@startuml

' Базовый интерфейс TestExecutor

interface TestExecutor {

+init(Map<String, Object> config) : void

+init(ExecutableEntity config) : void

+execute(Map<String, Object> config) : Map<String, Object>

+<T> execute(Map<String, Object> config, Class<T> clazz) : T

+execute(ExecutableEntity config) : ExecutableEntity

+logAction(String action, String message) : void

+getConfig() : Map<String, Object>

+getConfigAsEntity() : ExecutableEntity

+reset() : void

+<T> validateInput(T input) : void

+validateInput(Map<String, Object> input) : void

+validateInput(ExecutableEntity input) : void

+configure(Map<String, Object> config) : void

+configure(ExecutableEntity config) : void

+configure(Function<Map<String,Object>,Map<String,Object>> config) : void

+isReady() : boolean

+<T> getLastResult() : T

+getLastResultAsMap() : Map<String, Object>

+getLastResultAsEntity() : ExecutableEntity

+rollback() : void

+setTimeout(long timeout) : void

+setHeaders(Map<String, String> headers) : void

+<T> getResponseHeaders(T response) : Map<String, String>

+getResponseHeadersAsMap(Map<String, Object> response) : Map<String, String>

+getResponseHeadersAsEntity(ExecutableEntity response) : Map<String, String>

+executeBatch(List<Map<String, Object>> inputs) : List<Map<String, Object>>

+<T> executeBatch(List<Map<String, Object>> inputs, Class<T> clazz) : List<T>

+executeBatch(List<ExecutableEntity> inputs) : List<ExecutableEntity>

+executeStream(Map<String, Object> input) : Stream<Map<String, Object>>

+<T> executeStream(Map<String, Object> input, Class<T> clazz) : Stream<T>

+executeStream(ExecutableEntity input) : Stream<ExecutableEntity>

+executeAsync(Map<String, Object> input) : CompletableFuture<Map<String, Object>>

+<T> executeAsync(Map<String, Object> input, Class<T> clazz) : CompletableFuture<T>

