DocuCents.com

THE CENTSIBLE WAY TO SEND DOCUMENTS™

Vendor API Documentation

December 7, 2014

API Version 1.0

DocuCents.com
THE CENTSIBLE WAY TO SEND DOCUMENTS™

Table of Contents

DocuCents.com
THE CENTSIBLE WAY TO SEND DOCUMENTS™

Introduction:

Welcome to the Docucents vendor program. As a vendor with Docucents you will be responsible in developing the communication protocols with the Docucents system. Docucents uses XML-RPC web application programmatic internfaces (API). These API's will allow you to interact with the Docucents system to perform functions such as creating a submission and thereafter checking on the status and returning submission specific documentation. Our API's allow developers using different frameworks to connect to and interact with the Docucents system regardless of codebase. This means that if you code in PHP, .net, MS Visual Fox or another software language you will be able to utilize our services.

Getting Started:

Definitions:

1. **Vendor :** **The party who is providing the service for their customers**
2. **Customer/Account:** **The party who the vendor is submitting on behalf of. The account is the one that initiates the submittal with the vendor through an interface, such as a desktop app or a web graphical user interface**
3. **Payer:** **The party who is responsible for paying for the activity that is initiated by the customer/account.**

API Keys

Vendor Key

To begin you first must establish a Vendor account and be issued a vendor API Key. With this API key you will be able to connect to the URL and establish an XML-RPC Client. This key is unique to you and serves as a way to authenticate you to the system. Please make sure you safe guard it as you would with a username and password.

Account Key

Next you will need to register your customers with the systems. This registration lets docucents know who you will be creating submissions on behalf of. The unique combination of the Vendor (you), your customer and the party responsible for invoices (often the same entity as your customer) establishes an account. Each account will be issued a unique key.

Vendor: Account Key

Putting the vendor Key and the account Key together establishes a secure way to ensure that only you will be creating submissions for your customers. Your customer wont know your vendor key and the account key is unique to your relationship with the customer. Therefore if your customer provides your unique account key to anyone other than you the key will not validate and be rendered useless.

DocuCents.com
THE CENTSIBLE WAY TO SEND DOCUMENTS™

The Vendor:Account (API Key) is what is used to authenticate on every submission to the API. This composite key must be used for each submission and is not a static variable.

Example:

"Test" vendor Key: 37b5414502e1f1d7

Account "customer 1" : 9747843dc76658c475759f18653ec6ba

API Key: 37b5414502e1f1d7:9747843dc76658c475759f18653ec6ba

As seen in the examples each set of related customers, payers, and vendors are issued a unique key. For each submission the unique key will captured and passed to the URL as a parameter.

Failure to properly call the URL with the correct api_key will result in submission billing mistakes.

Creating a Submission:

Q: What is involved in making a submission?

A: Vendors will make several API calls that would encompass a complete submittal. The basic call set will compose of the following

1. **A Connection to the API for a particular vendor, customer, payer relationship. The URL that contains the API Key (**https://api.docucents.com/xmlrpc/?api_key=6bee4f82bcc1c060:47f3ba49a94d1f15ba55c58dd54ca323**)**

    ***NOTE FOR TESTING PLEASE USE: https://staging.api.docucents.com/xmlrpc/**
2. **The creation of a new submittal (Submittals.NewRecord)**
3. **The addition of parties to that submittal (PartyData.AddParty)**

4. **The addition of attachments (Submittals.AddAttachment)**
5. **Once complete marking the status submitted (Submittals.UpdateStatus)**

DocuCents.com THE CENTSIBLE WAY TO SEND DOCUMENTS™

API Calls:

For a complete listing of API calls please visit https://docs.docucents.com . This page is dynamic and automatically updated as the API calls are added, modified or removed.

Basic Calls:

To make multiple API calls in a single call

system.multicall

Multicall - boxcar feature of XML-RPC for calling multiple methods

in a single request.

Expects a an array of structs representing method calls, each element

having the keys:

- methodName

- *params*

Returns an array of responses, one for each method called, with the value returned by the method. If an error occurs for a given method, returns a struct with a fault response.

To check if a key is valid

dwcd.ValidateApiKey

Validate Api Key.

Parameters:

string api_key

Returns:

boolean returns true if the combination is in the table and membership is not disabled

To get an estimated cost for a submittal

dwcd.PrintCostEstimate

Estimate cost of printing.

