

---

**UTAH GOVPAY:  
THE OFFICIAL PAYMENT SOLUTION FOR UTAH GOVERNMENT**

---



**Technical Manual**



**Utah**Interactive

*"Extend the power of your website...by taking payments online"*

This Utah GovPay Technical Manual contains the following information:

<b>Utah GovPay:</b>	<b>1</b>
<b><i>The Official Payment Solution for Utah government</i></b>	<b>1</b>
WPS Web Service Overview:	3
Terminology	4
Features	4
Operations	5
Complex XML Types	5
Sample Web Service Java Code:	9

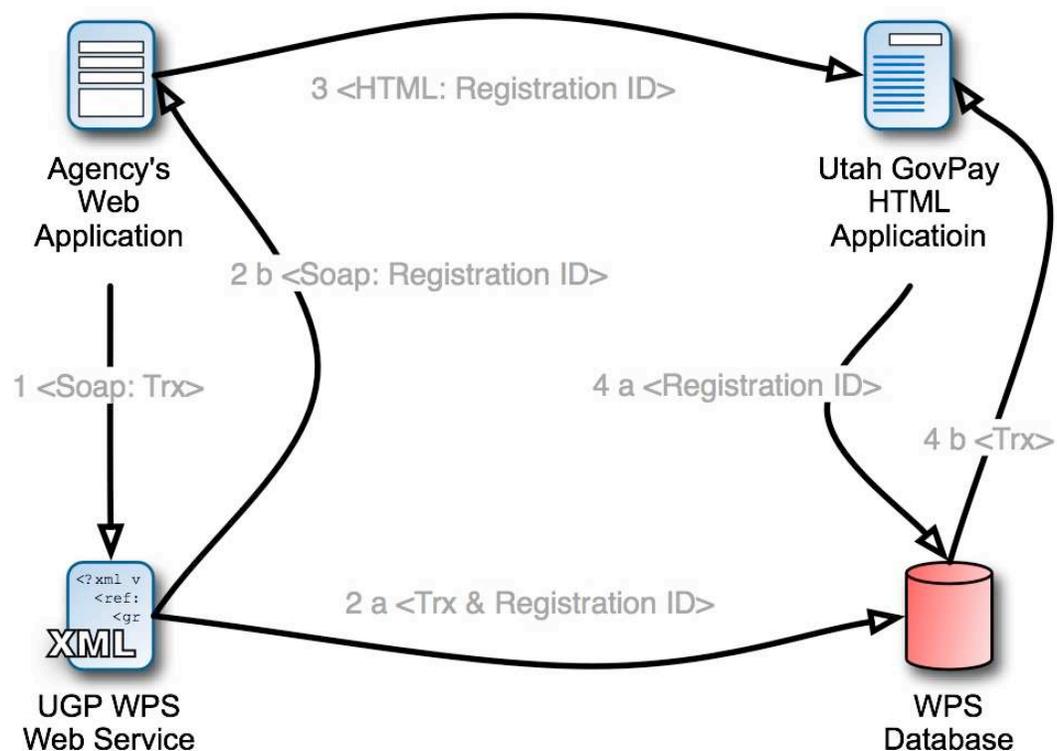
## WPS WEB SERVICE OVERVIEW:

The Utah GovPay WPS web service provides a secure method to pass customer transaction information between the Agency's web application and Utah GovPay.

The WPS is the backend link into the Utah GovPay system and was designed to prevent web users from fraudulently altering their own transaction data. WPS has two main functions, registering transactions and querying transactions. In the registration process, the agency's web application sends the transaction data to WPS and WPS returns a registration ID. The Agency's web application then forwards the user to Utah GovPay with the registration ID.

After a completed payment transaction, the Agency's web application can use the registration ID to query WPS to find out if the transaction was approved.

