

DDES Webinar Series: LMMD, *System Life Cycle Management (SLCM)*

Daniel L. Young, NRMRL/LMMD/ECEB



The screenshot shows the EPA website interface. At the top is the EPA logo and navigation links: Environmental Topics, Laws & Regulations, and About EPA. A search bar is on the right. Below the navigation is a 'Related Topics' section with a link to 'IM/IT Directives'. The main heading is 'Policy, Procedures and Guidance for System Life Cycle Management (SLCM)'. Below the heading is a paragraph stating the purpose of the policy.

Policy, Procedures and Guidance for System Life Cycle Management (SLCM)

The purpose of this policy is to establish a consistent framework across the Agency to ensure that EPA IT systems and applications are properly planned and managed, controllable, cost-effective and that they support the Agency's mission and business goals.

Overview

NRMRL and LMMD efforts to comply with:

- The OMB mandate,
- The [Federal Source Code Policy](#), and
- [EPA's Interim Open Source Software Policy](#).

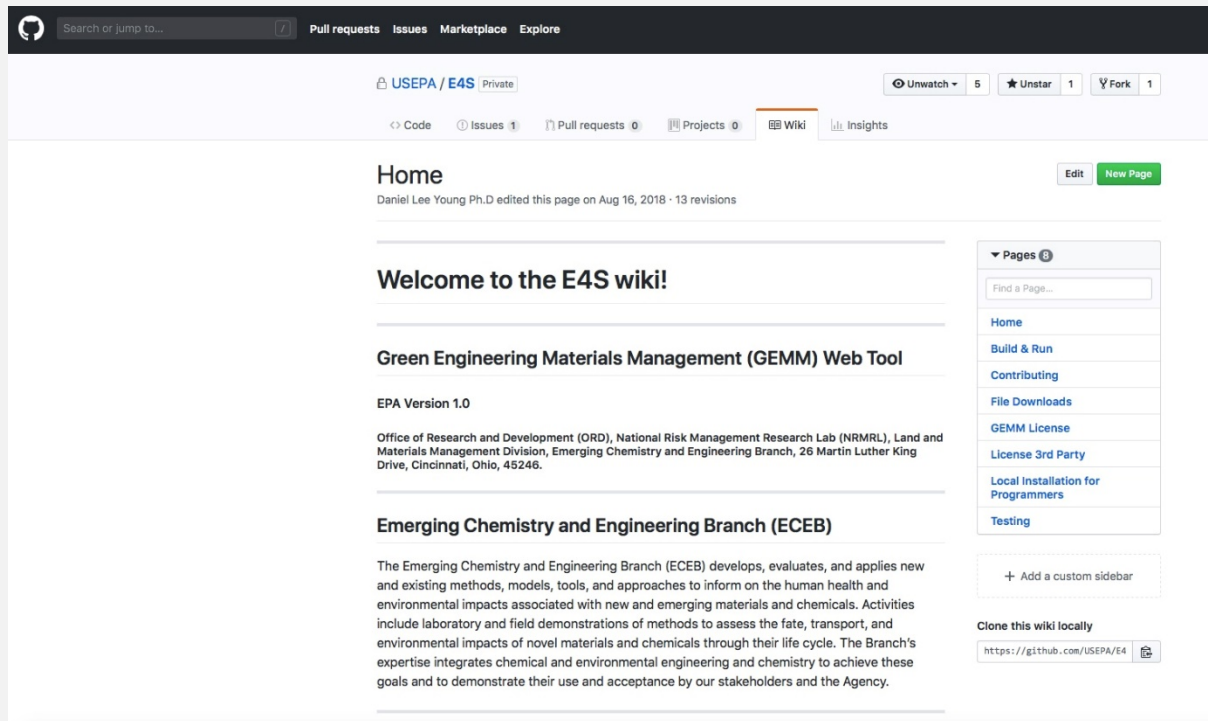
Introduction

- How we manage software in NRMRL... consistently throughout the Division.
- Steven Jones, NRMRL DQA, greatly supports a strong Software Quality Assurance (SQA) program with specific focus on:
 - System Life Cycle Management (SLCM)
 - Software Configuration Management
 - Verification & Validation (V&V) Testing
 - EPA Requirements for QA Project Plans, QA/R-5
 - Cat B, sub-set.

QAPP & System Life Cycle Management (SLCM) requirements.

- ORD QA TRACK
 - Scientific Data: Data-Sets
 - Scientific Data: Database
 - Scientific Data: Models
 - Computer Product (**CP**) removed... **QA/SW disconnect**?????
- US EPA GitHub – External Collaboration
 - Source Code
 - Dump Files
 - Wiki's
- US EPA Bitbucket – Internal Collaboration (requires PIV)

US EPA GitHub & Req. Wiki's



The screenshot shows the GitHub repository page for USEPA/E4S. The repository is private and has 5 unwatchers, 1 unstar, and 1 fork. The main content area displays the 'Home' page of the E4S wiki, which was last edited by Daniel Lee Young Ph.D. on August 16, 2018. The wiki content includes a welcome message, a link to the Green Engineering Materials Management (GEMM) Web Tool, and information about EPA Version 1.0. The sidebar on the right lists the wiki pages: Home, Build & Run, Contributing, File Downloads, GEMM License, License 3rd Party, Local Installation for Programmers, and Testing. There is also a section for cloning the wiki locally with a URL: https://github.com/USEPA/E4.

