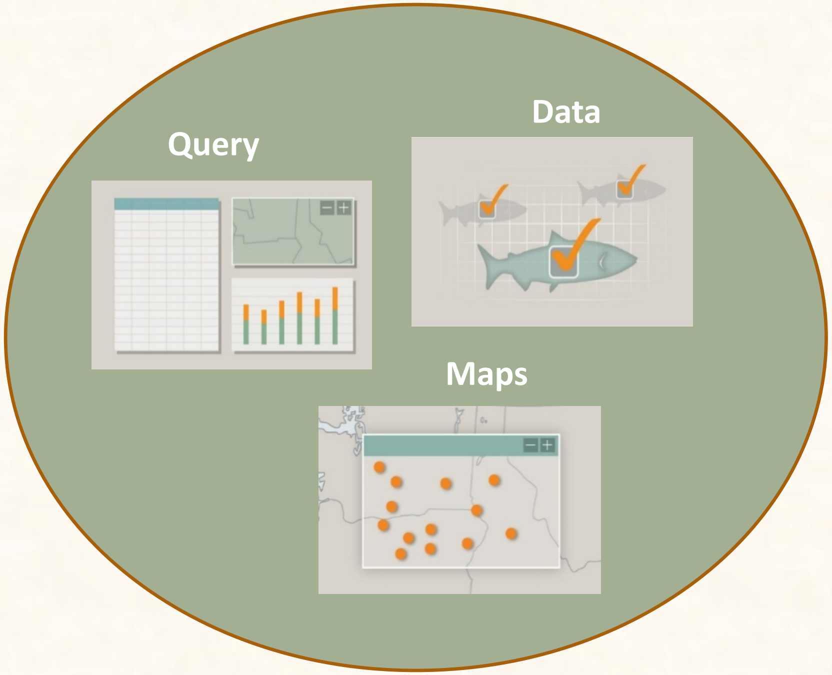
Efficiency and Opportunities Gained with a Data Exchange API

GREG WILKE,

PSMFC, STREAMNET PROJECT

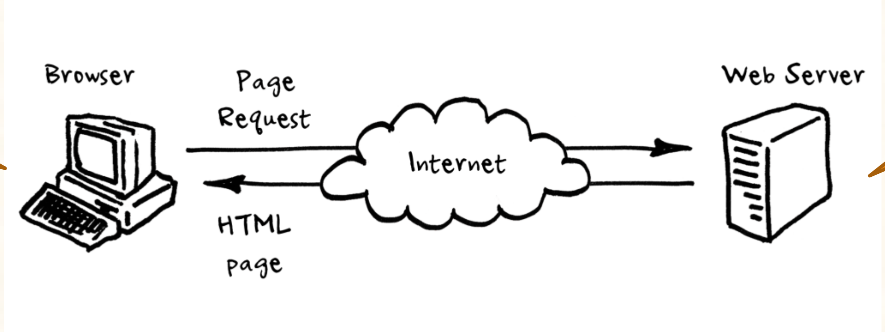
StreamNet



API and REST

Application **P**rogramming **I**nterface

I speak HTTP



Hey! Me too!

Google API

Weather API

GitHub API

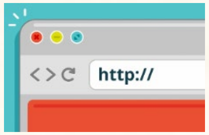
State Open Data Portal

GIS API

Twitter API

Amazon Cloud API

= REST API



- Speak HTTP
- Basic data format
- Easy for clients to use
- Data both directions
- CRUD functions

(**C**reate, **R**ead, **U**pdate, **D**elete)

REST APIs are Everywhere

Private sector:

ESRI - https://www.esri.com/en-us/arcgis/open-vision/standards/open-apis-specs
Google Maps - https://developers.google.com/maps/documentation/javascript/tutorial
IBM Watson - https://www.ibm.com/watson/developer/
Twitter - https://developer.twitter.com/
Netflix - https://www.programmableweb.com/api/netflix
Facebook - https://developers.facebook.com/
Zillow - https://www.zillow.com/howto/api/APIOverview.htm
Bloomberg - https://www.bloomberg.com/professional/support/api-library/
Weather Underground - https://www.wunderground.com/weather/api/
FullContact - https://www.fullcontact.com/developer/
Amazon S3 (Amazon cloud) - https://aws.amazon.com/s3/
Twilio - https://www.twilio.com/
MailChimp - https://developer.mailchimp.com/documentation/mailchimp/
Github - https://developer.github.com/v3/

Federal government:

- Data.gov - https://www.data.gov/developers/apis
- Of the roughly 250 federal agencies, approximately half now have public APIs.
- There are almost 100 developer hubs across the federal government.

State government:

