

## **Ohio Department of Public Safety**

### **Technical Specifications for Scrap Dealer Service API**

## TABLE OF CONTENTS

<b>1</b>	<b>Revision History</b> .....	<b>1</b>
<b>2</b>	<b>Overview</b> .....	<b>2</b>
	Getting Started .....	2
	HTTP Response Codes.....	2
	Error Format.....	3
<b>3</b>	<b>API Method Overview</b> .....	<b>4</b>
<b>4</b>	<b>API Method Details</b> .....	<b>5</b>
	POST v1/DoNotBuy/Search .....	5
	URI Parameters.....	5
	Request Information .....	5
	Response Information.....	5
	POST v1/Transaction/Validate.....	7
	URI Parameters.....	7
	Request Information .....	7
	Response Information.....	9
	POST v1/Transaction/Submit .....	9
	URI Parameters.....	9
	Request Information .....	9
	Response Information.....	11
<b>5</b>	<b>How To Consume?</b> .....	<b>12</b>
	C# Classes: .....	12
	Sample C# Code To Call The Service API (v1/DoNotBuy/Search).....	14
	Sample C# Code To Call The Service API (v1/Transaction/Validate).....	15
<b>6</b>	<b>Field Specifications</b> .....	<b>16</b>
	Driver License Data .....	16
	Scrap Transaction Data .....	16
<b>7</b>	<b>Code Lists</b> .....	<b>18</b>
	Recyclable Materials That Do Not Qualify as Special Purchase Articles .....	18
	Recyclable Materials That Qualify as Special Purchase Articles .....	19
	State Codes.....	19
<b>8</b>	<b>Versioning</b> .....	<b>21</b>

## 1 Revision History

---

Technical Specifications for Scrap Dealer Service API		
Date	Version	Description
07/15/2016	1.0	Original document

## 2 Overview

The scrap dealer service API (Application Programming Interface) is a REST service that can be used programmatically to implement a subset of scrap dealer activities over the internet.

### Getting Started

The scrap dealer service API is implemented as a REST-based Web Service using ASP.NET Web API. It is consumable by all modern development platforms and is offered in both JSON and XML formats. The API is available on the internet at the following URLs:

**Test URL** <https://servicesq.dps.ohio.gov/ScrapDealerServices.API/>

**Production URL** <https://services.dps.ohio.gov/ScrapDealerServices.API/>

### Authentication

Each registered dealer facility would have at least one authorized user that was registered as part of their initial setup. The user's credential information will need to be provided as part of each service call.

### Request Headers

All requests sent to the API must contain two HTTP Headers. If these headers are not supplied or invalid then a "**401/Unauthorized**" status is returned with a response message.

**Username** – Dealer's *username*

**Password** – Dealer's password

**Sample:**

```
[
{"Message":"Missing credentials."}
]
```

### HTTP Response Codes

HTTP	Description
<b>200 - OK</b>	Call successful. A response is returned.  <b>Action:</b> N/A
<b>400 - Bad Request</b>	A validation error has occurred with the data submitted to the API. This can occur if the user has made service call with a missing and invalid information.  A list of all error messages will be returned in the response.  <b>Action:</b> Correct the requested search parameters and retry.
<b>401 - Unauthorized</b>	Your request was unauthorized due to invalid credentials or your request is missing required HTTP headers  <b>Action:</b> Make sure to supply a valid scrap dealer credentials in the request http headers.
<b>404 - Not Found</b>	Invalid HTTP action was made on the request.  <b>Action:</b> Check the action method name being called.
<b>500 - Internal Error</b>	An unexpected error has occurred somewhere in the service. This could be the result of a bug in the system, unexpected downtime or just an issue with this particular user. Error details will be logged to the Service and the problem will be addressed by the ODPS IT team.

	<b>Action:</b> Try later and if issue persists contact the Scrap Metal Support
--	--

## Error Format

Error messages are returned in a standard format for easy consumption.

In JSON, the format is like:

```
[  
{"Message":"Facility registration number is required."},  
{"Message":"Seller's First Name can't be longer than 33 characters."}  
]
```

### 3 API Method Overview

---

API	Description
<a href="#"><u>POST v1/DoNotBuy/Search</u></a>	Searches and returns the matching records against the supplied driver license information.
<a href="#"><u>POST v1/Transaction/Validate</u></a>	Validates the supplied transaction and returns a 'Transaction Validated Successfully' message.
<a href="#"><u>POST v1/Transaction/Submit</u></a>	Saves the transaction and returns a 'Transaction Saved Successfully' message.

