

Python Virtual Environment per LabVIEW

Category: LabVIEW Talk, News

scritto da admin | Dicembre 22, 2024



Preparazione automatica di un ambiente Python per LabVIEW

La comunità LabVIEW ha ormai integrato con successo Python, sfruttandone i vantaggi complementari per ampliare le potenzialità dei propri progetti.

Anche in [Bytelabs](#) utilizziamo sempre più frequentemente script Python per completare e potenziare i progetti sviluppati in LabVIEW.

Ci siamo quindi posti l'interrogativo su quale sia il modo migliore per:

- Limitare dipendenze e problemi con le versioni di Python installate.
- Utilizzare al meglio i Python Node, puntando a versioni compatibili.
- Evitare troppe operazioni manuali.

Python Virtual Environment

La Virtual Environment (venv) è uno strumento che consente di creare ambienti Python isolati.

Ogni ambiente ha una propria installazione di Python, librerie e dipendenze, separate dal sistema principale.

Vantaggi:

Evita conflitti tra diverse versioni di librerie.

Garantisce compatibilità tra applicazioni.

Facilita il deployment di applicazioni standalone.

Creazione e Utilizzo

1. Creazione della Virtual Environment:

```
python -m venv percorso_della_venv
```

2. Attivazione:

Su Windows:

```
percorso_della_venv\Scripts\activate.bat
```

Su macOS/Linux:

```
source percorso_della_venv
```

Disattivazione:

Basta eseguire il comando: deactivate

La virtual environment garantisce che tutte le librerie installate tramite pip rimangano confinate all'ambiente qqqnj, evitando interferenze con altre applicazioni Python.

Batch File per Automatizzare il Processo

Per semplificare e automatizzare la configurazione dell'ambiente, si può utilizzare un file Batch.

Questo file esegue in sequenza i passaggi necessari per:

Creare una virtual environment nella directory dell'applicazione LabVIEW.

Aggiornare pip alla versione più recente.

Installare le librerie richieste tramite un file requirements.txt.

Esempio di Batch File

```
@echo off
REM Percorso della venv
set VENV_PATH=C:\Path\To\YourApp\venv
```

```
REM Creazione della venv
python -m venv %VENV_PATH%
```

```
REM Attivazione della venv  
call %VENV_PATH%\Scripts\activate.bat
```

```
REM Aggiornamento di pip  
%VENV_PATH%\Scripts\python.exe -m pip install --upgrade pip
```

```
REM Installazione dei pacchetti  
%VENV_PATH%\Scripts\python.exe -m pip install -r requirements.txt
```

```
REM Disattivazione  
deactivate
```

LabVIEW può generare dinamicamente questo file Batch e lanciarlo tramite System Exec.vi, garantendo così un setup automatico e trasparente per l'utente.

