

Класс	Предок	Модуль	Назначение
<b>ChemblActivityPipeline</b>	PipelineBase	<code>src/bioetl/ pipelines/chembl/ activity/run.py</code>	Реализация пайплайна выгрузки активности из ChEMBL <sup>1</sup> . Использует дескриптор для извлечения данных и реализует полный цикл ETL для сущности "activity".
<b>ChemblAssayPipeline</b>	PipelineBase	<code>src/bioetl/ pipelines/chembl/ assay/run.py</code>	Каркас пайплайна для выгрузки данных об биоактивности типа "assay" <sup>2</sup> . Наследует общий функционал ChEMBL-пайплайнов и переопределяет точки расширения для специфики assay.

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

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

Класс	Предок	Модуль	Назначение
<b>ConfiguredHttpClient</b>	BaseClient	<code>src/bioetl/clients/factory.py</code>	Базовая реализация клиента данных, настроенного конфигурацией источника и HTTP-бэкендом <sup>11</sup> . Делегирует выполнение запросов объекту <code>HttpBackend</code> .
<b>ChEMBLClient</b>	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> <sup>12</sup> , добавляя удобные методы для запросов ChEMBL.
<b>PubMedClient</b>	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 <sup>13</sup> .
<b>CrossrefClient</b>	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) <sup>14</sup> .

Класс	Предок	Модуль	Назначение
<b>SemanticScholarClient</b>	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" <sup>15</sup> .
<b>PubChemClient</b>	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 <sup>16</sup> .
<b>RequestsBackend</b>	HttpBackend	<code>src/bioetl/clients/chembl/factories.py</code>	Реализация транспортного HTTP-бэкенда с использованием библиотеки <code>requests</code> <sup>17</sup> . Отвечает за выполнение HTTP-запросов для клиентов данных.
<b>UnifiedOutputWriter</b>	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) <sup>18</sup> <sup>19</sup> .

Класс	Предок	Модуль	Назначение
<b>InMemoryTTLCacheImpl</b>	CacheABC (CacheStrategy)	<code>src/bioetl/core/http/cache.py</code>	Потокобезопасный кэш HTTP-ответов с истечением (TTL) в памяти (либо файле) <sup>20</sup> . Используется для кэширования результатов API вызовов в течение заданного времени.
<b>TokenBucketRateLimiterImpl</b>	RateLimiterABC	<code>src/bioetl/core/http/rate_limiter.py</code>	Реализация лимитера частоты запросов с алгоритмом токенов-бакета <sup>21</sup> . Ограничивает скорость обращений к внешним API, поддерживает блокирующее и неблокирующее получение токена.
<b>ExponentialBackoffRetryImpl</b>	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 <sup>22</sup> .
<b>CircuitBreakerImpl</b>	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 <sup>23</sup> <sup>24</sup> .

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

<sup>1</sup> <sup>9</sup> `run.py`

[https://github.com/SatoryKono/bioactivity\\_data\\_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/pipelines/chembl/activity/run.py](https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/pipelines/chembl/activity/run.py)

<sup>2</sup> `run.py`

[https://github.com/SatoryKono/bioactivity\\_data\\_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/pipelines/chembl/assay/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](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](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](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](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](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](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](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](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](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](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](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](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](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](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](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](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](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](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](https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/core/config/config_resolver.py)