

## Setup Guide for Great Expectations Framework in VSCode

# Setup Guide for Great Expectations Framework in VSCode

## Step 1: Install Great Expectations

Open VSCode.

Open the Integrated Terminal: Press Ctrl+` or navigate to View > Terminal.

Create a Python Virtual Environment (optional but recommended):

```
python -m venv venv
```

```
source venv/bin/activate # On Windows, use venv\Scripts\activate
```

Install Great Expectations:

```
pip install great_expectations
```

## Setup Guide for Great Expectations Framework in VSCode

### Step 2: Initialize a Great Expectations Project

Initialize Great Expectations:

```
great_expectations init
```

This command creates a `great_expectations` directory with several subdirectories and configuration files.

## Setup Guide for Great Expectations Framework in VSCode

### Step 3: Organize Your Project Structure

Create the necessary directories and files as described in the previous steps:

```
mkdir -p great_expectations_project/src great_expectations_project/scripts
great_expectations_project/config
touch great_expectations_project/src/__init__.py great_expectations_project/src/data_validation.py
great_expectations_project/src/models.py
touch great_expectations_project/scripts/run_validations.py
touch great_expectations_project/config/great_expectations.yml
great_expectations_project/requirements.txt great_expectations_project/setup.py
```

## Setup Guide for Great Expectations Framework in VSCode

### Step 4: Write the Code

src/models.py:

Define the SQLAlchemy models:

```
from sqlalchemy import create_engine, Column, Integer, String, ForeignKey, Text, DateTime, Boolean

from sqlalchemy.ext.declarative import declarative_base

from sqlalchemy.orm import sessionmaker

import datetime
```

```
Base = declarative_base()
```

```
class Datasource(Base):
```

```
    __tablename__ = 'datasources'

    id = Column(Integer, primary_key=True)

    name = Column(String, unique=True, nullable=False)

    connection_string = Column(String, nullable=False)
```

```
class ExpectationSuite(Base):
```

```
    __tablename__ = 'expectation_suites'

    id = Column(Integer, primary_key=True)

    name = Column(String, unique=True, nullable=False)
```

## Setup Guide for Great Expectations Framework in VSCode

```
description = Column(Text)
```

```
class Expectation(Base):
```

```
    __tablename__ = 'expectations'
```

```
    id = Column(Integer, primary_key=True)
```

```
    suite_id = Column(Integer, ForeignKey('expectation_suites.id'), nullable=False)
```

```
    expectation_type = Column(String, nullable=False)
```

```
    kwargs = Column(Text, nullable=False)
```

```
class Batch(Base):
```

```
    __tablename__ = 'batches'
```

```
    id = Column(Integer, primary_key=True)
```

```
    datasource_id = Column(Integer, ForeignKey('datasources.id'), nullable=False)
```

```
    batch_identifier = Column(String, nullable=False)
```

```
    timestamp = Column(DateTime, default=datetime.datetime.utcnow)
```

```
class Validation(Base):
```

```
    __tablename__ = 'validations'
```

```
    id = Column(Integer, primary_key=True)
```

```
    batch_id = Column(Integer, ForeignKey('batches.id'), nullable=False)
```

```
    suite_id = Column(Integer, ForeignKey('expectation_suites.id'), nullable=False)
```

```
    success = Column(Boolean, nullable=False)
```

```
    run_time = Column(DateTime, default=datetime.datetime.utcnow)
```

## Setup Guide for Great Expectations Framework in VSCode

```
class ValidationResult(Base):
```

```
    __tablename__ = 'validation_results'
```

```
    id = Column(Integer, primary_key=True)
```

```
    validation_id = Column(Integer, ForeignKey('validations.id'), nullable=False)
```

```
    expectation_id = Column(Integer, ForeignKey('expectations.id'), nullable=False)
```

```
    success = Column(Boolean, nullable=False)
```

```
    result = Column(Text)
```

# Setup Guide for Great Expectations Framework in VSCode

## Step 5: Create Configuration Files

Write the Configuration Files

requirements.txt:

great\_expectations

sqlalchemy

psycopg2-binary

setup.py:

```
from setuptools import setup, find_packages
```

```
setup(  
    name='great_expectations_project',  
    version='0.1.0',  
    packages=find_packages(where="src"),  
    package_dir={"": "src"},  
    install_requires=[  
        'great_expectations',  
        'sqlalchemy',  
        'psycopg2-binary',  
    ],
```



## Setup Guide for Great Expectations Framework in VSCode

```
entry_points={  
    'console_scripts': [  
        'run-validations=scripts.run_validations:main',  
    ],  
},  
)
```

config/great\_expectations.yml:

Configure your data context:

datasources:

your\_datasource\_name:

class\_name: PandasDatasource

module\_name: great\_expectations.datasource

validation\_operators:

action\_list\_operator:

class\_name: ActionListValidationOperator

action\_list:

- name: store\_validation\_result

action:

class\_name: StoreValidationResultAction

- name: store\_evaluation\_params

action:

## Setup Guide for Great Expectations Framework in VSCode

class\_name: StoreEvaluationParametersAction

- name: update\_data\_docs

action:

class\_name: UpdateDataDocsAction

## Setup Guide for Great Expectations Framework in VSCode

### Step 6: Install Dependencies

Open the Integrated Terminal: Press Ctrl+` or navigate to View > Terminal.

Navigate to your project directory:

```
cd great_expectations_project
```

Install dependencies:

```
pip install -r requirements.txt
```

## Setup Guide for Great Expectations Framework in VSCode

### Step 7: Run the Script

Package your project:

```
python setup.py sdist bdist_wheel
```

Install your project:

```
pip install dist/great_expectations_project-0.1.0-py3-none-any.whl
```

Run your validation script:

```
run-validations
```