# SIS Updates & Web Services

Prabha Acharya, Caltech

November 8, 2021

ANSS NetOps XI

# Features added in the last year

- Full Support for GNSS Stations
  - Import IGS Site Log into SIS for migration
  - Maintain GNSS stations and equipment using the SIS UI
  - Field actions for geodetic equipment
  - Publish IGS Site Log from SIS
- New web service endpoint: Channel level summary report
- Site Lists and Site List Groups
- Enhancements to device field actions
- Enhancements to equipment settings forms
- Category-wide settings

# Upcoming Features

Support for specialized equipment types

- Integrated sensor-logger models

  E.g., Basalt with internal episensor. The internal episensor will be integrated with the Basalt logger for configuration and installation in SIS.

- Digital sensors

  E.g., Temperature sensors that directly output in earth units

# SIS Web Services

# SIS Web Services

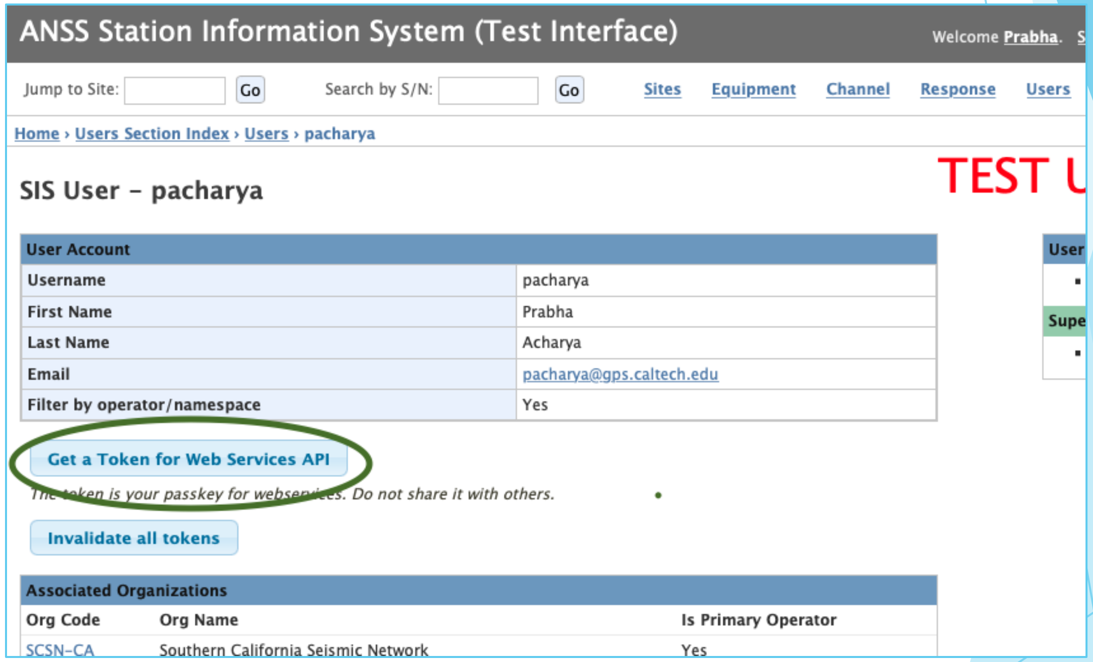
- You can write scripts to query SIS web services
  - to build custom reports
  - to build custom displays
  - to use in your website or app
  - to merge with information from other web services
- Read-only; no updating via web service
- Reads current state of SIS database as against published files that are a snapshot at time of publish
- Token based authentication
- JSON:API v1.0 compliant

# Token Authentication

- JSON Web Token (JWT) is used to authorize a user to the SIS web service
  - JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed. (From https://jwt.io/introduction)
- Improved security
  - SIS Token is a “one-day, read-only pass” for accessing the web service
  - User does not need to put in SIS username/password in the script
  - Users can invalidate their own existing tokens

# Token Authentication (cont’d)

- Get your token in SIS on your account page > “Get a Token”. If needed, invalidate all your existing tokens.
- SIS does not retain the token; it will generate a new one each time you “Get a token”. The token has an expiration time built into it (1 day).
- SIS provides an endpoint to ”refresh” the token to allow scripts to automatically fetch a new token each day and connect to the web service without daily user intervention. (Code snippet below)


ANSS Station Information System (Test Interface)
Welcome Prabha. S
Jump to Site:
Go
Search by S/N:
Go
Sites
Equipment
Channel
Response
Users
Home › Users Section Index › Users › pacharya
TEST U
User
Supe
Get a Token for Web Services API
Invalidate all tokens