▼ Pages 8
Find a Page...
Home
Build & Run
Contributing
File Downloads
GEMM License
License 3rd Party
Local Installation for Programmers
Testing

Software & Data Management Projects

- NRMRL currently manages / supports 32 Active SW Projects.
- LMMD currently manages / supports 21 Active SW Projects.

Log in

Email username@email.com

Password pass****

☐ Remember me

Log in

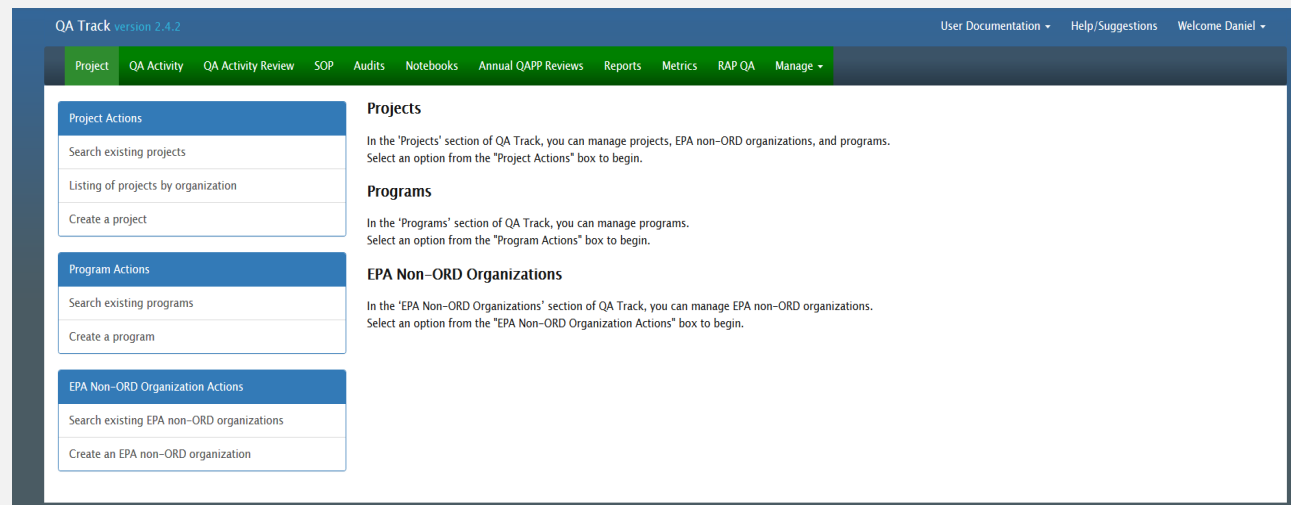
Sign up
Forgot your password?

Project in Production

- [SciNote](#). NRMRL sponsored Electronic Laboratory & Field Notebook.
 - Docker | Ruby Rails Development Environment
 - [Electronic Research Notebook](#)
 - EPA Link <https://ordscinote.epa.gov/>
 - Vendor Link <https://my.scinote.net/>

Project in Production

- ORD QA TRACK. Required QA TRACKING tool ORD.
 - Django | Python 3 Development Environment
 - Started as Division tool to replace Notes QLOG.
 - Adopted by NRMRL Fall 2015... then ORD 2016.



Welcome to GREENSCOPE

Please log in below

Username

Password

Submit

New User

[Forgot password](#) | [Forgot username](#)

For information on this app please contact: young.daniel@epa.gov

Project in Production

- GREENSCOPE.

- Django | Python 3 Development Environment
- The **G**auging **R**eaction **E**ffectiveness for the **E**Nvironmental **S**ustainability of **C**hemistries with a multi-**O**bjective **P**rocess **E**valuator (**GREENSCOPE**) tool allows for quantifying process sustainability with about 140 indicators in four main areas:
 - Material Efficiency (26),
 - Energy (14),
 - Economics (33), and
 - Environment (66).
- This set of indicators is capable of transmitting and translating process performance, feedstocks, utilities, equipment, and output information into a sustainability measurement scale.