## Utah GovPay Register Transaction



The Register Transaction process follows the following steps:

1. When the user is ready to make a payment, the agency's web application sends details of the payment in XML using the Soap format to the Utah GovPay Web Service or WPS.
2. The Utah GovPay WPS creates a Registration ID, stores the transaction information and registration ID in a database and returns a registration ID back to the agency's web application.

3. The agency's web application redirects the user to the Utah GovPay URL and includes the registration ID in the query string. This Utah GovPay URL will be created during the Utah GovPay setup.
4. The Utah GovPay website uses the registration ID to retrieve the transaction data and then takes the user through the payment process.

## Utah GovPay

### Query Payment Processing Result



The Query Transaction process:

1. The agency's web application sends a soap message with the registration ID
2. The Utah GovPay Web Service returns the results of the transaction. The details of the soap message are listed below in the Complex XML types under **PaymentprocessingResult**.

### TERMINOLOGY

- \* Registration Id - The unique identifier used generated by WPS. The calling application should pass this to WPS when the user is handed off.
- \* Calling Application – This is the outside application built by the agency.
- \* Web Service – A piece of software that can be accessed over the Internet by another application using XML to send or retrieve information.
- \* Web Application – A web application uses a web site as a front end to interact with users across the Internet.

### FEATURES

Current features of the WPS web service are:

- \* Register Transaction Details
- \* Retrieve Payment Processing Result in XML for a single transaction
- \* Retrieve Payment Processing Result in text for a single transaction

## OPERATIONS

The following operations are available in the WPS web service.

**Register a Transaction** – Information about the transaction is sent from the Calling Application to WPS and a registration ID is sent back the Calling Application. The name of this operation is “register”

**Query a Transaction** – The registration ID or a group of registration ID’s are sent from the Calling Application to WPS and the results of the transaction are returned to the Calling Application. There are two operations that can be used to retrieve information about the transaction after it has been processed.

\* getTransactionsByRegistrationID returns all the transaction information in an XML format.

\* getStatusByRegistrationID returns a string with one of four possible messages.

WSDL URLS PRODUCTION:

<https://secure.utah.gov/service-wps-v2-0/services/registerTransaction?wsdl>

<https://secure.utah.gov/service-wps-v2-0/services/queryTransactions?wsdl>

WSDL URLS TEST:

<https://test.secure.utah.gov/service-wps-v2-0/services/registerTransaction?wsdl>

<https://test.secure.utah.gov/service-wps-v2-0/services/queryTransactions?wsdl>

<i>Operation Name</i>	<i>Input</i>	<i>Output</i>	<i>Faults</i>
register	* WpsAccount * TransactionDetail	String – RegistrationId	Generic SOAP fault should an error occur
getTransactionsByRegistrationId	* WpsAccount * RegistrationID	PaymentProcessingResult	Generic SOAP fault should an error occur
getStatusByRegistrationId	* WpsAccount * RegistrationID	String – (not-found, error, successful, declined)	Generic SOAP fault should an error occur

## COMPLEX XML TYPES

<i>Field Name</i>	<i>Type</i>	<i>Length</i>	<i>Restrictio n</i>	<i>Description</i>
<b>WPSACCOUNT</b>				
username	String	128	Required	Assigned by Utah Interactive
password	String	128	Required	Assigned by Utah Interactive
id	Long		Null	Used by Utah Interactive- leave null
<b>TRANSACTIONDETAIL</b>				
allowedPaymentTypes	String[]	128	Optional	An array of values indicating the types of payment a user can make. Possible values: CREDITCARD,

