

## Speechtimer API

### 1.1.6.1

Can your rundown software send commands via HTTP? Speechtimer can also be controlled via HTTP API. A Streamdeck can also be used to control the timer. (API Request Plugin)  
The API does not support a running time display for your rundown software or Streamdeck/Companion.

Example calls

**Set time to 12 minutes:**

`http://127.0.0.1:55055/api?Function=SetTime&Value=12`

**Start timer:**

`http://127.0.0.1:55055/api?Function=Start`

**Stop timer:**

`http://127.0.0.1:55055/api?Function=Stop`

**Reset timer:**

`http://127.0.0.1:55055/api?Function=Reset`

**Adjust time (e.g., +30 seconds):**

`http://127.0.0.1:55055/api?Function=Adjust&Value=30`

**Adjust time (e.g., -30 seconds):**

`http://127.0.0.1:55055/api?Function=Adjust&Value=-30`

**blank screen ON**

`http://127.0.0.1:55055/api?Function=blankscreen&Value=on`

**blank screen OFF**

`http://127.0.0.1:55055/api?Function=blankscreen&Value=off`

To keep things simple, the API is not password protected. However, the API can be disabled. To do this, you need to edit app.py.

The API responses a command sent via http returns messages, for example:

```
{"message":"Time set to 12.0 min.,"status":"ok"}
{"message":"Timer started.,"status":"ok"}
{"message":"API access is disabled.,"status": "error"}
```

## Speehtimer API

Kann deine Rundown Software Kommandos via http senden? Speehtimer lässt sich auch via HTTP-API steuern. Auch ein Streamdeck lässt sich so einfach zur Steuerung des Timers verwenden. (API Request Plugin)  
Die API unterstützt keine laufende Zeitanzeige für deine Rundown Software oder Streamdeck/Companion.

Beispiel-Aufrufe

**Zeit setzen auf 12 Minuten:**

`http://127.0.0.1:55055/api?Function=SetTime&Value=12`

**Timer starten:**

`http://127.0.0.1:55055/api?Function=Start`

**Timer stoppen:**

`http://127.0.0.1:55055/api?Function=Stop`

**Timer zurücksetzen:**

`http://127.0.0.1:55055/api?Function=Reset`

**Zeit anpassen (z. B. +30 Sekunden):**

`http://127.0.0.1:55055/api?Function=Adjust&Value=30`

**Zeit anpassen (z. B. -30 Sekunden):**

`http://127.0.0.1:55055/api?Function=Adjust&Value=-30`

Um dies einfach zu halten, ist die API nicht per Passwort abgesichert. Die API lässt sich aber ausschalten. Dazu ist es nötig, die app.py zu editieren.

Ist API deaktiviert, schickt ein per http gesendeter Befehl Meldungen zurück:

```
{
  "message": "API-Zugriff ist deaktiviert.",
  "status": "error"
}
```

