



Класс

Предок

Модуль

Назначе

ChembITargetPipeline

ChembICommonPipeline

1

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

Основно
сущности
(включая
UniProt/I
общий С
определя
для тарг

Класс	Предок	Модуль	Назначение
-------	--------	--------	------------

ChembICommonPipeline

ChembIPipelineBase 8

[src/bioetl/pipelines/chembl/common/base.py](#)

Общий базовый класс для всех пайплайнов. Включает в себя проверку и создание записей и стандартов.

Класс	Предок	Модуль	Назначе
-------	--------	--------	---------

ChEMBLPipelineBase

PipelineBase 25

`src/bioetl/core/pipeline/
unified.py`

Базовый
ChEMBL,
выполне
специаль
извлечено
ChembIEx
выполне
из API.

Класс	Предок	Модуль	Назначение
ChEMBLExtractionServiceDescriptor	Generic[ChEMBLPipelineT] (типовoy класс) <small>38</small>	src/bioetl/core/pipeline/unified.py	Дескриптор параметров сущности, используемый для формирования запроса, возвращающего результаты. Использует <code>ServiceExtractor</code> , определенный в данном модуле.
ChEMBLExtractionDescriptor	object (dataclass) <small>42</small>	src/bioetl/pipelines/chembl/common/descriptor.py	Облегченный извлеченный класс, использующий стратегии идентификации параметров (<code>mode</code> , например, "all") и пакетные (батчевые) конвейеры <code>dataclass</code> (например, <code>Descriptor</code>).
ServiceExtractionStrategy	—	src/bioetl/pipelines/chembl/common:strategies.py	Стратегия извлечения сервисов. Поддерживает "service" (извлечение), "run_descriptors" (у пайплайна) и "descriptor" (дескриптор).

Класс	Предок	Модуль	Назначение
DataclassExtractionStrategy	—	<code>src/bioetl/pipelines/chembl/common/strategies.py</code>	Стратегии через <code>dataclass</code> Поддержка "dataclass" кастомные определения dataclass (например)
ExtractionStrategyFactory	—	<code>src/bioetl/pipelines/chembl/common/strategies.py</code>	Фабрика извлечения инициализации набор стратегий умолчания и Service возвращение поддержка типа дескрипторов условных выражений

Класс	Предок	Модуль	Назначение
ChEMBLExtractionService	—	<code>src/bioetl/pipelines/chembl/ common/ chembl_extraction_service.py</code>	Сервис, использующий общие шаблоны для обработки данных ChEMBL в контексте нормализации. Выбор параметров (клиентской конфигурации) определяется ChEMBL. Извлеченные данные используются для выполнения

Класс	Предок	Модуль	Назначение
ChembIDescriptorFactory	dataclass	src/bioetl/clients/chembl/descriptor_factory.py	<p>Фабрика дескрипторов Инкапсуляция <i>ChembI/Chebi</i> подключений параметров фабрики словаря данных</p> <p><code>build(chembl_id)</code> (реализация builder) в сконфигурированном экземпляре <i>ChembI/Chembl</i> или анализа указанных</p>
TargetSchema	pa.DataFrameSchema (Pandera) ⁷⁸	src/bioetl/core/schemas/target_schema.py	<p>Pandera- сущности структур данных т обязательные (<code>target</code>, организации business, business, <code>row_has</code>) условия <code>target_</code> ⁷⁸ ⁷⁹. проверки данных на</p>
DefaultValidationService	—	src/bioetl/core/pipeline/services/defaults.py	<p>Сервис валидации по схеме объектов обеспечения DataFrame этой схемы сортировки детерминированы Использует реализации</p>