Welcome to GEMM

Please log in below

Username

Password

[Submit](#)

[New User](#)

[Forgot password](#) | [Forgot username](#)

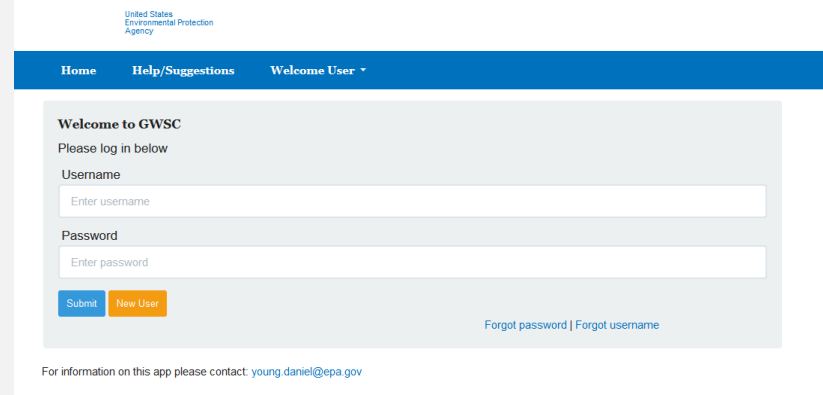
Project in Production

- GEMM.

- Django | Python 3 Development Environment

- Previously developed by Office of Chemical Safety and Pollution Prevention (OCSPP), the **Green Engineering Materials Management** model (**GEMM**) tool calculates and compares the potential environmental impacts of various material management options including destruction (e.g. via incineration), recovery, and reuse of chemicals either on-site or off-site at a different facility.

- The evaluated metrics are: lbs. of hazardous chemicals reduced, MBTU of energy saved, lbs. of water saved, and lbs. of air emissions (e.g. CO, N2O, SOX)
 - The tool shows the financial and environmental benefits of extending the life of chemicals.
 - GEMM has been applied to a subset of high-value secondary hazardous chemicals that can be managed as chemical products rather than waste. This demonstrates that extending the lives of these chemicals increases the potential to generate substantial economic benefits as well as lowers risks to human health and the environment.



The screenshot shows the login interface for the GWSC (Groundwater Seepage Calculator) application. At the top, there is a blue navigation bar with links for 'Home', 'Help/Suggestions', and 'Welcome User'. Below this, the main content area is titled 'Welcome to GWSC' and prompts the user to 'Please log in below'. It features two input fields: 'Username' with a placeholder 'Enter username' and 'Password' with a placeholder 'Enter password'. Below the password field are two buttons: 'Submit' (blue) and 'New User' (orange). At the bottom right of the login area, there are links for 'Forgot password' and 'Forgot username'. A footer note at the bottom of the page states: 'For information on this app please contact: young.daniel@epa.gov'.

Project in Production

- GWSC.
 - Django | Python 3 Development Environment
 - **GroundWater Seepage Calculator: Steady-State Single-Event Non-Redistribution and Transient Single-Event Non-Redistribution.**
 - This web application utilizes steady-state sediment temperature profile to estimate groundwater seepage flux as well as Transient Single-Event Non-Redistribution.

Planned SW Projects

- Multi-Criteria Decision Analysis or MCDA
 - Multi-Criteria Decision Analysis, or MCDA, is a valuable tool that we can apply to many complex decisions.
 - It is most applicable to solving problems that are characterized as a choice among alternatives.

Software Lint | Linting

- Lint or a Linter.
 - A program that supports linting (verifying code quality).
 - Available for most languages e.g., JavaScript, CSS, HTML, Python, etc..
 - Running a Lint program over your source code helps to ensure that source code is legible, readable, less polluted, and easier to maintain.