+executeAsync(ExecutableEntity input) : CompletableFuture<ExecutableEntity>

}

' Новый интерфейс для embedded-режима

interface IEmbeddedTestExecutor {

+startEmbedded() : void

+stopEmbedded() : void

+isEmbeddedRunning() : boolean

+configureEmbedded(Map<String, Object> embeddedConfig) : void

}

' Интерфейсы для конкретных систем

interface IRestTestExecutor {

+post(RequestPayload payload) : Map<String, Object>

+<T> post(RequestPayload payload, Class<T> clazz) : T

+post(RequestPayload payload) : ExecutableEntity

+post(Map<String, Object> payload) : Map<String, Object>

+<T> post(Map<String, Object> payload, Class<T> clazz) : T

+post(ExecutableEntity payload) : ExecutableEntity

+get(Map<String, String> params) : Map<String, Object>

+<T> get(Map<String, String> params, Class<T> clazz) : T

+get(Map<String, String> params) : ExecutableEntity

+put(RequestPayload payload) : Map<String, Object>

+<T> put(RequestPayload payload, Class<T> clazz) : T

+put(RequestPayload payload) : ExecutableEntity

+put(Map<String, Object> payload) : Map<String, Object>

+<T> put(Map<String, Object> payload, Class<T> clazz) : T

+put(ExecutableEntity payload) : ExecutableEntity

+delete(Map<String, String> params) : Map<String, Object>

+<T> delete(Map<String, String> params, Class<T> clazz) : T

+delete(Map<String, String> params) : ExecutableEntity

+patch(RequestPayload payload) : Map<String, Object>

+<T> patch(RequestPayload payload, Class<T> clazz) : T

+patch(RequestPayload payload) : ExecutableEntity

+patch(Map<String, Object> payload) : Map<String, Object>

+<T> patch(Map<String, Object> payload, Class<T> clazz) : T

+patch(ExecutableEntity payload) : ExecutableEntity

+setBasePath(String path) : void

+executeBatch(List<RequestPayload> payloads) : List<Map<String, Object>>

+<T> executeBatch(List<RequestPayload> payloads, Class<T> clazz) : List<T>

+executeBatch(List<ExecutableEntity> payloads) : List<ExecutableEntity>

+executeStream(RequestPayload payload) : Stream<Map<String, Object>>

+<T> executeStream(RequestPayload payload, Class<T> clazz) : Stream<T>

+executeStream(ExecutableEntity payload) : Stream<ExecutableEntity>

}

interface ISqlTestExecutor {

+executeUpdate(SqlQuery query) : Map<String, Object>

+<T> executeUpdate(SqlQuery query, Class<T> clazz) : T

+executeUpdate(ExecutableEntity query) : ExecutableEntity

+executeUpdate(Map<String, Object> query) : Map<String, Object>

+<T> executeUpdate(Map<String, Object> query, Class<T> clazz) : T

+executeQuery(SqlQuery query) : Map<String, Object>

+<T> executeQuery(SqlQuery query, Class<T> clazz) : T

+executeQuery(ExecutableEntity query) : ExecutableEntity

+executeQuery(Map<String, Object> query) : Map<String, Object>

+<T> executeQuery(Map<String, Object> query, Class<T> clazz) : T

+executeBatch(List<SqlQuery> queries) : List<Map<String, Object>>

+<T> executeBatch(List<SqlQuery> queries, Class<T> clazz) : List<T>

+executeBatch(List<ExecutableEntity> queries) : List<ExecutableEntity>

+beginTransaction() : void

+commitTransaction() : void

+executeStream(SqlQuery query) : Stream<Map<String, Object>>

+<T> executeStream(SqlQuery query, Class<T> clazz) : Stream<T>

+executeStream(ExecutableEntity query) : Stream<ExecutableEntity>

}

interface IKafkaTestExecutor {

+sendMessage(KafkaMessage message) : Map<String, Object>

+<T> sendMessage(KafkaMessage message, Class<T> clazz) : T

+sendMessage(ExecutableEntity message) : ExecutableEntity

+sendMessage(Map<String, Object> message) : Map<String, Object>

+<T> sendMessage(Map<String, Object> message, Class<T> clazz) : T

+consumeMessage() : Map<String, Object>

+<T> consumeMessage(Class<T> clazz) : T

+consumeMessage() : ExecutableEntity

+sendBatch(List<KafkaMessage> messages) : List<Map<String, Object>>

+<T> sendBatch(List<KafkaMessage> messages, Class<T> clazz) : List<T>

+sendBatch(List<ExecutableEntity> messages) : List<ExecutableEntity>

+resetOffsets() : void

+subscribeToTopic(String topic) : void

+executeStream(KafkaMessage message) : Stream<Map<String, Object>>

+<T> executeStream(KafkaMessage message, Class<T> clazz) : Stream<T>

+executeStream(ExecutableEntity message) : Stream<ExecutableEntity>

