
IECA2SDMX

frapercan

19 de julio de 2022

Introduccion

1. Modulos	1
Índice de Módulos Python	9
Índice	11

class Actividad(*configuracion_global, configuracion_actividad, plantilla_configuracion_actividad, actividad*)

Una actividad es definida por una lista de consultas a través de su ID en el fichero de configuración “**actividades.yaml**”.

Esta clase ejecutará las consultas y se encargará de hacer las transformaciones pertinentes al grupo completo para su correcta modelización en el estandar SDMX.

Parámetros

- **configuracion_global** (Diccionario) – Configuración común a todas las ejecuciones que se realicen.
- **configuracion_actividad** (Diccionario) – Configuración común para toda la actividad.
- **plantilla_configuracion_actividad** (Diccionario) – Configuración por defecto de la actividad. Este fichero de configuración extiende a **configuracion_actividad** con las configuraciones que no están explícitamente recogidas en este.
- **actividad** (Cadena de Texto) – Nombre de la actividad.

consultas

Diccionario que contiene las consultas con los datos y metadatos, cuya clave serán los **src.consulta.Consulta.id_consulta** correspondientes.

Type

Diccionario de **src.consulta.Consulta**

agrupar_consultas_SDMX()

Tras realizar las consultas, esta función genera un fichero de configuración en el sistema de información con respecto a la actividad completa, agrupando las consultas de la actividad por su título. De esta forma un grupo de consultas formaran un fichero de datos .CSV que contiene a todas. Los datos se guardarán siguiendo dos procedimientos:

- **EXTENSION_DISJUNTOS**: Estructura de datos (DSD) común para toda la actividad donde las columnas faltantes son añadidas y rellenas con la variable **_Z**

- **ORIGINAL:** Estructura de datos (DSD) por cada grupo de consulta. Si las consultas con el mismo título tienen dimensiones distintas, se creará una estructura común para todas las consultas y se mostrará una advertencia por consola.

comprobar_dimensiones_grupo_actividad(*columnas_grupo, grupo*)

Comprueba el modelado por títulos en BADEA y muestra por pantalla advertencias sobre las dimensiones para facilitar su depuración.

Parámetros

- **columnas_grupo** (Lista de `DataFrame.columns`) – Listado de columnas de cada consulta del grupo.
- **grupo** (Cadena de Texto) – Título del grupo

ejecutar()

Aplica las funciones configuradas en el fichero de configuración “**actividades.yaml**” bajo la clave **acciones_actividad_completa**.

generar_consultas()

Inicializa y ejecuta las consultas a la API de BADEA dentro del diccionario *consultas*.

class Consulta(*id_consulta, configuracion_global, configuracion_actividad, actividad*)

Este objeto al inicializarse consultará la API del IECA utilizando *id_consulta*. Esta clase se encargará de generar las estructuras de datos y metadatos y de ser su recipiente. Las medidas se tratarán como parte de una dimensión SDMX llamada **INDICATOR** y se manejarán dentro de la clase `src.datos.Datos`.

Parámetros

- **id_consulta** (Cadena de Texto) – ID de la consulta que se va a procesar.
- **configuracion_global** (Diccionario) – Configuración común a todas las ejecuciones que se realicen.
- **configuracion_actividad** (Diccionario) – Configuración común para toda la actividad.
- **actividad** (Cadena de Texto) – Nombre de la actividad.

id_consulta

Type

Cadena de Texto

metadatos

Metainformación de la consulta con los siguientes campos clave:

- id
- title
- subtitle
- activity
- source
- periodicity
- type
- notes

Type

Diccionario

jerarquias

Jerarquias utilizadas en los datos de la consulta

TypeLista de `src.jerarquia.Jerarquia`**datos**

Datos proporcionados en la consulta.

Type`src.datos.Datos`**ejecutar()**

Aplica las funciones configuradas en el fichero de configuración “**actividades.yaml**” bajo las claves **acciones_jerarquia** y ****acciones_datos***.

solicitar_informacion_api()

Utilizando *id_consulta* busca el JSON de la consulta en local, y si no, le manda la petición a la API del IECA. Si se ha alcanzado la API, se guarda el JSON para acelerar futuras consultas y no sobrecargar el sistema. Hemos de tener esto en cuenta, en caso de que las consultas de la API no sean inmutables.

Devuelve

- `metainfo` (Diccionario)
- `hierarchies` (Diccionario)
- `measures` (Diccionario)
- `data` (Diccionario)

class Jerarquia(*jerarquia, configuracion_global, actividad*)

Estructura de datos para manejar las jerarquias encontradas dentro de las consultas del IECA, es necesario hacer una petición HTTP para expandir la jerarquia y traernos los valores para generar las listas de código que usaremos en SDMX.

Parámetros

- **jerarquia** (Diccionario) – Información resumida de la jerarquia obtenida en una consulta anterior.
- **configuracion_global** (Diccionario) – Configuración común a todas la ejecución.
- **actividad** (Cadena de Texto) – Nombre de la actividad.

id_jerarquia

concatenación del alias y el código de la jerarquia.