## 4 API Method Details

---

### POST v1/DoNotBuy/Search

#### URI Parameters

N/A

#### Request Information

##### Parameters

Name	Description
DriverLicenseInfo	Driver License Information

If invalid or missing credentials are supplied in the request's http header, this method will return '**401/Unauthorized**' response.

All the fields in the **DriverLicenseInfo** parameter are **mandatory**. If any of the required fields are missing or if an invalid value is supplied, this method will return with a '**400/Bad Request**' response. A list of error messages will be returned.

If no matching records are found, this method will return an empty list.

The driver's license state field must be a code from the list in [State Codes](#) section. A '**400/Bad Request**' response is returned if an invalid state code is supplied.

#### Request body formats

##### application/json, text/json

##### Sample:

```
{
  "FacilityRegNumber": "SMBC-2012-0000004",
  "FirstName": "John",
  "LastName": "Smith",
  "DOB": "2016-07-15",
  "DLNumber": "XYZ0001",
  "DLState": "OH"
}
```

#### Response Information

Returns a list of matching records in the format below. A list of error messages is returned if any error occurs.

#### Response body formats

**application/json****Sample:**

```
[
  {
    "Age":31,
    "Alias":"","
    "ConvictionCaseNo":"1234567",
    "ConvictionCourt":"Xyz Municipal Court",
    "ConvictionDate":"2014-06-05",
    "ConvictionOffense":"Rec/Stolen Prop",
    "ConvictionOffenseClass":"MIS",
    "DLIssueState":"OH",
    "DLNumber":"ZZ0001",
    "Dob":"1988-05-14",
    "Ethnicity":"White/Caucasian",
    "EyeColor":"Black",
    "FirstName":"John",
    "FullName":"John Smith",
    "HairColor":"Red",
    "Height":"609",
    "LastName":"Smith",
    "MiddleName":"X",
    "ScarsMarksTattoos":"","
    "Sex":"M",
    "SourceAgencyName":"Xyz Co Sheriff's Office",
    "Suffix":"","
    "Weight":"159"
  },
  {
    "Age":41,
    "Alias":"","
    "ConvictionCaseNo":"1234566",
    "ConvictionCourt":"Xyz Municipal Court",
    "ConvictionDate":"2014-06-05",
    "ConvictionOffense":"Rec/Stolen Prop",
    "ConvictionOffenseClass":"MIS",
    "DLIssueState":"OH",
    "DLNumber":"ZZ0002",
    "Dob":"05/14/1974",
    "Ethnicity":"White/Caucasian",
    "EyeColor":"Brown",
    "FirstName":"Will",
    "FullName":"Will Smith",
    "HairColor":"Black",
    "Height":"509",
    "LastName":"Smith",
    "MiddleName":"X",
    "ScarsMarksTattoos":"","
    "Sex":"M",
    "SourceAgencyName":"Xyz Co Sheriff's Office",
    "Suffix":"","
    "Weight":"169"
  },
]
```

**Note:** Height is in feet and inches. Example: 509 is 5 feet and 9 inches.

## POST v1/Transaction/Validate

### URI Parameters

N/A

### Request Information

#### Parameters

Name	Description
Transaction	Scrap Transaction Information

This method should be supplied with all relevant fields required for the transaction. See [Field Specification](#) section.

If invalid or missing credentials supplied in the request's http header, this method will return '**401/Unauthorized**' response.

If any of the required fields are missing or if an invalid value is supplied, this method will return with a '**400/Bad Request**' response. A list of error messages will be returned.