				ECHECK
failUrl	String		Optional	The URL to send the user to upon failed payment. Can be set up as default. See account setup form
id	Long		Null	Used by Utah Interactive- leave null
items	ITEM[]			An array of Items, each describing what the end user is paying for. (see ITEM below)
paymentProcessingResultID	String		Null	Used by Utah Interactive- leave null
prePopAddressLine1	String	128	Optional	Value to use to pre-populate the credit card address line 1 field
prePopAddressLine2	String	128	Optional	Value to use to pre-populate the credit card address line 2 field
prePopCity	String	128	Optional	Value to use to pre-populate the credit card city field
prePopEmailAddress	String	128	Optional	Value to use to pre-populate the credit card email address field
prePopName	String	128	Optional	Value to use to pre-populate the credit card name field
prePopPostalCode	String	128	Optional	Value to use to pre-populate the credit card postal/zip code field
prePopStateProvince	String	128	Optional	Value to use to pre-populate the credit card state/province field
registrationId	String	128	Null	Created by Utah Interactive- leave null during registration. This ID will be returned during the Query Payment Operation
sharedSecretName	String	128	Required for eChecks	The name of the shared secret to display to the user. eChecks require users to confirm some information.
sharedSecretValue	String	128	Required for eChecks	A value that the user should know that is used to authenticate them when making an eCheck payment
successMessage	String	128	Optional	A message to be displayed upon successful payment. Can be set up as default. See account setup form
successUrl	String		Optional	The URL to send the user to upon successful payment. Can be set up as default. See account setup form
wpsAccountId	Long		Null	Used by Utah Interactive- leave null
<b>ITEM</b>				
amountEach	Long		Required	The amount of each item (See Note 1 below)
customerId	String	128	Required	A value that uniquely identifies the customer in the calling application. Examples include license or account numbers.

customFields	Custom Field[]		Optional	An array of custom fields that are passed into the Utah GovPay admin. (see CUSTOMFIELD below)
description	String	255	Required	A description of the item
id	Long		Null	Used by Utah Interactive- leave null
quantity	Long		Required	The quantity of this item
transactionDetailId	Long		Null	Used by Utah Interactive- leave null
transactionId	String	128	Required – Unique	Unique identifier for the transaction in the calling application
transactionType	String	128	Optional	A code identifying the type of transaction this is. If applicable the FINET code should be put here.
<b>CUSTOMFIELD</b>				
id	Long			Used by Utah Interactive- leave null
itemId	Long			Used by Utah Interactive- leave null
name	String	64	Required	The name of the custom field
value	String	128	Required	The value of the custom field
<b>PAYMENTPROCESSINGRESULT</b>				
addressLine1	String	128		Address used for payment
addressLine2	String	128		Address used for payment
authorizationCode	String	10		Code provided by payment processor
auxiliaryMessage	String	255		
city	String	128		City used for payment
completionDate	Timestamp			Date transaction completed
country	String	128		Country used for payment
error	String	10		Boolean (True or False). The false result could be caused by either a success or a declined transaction. (See Note 2 below)
gatewayTransactionId	String	128		
id	Long			Used by Utah Interactive- leave null
name	String	128		Name used for payment
orderId	String	16		Unique order ID created by Utah interactive.
paymentSuccessful	String	10		Boolean (True or False). The false result could be caused by either a technical error or a declined transaction (See Note 2 below)
postalCode	String	10		Zip used for payment
stateProvince	String	2		State used for payment
statusMessage	String	255		Message provided by payment processor
transactionDetail	transactionDetail[]			All the transaction information that was originally sent to WPS during the

			register operation. (See TRANSACTIONDETAIL above)
--	--	--	---

Note 1: The total payment amount for the transaction is calculated by multiplying each ITEM quantity by the ITEM's amountEach.

Note 2: There are three possible outcomes for Is\_Success and Is\_Error in the PaymentProcessingResult:

- If paymentSuccessful = True then the transaction was SUCCESSFUL
- If paymentSuccessful=False & Error=False then the transaction was DECLINED
- If Error=True then the transaction had a technical ERROR