}

interface INoSqlTestExecutor {

+insertDocument(MongoDocument doc) : Map<String, Object>

+<T> insertDocument(MongoDocument doc, Class<T> clazz) : T

+insertDocument(ExecutableEntity doc) : ExecutableEntity

+insertDocument(Map<String, Object> doc) : Map<String, Object>

+<T> insertDocument(Map<String, Object> doc, Class<T> clazz) : T

+findDocument(Bson filter) : Map<String, Object>

+<T> findDocument(Bson filter, Class<T> clazz) : T

+findDocument(Bson filter) : ExecutableEntity

+updateDocument(Bson filter, MongoDocument doc) : Map<String, Object>

+<T> updateDocument(Bson filter, MongoDocument doc, Class<T> clazz) : T

+updateDocument(Bson filter, ExecutableEntity doc) : ExecutableEntity

+updateDocument(Bson filter, Map<String, Object> doc) : Map<String, Object>

+<T> updateDocument(Bson filter, Map<String, Object> doc, Class<T> clazz) : T

+deleteDocument(Bson filter) : Map<String, Object>

+<T> deleteDocument(Bson filter, Class<T> clazz) : T

+deleteDocument(Bson filter) : ExecutableEntity

+insertMany(List<MongoDocument> docs) : List<Map<String, Object>>

+<T> insertMany(List<MongoDocument> docs, Class<T> clazz) : List<T>

+insertMany(List<ExecutableEntity> docs) : List<ExecutableEntity>

+executeStream(Bson filter) : Stream<Map<String, Object>>

+<T> executeStream(Bson filter, Class<T> clazz) : Stream<T>

+executeStream(ExecutableEntity filter) : Stream<ExecutableEntity>

}

interface IRedisTestExecutor {

+set(RedisEntry entry) : Map<String, Object>

+<T> set(RedisEntry entry, Class<T> clazz) : T

+set(ExecutableEntity entry) : ExecutableEntity

+set(Map<String, Object> entry) : Map<String, Object>

+<T> set(Map<String, Object> entry, Class<T> clazz) : T

+get(String key) : Map<String, Object>

+<T> get(String key, Class<T> clazz) : T

+get(String key) : ExecutableEntity

+delete(String key) : Map<String, Object>

+<T> delete(String key, Class<T> clazz) : T

+delete(String key) : ExecutableEntity

+setWithExpiry(RedisEntry entry, long expiry) : Map<String, Object>

+<T> setWithExpiry(RedisEntry entry, long expiry, Class<T> clazz) : T

+setWithExpiry(ExecutableEntity entry, long expiry) : ExecutableEntity

+hset(String key, RedisEntry entry) : Map<String, Object>

+<T> hset(String key, RedisEntry entry, Class<T> clazz) : T

+hset(String key, ExecutableEntity entry) : ExecutableEntity

+hget(String key, String field) : Map<String, Object>

+<T> hget(String key, String field, Class<T> clazz) : T

+hget(String key, String field) : ExecutableEntity

+lpush(String key, String value) : Map<String, Object>

+<T> lpush(String key, String value, Class<T> clazz) : T

+lpush(String key, ExecutableEntity value) : ExecutableEntity

+rpop(String key) : Map<String, Object>

+<T> rpop(String key, Class<T> clazz) : T

+rpop(String key) : ExecutableEntity

+executeStream(RedisEntry entry) : Stream<Map<String, Object>>

+<T> executeStream(RedisEntry entry, Class<T> clazz) : Stream<T>

+executeStream(ExecutableEntity entry) : Stream<ExecutableEntity>

}

' Кастомная обертка

class ExecutableEntity {

-data : Map<String, Object>

-metadata : Map<String, String>

+getData() : Map<String, Object>

+getMetadata() : Map<String, String>

+toHashMap() : Map<String, Object>

+fromHashMap(Map<String, Object>) : ExecutableEntity

}

' POJO-классы

class RequestPayload {

-id : String

-action : String

+getId() : String

+getAction() : String

+toHashMap() : Map<String, Object>

+fromHashMap(Map<String, Object>) : RequestPayload

}

class KafkaMessage {

-key : String

-value : String

+getKey() : String

+getValue() : String

+toHashMap() : Map<String, Object>

+fromHashMap(Map<String, Object>) : KafkaMessage

}

class SqlQuery {

-query : String

+getQuery() : String

+toHashMap() : Map<String, Object>

+fromHashMap(Map<String, Object>) : SqlQuery

}

class MongoDocument {

-document : Document

+getDocument() : Document

+toHashMap() : Map<String, Object>

+fromHashMap(Map<String, Object>) : MongoDocument

}

class RedisEntry {

-key : String

-value : String

+getKey() : String

+getValue() : String

+toHashMap() : Map<String, Object>

+fromHashMap(Map<String, Object>) : RedisEntry

}

' Реализации интерфейсов (обычные)

