POD Process workflow



Requests need to raise :

<https://carms.nsiam.vzwnet.com/>

1. EDN workstation access

i. EDN Workstation Access

ii. EDN Account for Application Access

2. <https://webtools.verizon.com/> >> ADOM >> ADOM GROUP SEARCH

3. Atoll user access

SDLC\_GYDV\_GIT\_DEVELOPER

4. Atoll tickets are created at this location

--> [https://atyourservice.verizon.com/ays?id=support HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229" HYPERLINK "https://atyourservice.verizon.com/ays?id=support HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229"& HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229"ci=6e88edf4db4a53487e09feb5ae961988 HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229"& HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229"kb\_number=KB0030229" HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229"& HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229" HYPERLINK "https://atyourservice.verizon.com/ays?id=support HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229"& HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229"ci=6e88edf4db4a53487e09feb5ae961988 HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229"& HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229"kb\_number=KB0030229" HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229"ci=6e88edf4db4a53487e09feb5ae961988 HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229" HYPERLINK "https://atyourservice.verizon.com/ays?id=support HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229"& HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229"ci=6e88edf4db4a53487e09feb5ae961988 HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229"& HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229"kb\_number=KB0030229" HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229"& HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229" HYPERLINK "https://atyourservice.verizon.com/ays?id=support HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229"& HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229"ci=6e88edf4db4a53487e09feb5ae961988 HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229"& HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229"kb\_number=KB0030229" HYPERLINK "https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb\_number=KB0030229"kb\_number=KB0030229](https://atyourservice.verizon.com/ays?id=support&ci=6e88edf4db4a53487e09feb5ae961988&kb_number=KB0030229) --> Create An Incident

AOD Process

Starts from requesting the coverage data by hitting the url : <http://10.194.198.57:10534/atoll/coverage_request>

With input sample data (post url with json data)

**[**

**{**

**"tx\_id":"013384 01 B5\_10MHz\_2560 RAINBOW (2-0000)",**

**"site": "0384 RAINBOW (2-0000)",**

**"market":"saltlake",**

**"request\_type":"PSD",**

**"request\_subtype":"",**

**"receiver\_height":"5",**

**"user\_id":"",**

**"output\_request\_list":[**

**{**

**"type":"Vector",**

**"SubType":".shp",**

**"Threshold":"-104",**

**"Legend":""**

**}**

**],**

**"modified\_attributes":{**

**"XGTRANSMITTERS": {**

**"CALC\_RADIUS": "64373",**

**"CALC\_RADIUS2":"241000",**

**"CALC\_RESOLUTION":"30",**

**"CALC\_RESOLUTION2":"90",**

**"MAX\_RANGE":"0"**

**}**

**}**

**}**

**]**

Which calls the function :

def coverage\_request()

NPP database (Function to connect to db getNDLconnection postgres)- Production database

database="nppcoverages",

user="nppcoverages\_user",

password="",

host="nts-npp-coverages-prod.cluster-cl9vgbtolm5s.us-east-1.rds.amazonaws.com",

port="5432

NDL database (Function to connect to db getNPPconnection oracle ) - Staging database

username = "vzcoverage"

password = ""

dsn\_tns = cx\_Oracle.makedsn('txslatollpd2v.nss.vzwnet.com', '1521', 'atolsmig')

**oracle prod database**

username = "VZCOVERAGE"

password = ""

dsn\_tns = cx\_Oracle.makedsn('txslatollsd2v.nss.vzwnet.com', '1521', 'atolprd')

Connects to the prod/staging databases with cx\_oracle python library

Process for /atoll/coverage\_request

1. Reads the post request json and Queries the Staging database (oracle), fetches the siteinfo, txinfo,cell\_LTE, cell\_NR dataframes

2. Adds new columns with CALC\_RADIUS and CALC\_RADIUS@ to txinfo dt and other columns based on the condition(certail values like XGCELLSLTE, XGCELLSNR, azimut, tilt in modified\_attributes value in post request json)

3. Inserts these dataframes data to Prod database(postgres) by merging all those dataframes to results json

user --> hits api end point /atoll/coverage\_request --> queries the NDL database --> fetches all the required dataframes --> inserts the data into NPPCoverages database and updates the record ith status "SENT TO ATOLL" or "SENT TO ATOLL TEST" -->

in the back end

--> ondemand.py --> calculates the coverage data --> gets the request data --> queries the npp database ith status "SENT TO ATOLL" or "FAILURE" -->

user --> hits api end point /atoll/coverage\_request --> queries the NDL database --> fetches all the required dataframes --> inserts the data into NPPCoverages database and updates the record ith status "SENT TO ATOLL" or "SENT TO ATOLL TEST" -->

in the back end

--> ondemand.py --> calculates the coverage data --> gets the request data --> queries the npp database ith status "SENT TO ATOLL" or "FAILURE" -->

class NDLDbConnection:

def \_\_init\_\_(self):

xxx

def open(self):

xxxx

def close(self):

xxxx

def get\_connection(self):

xxxx

class NPPDbConnection:

def \_\_init\_\_(self):

xxx

def open(self):

xxxx

def close(self):

xxxx

def get\_connection(self):

xxxx

class PredictionOnDemand:

def \_\_init\_\_(self):

\_npp\_db\_conn = NPPDbConnection()

\_ndl\_db\_conn = NDLDbConnection()