; parameter

integer total number of parties to be delivered

integer total number of pages

boolean true if duplex, false if simplex

Returns:

float estimated cost of printing.

To add a party to a submittal

PartyData.AddParty

Insert party info into DB.

If connect_to parameter is blank then id_number, user_name, and party_role are ignored.

Parameters:

string connect_to /^(|Case|Submittal)$/

string id_number Case Number or Vendor Submittal ID

string user_name NOT USED. Just leave as empty

string party_role ^(|ClaimAdmAtty|ClaimsAdmin|EmployAttny|Employer|Insurance|LienClaimant|LienClaimAtt|Worker|WorkerAttny|ER|CR|LX|LA|IW|AP)$/

struct party see below **

string usage_restrict -OPTIONAL /^(|Delivery|Form|,)*$/, one or both, empty implies both 'Delivery,Form', Used when connecting to Submittal

date state_update 'Y-m-d' (man strftime), or empty string for current time -OPTIONAL

integer role_index if there are more than 1 of the same role on a submittle this can set the order - OPTIONAL

Returns:

integer party_address_id (or worker_address_id), throws error

If party is an injured worker (Worker) then the structure needs to be in this form:

struct worker {

string dwc_id /^[0-9A-Z]{0,10}$/ ERN, old party ID, or other unique identifyer, or empty

string surname 1-20 char max

string given_name 0-30 char max

string middle_initial 0 or 1 char max

string suffix 0-3 char max

```
   string  address1       /^.{0,50}$/
   string  address2       /^.{0,50}$/, suite etc. allowed
   string  city          /^[a-zA-Z -]{0,35}$/   (city && state) || zip
   string  state         /^[A-Z]{0,2}$/        (city && state) || zip
   string  zip           /^[0-9A-Z -]{0,9}$/, 5 or 9 digits, or Canadian
   string  phone         /^[2-9][0-9]{9}$/, 8054840333, anything other than digits will be removed
   string  gender        /^(|[MF])$/
   date    date_of_birth
   string  ssn           /^[0-9]{0,9}$/
}
```

If party is any other party type then the structure needs to be in this form:

```
struct party {
   string  dwc_id        /^[0-9A-Z]{0,10}$/ ERN, old party ID, or other unique identifyer, or empty
   string  name1         /^(.{2,56}|[a-zA-Z -]{1,20})$/, company name
   string  name2         /^(.{0,40}|[a-zA-Z -]{1,30})$/, attn, c/o, etc.; if last name is separate from first &
MI then put last name here
   string  given_name     /^(.{0,25}|[a-zA-Z -]{1,30})$/, first name
   string  middle_initial  0 or 1 char max
   string  address1       /^.{0,50}$/
   string  address2       /^.{0,50}$/, suite etc. allowed
   string  city          /^[a-zA-Z -]{0,35}$/   (city && state) || zip
   string  state         /^[A-Z]{0,2}$/        (city && state) || zip
   string  zip           /^[0-9A-Z -]{0,9}$/, 5 or 9 digits, or Canadian
   string  phone         /^[2-9][0-9]{9}$/, 8054840333, anything other than digits will be removed
   string  delivery_pref   /^[UEF]$/, USMail, Email, Fax -OPTIONAL defaults to "U"
   string  role_sub_type   one character code used on submittals, ie.
Application___EmployerRoleTypeCode
}
```

Members that are omitted will be treated as an empty string.

"party_role" may be our 12-letter Dwcd_role_code, the EDEX3 two-letter code, or EAMS ten-character code.

A "Submittal" must include a pre-defined vendor_submittal_id in the id_number parameter, and the user_name that it belongs.

A "Case" must include a pre-defined case number in the id_number parameter. The user_name is optional for case ownership.

A party can be added to the DB without reference to a submittal or case. Just set those parameters to empty strings.

