import p	pandas as pd
from mod	cractors.direct_downloads import extract_with_request, extract_with_headless_browser dels.registry import Registry .ls.helpers import str_to_int, date_to_iso, build_address
	<pre>HE(Registry):    init (self, **kwargs):</pre>
dei	Child container for extracting, transforming, and loading registry data for Switzerland
	:param kwargs: see models.registry for details """
	# assign constant values for final registry
	<pre># assign registry input data source # assign registered aircraft input data filename</pre>
	# assign deregistered aircraft input data filename
	<pre># assign registration in progress aircraft input data filename # assign registry input data extraction method</pre>
	# assign updated-date scraping method for registry input data
	# assign registry input data reading method # assign registry data wrangling method
	# extract, transform, and load registry data
def	extract_registry(self): """  Extracts registry input data from remote source
	<pre>:return: effect - creates [DATA_DIR TEST_DIR]/[REGISTRY_INPUT_DIR]/[self.country_code]/[self.registered_filename] file;     effect - creates [DATA_DIR TEST_DIR]/[REGISTRY_INPUT_DIR]/[self.country_code]/[self.deregistered_filename] file;     effect - creates [DATA_DIR TEST_DIR]/[REGISTRY_INPUT_DIR]/[self.country_code]/[self.progress_filename] file; """</pre>
	# build test set if indicated
	# extract registered aircraft input data
	Microsoft Anna Control
	"Special Special Speci
	Management of the Control of the Con
	# extract deregistered aircraft input data
	# extract registration in progress aircraft input data
def	<pre># read registry input data scrape registry updated(self):</pre>
	Scrapes and formats updated-date for registry input data from remote source
	:return: effect - modifies self.updated """
	# assign updated-date for registry input data
def	<pre>read_registry(self, registered=None, deregistered=None, progress=None): """</pre>
	Reads registry input data  :param (list) registered: set of key-value pairs representing registered aircraft input data :param (list) deregistered: set of key-value pairs representing deregistered aircraft input data
	:param (list) progress: set of key-value pairs representing delegistered affected input data :return: effect - modifies self.registry """
	# handle when registry input data has been extracted but not in current session
	<pre># prep registered aircraft input data</pre> # prep deregistered aircraft input data
	# prep registration in progress aircraft input data
	# assign registry input data
def	wrangle_registry(self): """ Transforms registry data
	<pre>:return: effect - modifies self.wrangled_records """</pre>
	# loop through raw registrations # initialize storage for wrangled representation of registration in focus
	*Special Control of the control of t
	**Commanda Commanda C
	*uniform*: Boss
	# update list of transformed values to populate final registry
	# update list of transformed values to record registrant and accommodate multiple registrants
def	add_registrant_copies(self, registrants, record):
	Adds copies of a wrangled registration with updated registrant information to reflect all registrants  :param (list) registrants: registrant information data corresponding to a particular registration
	<pre>:param (dict) record: wrangled representation of a registration indicated by 'owners' argument to duplicate :return: effect - modifies self.wrangled_records """</pre>
	<pre># loop through registrants # avoid modifying original wrangled representation of registration</pre>
	# update registrant-related values
	Second Company of the Standard Company of Company of Company Company of the Second Company of the Second Company of Compa
	# update list of transformed values to populate final registry
	get_is_active(status):get_is_active(status):
	Determines whether a registration is currently active  :param (str) status: registration to assess :return: 'True' if 'status' argument represents an active registration
0	aticmethod
def	convert_date(date): """  Converts date to ISO format
	:param (dict) date: key-value pairs of registration date to convert :return: parsed registration date in ISO format
	data - Mass

@staticmethod
def \_\_get\_address(address):

or an electric

Builds human-readable address in common format

# assign country-specific common address format

address - build, address frequency, responsibil

:param (dict) address: registrant address components to utilize :return: cleaned and formatted address of registrant """