Open Data Portals

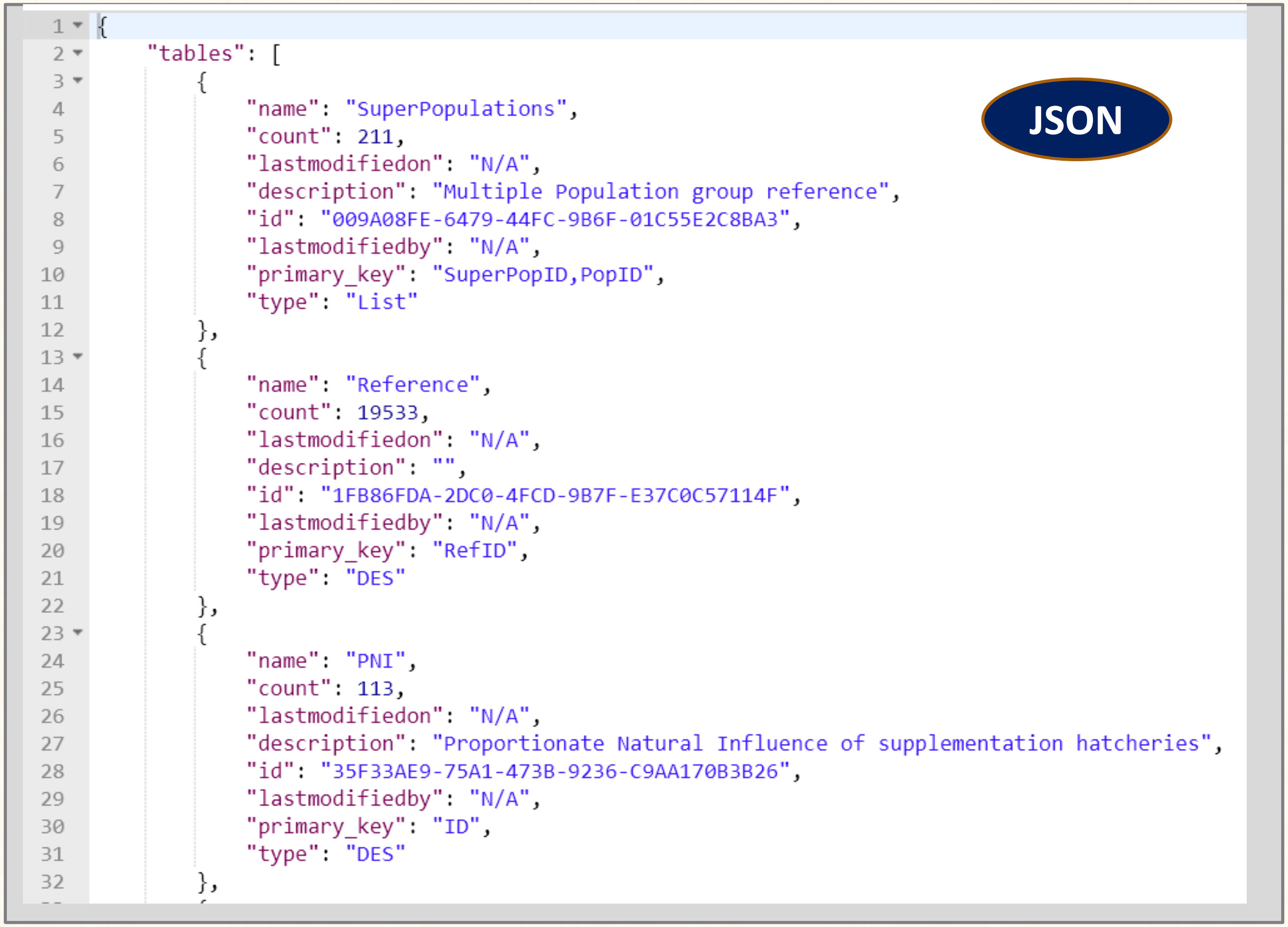
Programmable Web - https://www.programmableweb.com/

