



Объекты ETL-пайплайна ChEMBL/Assay

Класс/Объект	Предок	Модуль	Название
ChembIAssayPipeline	ChembICommonPipeline	<code>src/bioetl/pipelines/chembl/assay/run.py</code>	Основной пайплайн для обработки данных ChEMBL на этапе assay. Включает валидацию и первичную обработку, а также валидацию и обработка данных.
ChembICommonPipeline	ChembIPipelineBase	<code>src/bioetl/pipelines/chembl/common/base.py</code>	Базовый класс для ChEMBL (обобщенный legacy). Определяет стадии обработки, через которые проходит доменное валидирование. Содержит результаты.

Класс/Объект	Предок	Модуль	Название
ChEMLPipelineBase	PipelineBase	<code>src/bioetl/core/pipeline/unified.py</code>	Абстрактный класс ChEML для доменов извлечения. Содержит общие предикты для доменов извлечения. Умоляет исходные данные также в ChEML для з...
ChEMLExtractionService	object	<code>src/bioetl/pipelines/chembl/common/chembl_extraction_service.py</code>	Сервис общего извлечения ChEML, определяющий релизы постреликовых запросов функций (fetchers), данные пакетов извлечения ChEML.

Класс/Объект	Предок	Модуль	Название
ChEMBLExtractionServiceDescriptor	Generic[ChEMBLPipelineBase]	<code>src/bioetl/core/pipeline/unified.py</code>	Описывает извлечение ChEMBL подхвата фабрики build_контекст fetcher (возврат для пакета батчу finalizer (возврат для файла обработки)
ChEMBLExtractionDescriptor	-	<code>src/bioetl/pipelines/chembl/common/descriptor.py</code>	Лёгкий дескриптор ChEMBL, какие списки, полностью параметры, режимы или "разбивка" релиза, инициализации, выборки
BatchPlan	-	<code>src/bioetl/clients/chembl/descriptor_factory.py</code>	DataSources плана при выполнении Содействие (batch) чанка построения извлечения

Класс/Объект	Предок	Модуль	Название
ChembIDescriptorFactory	-	src/bioetl/clients/chembl/descriptor_factory.py	Фабрика ChEML конфигурации фасада (Chembl) набора получения (fetch) по типу функции фолда ошибки (failure) также для дескрипторов Использование пайплайна постройки через
ChembIContextFacade	-	src/bioetl/clients/chembl/descriptor_factory.py	"Фасад ChEML для пакета дескрипторов необходимо компоненты трансформации пагинации набора пагинации релиза Chembl либо
ChembIClient	ConfiguredHttpClient	src/bioetl/clients/chembl/client.py	API-клиент (новый) Наследует клиент методы форматы к ChEML инициализации загрузки источников Chembl для с

Класс/Объект	Предок	Модуль	Название
ChEMBLRequestBuilder	-	src/bioetl/clients/chembl/client.py	Класс запро 48 . И логик объект (марш фильтр пагинац Исполн Chem получен запро
RequestsBackend	HttpBackend (Protocol)	src/bioetl/clients/chembl/factories.py	Реализа бэкенда базе т реции за вы запро ChEMBL методы одной итери записи также содер
AssaySchema	pa.DataFrameSchema	src/bioetl/core/schemas/assay_schema.py	Схема ассая DataFrame Опреде обязат (assay др.) и испол провед привес данные транс

Класс/Объект	Предок	Модуль	Название
AssayNormalizer	-	<code>src/bioetl/pipelines/chembl/assay/normalizers.py</code>	Нормализатор для очистки assay для DataFrames
AssayPayloadParser	-	<code>src/bioetl/pipelines/chembl/assay/parsers.py</code>	Парсер сырой assay связанных структур (DataFrames)
ChEMBILWriteService	WriteService (Protocol)	<code>src/bioetl/pipelines/chembl/common/base.py</code>	Специализированные сервисы ChEMBL. Реализует детерминированное сохранение форматов выходных данных (датафреймы, метаданные, супфиксные файлы), сохраняющие CSV и JSON форматы, запускающие

Класс/Объект	Предок	Модуль	Название
PipelineOutputService	-	<code>src/bioetl/core/io/output_service.py</code>	Сервис для загрузки пайплайна вх артеф настрой конф вызы атома датас успеш генера (при г emit Испол Chem сохра использу или о Chem
WriteArtifacts	-	<code>src/bioetl/core/io/artifacts.py</code>	Струк путей пайпл поля датас data данные qual (CSV с качес (YAML запус мани Заполн испол перед запис

1 2 59 **run.py**

https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/pipelines/chembl/assay/run.py

3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 55 56 57 **base.py**

https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/pipelines/chembl/common/base.py

21 22 23 24 25 26 41 **unified.py**

https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/core/pipeline/unified.py

27 28 29 30 31 32 33 34 35 36 37 38 39 40 **chembl_extraction_service.py**

https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/pipelines/chembl/common/chembl_extraction_service.py

42 **descriptor.py**

https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/pipelines/chembl/common/descriptor.py

43 44 45 **descriptor_factory.py**

https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/clients/chembl/descriptor_factory.py

46 47 48 **client.py**

https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/clients/chembl/client.py

49 50 51 **factories.py**

https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/clients/chembl/factories.py

52 **assay_schema.py**

https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/core/schemas/assay_schema.py

53 **normalizers.py**

https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/pipelines/chembl/assay/normalizers.py

54 **parsers.py**

https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/pipelines/chembl/assay/parsers.py

58 60 61 **output_service.py**

https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/core/io/output_service.py

62 **artifacts.py**

https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/core/io/artifacts.py