To create a new submittal for delivery only (no E-filing or jet-filing)

submittals.AddForDelivery

Create a new submittal for delivery

Parameters:

```
struct    parameters {

string|int case_number any type case number, empty or zero

string                  billing_code            -OPTIONAL

string                  file_number                     -OPTIONAL

string                  charge                          -OPTIONAL  /^(Standard|None|Special)$/,
defaults to Standard

string    comment       -OPTIONAL  any string up to 128 characters

string    pos_wording   -OPTIONAL  any string up to 128 characters}
```

Returns:

```
        string vendor_submittal_id
```

DocuCents.com
THE CENTSIBLE WAY TO SEND DOCUMENTS™

Sample Code (.net):

```
public class Submit {

string _vendor_submittal;
string _submittalStatus;

       publi void Submit(bool duplex, IEnumerable<AttachmentFile> attachments, 
IEnumerable<PartyAddress> parties, string billingCode, string fileNumber, string pos, 
string author)
        {

                ////add new submittal
                ////save the vendor id returned here, you may need it for subsequent 
calls
                //$this->api->NewRecord();
                New(ct, billingCode, fileNumber, pos))

                ////add new party
                //$this->api->AddParty();
                AddParties(ct, parties))

                ////Submittals.AddAttachment
                //$this->api->AddAttachment();
                AddAttacments(ct, attachments, author, duplex))

                ////set status
                //$this->api->SetSubmittalStatus();
                SetStatus(ct, SUBMIT_COMPLETE))

                this.txtConfirmationNum.Text = _vendor_submittal_id;
        }



        //$this->api->NewRecord();
        private void New(string billingCode, string fileNumber, string pos)
        {
            string case_number = "";
            string billing_code = billingCode;
            string file_number = fileNumber;
            string charge = "Standard";
            string comment = "";
            string pos_wording = pos;

            _vendor_submittal_id = Api.AddForDelivery(case_number, billing_code, 
file_number, charge, comment, pos_wording);

        }


        //$this->api->AddParty();
        private void AddParties(IEnumerable<PartyAddress> parties)
        {
            foreach (PartyAddress pa in parties)
            {
                Party p = new Party("", pa.Name1, pa.Name2, "", "", pa.Address1,
pa.Address2, pa.City, pa.State, pa.Zip, "", "", "");
                string status = Api.AddParty("Submittal", _vendor_submittal_id, "LX", p);
            }

        }

        //$this->api->AddAttachment();
        private void AddAttacments(IEnumerable<AttachmentFile> attachments, string
author, bool duplex)
        {
            foreach (AttachmentFile a in attachments)
            {
                Attachment att = a.GetAttacment("", "", "", DateTime.Today, author);
                int status = Api.AddAttachment(_vendor_submittal_id, att, duplex);
            }

        }


        //$this->api->SetSubmittalStatus();
        private void SetStatus(string status)
        {
            _submittalStatus = Api.SetSubmittalStatus(_vendor_submittal_id, "Submitted");
        }
}
```

```
    public class Party
    {
        public Party(string dwc_id, string name1, string name2, string given_name, string
middle_initial, string address1, string address2, string city, string state, string zip,
string phone, string delivery_pref, string role_sub_type)
        {

            //If party is any other party type then the structure needs to be in this
form:
            //struct party {
            //string dwc_id /^[0-9A-Z]{0,10}$/ ERN, old party ID, or other unique
identifyer, or empty
            //string name1 /^(.{2,56}|[a-zA-Z -]{1,20})$/, company name
            //string name2 /^(.{0,40}|[a-zA-Z -]{1,30})$/, attn, c/o, etc.; if last name
is separate from first & MI then put last name here
            //string given_name /^(.{0,25}|[a-zA-Z -]{1,30})$/, first name
            //string middle_initial 0 or 1 char max
            //string address1 /^.{0,50}$/
            //string address2 /^.{0,50}$/, suite etc. allowed
            //string city /^[a-zA-Z -]{0,35}$/ (city && state) || zip
            //string state /^[A-Z]{0,2}$/ (city && state) || zip
            //string zip /^[0-9A-Z -]{0,9}$/, 5 or 9 digits, or Canadian
            //string phone /^[2-9][0-9]{9}$/, 8054840333, anything other than digits will
be removed
            //string delivery_pref /^[UEF]$/, USMail, Email, Fax -OPTIONAL defaults to
"U"
            //string role_sub_type one character code used on submittals, ie.
Application___EmployerRoleTypeCode
            //}



            this.dwc_id = dwc_id;
            this.name1 = name1;
            this.name2 = name2;
            this.given_name = given_name;
            this.middle_initial = middle_initial;
            this.address1 = address1;
            this.address2 = address2;
            this.city = city;
            this.state = state;
            this.zip = zip;
            this.phone = phone;
            this.delivery_pref = delivery_pref;
            this.role_sub_type = role_sub_type;

        }

        public string dwc_id;
        public string name1;
        public string name2;
        public string given_name;
        public string middle_initial;
        public string address1;
        public string address2;
        public string city;
        public string state;
        public string zip;
        public string phone;
        public string delivery_pref;
        public string role_sub_type;

    }
```

```
    public class AttachmentFile

    {
        private const long MB64 = 65 * 1024 * 1024; //64MB is max size

        public AttachmentFile(string filename, bool duplex)
        {
            FilePath = filename;
            if (!Exists)
            {
                return;
            }

            IsValidFileType = false;
            IsTooLarge = false;



            byte[] bBuf = new byte[5];
            using (FileStream strm = new FileStream(filename, FileMode.Open,
FileAccess.Read))
            {
                strm.Read(bBuf, 0, 4);

                if ((strm.Length * 1.37) < MB64)
                {
                    IsTooLarge = false;
                }
                else
                {
                    IsTooLarge = false;
                }

            }


            string sBuf = Encoding.Default.GetString(bBuf);
            if (sBuf.StartsWith("%PDF"))
            {
                IsValidFileType = true;
            }

            if (sBuf.StartsWith("II"))
            {
                IsValidFileType = false;
            }



            Duplex = duplex;
        }


        public bool Duplex { get; private set; }

        public string FilePath { get; private set; }
        public string FileNameWithoutDirectory { get { return Path.GetFileName(FilePath);
} }
        private Tuple<bool, FileInfo> Info
        {
            get
            {
                FileInfo fi;
                try
                {
                    fi = new FileInfo(FilePath);
                    return new Tuple<bool,FileInfo>(true, fi);
                }
                catch (Exception)
                {
                    return new Tuple<bool, FileInfo>(false, null);
                }

            }
        }

        public bool IsValid { get { Tuple<bool, FileInfo> filo = Info; return filo.Item1;
} }
        public bool Exists { get { Tuple<bool, FileInfo> filo = Info; return filo.Item1
&& filo.Item2.Exists; } }
        public long Length { get { Tuple<bool, FileInfo> filo = Info; if (filo.Item1 &&
filo.Item2.Exists) return filo.Item2.Length; else return -1; } }


        public bool IsValidFileType { get; private set; }
        public bool IsTooLarge { get; private set; }


        public string ToBase64String()
        {
            using (FileStream fs = new FileStream(FilePath, FileMode.Open,
FileAccess.Read))
            {

                byte[] filebytes = new byte[fs.Length];
                fs.Read(filebytes, 0, Convert.ToInt32(fs.Length));
                string encodedData = Convert.ToBase64String(filebytes);
                if (encodedData.Length > MB64) throw new ApplicationException("file too
large");

                return encodedData;

            }

        }

        public Attachment GetAttacment(string type, string title, string unit, DateTime
date, string author)
        {
            Tuple<bool, FileInfo> filo = Info;
            Attachment a = new Attachment(filo.Item2.Name, type, title, unit, date,
author, ToBase64String());
```

```
            return a;
        }
    }
```

