

Explicación de todos los pasos para apagar y encender la Bombilla Inteligente TP-Link Kasa Regulable KL110 desde PowerShell

Obtener el identificador universally unique identifier (UUID)

- <https://www.jesusninoc.com/12/11/crear-un-identificador-unico-universal-o-universally-unique-identifier-uuid-desde-node-js/>

Obtener el token y el deviceId que tiene la bombilla mediante Node.js

El token es un valor que establece una relación entre la bombilla y la aplicación Kasa. El deviceId es el identificador de la bombilla.

Para obtener el token y el deviceId de la bombilla es necesario el identificador universal obtenido en el paso anterior además del correo electrónico y la contraseña con la que se dió de alta en Kasa.

[crayon-6a9d4e8e720fa735087627/]

Apagar y encender la Bombilla Inteligente TP-Link Kasa Regulable KL110 desde PowerShell

Es necesario indicar el token y el deviceID.

```
$Token = "adsfsadf-adsfasdf"
```

```
# URI del servicio appServerUrl:  
'https://eu-wap.tplinkcloud.com'
```

```
$URI = 'https://eu-wap.tplinkcloud.com/?token='+$Token
```

```
# Cabeceras  
$Cabeceras = @{  
    'Content-type' = 'application/json'  
}
```

```
# Parámetros para encender y apagar
```

```
# Encender
```

```
## '{"method":"passthrough", "params": {"deviceId":  
"asdfsadf", "requestData":  
"{\"smartlife.iot.smartbulb.lightingservice\":{\"transition_li  
ght_state\": {\"on_off\": 1,\"transition_period\": 0}}}\"}'
```

```
# Apagar
```

```
## '{"method":"passthrough", "params": {"deviceId":  
"asdfsadf", "requestData":  
"{\"smartlife.iot.smartbulb.lightingservice\":{\"transition_li  
ght_state\": {\"on_off\": 0,\"transition_period\": 0}}}\"}'
```

```
$Body = '{"method":"passthrough", "params": {"deviceId":  
"asdfsadf", "requestData":  
"{\"smartlife.iot.smartbulb.lightingservice\":{\"transition_li  
ght_state\": {\"on_off\": 0,\"transition_period\": 0}}}\"}'
```

```
# Preparar la petición y obtener respuesta
```

```
$Respuesta = Invoke-RestMethod -Method Post -Uri $URI -Headers  
$Cabeceras -Body $Body
```

```
# Convertir respuesta en formato JSON y ver el resultado de la  
petición anterior
```

```
ConvertTo-Json $Respuesta.result
```