laforge

Release 0.2.0.dev0

Matt VanEseltine

2021 May 19

CONTENTS

laforge: a low-key build system for data work.

CHAPTER

ONE

OVERVIEW

laforge is a low-key build system designed to interoperate Python and SQL data work, originally developed internally for IRIS, the Institute for Research on Innovation and Science at the University of Michigan's Institute for Social Research.

Features:

- Interoperable: Read, write, and execute Python and SQL scripts/data.
- Straightforward: Simple build INI files designed for a one-click build.
- No lock-in: Maintain scripts independent from *laforge*.

CHAPTER

TWO

AN EXAMPLE BUILD

2.1 Directory Organization

Because *laforge* needs to find the scripts, and because scripts will likely interact with output from other scripts, I tend to keep related project/sub-project files together:

```
project
├──input
│   ├──a.csv
│   ├──b.xlsx
│   └──c.csv
├──output
│   ├──results_d.csv
│   ├──results_e.csv
│   └──results_f.csv
├──.env
├──build.ini
├──g.py
├──h.sql
├──i.py
└──j.py
```

2.2 .env

```
distro = mssql
server = MSSQL
database = testdb
schema = laforge
```

2.3 build.ini

```
[DEFAULT]
```

CHAPTER

THREE

MODULE DOCUMENTATION

3.1 builder

Builder reads and executes tasks and lists of tasks.

laforge.builder.**show_env**(*path*)
Show the calculated generic section environment

class laforge.builder.**Verb**
An enumeration.

class laforge.builder.**Target**
An enumeration.

class laforge.builder.**FileCall**(*method*, *kwargs*)

property kwargs
Alias for field number 1

property method
Alias for field number 0

exception laforge.builder.**TaskConstructionError**

exception laforge.builder.**TaskExecutionError**

class laforge.builder.**TaskList**(*from_string*, *location='.'*)

Todo: Implement cache_results=False

load_section_config(*section='DEFAULT'*)
Put together config from env, TaskList config, section config

execute()
Execute each task in the list.

Todo: Restore quiet?

dry_run()
List each task in the list.

class laforge.builder.**BaseTask**(*, *identifier*, *verb*, *target*, *content*, *config*)
Create a task to (verb) (something)

Todo:

if ":" in self.content: previous_result_key, actual_path_content = self.content.split(":")

---

property path
For handlers where dir[verb] + content = path

class laforge.builder.**FileReader**(*, *identifier*, *verb*, *target*, *content*, *config*)

class laforge.builder.**InternalPythonExecutor**(*, *identifier*, *verb*, *target*, *content*, *config*)
Execute (without importing) Python script by path

Allows script adjustment via setting the run name: `__main__ = 'laforge'`

..todo

```
Allow implict/explicit return of results.
```

class laforge.builder.**Echoer**(*, *identifier*, *verb*, *target*, *content*, *config*)

class laforge.builder.**SQLQueryReader**(*, *identifier*, *verb*, *target*, *content*, *config*)

class laforge.builder.**SQLExecutor**(*, *identifier*, *verb*, *target*, *content*, *config*)

class laforge.builder.**SQLReaderWriter**(*, *identifier*, *verb*, *target*, *content*, *config*)

class laforge.builder.**FileWriter**(*, *identifier*, *verb*, *target*, *content*, *config*)
Handles all tasks writing to file.

class laforge.builder.**ExistenceChecker**(*, *identifier*, *verb*, *target*, *content*, *config*)

laforge.builder.**load_env**(*path*)
Get .env values without dotenv's default to silently pull package dir

3.2 command

Command-line interface for laforge.

3.3 distros

exception laforge.distros.**SQLDistroNotFound**

class laforge.distros.**Distro**(_)
Base class for SQL Distros

..note:: http://troels.arvin.dk/db/rdbms/

class laforge.distros.**MySQL**(_)

class laforge.distros.**PostgresQL**(_)

class laforge.distros.**MSSQL**(_)

static add_fast_executemany(*engine: sqlalchemy.engine.base.Engine*)
Dramatically improve pyodbc upload performance

Theoretically, just "fast_executemany": "True" should be sufficient in newer versions of the driver.

Note: Improved 1m row upload from over 7 minutes to less than 1 under pyodbc==4.0.26, SQLAlchemy==1.3.1, pandas==0.24.2.

class laforge.distros.**SQLite**(_)

determine_dtypes(*df*)
SQlite does not make gradations in integers or text, so don't try.

laforge.distros.**round_up**(*n*, *nearest=1*)
Round up `n` to the nearest `nearest`.

Parameters

- **n** –
- **nearest** – (Default value = 1)

3.4 sql

SQL utilities for mid-level interaction. Inspired by pathlib; powered by SQLALchemy.

Note: Supported: MSSQL, MariaDB/MySQL, PostgreSQL, SQLite. Supportable: Firebird, Oracle, Sybase.

exception laforge.sql.**SQLTableNotFound**

exception laforge.sql.**SQLChannelNotFound**

exception laforge.sql.**SQLIdentifierProblem**

class laforge.sql.**Channel**(*distro*, ***, *server=None*, *database=None*, *schema=None*, ***engine_kwargs*)
Abstraction from Engine, other static details.

execute_statement(*statement*, *fetch=False*)
Execute SQL (core method)

Todo: De-messify

laforge.sql.**execute**(*statement*, *fetch=False*, *channel=None*)
Convenience method, autofetches Channel if possible

class laforge.sql.**Script**(*query*, *channel=None*)
SQL query string, parsable by 'go' separation and execute()able.

execute(*statements=None*)
Execute itsel(flves)

to_table()
Executes all and tries to return a DataFrame for the result of the final query.