### Request body formats

#### application/json

##### Sample:

```
{
  "FacilityRegNumber": "SMBC-2012-0000004",
  "TranNumber": "TRAN000001",
  "TranDate": "2016-06-05T18:25:43",
  "SellerInfo": {
    "FirstName": "John",
    "MiddleName": "A",
    "LastName": "Smith",
    "AddressLine1": "12 High Street",
    "AddressLine2": "",
    "City": "Columbus",
    "State": "OH",
    "Zip": "43017",
    "IdCardImage": {
      "image": "QEBA"
    },
    "Photo": {
      "image": "QEBA"
    },
    "LicensePlateNumber": "ZY001",
    "LicensePlateIssueState": "OH"
  },
  "BulkContainerInfo": {
    "Description": "Bulk Container",
    "Items": [
      {
```

```
    "Photo": {
      "image": "QEBA"
    },
    "Weight": 1.1
  },
  {
    "Photo": {
      "image": "QEBA"
    },
    "Weight": 1.1
  },
  {
    "Photo": {
      "image": "QEBA"
    },
    "Weight": 1.1
  }
]
},
"MetalRecyclableInfo": {
  "Description": "",
  "Weight": 1.1
},
"RecycMaterialsNotSpPurchaseInfo": {
  "items": [
    {
      "code": 1,
      "weight": 2.1
    },
    {
      "code": 1,
      "weight": 2.1
    },
    {
      "code": 1,
      "weight": 2.1
    }
  ]
},
"RecycMaterialsSpPurchaseInfo": {
  "items": [
    {
      "code": 1,
      "weight": 2.1,
      "Photo": {
        "image": "QEBA"
      }
    },
    {
      "code": 1,
      "weight": 2.1,
      "Photo": {
        "image": "QEBA"
      }
    }
  ],
  {
    "code": 1,
```

```

    "weight": 2.1,
    "Photo": {
      "image": "QEBA"
    }
  ]
}

```

**Note:** The image data should be in 'image/jpeg;base64' format

### Response Information

If successful returns "**200/OK**" response else returns a list error messages.

### Response body formats

#### application/json

##### Sample:

```

[
  {
    "Message": "First Name Missing"
  },
  {
    "Message": "Last Name Missing"
  }
]

```

### POST v1/Transaction/Submit

**Test URL** (Will be published once operational)

**Production URL** (Will be published once operational)

### URI Parameters

N/A

### Request Information

#### Parameters

Name	Description
Transaction	Scrap Transaction Information

This method should be supplied with all relevant fields required for the transaction. See [Field Specification](#) section.

If invalid or missing credentials are supplied in the request's http header, this method will return '**401/Unauthorized**' response.

If any of the required fields are missing or if an invalid value is supplied, this method will return with a '**400/Bad Request**' response. A list of error messages will be returned.

## Request body formats

### application/json

#### Sample:

```
{
  "FacilityRegNumber": "SMBC-2012-0000004",
  "TranNumber": "TRAN000001",
  "TranDate": "2016-06-05T18:25:43",
  "SellerInfo": {
    "FirstName": "John",
    "MiddleName": "A",
    "LastName": "Smith",
    "AddressLine1": "12 High Street",
    "AddressLine2": "",
    "City": "Columbus",
    "State": "OH",
    "Zip": "43017",
    "IdCardImage": {
      "image": "QEBA"
    },
    "Photo": {
      "image": "QEBA"
    },
    "LicensePlateNumber": "ZY001",
    "LicensePlateIssueState": "OH"
  },
  "BulkContainerInfo": {
    "Description": "Bulk Container",
    "Items": [
      {
        "Photo": {
          "image": "QEBA"
        },
        "Weight": 1.1
      },
      {
        "Photo": {
          "image": "QEBA"
        },
        "Weight": 1.1
      },
      {
        "Photo": {
          "image": "QEBA"
        },
        "Weight": 1.1
      }
    ]
  }
}
```

```
},
"MetalRecyclableInfo": {
  "Description": "",
  "Weight": 1.1
},
"RecycMaterialsNotSpPurchaseInfo": {
  "items": [
    {
      "code": 1,
      "weight": 2.1
    },
    {
      "code": 1,
      "weight": 2.1
    },
    {
      "code": 1,
      "weight": 2.1
    }
  ]
},
"RecycMaterialsSpPurchaseInfo": {
  "items": [
    {
      "code": 1,
      "weight": 2.1,
      "Photo": {
        "image": "QEBA"
      }
    },
    {
      "code": 1,
      "weight": 2.1,
      "Photo": {
        "image": "QEBA"
      }
    },
    {
      "code": 1,
      "weight": 2.1,
      "Photo": {
        "image": "QEBA"
      }
    }
  ]
}
}
```

Note: The image data should be in 'image/jpeg;base64' format

## Response Information

If successful returns "200/OK" response else returns a list error messages.