SIS User – pacharya

| User Account | |
|---|---|
| Username | pacharya |
| First Name | Prabha |
| Last Name | Acharya |
| Email | pacharya@gps.caltech.edu |
| Filter by operator/namespace | Yes |

*The token is your passkey for webservices. Do not share it with others.*

| Associated Organizations | | |
|---|---|---|
| Org Code | Org Name | Is Primary Operator |
| SCSN-CA | Southern California Seismic Network | Yes |

```
res = requests.post('https://anss-sis.scsn.org/sistest/api/v1/token/refresh', data={'token': ..})
res.raise_for_status()          # Check if response is valid. Handle error
token = res.json()['token']
```

# JSON:API

- A specification for how a client should request that resources be fetched or modified, and how a server should respond to those requests. (From https://jsonapi.org)
- Content-type: application/vnd.api+json
- Document structure contains links, data, attributes, relationships, meta
- Client implementations are available in many languages to request and parse the response. See https://jsonapi.org/implementations/
  - These libraries can simplify building the request, working with paginated data, converting JSON to objects, and traversing through the relationships to get to related information

# SIS Web Service Endpoints

- Code automatically builds the documentation using the underlying web service API
- Latest documentation always at https://anss-sis.scsn.org/sis/api/v1/docs/
- These endpoints produce lists of the items indicated in JSON:API format
  - equipment
  - equipment-categories
  - equipment-installations
  - equipment-models
  - organizations
  - site-epochs
  - telemetry-connections
  - telemetry-nodes
- End points that do not produce JSON:API
  - fdsnws/channel – FDSN station web service text format. Matches the output format at fdsnws/station/1/query?...level=channel&format=text. Use to check data load by comparing differences against another datacenter like IRIS.
  - token – Output is a simple JSON object

# Endpoint: equipment

- https://anss-sis.scsn.org/sis/api/v1/docs/#equipment-list
- Server-side filtering done using query parameters. See table on the right.
- Output is paginated. Default page size is 100. Can go up to 500.
- Additional filtering can be done by the client

| | |
|---|---|
| page[number] | A page number within the paginated result set. |
| page[size] | Number of results to return per page. |
| category | Equipment Category: accepts comma separated values, case insensitive |
| categorygroup | Equipment Group: accepts comma separated values, case insensitive |
| modelname | Model name: accepts comma separated values, case insensitive |
| serialnumber | Serial number: accepts comma separated values |
| operatorcode | Operator: accepts comma separated values, case insensitive |
| ownercode | Owner/CoOwner: accepts comma separated values, case insensitive |
| inventory | Inventory: accepts comma separated values, case insensitive |
| sort | Field to sort by. Choices: category, modelname, serialnumber, categorygroup |

# SIS Resources

- Check out the SIS Wiki: https://wiki.anss-sis.scsn.org/SIStrac
    - Contains glossary, tutorials, FAQs, Tips & Tricks
- Email the SIS Development Team at sis-help@gps.caltech.edu
- SIS Training via Zoom, send email to schedule

# Demo Script

- Python script “api_demo.py’ shows how to send a request, use server-side filters, use client-side filters, handle the json response, combine paged data and write out a csv file
  - Connect to ‘equipment-models’ and write out csv file
  - Connect to ‘equipment’ and write out csv file
  - Refresh a token and save the new token to file
- Dependencies:
  - python requests library
  - Token(s) saved in text file(s)
- File is available at https://github.com/anss-sis/sis-api

# Script Usage

```
$ python3 api_demo.py -h
    usage: api_demo.py [-h] [--refreshtoken]
                       {prod,test} {getloggermodel,getequipment} ...
    SIS Webservice Reports
    positional arguments:
      {prod,test}           Connect to SIS test or production
    optional arguments:
      -h, --help            show this help message and exit
      --refreshtoken        Refresh the token
    Report type:
      {getloggermodel,getequipment}
```

# Script Usage (cont'd)

getloggermodel and getequipment are sub-commands

```
$ python3 api_demo.py test getequipment -h

   usage: api_demo.py {prod,test} getequipment [-h] --modelnames MODELNAMES
   [MODELNAMES ...] --operatorcodes OPERATORCODES [OPERATORCODES ...] [--inventory
   INVENTORY [INVENTORY ...]] outfilename

   positional arguments:

     outfilename            Path and name of output csv file

   optional arguments:

     -h, --help             show this help message and exit

     --modelnames MODELNAMES [MODELNAMES ...]

                            Equipment Modelnames

     --operatorcodes OPERATORCODES [OPERATORCODES ...]

                            Operator codes

     --inventory INVENTORY [INVENTORY ...]

                            Inventory states
```

# Questions