Apigee.com - API services interface with list of available services

R is the preferred letter in CRUD

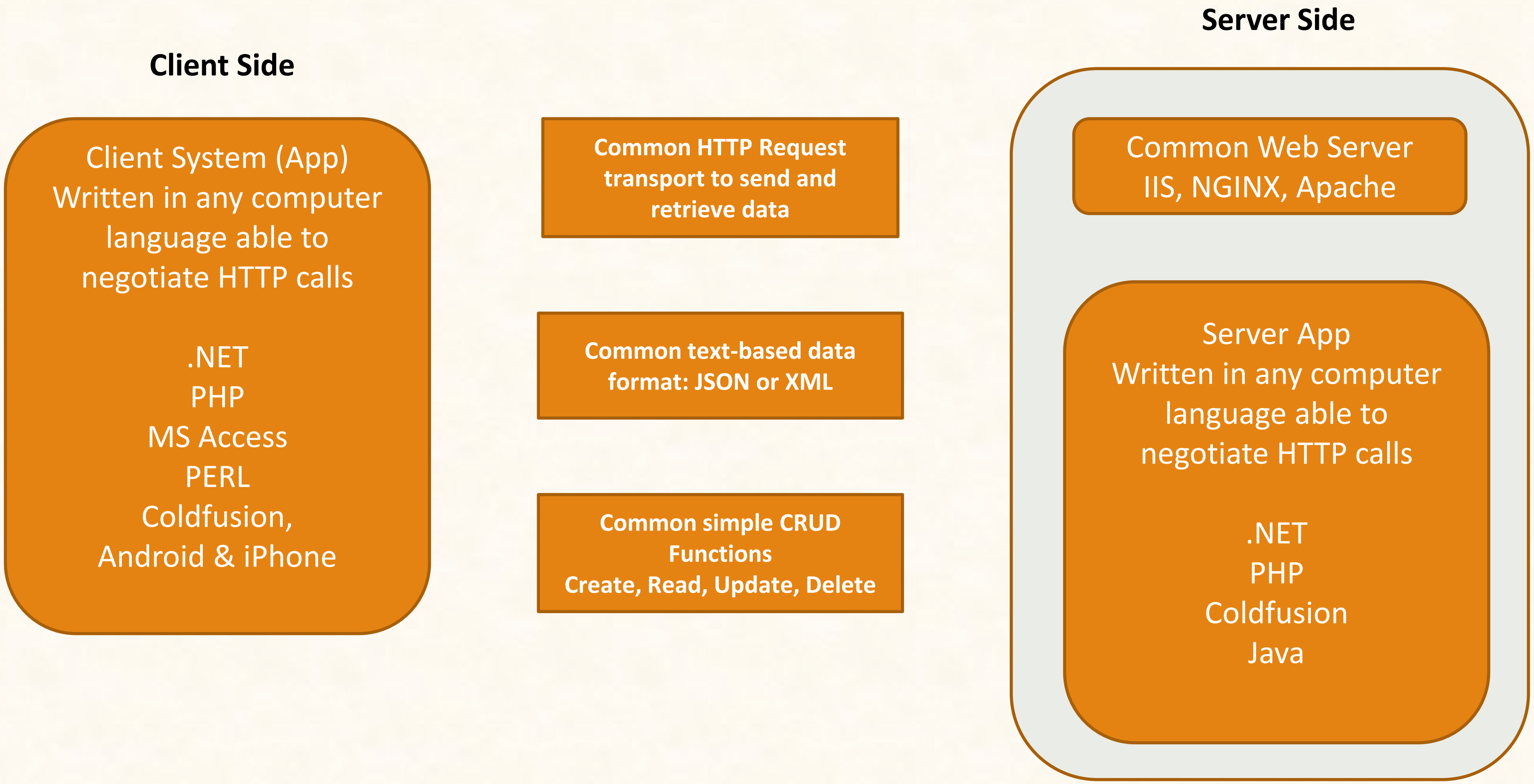
REST Data Format

https://api.streamnet.org/api/v1/catables



```
{
    "tables": [
        {
            "name": "SuperPopulations",
            "count": 211,
            "lastmodifiedon": "N/A",
            "description": "Multiple Population group reference",
            "id": "009A08FE-6479-44FC-9B6F-01C55E2C8BA3",
            "lastmodifiedby": "N/A",
            "primary_key": "SuperPopID,PopID",
            "type": "List"
        },
        {
            "name": "Reference",
            "count": 19533,
            "lastmodifiedon": "N/A",
            "description": "",
            "id": "1FB86FDA-2DC0-4FCD-9B7F-E37C0C57114F",
            "lastmodifiedby": "N/A",
            "primary_key": "RefID",
            "type": "DES"
        },
        {
            "name": "PNI",
            "count": 113,
            "lastmodifiedon": "N/A",
            "description": "Proportionate Natural Influence of supplementation hatcheries",
            "id": "35F33AE9-75A1-473B-9236-C9AA170B3B26",
            "lastmodifiedby": "N/A",
            "primary_key": "ID",
            "type": "DES"
        },
```

Basic REST API Functional Components



Server Side

Client Side

Client System (App)
Written in any computer language able to negotiate HTTP calls

.NET
PHP
MS Access
PERL
Coldfusion,
Android & iPhone

Common HTTP Request transport to send and retrieve data

Common text-based data format: JSON or XML

Common simple CRUD Functions
Create, Read, Update, Delete

Common Web Server
IIS, NGINX, Apache

Server App
Written in any computer language able to negotiate HTTP calls

.NET
PHP
Coldfusion
Java

StreamNet API Design

Let the client development begin!

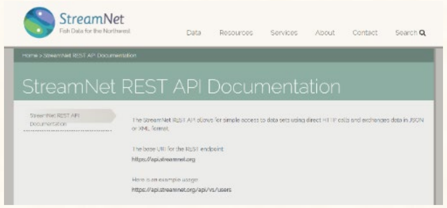
Design goals

- Minimal documentation necessary
- Server-side data abstraction
- Standard REST convention of collections & members
- Robust data validation

Kickoff

API Version 1

Docs


StreamNet
Fish Data for the Northwest
Data
Resources
Services
About
Contact
Search
Home > StreamNet REST API Documentation
StreamNet REST API Documentation

The StreamNet REST API allows for simple access to data sets using direct HTTP calls and exchanges data in JSON or XML format.