Класс	Предок	Модуль	Назначение
ChEMBLWriteService	—	<code>src/bioetl/pipelines/chembl/common/base.py</code>	<p>Сервис записи ChEMBL-данных в детерминированные именованные CSV-файлы с названием, качеством, метаданными, затем со CSV и загружает сопутствующий В режиме вызывает запись в пайплайн</p>
PipelineOutputService	—	<code>src/bioetl/core/io/output_service.py</code>	<p>Сервис-указатель результатов, использует унифицированную конфигурацию, настроенную UnifiedOutput конфигуратором (<code>resolver</code>) и задан - DataFrame артефакт бросает настройки пайплайна ChemBLTools, используя атомарные</p>

Класс	Предок	Модуль	Назначение
WriteArtifacts	dataclass	<code>src/bioetl/core/io/artifacts.py</code>	Структур хранения файлов Г поля: <code>data</code> ОСНОВНОГО <code>data_ра</code> данных), YAML-фа <code>manifest</code> манифес <code>quality</code> CSV файл <code>qc_summ</code> сводки Q для допо артефакт заполнян сохранен (наприме формиру
ChembIClient	ConfiguredHttpClient <small>95</small>	<code>src/bioetl/clients/chembl/client.py</code>	Клиент д унифици архитекту базовый (<code>ConfiguredHttpClient</code>) обеспечи запросов (наприме с поддер фильтров инициал конфигур ChEMBL и <code>HttpBackend</code> Requests выполне

Класс	Предок	Модуль	Назначение
ChEMBLRequestBuilder	—	<code>src/bioetl/clients/chembl/client.py</code>	Вспомогательный класс, формирующий запросы к ChEMBL API в соответствии с dataclass-конфигуратором (<code>SourceConfig</code>). Маршруты и объекты СУБД выполняются в 101 . Используется для выполнения запросов на ресурсы ChEMBL.
UniProtClient	ConfiguredHttpClient 104	<code>src/bioetl/clients/uniprot/client.py</code>	Клиент для работы с UniProt API. Реализован на основе архитектуры <code>HttpBackend</code> . Наследует класс <code>ConfiguredHttpClient</code> , аналогичный созданному для источника ChEMBL. Предоставляет методы для формирования и отправки запросов на получение данных из UniProt.

Класс	Предок	Модуль	Назначение
RequestsBackend	HttpBackend 108	<code>src/bioetl/clients/chembl/factories.py</code>	<p>Реализация клиента на основе библиотеки requests. Предоставляет низкоуровневые выполнения API ChEMBL и получения ответа. Использует умолчания (и аналогичные фактические) HTTP-запросов. Реализует единичный интерфейс по записи и получению, закрытию соединения.</p>
ClientFactoryRegistry	—	<code>src/bioetl/pipelines/client_registry.py</code>	<p>Реестр фабрик клиентов, используемый для хранения (ClientFactory) (namespace) "chembl". Клиенты в реестре инициализируются pipeline chembl_factory в реестре, получая конкретные get("client") дескрипторы.</p>

Класс	Предок	Модуль	Назначение
TargetNormalizer	—	<code>src/bioetl/pipelines/chembl/target/normalizers.py</code>	Класс-шаблон для нормализации целевых молекул на основе словарей. Предполагается, что класс является стандартным (например, организовано извлечение), однако требует заполнения NotImplen...

1 2 3 4 5 6 7 **run.py**

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

8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 85 86 87 88 **base.py**

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

25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 **unified.py**

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

42 43 **descriptor.py**

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

44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 **strategies.py**

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

60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 **chembl_extraction_service.py**

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

76 77 **descriptor_factory.py**

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

78 79 **target_schema.py**

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

80 81 82 83 84 **defaults.py**

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

89 90 91 92 93 **output_service.py**

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

94 **artifacts.py**

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

95 96 97 98 99 100 101 102 103 **client.py**

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

104 105 106 107 **client.py**

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

108 109 110 111 112 113 114 115 116 117 118 **factories.py**

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

119 120 121 122 **client_registry.py**

https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/pipelines/client_registry.py

123 124 **normalizers.py**

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