Software Lint | Linting

- Python Linters

- PEP 8 -- Style Guide for Python Code

- <https://www.python.org/dev/peps/pep-0008/>

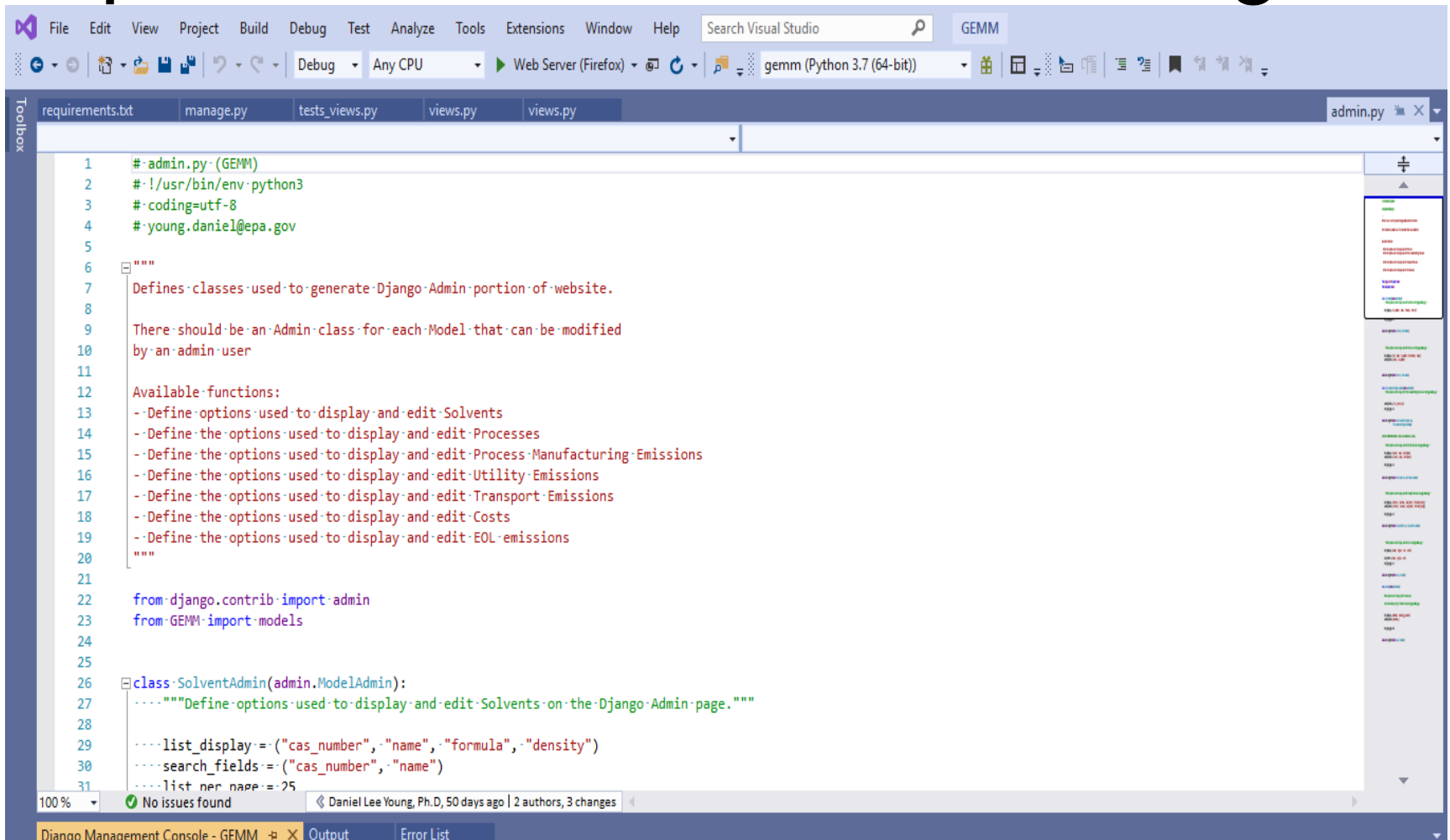
- PEP 257 -- docstring Conventions

- <https://www.python.org/dev/peps/pep-0257/>

Commenting Code

- Triple quote strings
- `#` Comment using pound sign inline that python ignores
- Docstrings (uses triple quotes)