Sample Code 2:

```
Public Interface ITest

    Inherits IXmlRpcProxy


    <XmlRpcMethod("dwcd.ValidateApiKey")> _

    Function CheckApiKey(ByVal ApiKey As String) As Boolean


    <XmlRpcMethod("dwcd.PrintCostEstimate")> _

    Function GetAmount(ByVal iParty As Integer, ByVal iPages As Integer, ByVal bSimplex
As Boolean)


End Interface




    Private Sub Button2_Click(sender As System.Object, e As System.EventArgs) Handles
Button2.Click

        TestProxy = CType(XmlRpcProxyGen.Create(GetType(ITest)), ITest)

        ClientProtocolTest = CType(TestProxy, XmlRpcClientProtocol)

        TxtUrlTest = "https://api.docucents.com/xmlrpc/?"

        ClientProtocolTest.Url = TxtUrlTest

        Cursor = Cursors.WaitCursor


        Try

            ServicePointManager.ServerCertificateValidationCallback = Function() True

            Dim result As Boolean =
TestProxy.CheckApiKey("a383953352a298d7:51a53d5c98ffc0ddfb246b18622b4845")


            'Dim iAmt As Double = TestProxy.GetAmount(3, 4, False)
            'Me.Lbl_Result.Text = result.ToString

            Dim sResult As String = "".ToString

        Catch ex As Exception

            MsgBox(ex.Message.ToString)

        End Try

        Cursor = Cursors.Default

    End Sub
```