The base URI for the REST endpoint

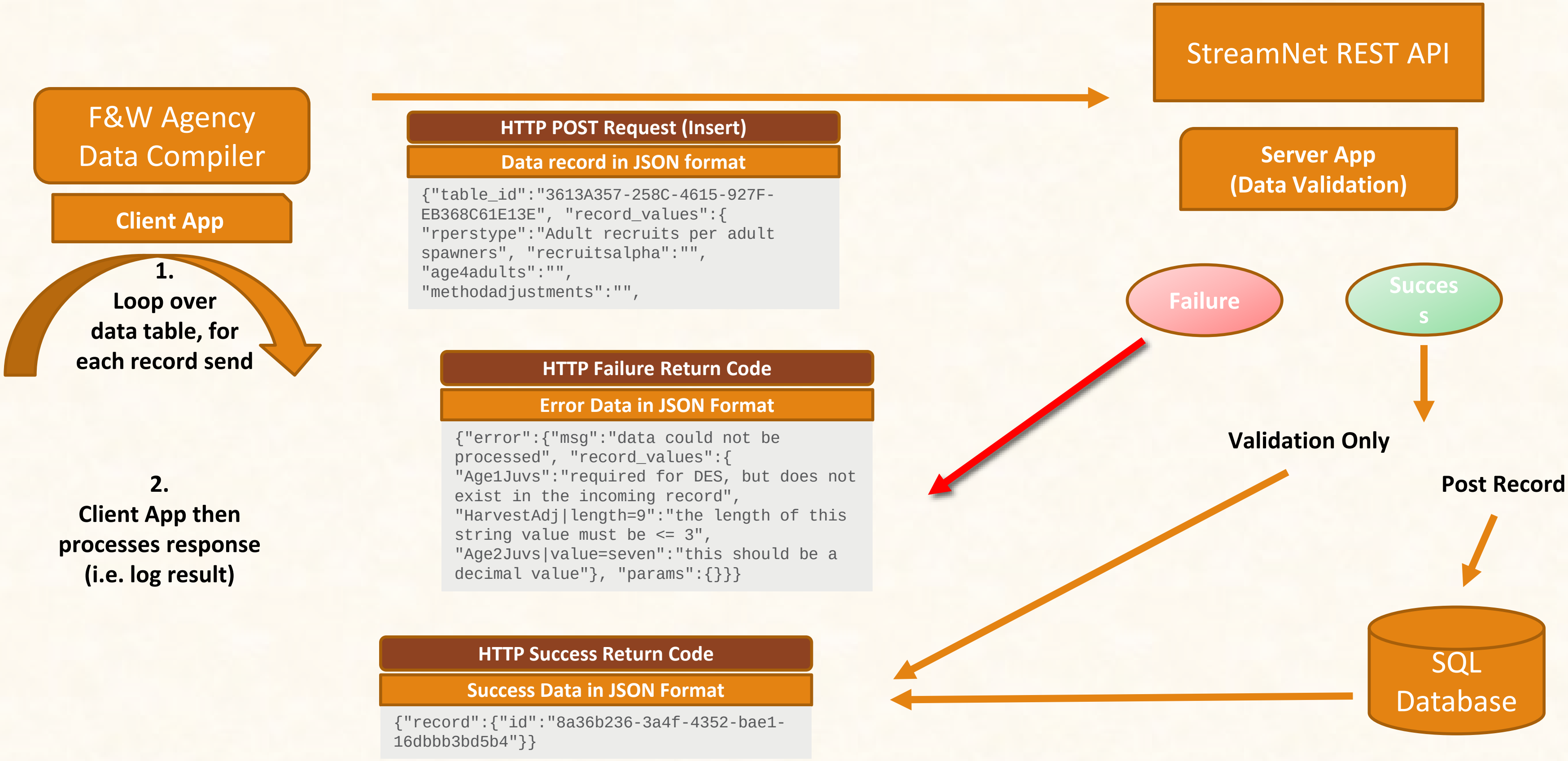
https://api.streamnet.org

Here is an example usage:

https://api.streamnet.org/api/v1/users

Demo Client App

StreamNet API Data Flow



F&W Agency Data Compiler

Client App

1.
Loop over data table, for each record send

2.
Client App then processes response (i.e. log result)

StreamNet REST API

Server App
(Data Validation)

Failure

Success

Validation Only

Post Record

SQL Database

HTTP POST Request (Insert)

Data record in JSON format

```
{"table_id":"3613A357-258C-4615-927F-
EB368C61E13E", "record_values":{
"rperstype":"Adult recruits per adult
spawners", "recruitsalpha":"",
"age4adults":"",
"methodadjustments":"",
```

HTTP Failure Return Code

Error Data in JSON Format

```
{"error":{"msg":"data could not be
processed", "record_values":{
"Age1Juvs":"required for DES, but does not
exist in the incoming record",
"HarvestAdj|length=9":"the length of this
string value must be <= 3",
"Age2Juvs|value=seven":"this should be a
decimal value"}, "params":{}}}
```

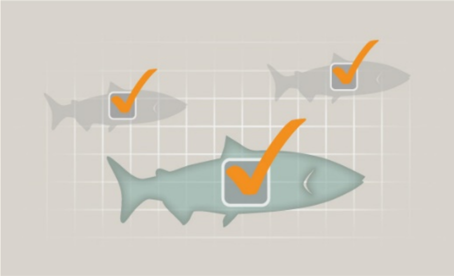
HTTP Success Return Code

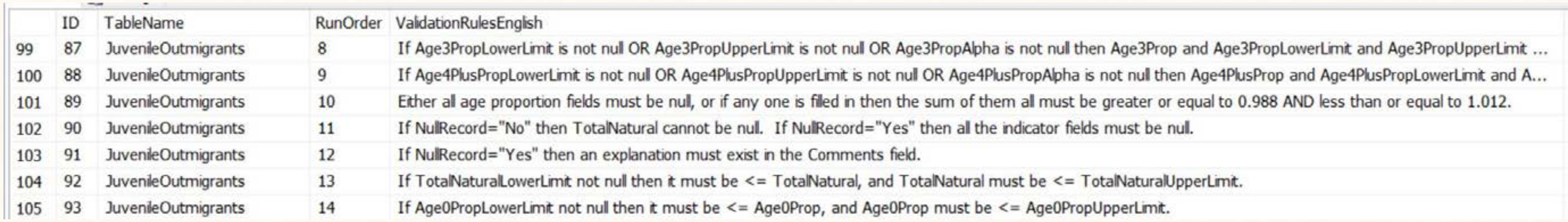
Success Data in JSON Format

```
{"record":{"id":"8a36b236-3a4f-4352-bae1-
16dbbb3bd5b4"}}
```