Triple Quote & Module docstring

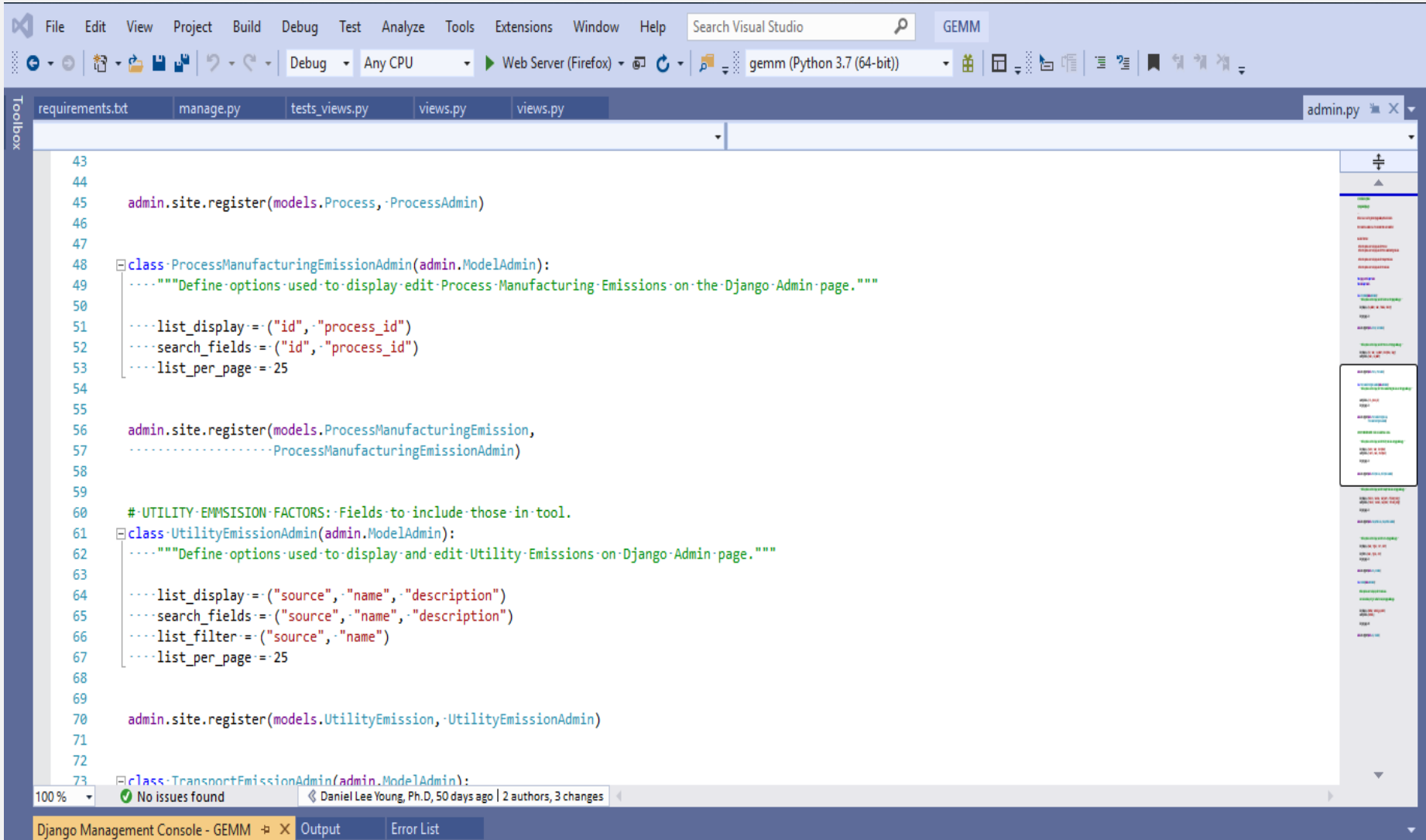


```
1  # admin.py (GEMM)
2  #!/usr/bin/env python3
3  # coding=utf-8
4  # young.daniel@epa.gov
5
6  """
7  Defines classes used to generate Django Admin portion of website.
8
9  There should be an Admin class for each Model that can be modified
10 by an admin user
11
12 Available functions:
13 - Define options used to display and edit Solvents
14 - Define the options used to display and edit Processes
15 - Define the options used to display and edit Process Manufacturing Emissions
16 - Define the options used to display and edit Utility Emissions
17 - Define the options used to display and edit Transport Emissions
18 - Define the options used to display and edit Costs
19 - Define the options used to display and edit EOL emissions
20 """
21
22 from django.contrib import admin
23 from GEMM import models
24
25
26 class SolventAdmin(admin.ModelAdmin):
27     """Define options used to display and edit Solvents on the Django Admin page."""
28
29     list_display = ("cas_number", "name", "formula", "density")
30     search_fields = ("cas_number", "name")
31     list_per_page = 25
```

100 % No issues found Daniel Lee Young, Ph.D, 50 days ago | 2 authors, 3 changes

Django Management Console - GEMM Output Error List

Comment & Class docstring



```
43
44
45 admin.site.register(models.Process, ProcessAdmin)
46
47
48 class ProcessManufacturingEmissionAdmin(admin.ModelAdmin):
49     """Define options used to display/edit Process Manufacturing Emissions on the Django Admin page."""
50
51     list_display = ("id", "process_id")
52     search_fields = ("id", "process_id")
53     list_per_page = 25
54
55
56 admin.site.register(models.ProcessManufacturingEmission,
57                     ProcessManufacturingEmissionAdmin)
58
59
60 # UTILITY EMISSIONS FACTORS: Fields to include those in tool.
61 class UtilityEmissionAdmin(admin.ModelAdmin):
62     """Define options used to display and edit Utility Emissions on Django Admin page."""
63
64     list_display = ("source", "name", "description")
65     search_fields = ("source", "name", "description")
66     list_filter = ("source", "name")
67     list_per_page = 25
68
69
70 admin.site.register(models.UtilityEmission, UtilityEmissionAdmin)
71
72
73 class TransportEmissionAdmin(admin.ModelAdmin):
```

100 % No issues found Daniel Lee Young, Ph.D, 50 days ago | 2 authors, 3 changes

Django Management Console - GEMM Output Error List

Software Lint | Linting

- Other Linters
 - Ruby: RuboCop
 - C#: Stylecop, FxCop

Software Development

- Preferred IDE's are:
 - Microsoft Visual Studio Professional 2017/19*.
 - Fully integrated Python/Anaconda*.
 - Microsoft Visual Studio Code.

VS Professional...

- Visual Studio Professional 2019
 - Run our 51 tests... immediate results!
 - 3 Tests Failed? Some work left...
 - 48 Test Passed!
 - Run our Linter... PyLint... immediate results!
 - 2 Errors | 2 Warnings | 16 Messages
 - Build & Run app localhost... immediate results!
 - No need to wait on a compiler!

