



Класс

Предок

Модуль

Назначение

ChEMBLActivityPipeline

PipelineBase

```
src/bioetl/  
pipelines/chembl/  
activity/run.py
```

Реализация пайплайна выгрузки активности из ChEMBL ¹. Использует дескриптор для извлечения данных и реализует полный цикл ETL для сущности "activity".

ChEMBLAssayPipeline

PipelineBase

```
src/bioetl/  
pipelines/chembl/  
assay/run.py
```

Каркас пайплайна для выгрузки данных об биоактивности типа "assay" ². Наследует общий функционал ChEMBL-пайплайнов и переопределяет точки расширения для специфики assay.

Класс	Предок	Модуль	Назначение
ChEMBLTargetPipeline	PipelineBase	<code>src/bioetl/ pipelines/chembl/ target/run.py</code>	Каркас пайплайна для сущности “target” в ChEMBL с обогащением данными UniProt/IUPHAR ³ . Выполняет стандартный ETL-процесс для target, добавляя проверку на заполненность идентификаторов.
ChEMBLDocumentPipeline	PipelineBase	<code>src/bioetl/ pipelines/chembl/ document/run.py</code>	Скелет пайплайна для сущности “document” ChEMBL с дополнительным обогащением из внешних источников ⁴ . Реализует стандартные стадии ETL и дополняет этап преобразования внешними данными.
TestItemChEMBLPipeline	PipelineBase	<code>src/bioetl/ pipelines/chembl/ testitem/run.py</code>	Скелет пайплайна “testitem” для тестовых веществ, включает обогащение данными PubChem ⁵ . Реализует ETL конвейер с кастомной трансформацией (обогащение через PubChem).
ActivityExtractor	StageABC	<code>src/bioetl/ pipelines/chembl/ activity/stages.py</code>	Класс этапа извлечения данных активности из API ChEMBL ⁶ . Вызывает клиент ChEMBL батчами и формирует результирующий DataFrame с метаданными.

Класс	Предок	Модуль	Назначение
ActivityTransformer	StageABC	<code>src/bioetl/pipelines/chembl/activity/stages.py</code>	Класс этапа трансформации для активности ChEMBL ⁷ . Нормализует и обогащает извлечённые данные (например, добавляет номер релиза ChEMBL).
ActivityWriter	StageABC	<code>src/bioetl/pipelines/chembl/activity/stages.py</code>	Класс этапа сохранения результатов пайплайна активности ⁸ . Записывает финальный набор данных на диск и генерирует артефакты контроля качества (QC).
ActivityWriteService	WriteService	<code>src/bioetl/pipelines/chembl/activity/run.py</code>	Сервис вывода данных для пайплайна активности ⁹ . Оборачивает ActivityWriter в интерфейс WriteService для совместимости с оркестрацией.
ChEMBLWriteService	WriteService	<code>src/bioetl/pipelines/chembl/common/base.py</code>	Сервис детерминированной записи для всех ChEMBL-пайплайнов ¹⁰ . Выполняет сохранение основного датасета, метаданных и отчёта качества с единообразным именованием.

Класс	Предок	Модуль	Назначение
ConfiguredHttpClient	BaseClient	<code>src/bioetl/clients/factory.py</code>	Базовая реализация клиента данных, настроенного конфигурацией источника и HTTP-бэкендом ¹¹ . Делегирует выполнение запросов объекту <code>HttpBackend</code> .
Chemblient	BaseClient	<code>src/bioetl/clients/chembl/client.py</code>	Высокоуровневый клиент API ChEMBL, использующий унифицированный контракт. Наследует всю базовую функциональность <code>BaseClient</code> через <code>ConfiguredHttpClient</code> ¹² , добавляя удобные методы для запросов ChEMBL.
PubMedClient	BaseClient	<code>src/bioetl/clients/pubmed/client.py</code>	Клиент для API PubMed на базе унифицированного интерфейса. Наследует <code>ConfiguredHttpClient</code> (то есть <code>BaseClient</code>) и предоставляет фабричный метод формирования запроса для статей PubMed ¹³ .
CrossrefClient	BaseClient	<code>src/bioetl/clients/crossref/client.py</code>	Клиент для API CrossRef, реализующий общий контракт клиентов данных. Наследует базовую реализацию и предоставляет метод для формирования запроса по рабочим записям (works) ¹⁴ .