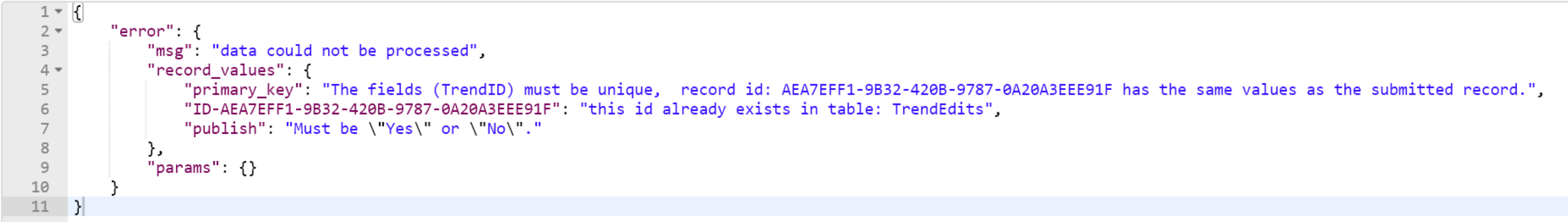
Data Validation

- **Field level**
- **Record level**
- **Table level**





	ID	TableName	RunOrder	ValidationRulesEnglish
99	87	JuvenileOutmigrants	8	If Age3PropLowerLimit is not null OR Age3PropUpperLimit is not null OR Age3PropAlpha is not null then Age3Prop and Age3PropLowerLimit and Age3PropUpperLimit ...
100	88	JuvenileOutmigrants	9	If Age4PlusPropLowerLimit is not null OR Age4PlusPropUpperLimit is not null OR Age4PlusPropAlpha is not null then Age4PlusProp and Age4PlusPropLowerLimit and A...
101	89	JuvenileOutmigrants	10	Either all age proportion fields must be null, or if any one is filled in then the sum of them all must be greater or equal to 0.988 AND less than or equal to 1.012.
102	90	JuvenileOutmigrants	11	If NullRecord="No" then TotalNatural cannot be null. If NullRecord="Yes" then all the indicator fields must be null.
103	91	JuvenileOutmigrants	12	If NullRecord="Yes" then an explanation must exist in the Comments field.
104	92	JuvenileOutmigrants	13	If TotalNaturalLowerLimit not null then it must be <= TotalNatural, and TotalNatural must be <= TotalNaturalUpperLimit.
105	93	JuvenileOutmigrants	14	If Age0PropLowerLimit not null then it must be <= Age0Prop, and Age0Prop must be <= Age0PropUpperLimit.



```
{
    "error": {
        "msg": "data could not be processed",
        "record_values": {
            "primary_key": "The fields (TrendID) must be unique,  record id: AEA7EFF1-9B32-420B-9787-0A20A3EEE91F has the same values as the submitted record.",
            "ID-AEA7EFF1-9B32-420B-9787-0A20A3EEE91F": "this id already exists in table: TrendEdits",
            "publish": "Must be \"Yes\" or \"No\"."
        },
        "params": {}
    }
}
```

Efficiencies & Opportunities Gained

- Faster Data Validation
- Better Quality Data
- Easy to Share Data with Others
- Integration with Other APIs

Current Status

- ✓ High Level Indicator data all submitted through the API for roughly 2 years now.
- ✓ Primary time-series data started being submitted through the API as of a few months ago. I said earlier that this type is more complicated with normalized data, but we implemented a system that works for our project.
- ✓ Some data with GIS information embedded do not go live immediately because they must still be processed and validated by a person using GIS tools and StreamNet's maps and layers. We have no plans to automate this at present.
- ✓ Our long-toiling Data Manager was finally able to retire this year at the age of 67

Publishing Data with a REST API

Most REST APIs are used for PUBLISHING data. That is, the API is made available over the internet for clients to consume the data. This is the context you mostly hear about. State open data initiatives are about making data available to the public or specific users. Companies have APIs that allow custom software to have authorized access to data.

But, a REST API can also be developed to receive data.

The StreamNet REST API publishes fisheries data compiled by state and tribal partners. The API is also the mechanism for partners to submit and validate data.

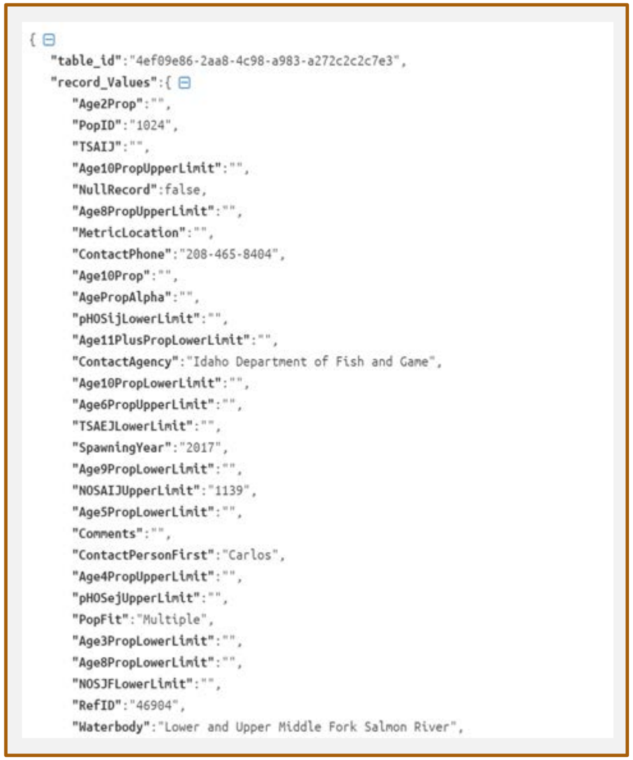
https://api.streamnet.org

A significant efficiency was gained by establishing a mechanism whereby compilers validate their own data using the API. Compilers can submit their data to the API as much as they want with the “validate=true” parameter with no changes to the regional database. We also use a Boolean field called Publish that defaults to “No” on every record. This allows data to be submitted and the database updated, but no record with Publish=No will show in any Web query tools or public/anonymous API calls.