## Response body formats

---

**application/json****Sample:**

```
[
  {
    "Message": "Invalid Facility Registration Number Supplied"
  },
  {
    "Message": "Tran Number Missing"
  }
]
```

## 5 How To Consume?

---

**C# Classes:**

```
public class DriverLicenseData
{
    public string FacilityRegNumber { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public string DOB { get; set; }
    public string DLNumber { get; set; }
    public string DLState { get; set; }
}

public class DoNotBuyData
{
    public string FirstName { get; set; }
    public string MiddleName { get; set; }
    public string LastName { get; set; }
    public string Suffix { get; set; }
    public string Sex { get; set; }
    public string Height { get; set; }
    public string Weight { get; set; }
    public string EyeColor { get; set; }
    public string HairColor { get; set; }
    public string Ethnicity { get; set; }
    public string ScarsMarksTattoos { get; set; }
    public string Dob { get; set; }
    public int Age { get; set; }
    public string SourceAgencyName { get; set; }
    public string FullName { get; set; }
    public string DLNumber { get; set; }
    public string DLIssueState { get; set; }
    public string Alias { get; set; }
}
```

```
    public string ConvictionCaseNo { get; set; }
    public string ConvictionDate { get; set; }
    public string ConvictionOffense { get; set; }
    public string ConvictionOffenseClass { get; set; }
    public string ConvictionCourt { get; set; }
}

public class TransactionData
{
    public string FacilityRegNumber { get; set; }
    public string TranNumber { get; set; }
    public string TranDate { get; set; }
    public Seller SellerInfo { get; set; }
    public BulkContainer BulkContainerInfo { get; set; }
    public MetalNonRecyclable MetalRecyclableInfo { get; set; }
    public RecycMaterialsNotSpPurchase RecycMaterialsNotSpPurchaseInfo { get; set; }
    public RecycMaterialsSpPurchase RecycMaterialsSpPurchaseInfo { get; set; }
}

public class Seller
{
    public string FirstName { get; set; }
    public string MiddleName { get; set; }
    public string LastName { get; set; }
    public string AddressLine1 { get; set; }
    public string AddressLine2 { get; set; }
    public string City { get; set; }
    public string State { get; set; }
    public string Zip { get; set; }
    public Photo IdCardImage { get; set; }
    public Photo Photo { get; set; }
    public string LicensePlateNumber { get; set; }
    public string LicensePlateIssueState { get; set; }
}

public class MetalNonRecyclable
{
    public string Description { get; set; }
    public double Weight { get; set; }
}

public class BulkContainer
{
    public string Description { get; set; }
    public List<BulkContainerItem> items { get; set; }
}

public class BulkContainerItem
{
    public Photo Photo { get; set; }
    public double Weight { get; set; }
}

public class RecycMaterialsNotSpPurchase
{
    public List<NonSpecialPurchase> items { get; set; }
}
```

```
public class RecycMaterialsSpPurchase
{
    public List<SpecialPurchase> items { get; set; }
}

public class SpecialPurchase
{
    public int code { get; set; }
    public double weight { get; set; }
    public Photo Photo { get; set; }
}

public class NonSpecialPurchase
{
    public int code { get; set; }
    public double weight { get; set; }
}

public class Photo
{
    public byte[] image { get; set; }
}

public class Error
{
    public string Message { get; set; }
}
```

### Sample C# Code To Call The Service API (v1/DoNotBuy/Search)

```
static void Main(string[] args)
{
    PostAsync().Wait();
}

private static async Task PostAsync()
{
    using (var client = new HttpClient())
    {
        client.BaseAddress = new Uri("https://servicesq.dps.ohio.gov/ScrapDealerServices.API/");
        client.DefaultRequestHeaders.Accept.Clear();
        client.DefaultRequestHeaders.Accept.Add(new
            MediaTypeWithQualityHeaderValue("application/json"));

        //Valid scrap dealer credentials must be supplied
        client.DefaultRequestHeaders.Add("username", "xxx");
        client.DefaultRequestHeaders.Add("password", "xxx");

        var dl = new DriverLicenseData();

        //All fields are mandatory
        dl.FacilityRegNumber = "SMBC-2012-0000004";
        dl.FirstName = "John";
        dl.LastName = "Smith";
    }
}
```

```
dl.DOB = "1989-04-18";
dl.DLNumber = "TX329SD";
dl.DLState = "OH";

var response = await client.PostAsJsonAsync("v1/DoNotBuy/Search", dl);
if (response.IsSuccessStatusCode)
{
    var dnbList = await response.Content.ReadAsAsync<List<DoNotBuyData>>();

    if (dnbList.Any())
    {
        //Found some matches
    }
    else
    {
        //No matches found
    }
}
else
{
    //Invalid or missing input header/parameter information
    var errorList = await response.Content.ReadAsAsync<List<Error>>();
}
}
```

