

Schema Repository Setup Guide (Simplified Version)

1. Repository Name & Structure

Recommended repo name: largegroup-schemas

Folder layout:

```
largegroup-schemas/  
  schemas/  
    json/  
      account/v1.0.0/account.schema.json  
      openapi/v1.0.0/openapi.yaml  
  scripts/prepush.py  
  .github/workflows/validate.yml  
  requirements.txt  
  README.md
```

2. Python Validation Script (prepush.py)

```
import os, sys, subprocess, json, yaml  
from jsonschema import Draft202012Validator  
from openapi_spec_validator import validate_spec  
  
def get_files():  
    if len(sys.argv) > 1:  
        return sys.argv[1:] # files passed from CI  
    result = subprocess.run(["git", "diff", "--cached", "--name-only"],  
                            capture_output=True, text=True)  
    return result.stdout.strip().split("\n")  
  
def validate_json(path):  
    with open(path, "r") as f:  
        schema = json.load(f)  
    Draft202012Validator.check_schema(schema)  
    print(f"[OK] JSON Schema valid: {path}")  
  
def validate_openapi(path):  
    with open(path, "r") as f:  
        spec = yaml.safe_load(f)  
    validate_spec(spec)  
    print(f"[OK] OpenAPI spec valid: {path}")  
  
if __name__ == "__main__":  
    files = get_files()  
    has_error = False  
  
    for path in files:  
        if not os.path.exists(path):  
            continue  
        try:  
            if path.endswith(".schema.json"):
```

```

        validate_json(path)
    elif path.endswith((".yaml", ".yml")):
        validate_openapi(path)
except Exception as e:
    print(f"[ERROR] {path} failed validation: {e}")
    has_error = True

if has_error:
    print("■ Some files are invalid. Fix before push.")
    sys.exit(1)
else:
    print("■ All checked files are valid!")

```

3. requirements.txt

```

jsonschema==4.23.0
openapi-spec-validator==0.7.1
PyYAML==6.0.2

```

4. VS Code Tasks (.vscode/tasks.json)

```

{
  "version": "2.0.0",
  "tasks": [
    {
      "label": "Validate All (pre-push)",
      "type": "shell",
      "command": "python scripts/prepush.py",
      "group": { "kind": "build", "isDefault": true }
    }
  ]
}

```

5. Simplified GitHub Actions Workflow (.github/workflows/validate.yml)

```

name: Validate Schemas
on:
  push:
    branches: [ "main", "develop" ]
  pull_request:
    branches: [ "main", "develop" ]

jobs:
  validate:
    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-python@v5
        with:
          python-version: "3.11"
      - run: pip install -r requirements.txt

```

```

- name: Get changed files
run: |
  git fetch origin ${ github.base_ref } --depth=1 || true
  git diff --name-only origin/${ github.base_ref }...HEAD > changed_files.txt
  cat changed_files.txt

- name: Run validation
run: |
  if [ -s changed_files.txt ]; then
    python scripts/prepush.py $(cat changed_files.txt)
  else
    echo "No schema changes detected."
  fi

```

6. README.md (Simplified)

Large Group Schemas

This repo contains JSON Schema and OpenAPI models for large group systems.

How to Use

1. Install requirements:

```
pip install -r requirements.txt
```
2. Validate before push:

```
python scripts/prepush.py
```
3. GitHub Actions automatically validates changed files on PRs.

Structure

schemas/json/, schemas/openapi/, scripts/, .github/

License

MIT © 2025 Large Group Schema Team