

```
import os
import pandas as pd
import re

from extractors.direct_downloads import extract_with_request, extract_with_headless_browser
from models.registry import Registry
from utils.helpers import clean_str, date_to_iso
```

```

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

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

        # assign constant values for final registry
        self.country_code = "FIN"
        self.country_name = "Finland"
        self.country_flag = "🇫🇮"
        self.registry_code = "FIN"

        # assign registry input data source and filename
        self.registry_dir = "data/registries/fin"
        self.registry_filename = "fin_registry_2018.csv"

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

        # assign updated-date scraping method for registry input data
        self.scrape_registry_updated = self.scrape_registry_updated

        # 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 registry input data
        self.registry_dir = self.registry_dir
        self.registry_filename = self.registry_filename

    def scrape_registry_updated(self):
        """
        Scrapes and formats updated-date for registry input data from remote source

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

        # assign updated-date for registry input data
        self.updated = self.updated

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

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

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

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

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

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

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

        # loop through raw registrations
        self.registry_dir = self.registry_dir
        self.registry_filename = self.registry_filename

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

    @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
        """

        return colname

```