For the purpose of getting data updated in a timely manner, this method actually works ok because the bulk of time involved with getting fisheries data from the stream to the regional database is ***in the field*** and in Fish & Wildlife agency regional offices where data is compiled and checked, and indicators calculated. The “last-mile” is getting the data submitted to a regional database like StreamNet.

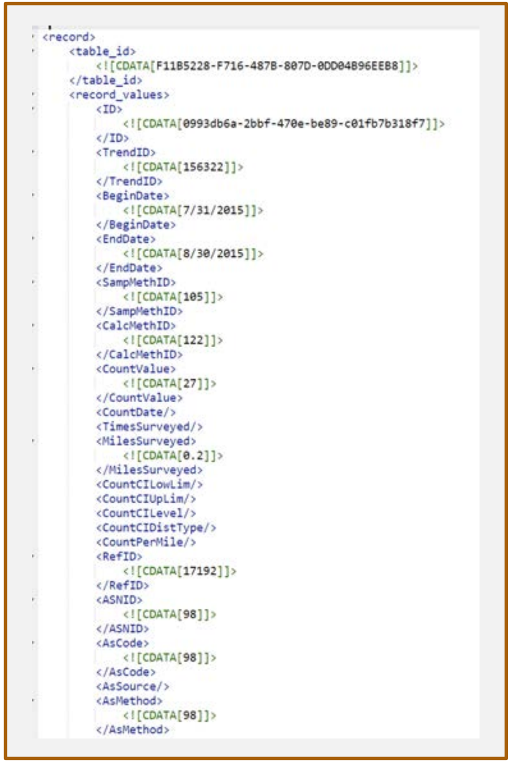
Basic REST API Data Formats

JSON

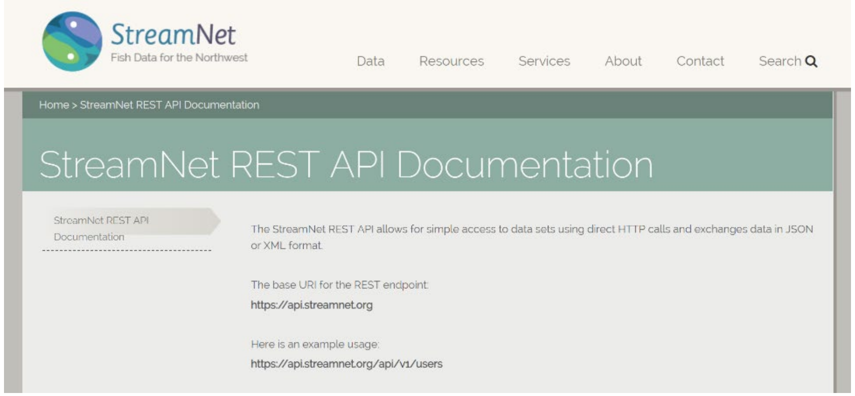


```
{ ⊟
   "table_id":"4ef09e86-2aa8-4c98-a983-a272c2c2c7e3",
   "record_Values":{ ⊟
      "Age2Prop":"",
      "PopID":"1024",
      "TSAIJ":"",
      "Age10PropUpperLimit":"",
      "NullRecord":false,
      "Age8PropUpperLimit":"",
      "MetricLocation":"",
      "ContactPhone":"208-465-8404",
      "Age10Prop":"",
      "AgePropAlpha":"",
      "pHOSijLowerLimit":"",
      "Age11PlusPropLowerLimit":"",
      "ContactAgency":"Idaho Department of Fish and Game",
      "Age10PropLowerLimit":"",
      "Age6PropUpperLimit":"",
      "TSAEJLowerLimit":"",
      "SpawningYear":"2017",
      "Age9PropLowerLimit":"",
      "NOSAIJUpperLimit":"1139",
      "Age5PropLowerLimit":"",
      "Comments":"",
      "ContactPersonFirst":"Carlos",
      "Age4PropUpperLimit":"",
      "pHOSejUpperLimit":"",
      "PopFit":"Multiple",
      "Age3PropLowerLimit":"",
      "Age8PropLowerLimit":"",
      "NOSJFLowerLimit":"",
      "RefID":"46904",
      "Waterbody":"Lower and Upper Middle Fork Salmon River",
```