PyLint & Testing

The screenshot shows a Visual Studio IDE with a Python project named 'GEMM'. The main editor displays a file named 'requirements.txt' with a list of Python dependencies. The bottom-left pane shows the 'Error List' with 16 messages, including 'Bad option value 'undefined-variable.' [Ebad-option-value]' and 'Module name '0001_initial' doesn't conform to snake_case naming style [Cinvalid-name]'. The bottom-right pane shows the 'Test Explorer' with a summary of 51 tests: 3 failed and 48 passed. The word 'PyLint' is highlighted in a box on the left, and 'Tests' is highlighted in a box on the right.

PyLint

Tests

PyLint Code Rating | Req's

Requirements.txt

```

1 # Check Python versions - "version Python 3.7.3 (64-bit)."
2 # This requirements document was generated in the following way:
3 # 1. Create a new python virtual environment.
4 # 2. Install packages in this environment from the previous requirements file in Github
5 # 3. Delete the existing requirements file.
6 # 4. Generate this requirements file.
7 # This method ensures that all installed packages are in the requirements file, which
8 # includes packages that are sub-requirements of other packages.
9
10 .....#
11 # All of these packages appear to be necessary for both Windows and Linux environments. #
12 .....#
13 astroid==2.2.5 .....# Abstract syntax tree for Python with inference support.
14 atomicwrites==1.3.0 .....# Temporary file in the same directory as the given path.
15 autopep8==1.4.3 .....# Automatically formats Python code to conform to the PEP 8 style guide.
16 bcrypt==3.1.6 .....# Modern password hashing for your software and your servers. *May be required for RTP RHEL server?
17 certifi==2019.3.9 .....# Collection Root Certificates for validating trustworthiness SSL certificates verifying identity TLS hosts.
18 chardet==3.0.4 .....# Universal encoding detector for Python 2 and 3.
19 colorama==0.4.1 .....# Cross-platform colored terminal text.
20 coverage==4.5.3 .....# Provides in-depth look at unit testing coverage.
21 decorator==4.4.0 .....# Better living through Python with decorators.
22 django-docs==0.3.1 .....# Allows to serve Sphinx generated docs directly from Django.
23 django-filter==2.1.0 .....# Reusable Django application allows users to filter querysets dynamically.
24 django-grappelli==2.12.2 .....# A jazzy skin for the Django Admin-Interface
25 django-jet==1.0.8 .....# Modern template for Django admin interface with improved functionality.
26 django-session-security==2.6.5 .....# Provides a mechanism to logout inactive authenticated users.
27 django-wkhtmltopdf==3.2.0 .....# Allows a Django site to output dynamic PDFs.
28 Django==2.2 .....# Python Modules.
29 django-rest-framework==3.9.2 .....# Powerful and flexible toolkit for building Web APIs.
30 docutils==0.14 .....# Modular system for processing documentation into formats such as HTML, XML, and LaTeX.
31 et-xmlfile==1.0.1 .....# Low memory library for creating large XML files.