class RestTestExecutorImpl {

-requestSpec : RequestSpecification

-controller : TestRestController

-lastResult : Object

}

class PostgresTestExecutorImpl {

-connection : Connection

-lastResult : Object

}

class KafkaTestExecutorImpl {

-producer : KafkaProducer

-consumer : KafkaConsumer

-lastResult : Object

}

class MongoTestExecutorImpl {

-mongoClient : MongoClient

-database : MongoDatabase

-lastResult : Object

}

class RedisTestExecutorImpl {

-jedis : Jedis

-lastResult : Object

}

' Реализации интерфейсов (embedded)

class EmbeddedRestTestExecutorImpl {

-requestSpec : RequestSpecification

-controller : TestRestController

-lastResult : Object

}

class EmbeddedPostgresTestExecutorImpl {

-connection : Connection

-postgresContainer : PostgreSQLContainer

-lastResult : Object

}

class EmbeddedKafkaTestExecutorImpl {

-producer : KafkaProducer

-consumer : KafkaConsumer

-embeddedKafka : EmbeddedKafkaBroker

-lastResult : Object

}

class EmbeddedMongoTestExecutorImpl {

-mongoClient : MongoClient

-database : MongoDatabase

-mongoContainer : MongoDBContainer

-lastResult : Object

}

class EmbeddedRedisTestExecutorImpl {

-jedis : Jedis

-redisContainer : RedisContainer

-lastResult : Object

}

' Наследование интерфейсов

IRestTestExecutor <|.. RestTestExecutorImpl

ISqlTestExecutor <|.. PostgresTestExecutorImpl

IKafkaTestExecutor <|.. KafkaTestExecutorImpl

INoSqlTestExecutor <|.. MongoTestExecutorImpl

IRedisTestExecutor <|.. RedisTestExecutorImpl

IEmbeddedTestExecutor <|.. EmbeddedRestTestExecutorImpl

IRestTestExecutor <|.. EmbeddedRestTestExecutorImpl

IEmbeddedTestExecutor <|.. EmbeddedPostgresTestExecutorImpl

ISqlTestExecutor <|.. EmbeddedPostgresTestExecutorImpl

IEmbeddedTestExecutor <|.. EmbeddedKafkaTestExecutorImpl

IKafkaTestExecutor <|.. EmbeddedKafkaTestExecutorImpl

IEmbeddedTestExecutor <|.. EmbeddedMongoTestExecutorImpl

INoSqlTestExecutor <|.. EmbeddedMongoTestExecutorImpl

IEmbeddedTestExecutor <|.. EmbeddedRedisTestExecutorImpl

IRedisTestExecutor <|.. EmbeddedRedisTestExecutorImpl

' Наследование от базового интерфейса TestExecutor

IRestTestExecutor ..|> TestExecutor

ISqlTestExecutor ..|> TestExecutor

IKafkaTestExecutor ..|> TestExecutor

INoSqlTestExecutor ..|> TestExecutor

IRedisTestExecutor ..|> TestExecutor

' Связь POJO и ExecutableEntity с реализациями

RestTestExecutorImpl --> RequestPayload

RestTestExecutorImpl --> ExecutableEntity

PostgresTestExecutorImpl --> SqlQuery

PostgresTestExecutorImpl --> ExecutableEntity

KafkaTestExecutorImpl --> KafkaMessage

KafkaTestExecutorImpl --> ExecutableEntity

MongoTestExecutorImpl --> MongoDocument

MongoTestExecutorImpl --> ExecutableEntity

RedisTestExecutorImpl --> RedisEntry

RedisTestExecutorImpl --> ExecutableEntity

EmbeddedRestTestExecutorImpl --> RequestPayload

EmbeddedRestTestExecutorImpl --> ExecutableEntity

EmbeddedPostgresTestExecutorImpl --> SqlQuery

EmbeddedPostgresTestExecutorImpl --> ExecutableEntity

EmbeddedKafkaTestExecutorImpl --> KafkaMessage

EmbeddedKafkaTestExecutorImpl --> ExecutableEntity

EmbeddedMongoTestExecutorImpl --> MongoDocument

EmbeddedMongoTestExecutorImpl --> ExecutableEntity

EmbeddedRedisTestExecutorImpl --> RedisEntry

EmbeddedRedisTestExecutorImpl --> ExecutableEntity

@enduml