D:\NHRI\AMR\amr-main\datasets\clean_SOURCE_2.py

```python
import pandas as pd
import numpy as np
import os

from cod_prep.claude.configurator import Configurator
from cod_prep.downloaders import (get_ages, get_cod_ages,
    get_current_location_hierarchy)
from cod_prep.utils.formatting import ages
from cod_prep.utils import report_if_merge_fail, print_log_message

CONF = Configurator()
SOURCE = "SOURCE_NAME"

L_DIR = "FILEPATH"

def read_in_data():

    df = pd.read_csv(L_DIR + 'FILENAME')

    df['cases'] = 1

    df['raw_specimen'] = df['source']
    df['raw_pathogen'] = df['species']

    df['sample_id'] = 'SOURCE-' + df['isolateid'].astype(str) + '-' + df['raw_specimen']

    df['year_id'] = df['year']

    return df


def format_age_group_id(df):

    age_dict = {
    '0 to 2 Years': 244,
    '3 to 12 Years': 291,
    '13 to 18 Years': 15,
    '19 to 64 Years': 163,
    '65 to 84 Years': 281,
    '85 and Over': 160,
    'Unknown': 283}

    df['age_group_id'] = df['agegroup'].map(age_dict)

    assert df.age_group_id.notnull().values.all()

    return df


def format_sex_id(df):

    df["sex_id"] = df["gender"].map({"Male": 1, "Female": 2, np.nan: 9})
    report_if_merge_fail(df, "sex_id", "gender")

    return df


def format_location_id(df):

    location = get_current_location_hierarchy()

    df['location_name'] = df['country']
    df.loc[(df['location_name'] == 'United States') & (df['state'].notnull()), 'location_name'] = df['state']

    df = df.merge(location[['location_name','location_id']], how = 'left', on = 'location_name')

    df.loc[df['location_id'].isna(), 'location_name'].unique()

    df.loc[df['location_name'] == 'United States', 'location_id'] = 102
    df.loc[df['location_name'] == 'Hong Kong', 'location_id'] = 354
    df.loc[df['location_name'] == 'Czech Republic', 'location_id'] = 47
    df.loc[df['location_name'] == 'Russia', 'location_id'] = 62
    df.loc[df['location_name'] == 'Venezuela', 'location_id'] = 133
    df.loc[df['location_name'] == 'Korea, South', 'location_id'] = 68
    df.loc[df['location_name'] == 'Taiwan', 'location_id'] = 8
    df.loc[df['location_name'] == 'Vietnam', 'location_id'] = 20
    df.loc[df['location_name'] == 'Slovak Republic', 'location_id'] = 54

    assert df.location_id.notnull().values.all()

    return df


def clean_drug(df):

    antibio = df.iloc[:, 13:101]
    antibio_names = [x for x in list(antibio) if "_i" in x]
    antibio_numbers = [x for x in list(antibio) if (x not in list(antibio_names))]

    df = df.drop(antibio_names, axis=1)

    df.loc[df[antibio_numbers].isnull().apply(lambda x: all(x), axis=1), antibio_numbers] = 'unknown'

    keep = [x for x in list(df) if (x not in list(antibio_numbers))]

    df = pd.melt(df, id_vars= keep, value_vars=antibio_numbers)

    df = df[df['value'].notnull()]

    df.rename(columns={'variable': 'raw_antibiotic', 'value': 'resistance'}, inplace=True)

    assert df.raw_antibiotic.notnull().values.all()
    assert df.resistance.notnull().values.all()

    return df


def clean_SOURCE():
    df = read_in_data()
    df = format_age_group_id(df)
    df = format_sex_id(df)
    df = format_location_id(df)
    df = clean_drug(df)

    df["nid"] = 410524

    return df


if __name__ == '__main__':
    df = clean_SOURCE()
    demo_cols = ['nid', 'sample_id', 'location_id', 'year_id', 'age_group_id', 'sex_id']
    biology = ['raw_pathogen', 'raw_antibiotic', 'resistance', 'raw_specimen']
    other_values = ['cases']
    lowercase_cols = ['raw_pathogen', 'raw_antibiotic', 'raw_specimen']
    df = df[demo_cols + biology + other_values]

    # Lower case a few mapped columns
    for col in lowercase_cols:
        df[col] = df[col].str.lower().str.strip()
        assert df[col].notnull().values.all()

    df.to_csv("FILEPATH", index=False)
```