

```
{  
  "epics_variables": {  
    "pv_a": "fc",  
    "pv_b": "shutdown"  
  },
```

```
"epics_database": {  
  "fc": 0,  
  "shutdown": 0  
},
```

```
"commands": {  
  "master off": {  
    "print": "Master turned OFF",  
    "pv": "pv_a",  
    "value": 1  
  },
```

```
  "master on": {  
    "print": "Master turned ON",  
    "pv": "pv_a",  
    "value": 0  
  },
```

```
"shutdown system": {  
  "print": "System Shutdown Triggered",  
  "pv": "pv_b",  
  "value": 1  
}
```

```
}
```

```
Epiccsss
```

```
import json
```

```
CONFIG_FILE = "config.json"
```

```
def update_epics(command_name):
```

```
    with open(CONFIG_FILE, "r") as f:
```

```
        config = json.load(f)
```

```
    commands = config["commands"]
```

```
    epics_vars = config["epics_variables"]
```

```
    epics_db = config["epics_database"]
```

```
if command_name not in commands:
```

```
    return False, None, None, None
```

```
cmd = commands[command_name]
```

```
pv_name = cmd["pv"]      # pv_a
```

```
epics_key = epics_vars[pv_name] # fc
```

```
value = cmd["value"]
```

```
message = cmd["print"]
```

```
# Update value
epics_db[epics_key] = value

# Save back to JSON
with open(CONFIG_FILE, "w") as f:
    json.dump(config, f, indent=4)

return True, epics_key, value, message
```

loggeerrr

```
import json
from datetime import datetime
import os
```

LOG_FILE = "logs.json"

```
def log_event(user, command, epics_key=None, value=None, status="success"):
```

```
    log_entry = {
        "timestamp": datetime.now().isoformat(),
        "user": user,
        "command": command,
        "epics_updated": epics_key,
        "new_value": value,
        "status": status
    }
```

```
}
```

```
if not os.path.exists(LOG_FILE):
```

```
    with open(LOG_FILE, "w") as f:
```

```
        json.dump({"logs": []}, f)
```

```
    with open(LOG_FILE, "r") as f:
```

```
        data = json.load(f)
```

```
        data["logs"].append(log_entry)
```

```
    with open(LOG_FILE, "w") as f:
```

```
        json.dump(data, f, indent=4)
```

.....in place of if else

```
with open("config.json", "r") as f:
```

```
    config = json.load(f)
```

```
commands = config["commands"]
```

```
epics_vars = config["epics_variables"]
```

```
epics_db = config["epics_database"]
```

```
if predicted_command in commands:
```

```
    cmd = commands[predicted_command]
```

```
# Print message
```

```
    print(cmd["print"])
```

```
# Get pv (like pv_a)
pv_name = cmd["pv"]

# Convert to actual EPICS key (like fc)
epics_key = epics_vars[pv_name]

# Update value
epics_db[epics_key] = cmd["value"]

# Save changes
with open("config.json", "w") as f:
    json.dump(config, f, indent=4)

print(f"{epics_key} updated to {cmd['value']}")

else:
    print("Unknown command")
```