

```
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, date_to_iso

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

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

        # assign constant values for final registry
        self.country_code = "BGH"
        self.registry_code = "BGH"

        # assign registry input data source and filename
        self.registry_dir = "BGH"
        self.registry_filename = "BGH"

        # assign registry input data extraction method
        self.extract_registry = self.extract_registry

        # assign registry input data reading method
        self.read_registry = self.read_registry

        # assign registry data wrangling method
        self.wrangle_registry = self.wrangle_registry

        # extract, transform, and load registry data
        self.extract_registry()

    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 aircraft registry input data
        self.registry_dir = "BGH"
        self.registry_filename = "BGH"

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

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

        # read registry input data
        self.registry_dir = "BGH"
        self.registry_filename = "BGH"

        # prep registry input data
        self.registry_dir = "BGH"
        self.registry_filename = "BGH"

        # assign registry input data
        self.registry_dir = "BGH"
        self.registry_filename = "BGH"

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

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

        # loop through raw registrations
        for registration in self.registry_dir:
            # initialize storage for wrangled representation of registration in focus
            wrangled_registration = {}

            # update list of transformed values to populate final registry
            self.registry_dir = "BGH"
            self.registry_filename = "BGH"

    @staticmethod
    def __format_registry(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
        tables = []

        # initialize storage for column names
        column_names = []

        # loop through pdf pages
        for table in registry:
            # assign table on the page in focus
            table = table[0]

            # assign column names once using the table on the first page
            column_names = table.columns

            # drop header row from table on the first page
            table = table[1:]

        # merge tables across multiple pages together
        registry = pd.concat(tables)

    @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
        """

    @staticmethod
    def __convert_date(date):
        """
        Converts date to ISO format

        :param (str) date: registration date to convert
        :return: copy of 'date' argument in ISO format
        """

        # handle dates with erroneous trailing "." character
```