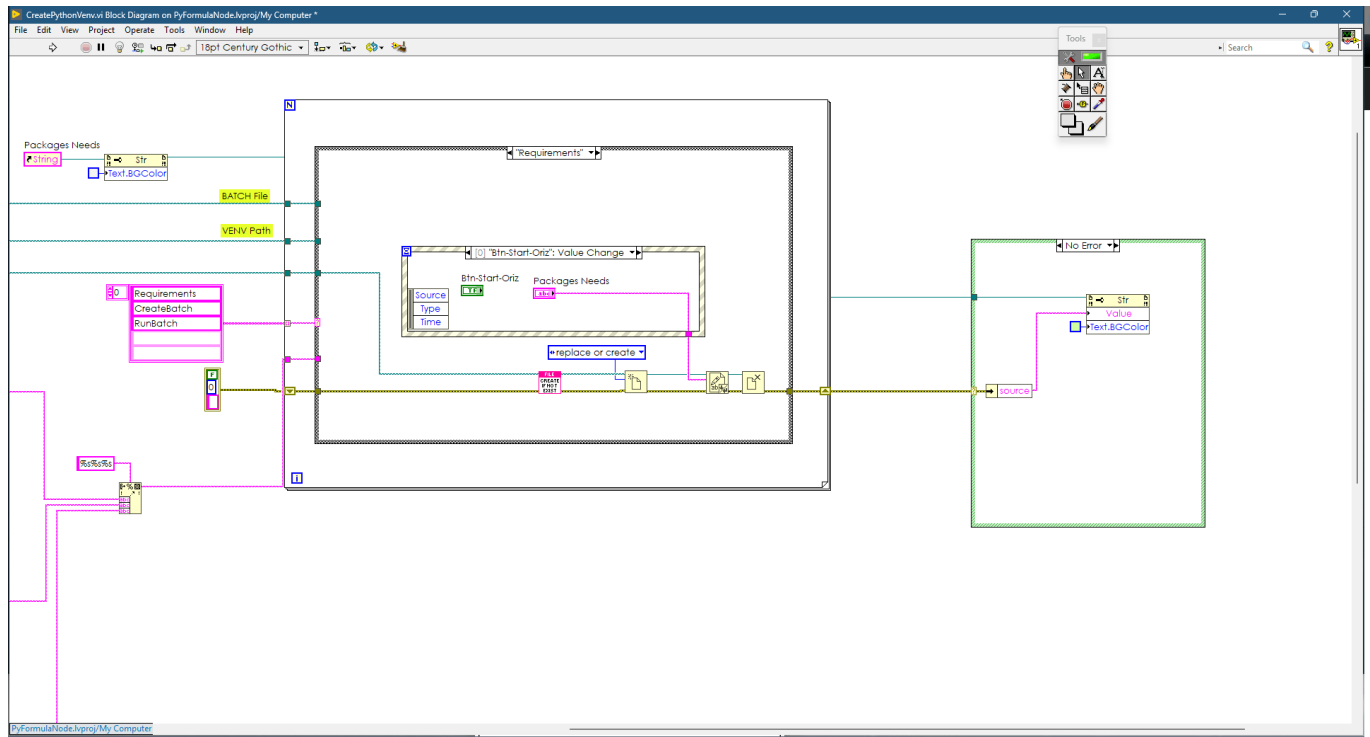
Creazione del configuratore in LabVIEW

Packages Needs

```
Creazione della virtual environment nel percorso: C:\BytelabsRepository\Lab_PythonFormulaNode\Source\my_venv  
Attivazione della virtual environment...  
aggiornamento PIP...  
Requirement already satisfied: pip in c:\bytelabsrepository\lab_pythonformulanode\source\my_venv\lib\site-packages (24.3.1)  
Installazione dei pacchetti richiesti...  
C:\BytelabsRepository\Lab_PythonFormulaNode\Source\requirements.txt  
Requirement already satisfied: numpy in c:\bytelabsrepository\lab_pythonformulanode\source\my_venv\lib\site-packages (from -r C:\BytelabsRepository\Lab_PythonFormulaNode\Source\requirements.txt (line 1)) (2.2.0)  
Virtual environment configurata con successo!
```



NEXT



Nome	Ultima modifica	Tipo
FileTools	21/12/2024 10:43	Cartella di file
my_venv	21/12/2024 11:45	Cartella di file
PythonTools	21/12/2024 11:24	Cartella di file
TestUnit	21/12/2024 12:52	Cartella di file
PyFormulaNode.aliases	23/12/2024 12:48	File ALIASES
PyFormulaNode.lvpls	21/12/2024 13:46	File LVLPS
PyFormulaNode.lvproj	21/12/2024 13:46	LabVIEW Project
PyFormulaNode.py	21/12/2024 13:10	File PY
requirements.txt	23/12/2024 12:49	Documento di test
setup_venv.bat	23/12/2024 12:49	File batch Window