## SAMPLE WEB SERVICE JAVA CODE:

```
import gov.utah.secure.wps.model.TransactionDetail;
import gov.utah.secure.wps.model.WpsAccount;
import gov.utah.secure.wps.model.Item;
import gov.utah.secure.wps.model.CustomField;
import gov.utah.secure.wps.service.RegisterTransaction;
import gov.utah.secure.wps.service.RegisterTransactionLocator;
import gov.utah.secure.wps.service.RegisterTransactionWebService;

import java.net.URL;
import java.net.MalformedURLException;
import java.rmi.RemoteException;

public class GovPayRegister
{
    private static final String wpsAccountUsername = "";
    private static final String wpsAccountPassword = "";

    public static void main(String[] args)
    {
        /* URL for Utah Interactive's GovPay Registration Web Service */
        String url = "https://test.secure.utah.gov/service-wps-v2-0/services/registerTransaction?wsdl";
        URL registrationWebServiceUrl;
        try
        {
            registrationWebServiceUrl = new URL(url);
        }
        catch (MalformedURLException e)
        {
            String s = "Unable to recognize govpay web service url " + url;
            System.err.println(s);
            e.printStackTrace();
            return;
        }
        System.out.println("Registering request");

        WpsAccount wpsAcct = constructWpsAccount();
        TransactionDetail trxDetail = constructTransactionDetail();

        RegisterTransaction service = new RegisterTransactionLocator();
        RegisterTransactionWebService webService;
        try
        {
            webService = service.getregisterTransaction(registrationWebServiceUrl);
        }
        catch (Exception e)
        {
            String s = "Unable to connect to registration web service at " + registrationWebServiceUrl;
```

```

        System.err.println(s);
        e.printStackTrace();
        return;
    }
    String registrationId;
    try
    {
        registrationId = webService.register(wpsAcct, trxDetail);
    }
    catch (RemoteException e)
    {
        String s = "Unable to register request";
        System.err.println(s + "\nWpsAccount: " + wpsAcct + "\nTransactionDetail: " + trxDetail +
"\n");
        e.printStackTrace();
        return;
    }
    System.out.println("Registration Id: " + registrationId);
}

```

```

private static WpsAccount constructWpsAccount()
{
    WpsAccount wpsAcct = new WpsAccount();

    wpsAcct.setUsername(wpsAccountUsername);
    wpsAcct.setPassword(wpsAccountPassword);

    return wpsAcct;
}

```

```

private static TransactionDetail constructTransactionDetail()
{
    CustomField customField1 = new CustomField();
    customField1.setName("First Field Name");
    customField1.setValue("First Field Value");

    CustomField customField2 = new CustomField();
    customField2.setName("Second Field Name");
    customField2.setValue("Second Field Value");

    CustomField customField3 = new CustomField();
    customField3.setName("Third Field Name");
    customField3.setValue("Third Field Value");

    Item item = new Item();

    item.setAmountEach(12.00);
    item.setCustomerId("12345");
    item.setDescription("My Cool Item");
}

```

```
item.setQuantity(1);
String transactionId = String.valueOf(System.currentTimeMillis());
item.setTransactionId(transactionId);
item.setTransactionType("Finet-001");
item.setCustomFields(new CustomField[] {customField1, customField2, customField3});

TransactionDetail trxDetail = new TransactionDetail();

trxDetail.setAllowedPaymentTypes(new String[] {"CREDITCARD", "ECHECK"});
trxDetail.setFailUrl("");
trxDetail.setItems(new Item[] {item});
trxDetail.setPrePopAddressLine1("123 Fake Street");
trxDetail.setPrePopCity("Springfield");
trxDetail.setPrePopEmailAddress("KnifeyWifey@thesimpsons.tv");
trxDetail.setPrePopName("Knifey Wifey");
trxDetail.setPrePopPostalCode("84000-1234");
trxDetail.setPrePopStateProvince("UT");
trxDetail.setSharedSecretName("");
trxDetail.setSharedSecretValue("");
trxDetail.setSuccessMessage("Thank you for purchasing 'My Item'");
trxDetail.setSuccessUrl("https://example.com/coolapp/?trxId=" + transactionId);

return trxDetail;
}
}
```