XML



```
<record>
    <table_id>
        <![CDATA[F11B5228-F716-487B-807D-0DD04B96EEB8]]>
    </table_id>
    <record_values>
        <ID>
            <![CDATA[0993db6a-2bbf-470e-be89-c01fb7b318f7]]>
        </ID>
        <TrendID>
            <![CDATA[156322]]>
        </TrendID>
        <BeginDate>
            <![CDATA[7/31/2015]]>
        </BeginDate>
        <EndDate>
            <![CDATA[8/30/2015]]>
        </EndDate>
        <SampMethID>
            <![CDATA[105]]>
        </SampMethID>
        <CalcMethID>
            <![CDATA[122]]>
        </CalcMethID>
        <CountValue>
            <![CDATA[27]]>
        </CountValue>
        <CountDate/>
        <TimesSurveyed/>
        <MilesSurveyed>
            <![CDATA[0.2]]>
        </MilesSurveyed>
        <CountCILowLim/>
        <CountCIUpLim/>
        <CountCILevel/>
        <CountCIDistType/>
        <CountPerMile/>
        <RefID>
            <![CDATA[17192]]>
        </RefID>
        <ASNID>
            <![CDATA[98]]>
        </ASNID>
        <AsCode>
            <![CDATA[98]]>
        </AsCode>
        <AsSource/>
        <AsMethod>
            <![CDATA[98]]>
        </AsMethod>
```

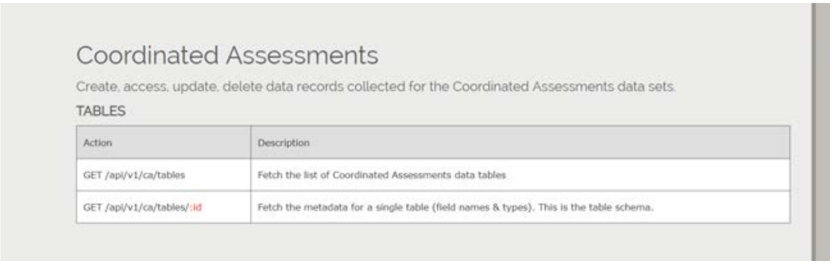

StreamNet
Fish Data for the Northwest
Data
Resources
Services
About
Contact
Search
Home > StreamNet REST API Documentation
StreamNet REST API Documentation

StreamNet REST API Documentation

The StreamNet REST API allows for simple access to data sets using direct HTTP calls and exchanges data in JSON or XML format.

The base URI for the REST endpoint:
https://api.streamnet.org

Here is an example usage:
https://api.streamnet.org/api/v1/users



Coordinated Assessments

Create, access, update, delete data records collected for the Coordinated Assessments data sets.

TABLES

Action	Description
GET /api/v1/ca/tables	Fetch the list of Coordinated Assessments data tables
GET /api/v1/ca/tables/:id	Fetch the metadata for a single table (field names & types). This is the table schema.

REST and HTTP

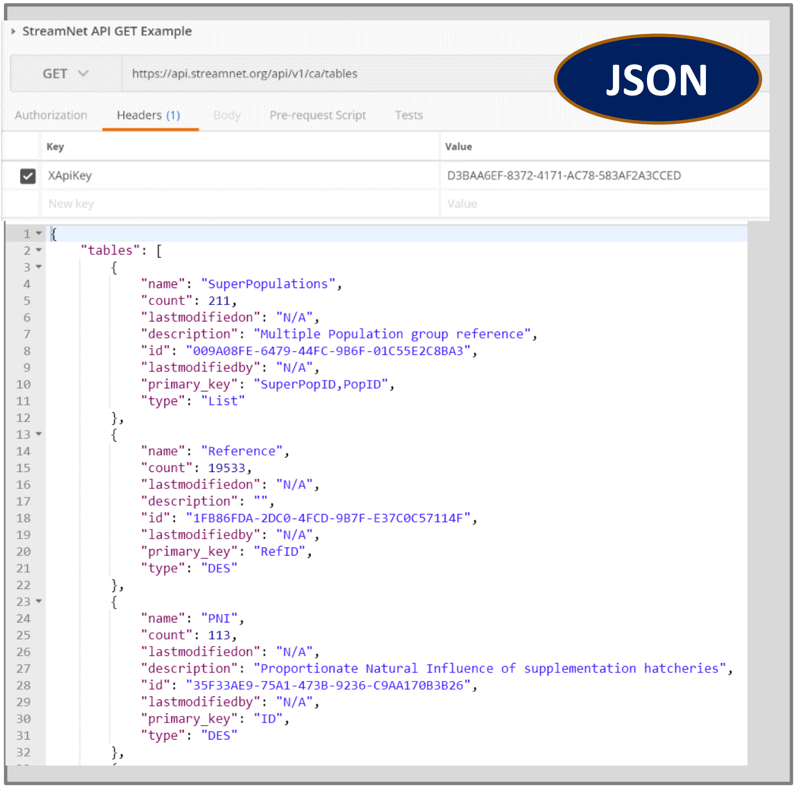
Now, it doesn't have to be the case, but usually a REST API works pretty much the same way a website does.

When a client browser like Chrome or Firefox makes a call from your client computer system to a server, you get data back over the HTTP protocol. The HTTP protocol is one of the primary mechanisms or specifications for internet communication. In the case of the browser, the server delivers data which the browser formats into a web page.

In the case of a REST API, a software client requests and receives data from a server (or sends data to the server) using the same underlying HTTP protocol as the browser, but what the client software does with the response depends on the application.

In the case of a browser, it displays information on a web page, but in the case of a GIS client application, a REST API could deliver a GIS layer that the client displays as a polygon on a map. Or an API could deliver tabular data to be imported into a SQL table or shown in a grid for the user.