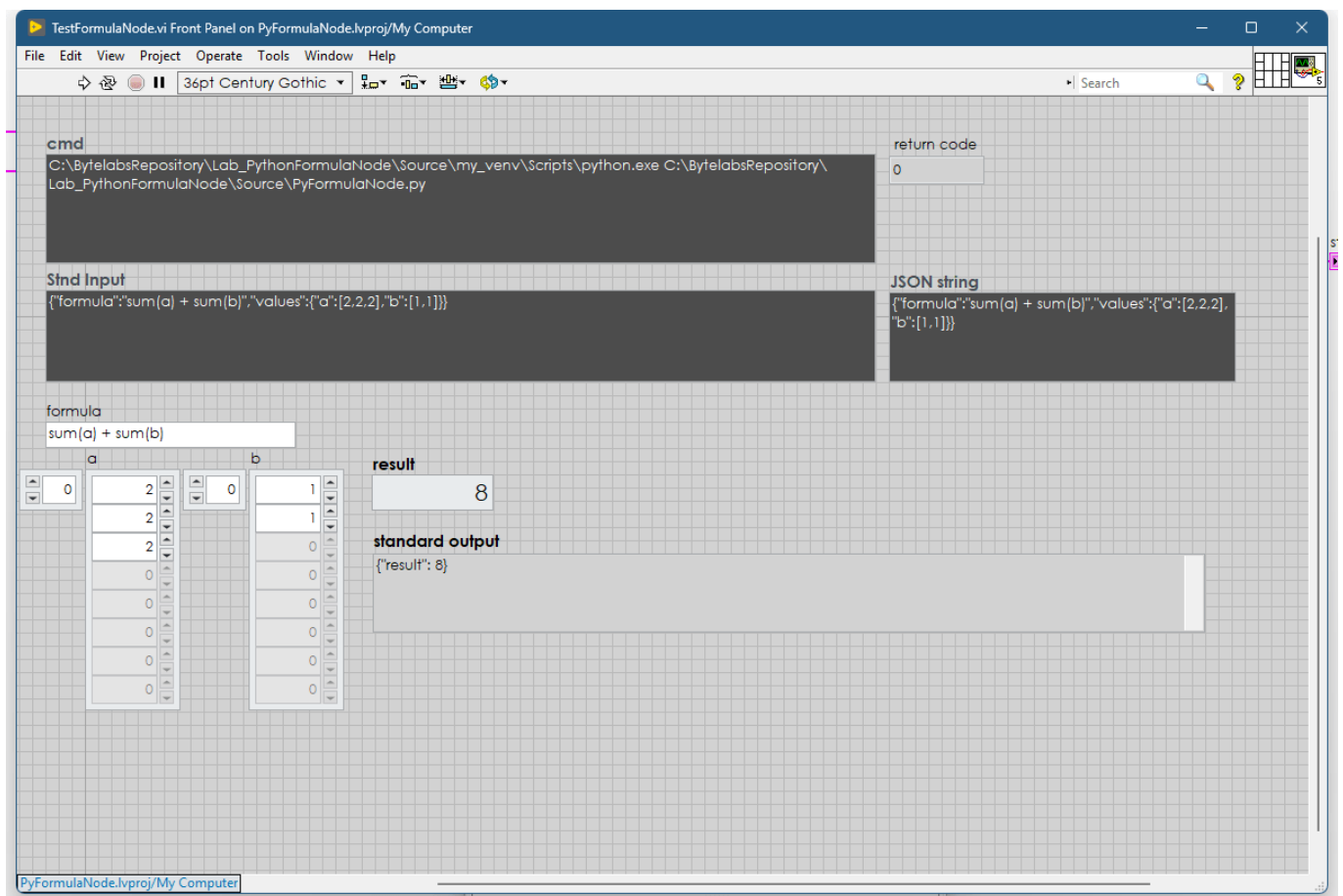
Test dell'eseguibile

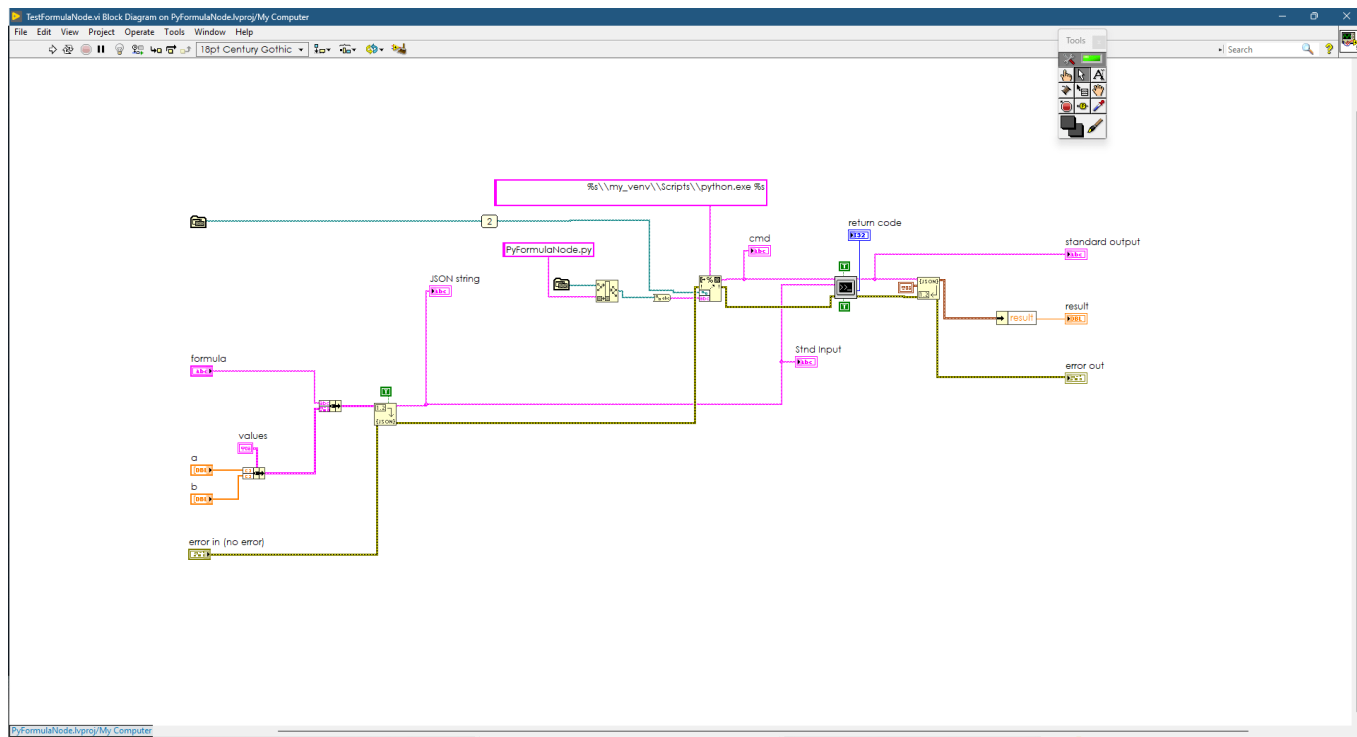
Per testare l'eseguibile abbiamo creato uno script che usa numpy per eseguirsi in LabVIEW come Formula Node.