Sample Code PHP:

```
function AddForDelivery() {

    $charge = array("Standard","None","Special");
    $params = new Struct(array(
      'case_number' => preg_replace("/[^\d]+/", "", uniqid()),
      'billing_code' => "code-".uniqid(),
      'file_number' => "file-".uniqid(),
      'charge' => $this->pickRand($charge),
      'comment' => "comment ".uniqid(),
      'pos_wording' => "These are the documents I served on ".date("M D d Y", time())." at ".date("H:i:s.u a",
time()).".",
    ));

    $result = $this->client->Submittals->AddForDelivery($params);


    //set id for other calls
    $this->vendorSubmittalID = $result;


    //assert
    $assert = "danger";
    if (is_string($result) && $result != "") $assert = "success";


    return array(
      "params" => $params,
      "result" => $result,
      "assert" => $assert,
    );

}
```

```
function AddPartyForDelivery() {

    $params = new Struct(array(
      'id_number'=>"6E99E1A8DF902D78",   //string   id_number      Case Number or Vendor Submittal ID
      'company_name'=>"yamaha racing",   //string   company_name   1-30 char max
      'last_name'=>"josh",          //string   last_name      1-20 char max
      'first_name'=>"hayes",         //string   first_name     0-30 char max
      'middle_initial'=>"j",         //string   middle_initial   0 or 1 char max
      'address1'=>"1384 N Moorpark Rd",   //string   address1      /^.{0,50}$/
      'address2'=>"",               //string   address2      /^.{0,50}$/, suite etc. allowed
      'city'=>"Thousand Oaks",        //string   city         /^[a-zA-Z -]{0,35}$/  (city && state) || zip
      'state'=>"ca",                //string   state        /^[A-Z]{0,2}$/      (city && state) || zip
      'zip'=>"91360",               //string   zip          /^[0-9A-Z -]{0,9}$/, 5 or 9 digits, or Canadian
      'phone'=>"8182922123",          //string   phone        /^[2-9][0-9]{9}$/, 8054840333, anything
other than digits will be removed
    ));

    $result = $this->api->call("PartyData.AddForDelivery", $params);


    $assert = "danger";
    if (is_string($result) || is_int($result)) $assert = "success";


    return array(
      "params" => $params,
      "result" => $result,
      "assert" => $assert,
    );
    }

function AddAttachment() {

    $file_name = "FILE_".rand(1,5).".pdf";
    $file_path = "/srv/data/uploads/pdfs/".$file_name;
    $file = new Zend\XmlRpc\Value\Base64(file_get_contents($file_path));

    $attachment = array(
      'file_name' => $file_name,
      'type' => "title-".uniqid(),
      'title' => "type-".uniqid(),
      'unit' => "ADJ",
      'date' => date("Y-m-d", time()), //"2014-09-23",
      'author' => "author-".uniqid(),
      'base64' => $file,
    );

    $result = $this->client->Submittals->AddAttachment(
      $this->vendorSubmittalID, //$this->vendorSubmittalID, //vendor_submittal_id     globally unique ID
number
      $attachment, //structure   attachment
      (bool)rand(0, 1) //boolean     duplex      -OPTIONAL false
    );


    return array(
      "params" => array(),
      "result" => $result,
      "assert" => "success",
    );
}

function SetSubmittalStatus() {

    /**
     * Submittals.SetSubmittalStatus
     * TODO: Get the rest of actions here
     */
    $action = array(
      'DeletedRecord',
      'Submitted',
      'Recalled',
      'RetrievedForOFile',
      'SubmitalRetrieved',
      'RetrievedForPrintDeliver',
      'RetrievedForEFile',
      'RetrievedForJetFile',
    );

    $r = $this->client->Submittals->SetSubmittalStatus(
      $this->vendorSubmittalID, //string|integer  vendor_submittal_id     globally unique ID number
      'Submitted', //$this->pickRand($action), //string         action              This is one of the "action_name"
values from "status codes" on Google Docs.
      array() //array|object    baf                  -OPTIONAL flattened BAF contents
    );


    //integer        submittal_status_id
    return $r;
}
```