Type

Cadena de Texto

metadatos

Metainformación de la jerarquia con los siguientes campos clave:

- `url`
- `cod` (Codificación)
- `des` (Descripción)

- position (En desuso)
- order (En desuso)
- alias (Las jerarquias pueden tener distintos datos, usamos estos alias para concretizarlos)
- levels (En desuso)

Type

Diccionario

datos

La jerarquia en un cuadro de datos que posteriormente puede ser exportada a .CSV para importarse en SDMX.

Type`pandas.DataFrame`**convertir_jerarquia_a_dataframe(*datos_jerarquia*)**

Transforma el diccionario con los datos de la jerarquia a formato tabular, borrando los valores con Código duplicado además de añadir el valor **_Z**.

Devuelve

La jerarquia en un cuadro de datos.

Tipo del valor devuelto

datos (`pandas.DataFrame`)

guardar_datos()

Accion que guarda la jerarquia en formato .CSV de dos formas:

- Con el Còdigo de BADEA (No admitido por nuestro framework de SDMX)
- Sin el código de BADEA (Admitido por nuestro framework de SDMX)

La salida de la lista de código para importar en SDMX se deshará de los valores de las jerarquias que no han sido utilizados en la consulta.

solicitar_informacion_jerarquia()

Realiza la petición HTTP a la API si la jerarquía no se encuentra en nuestro directorio local, automáticamente se convierte la jerarquia a dataframe haciendo uso de `src.jerarquia.Jerarquia.convertir_jerarquia_a_dataframe`.

Devuelve

La jerarquia en un cuadro de datos.

Tipo del valor devuelto

datos (`pandas.DataFrame`)

class Datos(*id_consulta, configuracion_global, actividad, periodicidad, datos, jerarquias, medidas*)

Estructura de datos para manejar los datos encontrados dentro de las consultas del IECA. El proceso de inicialización de esta estructura realiza los siguientes acciones de forma configurable:

- Convierte de JSON a DataFrame utilizando las medidas y jerarquias para generar las dimensiones/columnas.
- Desacopla las observaciones en base a las medidas utilizando la dimension **INDICATOR**
- Añade la dimension **FREQ** de SDMX en base a la periodicidad de la consulta.

Parámetros

- **id_consulta** (Cadena de Texto) – ID de la consulta que se va a procesar.
- **configuracion_global** (Diccionario) – Configuración común a todas las ejecuciones que se realicen.
- **actividad** (Cadena de Texto) – Nombre de la actividad.
- **periodicidad** (Cadena de Texto) – Periodicidad de las observaciones.
- **datos** (Diccionario) – Datos de la consulta.
- **jerarquias** (Lista de src.jerarquia.Jerarquia) – Jerarquias de la consulta
- **medidas** (src.consulta.medidas) – Medidas de la consulta

datos

Los datos en un cuadro de datos

Type

`pandas.DataFrame`

datos_por_observacion

Los datos desacoplados por medidas en columnas

Type

`pandas.DataFrame`

datos_por_observacion_extension_disjuntos

Los datos desacoplados pero con todas las columnas necesarias para crear un DSD para toda la actividad.

Type

`pandas.DataFrame`

borrar_datos_duplicados()

Accion que borra las filas duplicadas sin tener en cuenta **OBS_VALUE**.

borrar_filas(dics_columna_valor_a_borrar)

Accion que elimina del cuadro de datos las filas que contengan los valores proporcionados.

Parámetros

dics_columna_valor_a_borrar (Lista de Cadena de Texto) – Lista de diccionarios con cuyo par clave-valor es la columna-valor deseado para la eliminación.

convertir_datos_a_dataframe_sdmx(datos)

Transforma el diccionario con los datos de las observaciones a formato tabular válido para SDMX de la siguiente forma:

1. Recorremos el diccionario accediendo de forma ordenada a las dimensiones correspondientes tomando las jerarquias y las medidas como referencia.
2. Los datos en el JSON están indexados por “COD” en lugar de por “ID” que sería más apropiado, aquí cruzamos la información pertinente para mapear COD -> ID.
3. Rellenamos la dimension **FREQ** de SDMX en base a la periodicidad de la consulta.

Parámetros

datos (Diccionario) – Datos de la consulta.

Devuelve

Las observaciones en un cuadro de datos con la forma tabular original del modelado hecho en BADEA.

Tipo del valor devueltodatos (`pandas.DataFrame`)**desacoplar_datos_por_medidas()**

El formato tabular proporcionado por la API tiene una dimension para cada medida, nuestro modelado SDMX consistirá en crear una dimension **INDICATOR** cuyo valor será la medida en sí.

Esto quiere decir que nuestro cuadro de datos tendrá menos columnas, pero mayor número de filas (Numero de observaciones * Numero de medidas).

Parámetros**datos** (`pandas.DataFrame`) – Datos de la consulta en un cuadro de datos.**Devuelve**

Las observaciones en un cuadro de datos adaptado al estandar SDMX.