```

PyLint code rating 9.93/10

Test Explorer Summary

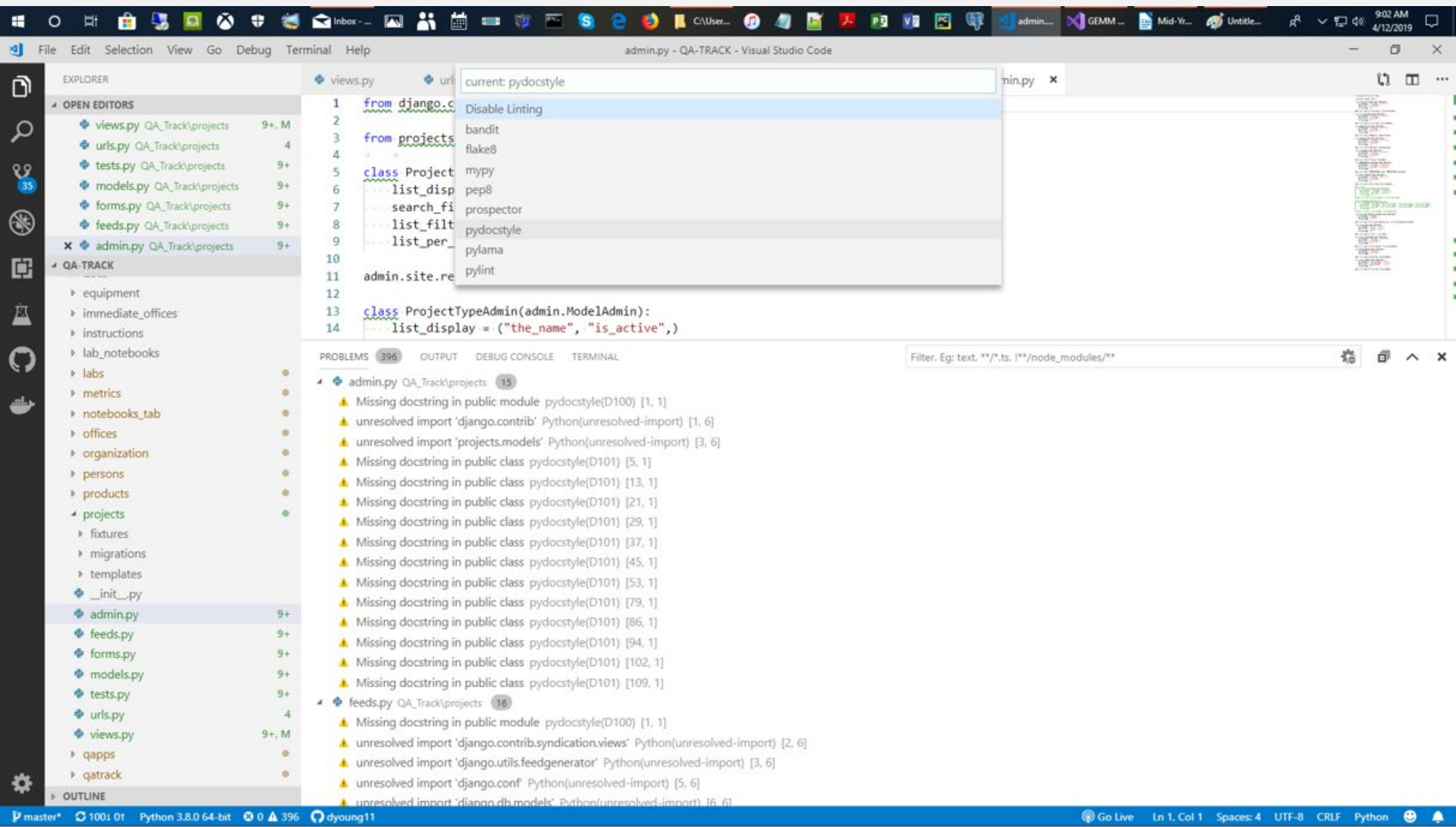
Test Name	Duration	Status
GEMM (51)	15 sec	Failed
GEMM/tests_calculate.py (6)	373 ms	Failed
TestUtilsCalculate (6)	373 ms	Failed
test_acetonitrile_exists		Passed
test_calculate	91 ms	Passed
test_calculate_additional...	75 ms	Passed
test_get_dist_emissions	70 ms	Passed
test_get_incin_emissions	67 ms	Passed
test_get_transport_emis...	70 ms	Failed
GEMM/tests_utils.py (18)	1 sec	Passed
TestUtilsMethods (18)	1 sec	Passed
GEMM/tests_views.py (27)	14 sec	Failed
TestViewsMethods (27)	14 sec	Failed

Last Test Run Failed (Total Run Time 0:00:17.840)
3 Tests Failed
48 Tests Passed

VS Code...

Switching between lint tools...
CTRL+SHIFT+P

pydocstyle



The screenshot shows the Visual Studio Code interface with the 'admin.py' file open. A context menu is visible over the code, listing various linting tools: Disable Linting, bandit, flake8, mypy, pep8, prospector, pydocstyle (selected), pylama, and pylint. The Explorer sidebar on the left shows the project structure, including 'QA-TRACK' and its subdirectories. The Problems panel at the bottom displays 396 errors, with a filter set to 'text: **/*.ts, !**/node_modules/**'. The errors are categorized by file: 'admin.py' (15 errors) and 'feeds.py' (16 errors). The errors for 'admin.py' include missing docstrings for the 'pydocstyle' module and several classes, and unresolved imports for 'django.contrib', 'projects.models', and 'django.conf'. The errors for 'feeds.py' include missing docstrings for the 'pydocstyle' module and unresolved imports for 'django.contrib.syndication.views', 'django.utils.feedgenerator', and 'django.db.models'.

```
1 from django.conf import settings
2 from django.contrib import admin
3 from projects.models import Project
4
5 class ProjectTypeAdmin(admin.ModelAdmin):
6     list_display = (
7         'search_filter',
8         'list_filter',
9         'list_per_page',
10    )
11
12 admin.site.register(ProjectTypeAdmin)
```

PROBLEMS 396 OUTPUT DEBUG CONSOLE TERMINAL

admin.py QA_Track/projects 15

- Missing docstring in public module pydocstyle(D100) [1, 1]
- unresolved import 'django.contrib' Python(unresolved-import) [1, 6]
- unresolved import 'projects.models' Python(unresolved-import) [3, 6]
- Missing docstring in public class pydocstyle(D101) [5, 1]
- Missing docstring in public class pydocstyle(D101) [13, 1]
- Missing docstring in public class pydocstyle(D101) [21, 1]
- Missing docstring in public class pydocstyle(D101) [29, 1]
- Missing docstring in public class pydocstyle(D101) [37, 1]
- Missing docstring in public class pydocstyle(D101) [45, 1]
- Missing docstring in public class pydocstyle(D101) [53, 1]
- Missing docstring in public class pydocstyle(D101) [79, 1]
- Missing docstring in public class pydocstyle(D101) [86, 1]
- Missing docstring in public class pydocstyle(D101) [94, 1]
- Missing docstring in public class pydocstyle(D101) [102, 1]
- Missing docstring in public class pydocstyle(D101) [109, 1]