Ecco il codice PyFormulaNode.py

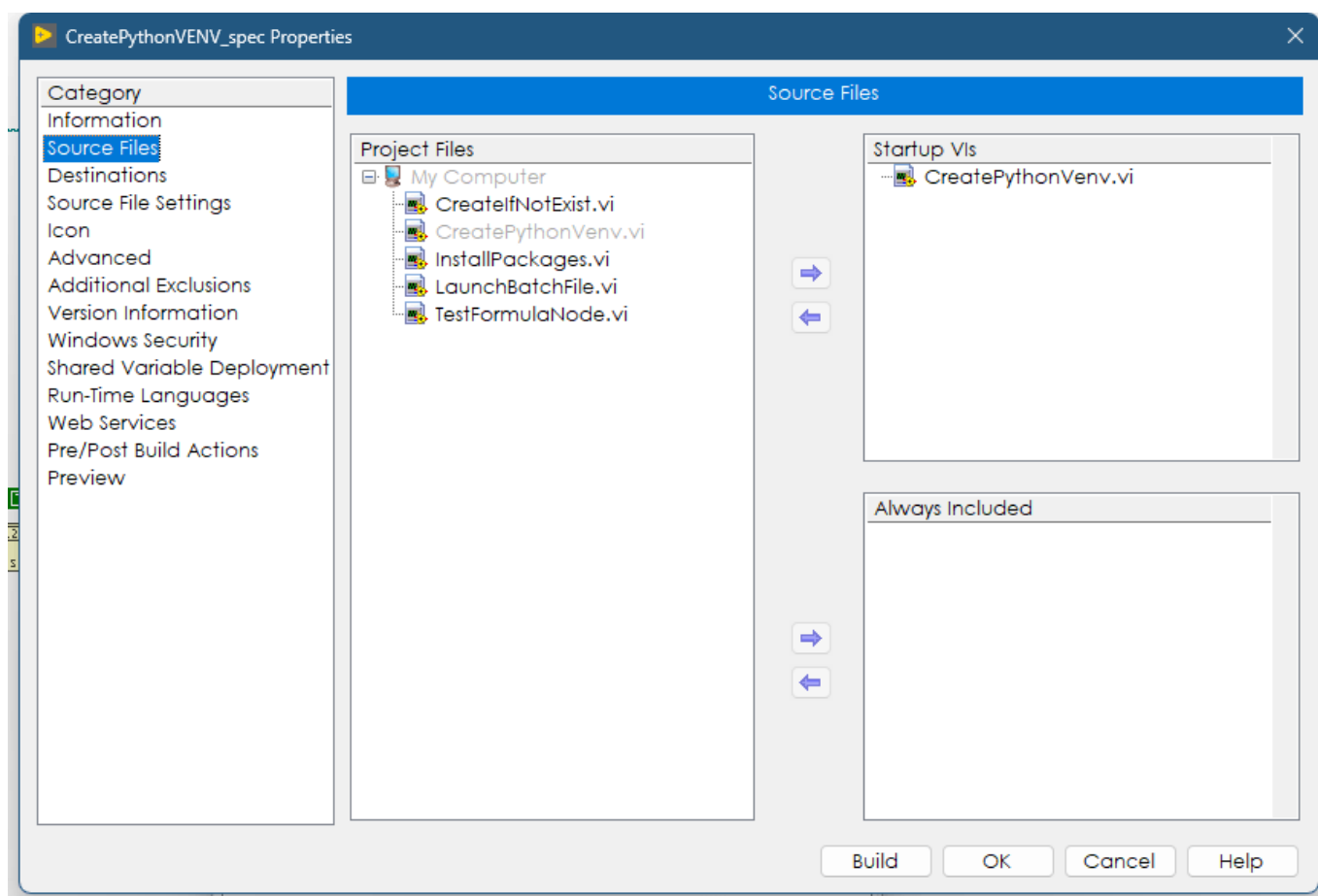
```
21 "arctan": np.arctan,
22 }
23
24 def evaluate_formula(formula, variables):
25     """
26     Valuta una formula in un contesto sicuro utilizzando NumPy.
27
28     Args:
29         formula (str): La formula da valutare, ad esempio "a + b * c".
30         variables (dict): Un dizionario di variabili con scalari o array.
31
32     Returns:
33         Risultato della valutazione della formula o un messaggio di errore.
34     """
35     # Combina funzioni consentite e variabili nel contesto
36     context = {"allowed_functions": **variables}
37     try:
38         # Valuta la formula
39         result = eval(formula, {"__builtins__": None}, context)
40         return result
41     except Exception as e:
42         return f"Error: {e}"
43
44 def convert_numpy_to_python(value):
45     """
46     Converte tipi NumPy in tipi Python standard.
47
48     Args:
49         value: Un valore da convertire.
50
51     Returns:
52         Il valore convertito in un tipo Python standard.
53     """
54     if isinstance(value, np.ndarray):
55         return value.tolist() # Converte array NumPy in lista
56     elif isinstance(value, (np.int32, np.int64)):
57         return int(value) # Converte intero NumPy in int Python
58     elif isinstance(value, (np.float32, np.float64)):
59         return float(value) # Converte Float NumPy in float Python
60     else:
61         return value # Ritorna valori di altri tipi senza modifiche
62
63 if __name__ == "__main__":
64     try:
65         # Leggi il JSON dallo standard input
66         json_data = sys.stdin.read()
67         variables = json.loads(json_data)
68         formula = variables.get("formula", "")
69         result = evaluate_formula(formula, variables)
70         # Converte il risultato a Python standard
71         python_result = convert_numpy_to_python(result)
72         # Stampa il risultato
73         print(json.dumps({"result": python_result}))
```

Il Front Panel del VI TestFormulaNode.vi





Creazione dell'eseguibile per lanciare l'automazione della VENV.



Ora è sufficiente, eseguire questo eseguibile dalla cartella di progetto o da una cartella di esecuzione, per generare una VENV predisposta all'esecuzione con le path relative di progetto/applicazione.

Discussione

Stiamo ancora facendo qualche esperimento sui progetti (sorgenti) e sugli installer per capire potenziali miglioramenti e testare l'efficacia.