### Sample C# Code To Call The Service API (v1/Transaction/Validate)

```
static void Main(string[] args)
{
    PostAsync().Wait();
}

private static async Task PostAsync()
{
    using (var client = new HttpClient())
    {
        client.BaseAddress = new Uri("https://servicesq.dps.ohio.gov/ScrapDealerServices.API/");
        client.DefaultRequestHeaders.Accept.Clear();
        client.DefaultRequestHeaders.Accept.Add(new
            MediaTypeWithQualityHeaderValue("application/json"));
        client.DefaultRequestHeaders.Add("username", "xxx");
        client.DefaultRequestHeaders.Add("password", "xxx");

        var scrapTran = new TransactionData();

        scrapTran.FacilityRegNumber = "SMBC-2012-0000004";
        scrapTran.TranNumber = "TRAN0001";
        scrapTran.TranDate = "2016-05-06T12:45:12";

        var seller = new Seller();
        seller.FirstName = "Johns";
        seller.LastName = "Smith";
        seller.AddressLine1 = "12 High Street";
        seller.City = "Columbus";
    }
}
```

```

seller.Zip = "43223";
seller.LicensePlateNumber = "SH487329";
seller.LicensePlateIssueState = "OH";
var photo = new Photo();
photo.image = File.ReadAllBytes(@"\images\image1.jpg");
seller.IdCard = photo;
photo.image = File.ReadAllBytes(@"\images\image2.jpg");
seller.Photo = photo;
scrapTran.SellerInfo = seller;
var metalRecyclable = new MetalNonRecyclable ();
metalRecyclable.Description = "Metal Article 1";
metalRecyclable.Weight = 34.4;
scrapTran.MetalNonRecyclableInfo = metalRecyclable;

var response = await client.PostAsJsonAsync("v1/Transaction/Validate", scrapTran);
if (response.IsSuccessStatusCode){
    //Validation Successfull
}
Else {
    //Read and display all the error messages returned.
    var errors = await response.Content.ReadAsAsync<List<Error>>();
}
}
}

```

## 6 Field Specifications

### Driver License Data

Field Name	Description	
FacilityRegNumber	Facility registration number	Required
FirstName	The first name of the seller	Required
LastName	The last name of the seller	Required
DOB	The date of birth of the seller.	Required
DLNumber	The Driver License number of the seller	Required
DLState	The Driver License issue state of the seller. See <a href="#">state codes</a>	Required

### Scrap Transaction Data

Field Name	Description	Size	
FacilityRegNumber	Facility registration number	17	Required
TranNumber	Transaction number	20	Required
TranDate	The date and time of the sale Format: (yyyy-MM-ddTHH:mm)		Required
<b>Seller</b>			
FirstName	The first name of the seller	33	Required

Field Name	Description	Size	
MiddleName	The middle name of the seller	31	Optional
LastName	The last name of the seller	33	Required
Suffix	If applicable, the suffix in the seller's name	5	Optional
AddressLine1	The street address of the seller	40	Required
AddressLine2	The street address of the seller	40	Optional
City	The seller's city of residence	30	Required
State	The seller's state of residence For the allowed values, see section <a href="#">state codes</a>	2	Required
Zip	The seller's zip code of residence	9	Required
IdCardImage	Image of the seller's state issued identification card, driver's license or military identification card. Image should be in JPEG format and the size should not exceed 1MB.		Required
Photo	Photograph of the seller Photo should be in JPEG format and the size should not exceed 1MB.		Required
LicensePlateNumber	The license plate number of the seller's motor vehicle	20	Optional
LicensePlateIssueState	The state that issued the license plate of the seller's motor vehicle For the state codes, see section <a href="#">state codes</a>	2	Optional
<b>Bulk Container</b>			
Description	A description of the bulk container(s)	255	Optional
<b>Items (Collection)</b>			
Photo	A photograph of the bulk container item. Photos should be in JPEG format and the size of each photo should not exceed 1MB.		Required if Description field is supplied
Weight	Weight of the item in above supplied container.	7	Required if Description field is supplied
<b>Metal Articles – Not Recyclable</b>			

Field Name	Description	Size	
Description	Full and accurate description of each article purchased or received by the dealer. Include identifying marks, letters written, inscribed, or otherwise included on the article. Include the name and manufacturer of the article if known	255	Optional
Weight	Total weight of the Metal Articles that are not Recyclable Example: 103.00	7	Conditional (Required if above field is supplied)
<b>Recyclable – Not Special Purchase (Collection)</b>			
Code	Supply recyclable Not special purchase code. For a list of articles, see <a href="#">Codes</a> .  Example: 1 (i.e. Beer Keg)	2	Optional
Weight	Weight of above supplied item.  Example: 20.00 (i.e. Beer Kegs of 20 LB)	7	Conditional (Required if above field is supplied)
<b>Recyclable – Special Purchase (Collection)</b>			
Code	Supply recyclable special purchase code. For a list of articles, see <a href="#">Codes</a> .  Example: 1 (i.e. Beer Keg)	2	Optional
Weight	Weight of above supplied item.  Example: 20.00 (i.e. Beer Kegs of 20 LB)	7	Conditional (Required if above field is supplied)
Photo	Photo of the above supplied item.  Photos should be of JPG format and size of each photo should not exceed 1MB.		Conditional (Required if above field is supplied)