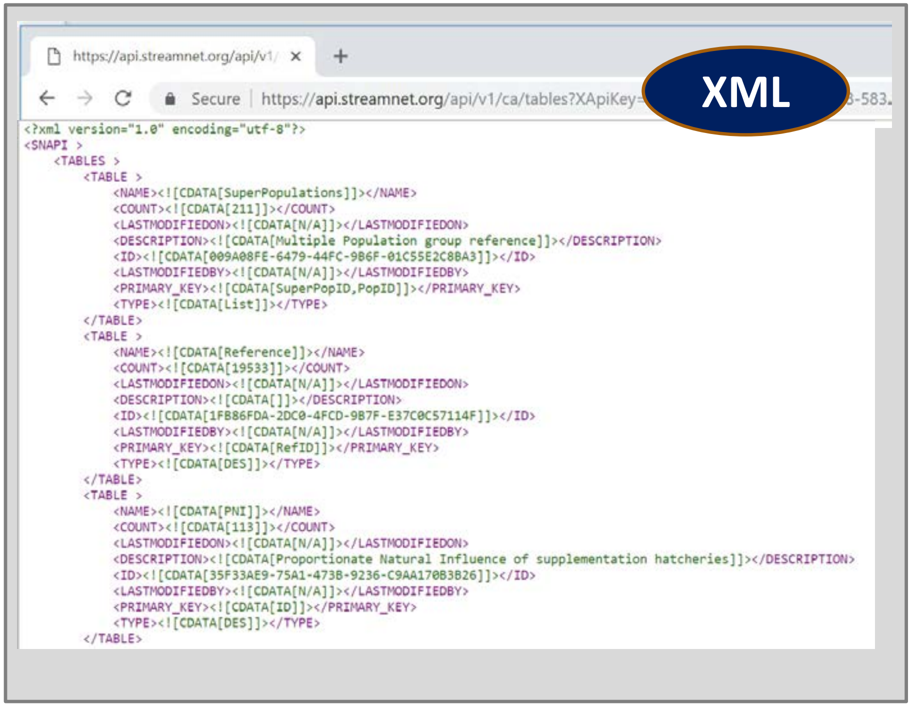
REST Data Formats



https://api.streamnet.org/api/v1/ca/tables

	Key	Value
☑	XApiKey	D3BAA6EF-8372-4171-AC78-583AF2A3CCED
	New key	Value

```
{
    "tables": [
        {
            "name": "SuperPopulations",
            "count": 211,
            "lastmodifiedon": "N/A",
            "description": "Multiple Population group reference",
            "id": "009A08FE-6479-44FC-9B6F-01C55E2C8BA3",
            "lastmodifiedby": "N/A",
            "primary_key": "SuperPopID,PopID",
            "type": "List"
        },
        {
            "name": "Reference",
            "count": 19533,
            "lastmodifiedon": "N/A",
            "description": "",
            "id": "1FB86FDA-2DC0-4FCD-9B7F-E37C0C57114F",
            "lastmodifiedby": "N/A",
            "primary_key": "RefID",
            "type": "DES"
        },
        {
            "name": "PNI",
            "count": 113,
            "lastmodifiedon": "N/A",
            "description": "Proportionate Natural Influence of supplementation hatcheries",
            "id": "35F33AE9-75A1-473B-9236-C9AA170B3B26",
            "lastmodifiedby": "N/A",
            "primary_key": "ID",
            "type": "DES"
        },
```



https://api.streamnet.org/api/v1/ca/tables?XApiKey=

```
<?xml version="1.0" encoding="utf-8"?>
<SNAPI >
    <TABLES >
        <TABLE >
            <NAME><![CDATA[SuperPopulations]]></NAME>
            <COUNT><![CDATA[211]]></COUNT>
            <LASTMODIFIEDON><![CDATA[N/A]]></LASTMODIFIEDON>
            <DESCRIPTION><![CDATA[Multiple Population group reference]]></DESCRIPTION>
            <ID><![CDATA[009A08FE-6479-44FC-9B6F-01C55E2C8BA3]]></ID>
            <LASTMODIFIEDBY><![CDATA[N/A]]></LASTMODIFIEDBY>
            <PRIMARY_KEY><![CDATA[SuperPopID,PopID]]></PRIMARY_KEY>
            <TYPE><![CDATA[List]]></TYPE>
        </TABLE>
        <TABLE >
            <NAME><![CDATA[Reference]]></NAME>
            <COUNT><![CDATA[19533]]></COUNT>
            <LASTMODIFIEDON><![CDATA[N/A]]></LASTMODIFIEDON>
            <DESCRIPTION><![CDATA[]]></DESCRIPTION>
            <ID><![CDATA[1FB86FDA-2DC0-4FCD-9B7F-E37C0C57114F]]></ID>
            <LASTMODIFIEDBY><![CDATA[N/A]]></LASTMODIFIEDBY>
            <PRIMARY_KEY><![CDATA[RefID]]></PRIMARY_KEY>
            <TYPE><![CDATA[DES]]></TYPE>
        </TABLE>
        <TABLE >
            <NAME><![CDATA[PNI]]></NAME>
            <COUNT><![CDATA[113]]></COUNT>
            <LASTMODIFIEDON><![CDATA[N/A]]></LASTMODIFIEDON>
            <DESCRIPTION><![CDATA[Proportionate Natural Influence of supplementation hatcheries]]></DESCRIPTION>
            <ID><![CDATA[35F33AE9-75A1-473B-9236-C9AA170B3B26]]></ID>
            <LASTMODIFIEDBY><![CDATA[N/A]]></LASTMODIFIEDBY>
            <PRIMARY_KEY><![CDATA[ID]]></PRIMARY_KEY>
            <TYPE><![CDATA[DES]]></TYPE>
        </TABLE>
```