

## Объекты ETL-пайплайна ChEMBL/Assay

Класс/Объект	Предок	Модуль	Назначение
ChemblAssayPipeline	ChemblCommonPipeline	<code>src/bioetl/pipelines/chembl/assay/run.py</code>	Основной пайплайн для обработки данных ChEMBL. Включает этапы загрузки, валидации и переработки данных.
ChemblCommonPipeline	ChemblPipelineBase	<code>src/bioetl/pipelines/chembl/common/base.py</code>	Базовый класс для пайплайнов ChEMBL. Включает методы для загрузки данных (обобщенные), загрузки legacy данных, определения стадий, загрузки данных через трансформаторы, загрузки данных из pre_trained, загрузки данных из domain, загрузки данных из validated, загрузки данных из Schenck, загрузки данных из results.

Класс/Объект	Предок	Модуль	Назн
<b>ChemblPipelineBase</b>	PipelineBase	<code>src/bioetl/core/pipeline/unified.py</code>	Абстрактный класс ChEMO-домени извлечения. Содержит общие методы <code>pre_training</code> и <code>domain_adaptation</code> для упрощения исходного кода. Также содержит методы <code>ChemblPipelineBase</code> для загрузки
<b>ChemblExtractionService</b>	object	<code>src/bioetl/pipelines/chembl/common/chembl_extraction_service.py</code>	Сервис для извлечения ChEMO-определителей. Реализует построение запросов к базе данных (fetch) для извлечения ChEMO-определителей.

Класс/Объект	Предок	Модуль	Назн
<b>ChemblExtractionServiceDescriptor</b>	Generic[ChemblPipelineBase]	<code>src/bioetl/core/pipeline/unified.py</code>	Опис извле ChEM подхо фабри build_ конте fetche (возв для п батчу finaliz (возв для ф обраб
<b>ChemblExtractionDescriptor</b>	-	<code>src/bioetl/pipelines/chembl/common/descriptor.py</code>	Лёгки дескр ChEM какие списо полно парам режи или "a разби релиз иниц валид выбор
<b>BatchPlan</b>	-	<code>src/bioetl/clients/chembl/descriptor_factory.py</code>	Datac план при в Содер ( bat чанка постр извле

Класс/Объект	Предок	Модуль	Назн
<b>ChemblDescriptorFactory</b>	-	src/bioetl/clients/chembl/ descriptor_factory.py	Фабри ChEM конф фасад (Chem набор получ ( fet по ти функ фолб ошиб ( fal также для д Испол пайпл постр через
<b>ChemblContextFacade</b>	-	src/bioetl/clients/chembl/ descriptor_factory.py	"Фаса ChEM для п дескр необх компо транс пагин набор пагин релиз Chem либо
<b>ChemblClient</b>	ConfiguredHttpClient	src/bioetl/clients/chembl/ client.py	API-кл (нова Насле клиен метод форм к ChE иниц загру источ Chem для со

Класс/Объект	Предок	Модуль	Назн
<b>ChemblRequestBuilder</b>	-	<code>src/bioetl/clients/chembl/ client.py</code>	Класс запрос 48 . И логик объе (мар филь пагин Испол Chem получ запрос
<b>RequestsBackend</b>	HttpBackend (Protocol)	<code>src/bioetl/clients/chembl/ factories.py</code>	Реали бэкен база requ за вы запрос ChEM метод одной итери запис также соеди
<b>AssaySchema</b>	pa.DataFrameSchema	<code>src/bioetl/core/schemas/ assay_schema.py</code>	Схема ассая DataF Опре обяза (assay др.) и испол прове приве данны транс

Класс/Объект	Предок	Модуль	Назн
<b>AssayNormalizer</b>	-	src/bioetl/pipelines/chembl/ assay/normalizers.py	Норм assay для п очист т.ч. об влож чтобы DataF виду. заглу
<b>AssayPayloadParser</b>	-	src/bioetl/pipelines/chembl/ assay/parsers.py	Парсе 54 . Д сыро ChEM assay. связа струк (Data (Пока
<b>ChemblWriteService</b>	WriteService (Protocol)	src/bioetl/pipelines/chembl/ common/base.py	Спец серви ChEM Реали детер сохра форм выход (дата мета суффи сохра CSV и JSON запус

Класс/Объект	Предок	Модуль	Назн
<b>PipelineOutputService</b>	-	<code>src/bioetl/core/io/output_service.py</code>	Серви для за пайпл на вх артеф настр конф вызы атома датас успеш генер (при л <code>emit</code> Испол Chem сохра испол униф или o Chem
<b>WriteArtifacts</b>	-	<code>src/bioetl/core/io/artifacts.py</code>	Струк путей пайпл поля датас <code>data</code> данны <code>qual</code> (CSV с качес (YAML запус <code>manip</code> мани Запол испол перед запис

1 2 59 **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 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](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](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](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](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](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](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](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](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](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](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](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](https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/src/bioetl/core/io/artifacts.py)