

NGINX in-memory dynamic
Key/Value store

Keyval tables in memory

- Use the keyval module, built-in with NGINX+
- JSON format store loaded on startup
- Source of truth saved anywhere
- CRUD via REST API, POST/PATCH/GET/DELETE
- Records added dynamically, no restart needed
- API protected/hardened
- Memory chunks reserved and zoned by name space

Configuration in http block.

```
keyval_zone zone=one:10m type=ip state=/etc/nginx/one.keyval;  
keyval $remote_addr $ip_flag zone=one;
```

Client IP address is the key, `$remote_addr` system built-in variable

Value matching Client IP is stored in user specified `$ip_flag` variable

Store can be saved anywhere, on or off device.

Keyval CRUD Examples

- `curl -iX POST -d '{"10.11.12.13": "trusted"}' http://localhost/api/7/http/keyvals/one`
- `curl -iX PATCH -d '{"10.11.12.13": "suspect"}' http://localhost/api/7/http/keyvals/one`
- `curl http://localhost/api/7/http/keyvals/one #Dump entire store from memory`

Static string mappings

- Similar to keyval pairs
- Part of configuration files
- Changes require configuration reload which is non disruptive
- Can be used like Switch/Case in limited scenarios

NGINX map

Search string: `ip_flagagent_flag`. Concatinated values of two variables.
When found the corresponding value is assigned to `$policy_type`.

```
map $ip_flag$agent_flag $policy_type {  
    "trusted_IPsuspect_agent"    "default";  
    "suspect_IP*"                "strict";  
    "trusted_IPsuspect_agent"    "medium";  
    default                      "strict";  
}
```