Класс	Предок	Модуль	Назначение
SemanticScholarClient	BaseClient	<code>src/bioetl/clients/semantic_scholar/client.py</code>	Клиент для API Semantic Scholar, реализованный на базе <code>BaseClient</code> . Наследует типовой HTTP-клиент и определяет источник (<code>source</code>) как "semantic_scholar" ¹⁵ .
PubChemClient	BaseClient	<code>src/bioetl/clients/pubchem/client.py</code>	Клиент для PubChem, предоставляющий доступ к данным соединений. Реализован через общий <code>ConfiguredHttpClient</code> для унификации работы с REST API ¹⁶ .
RequestsBackend	HttpBackend	<code>src/bioetl/clients/chembl/factories.py</code>	Реализация транспортного HTTP-бэкенда с использованием библиотеки <code>requests</code> ¹⁷ . Отвечает за выполнение HTTP-запросов для клиентов данных.
UnifiedOutputWriter	WriterABC (UnifiedOutputWriter Protocol)	<code>src/bioetl/core/io/output.py</code>	Унифицированный писатель вывода данных пайплайна. Сохраняет результирующий DataFrame в CSV или Parquet, вычисляет хеши и записывает метаданные запуска (манифест, мета-файл) ¹⁸ ¹⁹ .

Класс	Предок	Модуль	Назначение
InMemoryTTLCacheImpl	CacheABC (CacheStrategy)	<code>src/bioetl/core/http/cache.py</code>	Потокобезопасный кэш HTTP-ответов с истечением (TTL) в памяти (либо файле) ²⁰ . Используется для кэширования результатов API вызовов в течение заданного времени.
TokenBucketRateLimiterImpl	RateLimiterABC	<code>src/bioetl/core/http/rate_limiter.py</code>	Реализация лимитера частоты запросов с алгоритмом токен-бакета ²¹ . Ограничивает скорость обращений к внешним API, поддерживает блокирующее и неблокирующее получение токена.
ExponentialBackoffRetryImpl	RetryPolicyABC	<code>src/bioetl/core/http/retry.py</code>	Реализация стратегии повторных попыток с экспоненциальной задержкой. Вычисляет интервал перед следующей попыткой с учётом случайного джиттера и максимального порога задержки ²² .
CircuitBreakerImpl	CircuitBreakerStrategyABC	<code>src/bioetl/core/http/circuit_breaker.py</code>	Реализация шаблона Circuit Breaker для устойчивости HTTP-запросов. Отслеживает число ошибок, открывает "предохранитель" при достижении порога и автоматически сбрасывается по таймеру ²³ ²⁴ .

Класс	Предок	Модуль	Назначение
DefaultPaginationStrategy	PaginatorABC	<code>src/bioetl/core/ http/pagination.py</code>	Простая стратегия постраничной навигации по API ²⁵ . Использует параметр "page" и поле ответа "next" для последовательного получения всех страниц результатов.
EnvSecretProvider	SecretProviderABC	<code>src/bioetl/core/ config/ config_resolver.py</code>	Провайдер секретов, основанный на переменных окружения и файлах <code>.env</code> ²⁶ . Обеспечивает доступ к секретным настройкам (пароли, ключи) и переменным окружения для конфигурации пайплайна.
FileConfigResolver	ConfigResolverABC	<code>src/bioetl/core/ config/ config_resolver.py</code>	Реализация резолвера конфигурации пайплайна из YAML-файлов ²⁷ . Загружает основной конфиг и профильные слоя, сливает их и применяет переопределения окружения и переменных для получения окончательного <code>PipelineConfig</code> .

1 9 `run.py`

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

2 `run.py`

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

3 **run.py**

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

4 **run.py**

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

5 **run.py**

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

6 7 8 **stages.py**

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

10 **base.py**

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

11 **factory.py**

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

12 **client.py**

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

13 **client.py**

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

14 **client.py**

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

15 **client.py**

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

16 **client.py**

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

17 **factories.py**

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

18 19 **output.py**

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

20 **cache.py**

https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/core/http/cache.py

21 **rate_limiter.py**

https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/core/http/rate_limiter.py

22 **retry.py**

https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/core/http/retry.py

23 24 **circuit_breaker.py**

https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/core/http/circuit_breaker.py

25 **pagination.py**

https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/core/http/pagination.py

26 27 **config_resolver.py**

https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/core/config/config_resolver.py