## 7 Code Lists

### Recyclable Materials That Do Not Qualify as Special Purchase Articles

Code	Description
1	Number 1 Copper
2	Number 2 Copper
3	Sheet Copper
4	Insulated Copper Wire
5	Aluminum or Copper Radiators
6	Red Brass
7	Yellow Brass

8	Aluminum Sheet
9	Aluminum Extrusions
10	Clean Aluminum Wire
11	Cast Aluminum
12	Unclean Aluminum Wire
13	Aluminum Exterior
14	Contaminated Aluminum
15	Stainless Steel
16	Large Appliances
17	Steel Structural
18	Miscellaneous Steel
19	Sheet Irons
20	Motor-Vehicles Non-body Parts
21	Catalytic Converters
22	Lead
23	Electric Motors
24	Electronic Scrap

### Recyclable Materials That Qualify as Special Purchase Articles

Code	Description
1	Beer kegs;
2	Cable, wire, electrical components, and other equipment used in providing cable service or any utility service, including, but not limited to, copper or aluminum coverings, housings, or enclosures related thereto
3	Grave markers, sculptures, plaques, and vases made out of metal, the appearance of which suggests that the articles have been obtained from a cemetery
4	Guard rails for bridges, highways, and roads; highway and street signs; street light poles and fixtures; worker access hole covers, water meter covers, and other similar types of utility access covers; traffic directional and control signs and light signals, metal marked with the name of a political subdivision of the state, and other metal articles that are purchased and installed for use upon authorization of the state or any political subdivision of the state
5	Historical, commemorative, and memorial markers and plaques made out of metal
6	Four-wheel metal carts, commonly referred to as "grocery carts," that are generally used by individuals to collect and transport consumer goods while shopping
7	Four-wheel metal carts, commonly referred to as "metal bossies," that are used to transport or merchandise food products that are stored in crates, shells, or trays
8	Railroad material, including journal brasses, rail spikes, rails, tie plates, frogs, and communication wire
9	Metal trays, merchandise containers, or similar transport containers used by a product producer, distributor, retailer, or an agent of a product producer, distributor, or retailer as a means for the bulk transportation, storage, or carrying of retail containers of milk, baked goods, eggs, or bottled beverage products
10	"Burnt wire," which is any coated metal wire that has been smelted, burned, or melted thereby removing the manufacturer's or owner's identifying marks

### State Codes

Code	Description
AL	Alabama
AK	Alaska
AZ	Arizona

---

<b>Code</b>	<b>Description</b>
AR	Arkansas
CA	California
CO	Colorado
CT	Connecticut
DE	Delaware
DC	District of Columbia
FL	Florida
GA	Georgia
HI	Hawaii
ID	Idaho
IL	Illinois
IN	Indiana
IA	Iowa
KS	Kansas
KY	Kentucky
LA	Louisiana
ME	Maine
MD	Maryland
MA	Massachusetts
MI	Michigan
MN	Minnesota
MS	Mississippi
MO	Missouri
MT	Montana
NE	Nebraska
NV	Nevada
NH	New Hampshire
NJ	New Jersey
NM	New Mexico
NY	New York
NC	North Carolina
ND	North Dakota
OH	Ohio
OK	Oklahoma
OR	Oregon

---

Code	Description
PA	Pennsylvania
RI	Rhode Island
SC	South Carolina
SD	South Dakota
TN	Tennessee
TX	Texas
UT	Utah
VT	Vermont
VA	Virginia
WA	Washington
WV	West Virginia
WI	Wisconsin
WY	Wyoming

## 8 Versioning

The “v1” in the URL refers to this being version 1 of the REST API. It is safe to assume that any changes made to the REST API will be “non-breaking” and will not require immediate action by the vendor.

A new version will be added for “breaking” changes – that is, enhancements or changes that would prevent the client-side code from working without changes. Vendors will be notified when a new version is released and have approximately 60 days before the previous version is deprecated.