```python
import camelot
import pandas as pd

from extractors.direct_downloads import extract_with_request
from models.registry import Registry
from utils.helpers import clean_str


class BLZ(Registry):
    def __init__(self, **kwargs):
        """
        Child container for extracting, transforming, and loading registry data for Belize

        :param kwargs: see models.registry for details
        """

        [redacted]

        # assign constant values for final registry
        [redacted]

        # assign registry input data source and filename
        [redacted]

        # assign registry input data extraction method
        [redacted]
        # assign registry input data reading method
        [redacted]
        # assign registry data wrangling method
        [redacted]

        # extract, transform, and load registry data
        [redacted]

    def extract_registry(self):
        """
        Extracts registry input data from remote source

        :return: effect - creates [DATA_DIR|TEST_DIR]/[REGISTRY_INPUT_DIR]/[self.country_code]/[self.registry_filename] file
        """

        # extract registry input data
        [redacted]

    def read_registry(self):
        """
        Reads registry input data

        :return: effect - modifies self.registry
        """

        # read registry input data
        [redacted]
        # prep registry input data
        [redacted]

        # assign registry input data
        [redacted]

    def wrangle_registry(self):
        """
        Transforms registry data

        :return: effect - modifies self.wrangled_records
        """

        # loop through raw registrations
        [redacted]
            # initialize storage for wrangled representation of registration in focus
            [redacted]

            [redacted]

            # assign airframe-related values
            [redacted]

            # update list of transformed values to populate final registry
            [redacted]

    def __format_registry(self, registry):
        """
        Cleans and merges tables across multiple pdf pages

        :param (camelot.core.TableList) registry: tables of registrations from registry pdf to process
        :return: cleaned and merged transformation of 'registry' argument
        """

        # initialize storage for tables on multiple pages
        [redacted]
        # initialize storage for column names
        [redacted]

        # loop through pdf pages
        [redacted]
            # assign table on page in focus
            [redacted]

            [redacted]
                # assign column names once using the table on the first page
                [redacted]

            # drop header row from each table
            [redacted]

            [redacted]

        # merge tables across multiple pages together
        [redacted]

        [redacted]

    @staticmethod
    def __extract_colnames(header):
        """
        Parses and validates table column names that are incorrectly captured by the camelot extraction process

        :param (list) header: table column names from camelot reading method to extract values from
        :return: formatted copy of 'header' argument to label table columns
        """

        # remove invalid column names
        [redacted]

        # initialize storage for output
        [redacted]
        # parse column names erroneously concatenated to others
        [redacted]

        [redacted]

    @staticmethod
    def __format_colname(colname):
        """
        Formats column name for namedtuple compatibility

        :param (str) colname: column name from registry input data to format
        :return: formatted copy of 'colname' argument suitable for namedtuple indexing
        """

        [redacted]

        [redacted]

    @staticmethod
    def __get_manufacturer_and_model(man_mod):
        """
        Parses and formats airframe manufacturer and model

        :param (str) man_mod: registration airframe to extract values from
        :return: formatted airframe manufacturer;
            formatted airframe model
        """

        # isolate airframe components
        [redacted]

        # format airframe manufacturer
        [redacted]

        [redacted]
            # format airframe model
            [redacted]
        [redacted]
            # handle when no airframe model information exists
            [redacted]

        [redacted]
```