Python solution that reads the provided CSV file, updates the JSON configuration file based on the environment parameter, and leaves other JSON elements untouched. The CSV contains key-value pairs that will update the corresponding environment's JSON configuration.

updatejson.py

Instructions to Run:

1. **Place the Files in the Same Directory**: Ensure that `updatejson.py`, `config.json`, and `input.csv` are in the same folder or specify the correct paths when running the script.
2. **Command Line Execution**: To run the script, use the following command:

   ```
   python3 updatejson.py --env <ENVIRONMENT> --json <PATH_TO_JSON> --csv
   <PATH_TO_CSV>
   ```

   For example, if you are updating the `DEV` environment, the command will look like:

   ```
   python3 updatejson.py --env DEV --json ./config.json --csv
   ./input.csv
   ```

CSV File Format:

The CSV file (`input.csv`) should have two columns, where the first column is the key and the second column is the value. For example:

```
host,127.0.0.1
port,8080
```

How It Works:

1. The script reads the JSON configuration file (e.g., `config.json`).
2. It reads the CSV file containing the key-value pairs.
3. It updates the specified environment in the JSON configuration file with the values from the CSV file.
4. Other environments and their configurations remain untouched.
5. The modified JSON file is written back to disk.

The code is now ready to use and will make updates based on the environment provided at runtime.

Here is an example of how your `input.csv` file might look, based on the keys available in the JSON configuration:

Example `input.csv`:

```
csv
Copy code
host,127.0.0.1
port,8080
dbname,new_database
user,new_user
```

```
password,new_password
```

Explanation:

- `host`: The new host value (e.g., `127.0.0.1`)
- `port`: The new port value (e.g., `8080`)
- `dbname`: The new database name (e.g., `new_database`)
- `user`: The new user value (e.g., `new_user`)
- `password`: The new password value (e.g., `new_password`)

When this CSV is processed with the script for the `DEV` environment, it will update the corresponding values in the `config.json` file while leaving the `PROD` environment unchanged.