feeds.py QA_Track/projects 16

- Missing docstring in public module pydocstyle(D100) [1, 1]
- unresolved import 'django.contrib.syndication.views' Python(unresolved-import) [2, 6]
- unresolved import 'django.utils.feedgenerator' Python(unresolved-import) [3, 6]
- unresolved import 'django.conf' Python(unresolved-import) [5, 6]
- unresolved import 'django.db.models' Python(unresolved-import) [16, 6]

Filter: Eg: text, **/*.ts, !**/node_modules/**

master* 1001:01 Python 3.8.0 64-bit 0 396 dyoung11 Go Live Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python

Online Linters & Other info...

Cut & Paste | <http://pep8online.com/>

PEP8 online

Check your code for PEP8 requirements

Just paste your code here

1

Check code

NRMRL SOPs for SW

- Software Configuration Management (SCM), Date: 04/11/2018, SOP Number (G-LMMD-SOP-1399-0).
 - This procedure establishes the responsibilities and process for implementing and documenting Software Configuration Management (SCM) activities within the Land and Materials Management Division (LMMD).

NRMRL SOPs for SW

- Software Verification & Validation (V&V), Date: 04/11/2018. SOP Number (G-LMMD-SOP-1396-0).
 - The purpose of this procedure is to provide the methodology for performing the testing, verification and validation (V&V), and / or independent verification and validation (IV&V) processes throughout the software life cycle.
 - This SOP has been developed to further assist researchers in the Land and Materials Management Division (LMMD) with documenting data that is analyzed during the testing, V&V, and independent verification & validation (IV&V) activities of a Software Development Life Cycle (SDLC), i.e., SLCM.

Django Admin Docs

- <https://docs.djangoproject.com/en/2.2/ref/contrib/admin/admindocs/>
- “Django’s admin docs app pulls documentation from the docstrings of models, views, template tags, and template filters for any app in `INSTALLED_APPS` and makes that documentation available from the Django admin.”

Django Admin

Grappelli

Jacob

[View site](#)

[Documentation](#)

[Home](#) > [Documentation](#) > [Views](#)

View documentation

Jump to site

[Empty namespace](#)

[admin](#)

Views by empty namespace

[/](#)

View function: GEMM.views.home_page

[/<drf_format_suffix:format>](#)

View function: GEMM.views.home_page

[/about/](#)

View function: GEMM.views.AboutView

[/about/<format>/](#)

View function: GEMM.views.AboutView

[/accounts/login/](#)

View function: accounts.views.login

[/accounts/login/<format>/](#)

View function: accounts.views.login

[/accounts/logout/](#)

View function: accounts.views.logout

[/accounts/logout/<format>/](#)

View function: accounts.views.logout

[/accounts/password/reset/](#)

View function: accounts.views.PasswordResetRequestView

Django Admin

Grappelli	Jacob	View site	Documentation
-----------	-------	-----------	---------------

[Home](#) > [Documentation](#) > [Views](#) > `accounts.views.login`

`accounts.views.login`

Displays the login form and handles the login action.

[< Back to Views Documentation](#)

Grappelli	Jacob	View site	Documentation
-----------	-------	-----------	---------------

[Home](#) > [Documentation](#) > [Views](#) > `accounts.views.PasswordResetRequestView`

`accounts.views.PasswordResetRequestView`

View for starting the password reset process. This view renders the form to enter a username or email address. Upon successful entry of a user email, an email is sent with password reset instructions and a confirmation message displayed.

[< Back to Views Documentation](#)

Closing Remarks

- If you have any questions please contact me @ (513) 569-7451 or young.daniel@epa.gov.
- If you would like to provide input, comments, or suggestions to the LMMD SCM or V&V SOP, contact Dr. Michael Gonzalez.

Resources

- [EPA Requirements for QA Project Plans \(QA/R-5\)](#)
- [Policy, Procedures and Guidance for System Life Cycle Management \(SLCM\)](#)
- [System Life Cycle Management \(SCM\) Procedure. CIO 2121-P-03.0. Issued by the EPA Chief Information Officer, Pursuant to Delegation 1-19, dated 07/07/2005.](#)

Resources

- EPA Developer Guidance.
 - <https://developer.epa.gov/guide/templates-guides/waterfall/>
 - <https://developer.epa.gov/guide/templates-guides/agile/agile-frameworks/>
- [GitHub Guidance](#).

Resources

- [ORD IT Resources, GitHub.](#)
- [United States Environmental Protection Agency, GitHub.](#)
- [United States Environmental Protection Agency, Web Guide.](#)