error occurred during payment submission. No funds were withdrawn from the user's account(s).



GetOrderURL

1. Purpose:

This function is used to get the order URL you will need to direct your users to for the checkout process.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sSeqidOrder – This is ScholarChip's unique identifier for the order. You must call the **DoesOrderExist** function with your unique identifier to retrieve this value.

3. Return Values:

This function returns the checkout URL. In the event of an error, this function will return a human-readable error message containing the word "ERROR".



GetToken

1. Purpose:

This function serves as your single point of entry into the StoreFront.asmx WebService.

2. Input Parameters:

sUserid – This is the username that ScholarChip provided to you. In order to make WebService calls with this user, it must be marked as having WebService access on the Payserv system.

sPassword – This is the username that ScholarChip provided to you.

3. Return Values:

This function returns a 32 alphanumeric string which you will need to store in order to call any other function in this WebService. In the event of an error, this function will return a human-readable error message containing the word "ERROR".



GetTransactionCallbackURL

1. Purpose:

This function is used to get the callback URL of the transaction. This callback URL is the site where the user will be redirected after completing the checkout process. To set the callback URL please see **SetTransactionCallbackURL**.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sSeqidTransaction – This is ScholarChip's unique identifier for the transaction you are trying to query. You must call the **DoesTransactionExist** function with your unique identifier to retrieve this value.

3. Return Values:

This function returns the callback URL of the transaction. In the event of an error, this function will return a human-readable error message containing the word "ERROR".



GetTransactionStatus

1. Purpose:

This function returns the status of a previously created transaction.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sSeqidTransaction – This is ScholarChip's unique identifier for the transaction you are trying to query. You must call the **DoesTransactionExist** function with your unique identifier to retrieve this value.

3. Return Values:

This function will return the status of the transaction. Please see the list of status codes below for a full list of return values. In the event of an error, this function will return a human-readable error message containing the word "ERROR".

Transaction Status Codes:

1. *Expired* – The transaction was created, however, the user failed to complete the checkout process within 30 minutes of creation.
2. *Failed* – The user has completed the checkout process, however an error occurred during payment submission. No funds were withdrawn from the user's account(s).
3. *Initialized* – The transaction has been completed, however, the user has not yet completed the checkout process.
4. *Pending* – The user has completed the checkout process, however, we are waiting on one, or all of their payment methods to clear. Usually this will happen if a user pays with an Electronic Check (ACH).
5. *Processed* – The user has completed the checkout process and all of their payment methods have cleared.



GetTransactionURL

1. Purpose:

This function is used to get the checkout URL for a transaction. You will need to direct your users to this URL to complete the checkout process.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sSeqidTransaction – This is ScholarChip's unique identifier for the transaction you are trying to query. You must call the **DoesTransactionExist** function with your unique identifier to retrieve this value.

3. Return Values:

This function returns the checkout URL. In the event of an error, this function will return a human-readable error message containing the word "ERROR".



IsValidToken

1. Purpose:

This function is used to check the validity of a token. A valid token is one that has been created or used within the last ten minutes. We strongly recommend that you check all tokens before making any WebService calls.

2. Input Parameters:

sToken – The token you are trying to validate. A valid token is one that was either created or used within the last ten minutes. After ten minutes of inactivity, your token will expire and you will need to call **GetToken** to retrieve another token.

3. Return Values:

This function will return “1” if the token is valid. In the event of an error, or if the token is not valid, this function will return a human-readable error message containing the word “ERROR”.



MarkOrderProcessed

1. Purpose:

This function is used to manually mark an order as processed after you have completed a manual order step (shipping). The order will need to be in a pending status in order to call this function.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sSeqidOrder – This is ScholarChip's unique identifier for the order. You must call the **DoesOrderExist** function with your unique identifier to retrieve this value.

3. Return Values:

This function will return "1" on successful execution. In the event of an error, this function will return a human-readable error message containing the word "ERROR".



SetOrderCallbackURL

1. Purpose:

This function is used to set the callback URL which the user will be redirected to after completing the checkout process.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sSeqidOrder – This is ScholarChip's unique identifier for the order. You must call the **DoesOrderExist** function with your unique identifier to retrieve this value.

sCallbackURL – This is the URL you would like the user to be redirected to after they complete the order checkout process.

3. Return Values:

This function will return “1” on successful execution. In the event of an error, this function will return a human-readable error message containing the word “ERROR”.



SetTransactionCallbackURL

1. Purpose:

This function is used to set the callback URL which the user will be redirected to after completing the checkout process.

2. Input Parameters:

sToken – This is the value you have previously retrieved from **GetToken**. To verify your token, you can call the **IsValidToken** function.

sSeqidTransaction – This is ScholarChip's unique identifier for the transaction. You must call the **DoesTransactionExist** function with your unique identifier to retrieve this value.

sCallbackURL – This is the URL you would like the user to be redirected to after they complete the order checkout process.

3. Return Values:

This function will return "1" on successful execution. In the event of an error, this function will return a human-readable error message containing the word "ERROR".