This is one of two ways that laforge retrieves tables.

> **Warning:** This is limited by the capacity of Pandas to retrieve only the final result. For Microsoft SQL Server, if a lengthy set of queries is desired, the most reliable approach appears to be a single final query after a 'go' as a batch gterminator.

> **Warning:** This will rename columns that do not conform to naming standards.

class laforge.sql.**Table**(*name*, *channel=None*, ***kwargs*)
Represents a SQL table, featuring methods to read/write DataFrames.

> **Todo:** Factor out to superclass to allow views

write(*df*, *if_exists='replace'*)
From DataFrame, create a new table and fill it with values

read()
Return the full table as a DataFrame

drop(*ignore_existence=False*)
Delete the table within SQL

class laforge.sql.**Scalar**(*prox*)
Little helper to produce clearly typed single (upper left) ResultProxy result.

class laforge.sql.**Identifier**(*user_input*, *extra=None*)
Single standardized variable/database/schema/table/column/anything identifier.

> **Todo:** class InvalidIdentifierError relay_id_problem(identifier, action, reason=None, replacement=None)

3.5 tech

laforge.tech.**make_first_upper**(*s*)
Uppercase the first letter of s, leaving the rest alone.

3.6 toolbox

Handful of utility functions

> **Note:** These intentionally *only* depend on builtins.

> **Note:** Some copyright information within this file is identified per-block below.

laforge.toolbox.**flatten**(*foo*)

Take any set of nests in an iterator and reduce it into one generator.

'Nests' include any iterable except strings.

Parameters **foo** –

Note: *flatten()* was authored by Amber Yust at https://stackoverflow.com/a/5286571. This function is not claimed under the laforge license.

CHAPTER

FOUR

CONTRIBUTING TO DEVELOPMENT

laforge supports Python 3.6+.

Process	Tool	Documentation
Automation	Nox	https://nox.readthedocs.io/
Test	pytest	https://docs.pytest.org/
Test coverage	pytest-cov	https://pytest-cov.readthedocs.io/
Format	Black	https://black.readthedocs.io/
Lint	Flake8	http://flake8.pycqa.org/
Lint more	Pylint	https://pylint.readthedocs.io/en/latest/
Document	Sphinx	https://www.sphinx-doc.org/

4.1 Suggested Environment

```
# Create virtual environment
python -m venv .venv

# Activate virtual environment with shell-specific script:
. .venv/bin/activate.fish           # fish
# $ source ./.venv/bin/activate     # bash
#  source ./.venv/bin/activate.csh  # csh
# Note that Python for Windows creates ./Scripts/ rather than ./bin/
# .\.venv\Scripts\Activate.ps1      # PowerShell
# .venv\Scripts\Activate.bat        # cmd

# Install packages
python -m pip install -r requirements.txt

# Install working copy

# If desired, optional packages for Excel or other DBs...
# python -m pip install -e .[excel]
# python -m pip install -e .[mysql]
# python -m pip install -e .[all]

# Run tests
python -m pytest

# Run the gauntlet
python -m nox
```

4.2 Embedded TODOs

Todo: Implement cache_results=False

original entry

Todo: Restore quiet?

original entry

Todo:

if ":" in self.content: previous_result_key, actual_path_content = self.content.split(":")

original entry

Todo: De-messify

original entry

Todo: Factor out to superclass to allow views

original entry

Todo: class InvalidIdentifierError relay_id_problem(identifier, action, reason=None, replacement=None)

original entry

4.3 Docstring Gaps

```
Undocumented Python objects
===========================
laforge.builder
---------------
Functions:
 * get_verb
 * is_verb

Classes:
 * BaseTask -- missing methods:

   - implement
   - validate_results
 * DirectoryVisit
 * Echoer -- missing methods:
```

(continues on next page)

(continued from previous page)

```
   - implement
 * ExistenceChecker -- missing methods:

   - implement
 * FileReader -- missing methods:

   - implement
 * FileWriter -- missing methods:

   - implement
   - write
 * InternalPythonExecutor -- missing methods:

   - implement
 * SQLExecutor -- missing methods:

   - implement
 * SQLQueryReader -- missing methods:

   - implement
 * SQLReaderWriter -- missing methods:

   - implement
 * Task
 * TaskList -- missing methods:

   - load_tasks
   - template_content

laforge.command
---------------
Functions:
 * find_build_config
 * get_package_logger
 * run_build
 * technobabble

laforge.distros
---------------
Classes:
 * Distro -- missing methods:

   - create_engine
   - create_spec
   - determine_dtypes
   - find
   - known
 * MSSQL -- missing methods:

   - create_spec
   - find
 * MySQL -- missing methods:

   - create_spec
 * PostgresQL -- missing methods:

   - create_spec
```

(continues on next page)

(continued from previous page)

```
 * SQLite -- missing methods:

   - create_spec
   - find

laforge.sql
-----------
Functions:
 * fix_bad_columns
 * is_reserved_word

Classes:
 * Channel -- missing methods:

   - clean_up_statement
   - find
   - grab
   - retrieve_engine
   - save_engine
 * Identifier -- missing methods:

   - check
 * Script -- missing methods:

   - read
 * Table -- missing methods:

   - exists
   - resolve

laforge.tech
------------
Functions:
 * capitalize_sentences

Classes:
 * ModifiableVerb
 * Technobabbler
```

CHAPTER

FIVE

FREE SOFTWARE LICENSE

- genindex
- search

PYTHON MODULE INDEX

l

INDEX

V

W