

# Scelte di Progetto

PW23/24 – Gruppo Boston23

## Web Service Remoto (PHP – Altervista)

Abbiamo realizzato un web service remoto in php sul nostro sito altervista che permette la visualizzazione di tutti i dati all'interno del database. L'esposizione dei dati è in formato JSON in modo tale da poterli trasportare in maniera corretta dal web service remoto a quello locale. La pagina php è stata arricchita con una visualizzazione semplice dei dati in tabelle corrispondenti a quelle sul DB.

Il web service è raggiungibile tramite il seguente indirizzo:

<https://programmazioneweblog.altervista.org/ProgettoMigrazioneDati/index.php>.

## Web Service Locale (Phyton – Django – PostgreSQL)

Il WebServiceLocale espone per ogni tabella all'interno del database remoto un'api che permette, data una tupla, di inserirla nel database locale in PostgreSQL. I modelli dei dati in django non permettono di definire chiavi composte, per questo, è stato necessario inserire in alcune tabelle una nuova chiave primaria che fungesse da "id". Una volta avviato si mette in ascolto sulla porta locale :8000 e gestisce le richieste alle api.

## ServletJava (Java – Tomcat)

La servlet è rilasciata in Tomcat e viene attivata tramite l'invio di una richiesta post. In seguito alla richiesta ottiene i dati presenti nel database di altervista dal Web Service Remoto e per ogni tupla fa una richiesta all'api della tabella corrispondente nel Web Service Locale. Inoltre tutte le dipendenze (servlet-api, gson, ...) vengono gestite con maven.

# Guida all'utilizzo

PW23/24 – Gruppo Boston23

## Prerequisiti

- Python, Tomcat e PostgreSQL.
- Clonare il repository git [https://github.com/lghisleni/ProgettoPW\\_Boston23](https://github.com/lghisleni/ProgettoPW_Boston23).

## Web Service Locale

- Creare un database PostgreSQL chiamato “MigrazioneDati”;
- Bisogna andare nel seguente percorso:  
**C:\...\ProgettoPW\_Boston23\ProgettoMigrazioneDati\WebServiceLocale\myproject\myproject\settings.py**

E modificare i settaggi del database locale PostgreSQL.

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.postgresql',  
        'NAME': 'MigrazioneDati',  
        'USER': 'postgres',  
        'PASSWORD': 'roberto',  
        'HOST': 'localhost',  
        'PORT': '5432',  
    }  
}
```

- Aprire un terminale **POWERSHELL** come **amministratore**. Muoversi al seguente percorso:  
**C:\..\ProgettoPW\_Boston23\ProgettoMigrazioneDati\WebServiceLocale**

Eseguire il seguente comando per cancellare l'env attuale:

**Remove-Item -Recurse -Force myenv.**

Creare un nuovo env con il comando: **python -m venv myenv**

Avviare l'env: **.\myenv\Scripts\Activate.ps1**

Installare i moduli: **pip install django djangoframework psycopg2-binary**

**NB:** sul terminale deve essere abilitata l'esecuzione degli script. Nel caso non lo fosse, digitare nel terminale: **Set-ExecutionPolicy RemoteSigned** e successivamente digitare **S**.

- Sempre nel terminale muoversi al percorso:  
**C:\..\ProgettoPW\_Boston23\ProgettoMigrazioneDati\WebserviceLocale\myproject**

Eseguire i seguenti comandi: **python manage.py makemigrations** e

**python manage.py migrate**

Al termine di questi comandi il Web Service Locale è funzionante e il database è stato predisposto correttamente.

- Avviare il Web Service con il seguente comando: **python manage.py runserver**

## Servlet

- Copiare il file che si trova nel seguente percorso:  
**C:\..\ProgettoPW\_Boston23\ProgettoMigrazioneDati\ServletJava\servlet-example\target\servlet-example-1.0.1.war**

Nella cartella **webapps** di Tomcat.

- Avviare TomCat in un nuovo terminale (come amministratore). Navigare nella directory bin di tomcat ed eseguire: **.\startup.bat**.

## Testare l'applicazione

- Inviare una richiesta POST con il seguente comando:

**curl -X POST http://localhost:8080/servlet-example-1.0.1/migrate** (questo è un comando per il prompt dei comandi e non per la powershell) oppure inviarla con Postman.

- Se tutto funziona correttamente dall'output del terminale su cui è avviato il Web Service Locale appariranno le chiamate alle api per trasferire i dati:

```
[13/Jul/2024 15:30:35] "POST /api/ricettareregionale/ HTTP/1.1" 201 33
[13/Jul/2024 15:30:35] "POST /api/ricettareregionale/ HTTP/1.1" 201 33
[13/Jul/2024 15:30:35] "POST /api/ricettareregionale/ HTTP/1.1" 201 33
[13/Jul/2024 15:30:35] "POST /api/ricettareregionale/ HTTP/1.1" 201 34
[13/Jul/2024 15:30:35] "POST /api/ricettareregionale/ HTTP/1.1" 201 34
[13/Jul/2024 15:30:35] "POST /api/ricettareregionale/ HTTP/1.1" 201 34
[13/Jul/2024 15:30:35] "POST /api/ricettareregionale/ HTTP/1.1" 201 34
[13/Jul/2024 15:30:35] "POST /api/ricettareregionale/ HTTP/1.1" 201 34
[13/Jul/2024 15:30:35] "POST /api/ricettareregionale/ HTTP/1.1" 201 34
[13/Jul/2024 15:30:35] "POST /api/ricettareregionale/ HTTP/1.1" 201 34
[13/Jul/2024 15:30:35] "POST /api/ricettareregionale/ HTTP/1.1" 201 34
```

Dopodiché i dati saranno presenti all'interno del database PostgreSQL nelle tabelle **data\_service\_[nometabella]**.