

Build-Benedictions

```
buildben init_proj
```

Managing Multiple (Python) Projects & Dependencies

By Dr. Martin Kuric



Icon of *Jesus Christ Pantokrator* by *Theophanes the Cretan*. His right hand is raised in benediction.

From Wikipedia:

*"A **benediction** (Latin: bene, 'well' + dicere, 'to speak') is a short **invocation** for divine help, blessing and guidance [...]."*

*"**Invocation** is the act of calling upon a deity, spirit, or supernatural force, typically through prayer, ritual, or **spoken formula**, to seek guidance, assistance, or presence."*

Build-Benedictions: Main Features

- Standardize setups with **template scaffolds**:
 - `buildben init_proj` : Create a new **project**. *(done)*
 - `buildben add_experiment` : Add a new **experiment** to a project. *(done)*
 - `buildben init_database` : Create a new central **database**. *(wip)*
- Integrate **popular tools**:
 - `direnv` : Virtual environment & variables.
 - `pip-tools` : Dependency management.
 - `just` : For summarizing tasks (upgrade, test, etc.).
 - `Docker` : Snapshot current state of project.

Common Practice: ``requirements.txt`` & ``.venv``

Common Practice: Installing from GitHub

```
git clone <repo-url>           # Download
python -m venv .venv           # Protect system packages
source .venv/bin/activate       # Activate virtual environment
pip install -r requirements.txt # Install dependencies
```

Why this is not enough:

- `requirements.txt` only holds dependencies, not the **project structure**.
 - Python can't **import** local packages.
 - VS Code (sometimes) struggles with **refactoring** & **typing** across packages.
- `requirements.txt` not up-to-date?
- `.venv` not activated?