Tipo del valor devueltodatos (`pandas.DataFrame`)**extender_con_disjuntos**(*dimensiones*)

Accion que añade las columnas al dataframe para que todas las dimensiones SDMX de la actividad sean contempladas en un mismo DSD, añadiendo el valor **_Z** a estas nuevas columnas.

Parámetros**dimensiones** (Lista de Cadena de Texto) – Lista de dimensiones únicas que se encuentran en el conjunto de la actividad sobre la que estamos trabajando.**extender_mapa_nuevos_terminos()**

Accion que crea/extiende el mapa para las columnas configuradas facilitando al técnico realizar la conversión y su posterior reutilización en distintas actividades.

guardar_datos(*clase*)

Accion que guarda la jerarquia en formato .CSV de dos formas bifurcando en directorios a traves del argumento clase:

- Con el Còdigo de BADEA (No admitido por nuestro framework de SDMX)
- Sin el código de BADEA (Admitido por nuestro framework de SDMX)

Parámetros**clase** (Cadena de Texto) – Nombre para el directorio en el que se almacenaran los datos.**mapear_columnas()**

Accion que realiza el mapeo de las columnas del cuadro de datos configuradas bajo el parámetro observaciones utilizando el mapa previamente relleno a través del campo `mapear_columnas` del fichero de configuracion global. Por último limpiamos las jerarquias de los prefijos y sufijos.

mapear_valores()

Accion que realiza el mapeo de los valores del cuadro de datos configuradas bajo el parámetro `dimensiones_a_mapear` observaciones utilizando los mapas previamente rellenos del fichero de configuracion global.

sumar_datos_duplicados()

Accion que agrupa los datos duplicados y devuelve los datos agregados de **OBS_VALUE**.

insertar_freq(*df, periodicidad*)

Añade los valores a la columna “**FREQ**” dependiendo de un mapa simple de periodicidad.

Parámetros

- **df** (`pandas.DataFrame`) – Cuadro de datos al que añadir la frecuencia.
- **periodicidad** (Cadena de Texto) – Periodicidad de la consulta.

transformar_formato_tiempo_segun_periodicidad(*serie, periodicidad*)

Transforma la dimension temporal de un cuadro de datos para que se adecue al formato de tiempo utilizado en SDMX.

Parámetros

- **serie** (`pandas.Series`) – Serie perteneciente al cuadro de datos a transformar
- **periodicidad** (Cadena de Texto) – Periodicidad de la consulta.

S

`src.ieca.actividad`, 1
`src.ieca.consulta`, 2
`src.ieca.datos`, 4
`src.ieca.jerarquia`, 3

A

Actividad (*clase en src.ieca.actividad*), 1
 agrupar_consultas_SDMX() (*método de Actividad*), 1

B

borrar_datos_duplicados() (*método de Datos*), 5
 borrar_filas() (*método de Datos*), 5

C

comprobar_dimensiones_grupo_actividad() (*método de Actividad*), 2
 Consulta (*clase en src.ieca.consulta*), 2
 consultas (*atributo de Actividad*), 1
 convertir_datos_a_dataframe_sdmx() (*método de Datos*), 5
 convertir_jerarquia_a_dataframe() (*método de Jerarquía*), 4

D

datos (*atributo de Consulta*), 3
 datos (*atributo de Datos*), 5
 datos (*atributo de Jerarquía*), 4
 Datos (*clase en src.ieca.datos*), 4
 datos_por_observacion (*atributo de Datos*), 5
 datos_por_observacion_extension_disjuntos (*atributo de Datos*), 5
 desacoplar_datos_por_medidas() (*método de Datos*), 6

E

ejecutar() (*método de Actividad*), 2
 ejecutar() (*método de Consulta*), 3
 extender_con_disjuntos() (*método de Datos*), 6
 extender_mapa_nuevos_terminos() (*método de Datos*), 6

G

generar_consultas() (*método de Actividad*), 2
 guardar_datos() (*método de Datos*), 6

guardar_datos() (*método de Jerarquía*), 4

I

id_consulta (*atributo de Consulta*), 2
 id_jerarquia (*atributo de Jerarquía*), 3
 insertar_freq() (*en el módulo src.ieca.datos*), 6

J

Jerarquía (*clase en src.ieca.jerarquia*), 3
 jerarquias (*atributo de Consulta*), 3

M

módulo
 src.ieca.actividad, 1
 src.ieca.consulta, 2
 src.ieca.datos, 4
 src.ieca.jerarquia, 3
 mapear_columnas() (*método de Datos*), 6
 mapear_valores() (*método de Datos*), 6
 metadatos (*atributo de Consulta*), 2
 metadatos (*atributo de Jerarquía*), 3

S

solicitar_informacion_api() (*método de Consulta*), 3
 solicitar_informacion_jerarquia() (*método de Jerarquía*), 4
 src.ieca.actividad
 módulo, 1
 src.ieca.consulta
 módulo, 2
 src.ieca.datos
 módulo, 4
 src.ieca.jerarquia
 módulo, 3
 sumar_datos_duplicados() (*método de Datos*), 6

T

transformar_formato_tiempo_segun_periodicidad() (*en el módulo src.ieca.datos*), 7