

Slide 2 - Ripasso HTTP ESP32

- Richiesta HTTP
 - Linea iniziale
 - Headers
 - Body

```
// Esempio richiesta HTTP
```

```
POST /login HTTP/1.1
```

```
Host: example.com
```

```
User-Agent: Mozilla/5.0
```

```
Content-Type: application/x-www-form-urlencoded
```

```
Content-Length: 32
```

```
username=johndoe&password=123456
```

```
// Esempio URL: https://www.site.com/login?/post
```

```
// URL = Dove si trova la tua risorsa
```

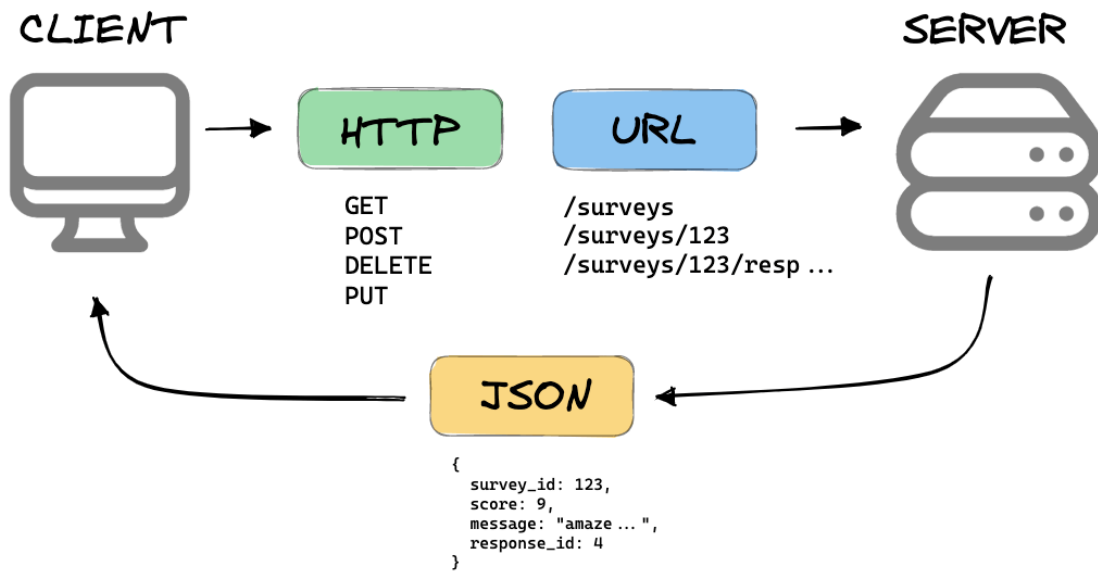
HTTP (Protocollo):

- Client (richiede servizi al fornitore - server)
 - Browser
 - HTTP-Request --> Pagina al server
- Server (fornisce servizi al cliente)
 - HTTP-Response --> Il contenuto della pagina

API (Application Programming Interface):

- Procedure automatiche che ti permettono di utilizzare come interfacce (servizi) facendo comunicare parti diverse
- Esempi
 - HTTP
 - POST
 - GET
 - Google Font
 - GoogleFontAPI.OpenSans
 - ChatGPT
 - ChatGPT api = new ChatGPT();
 - api.invoke();

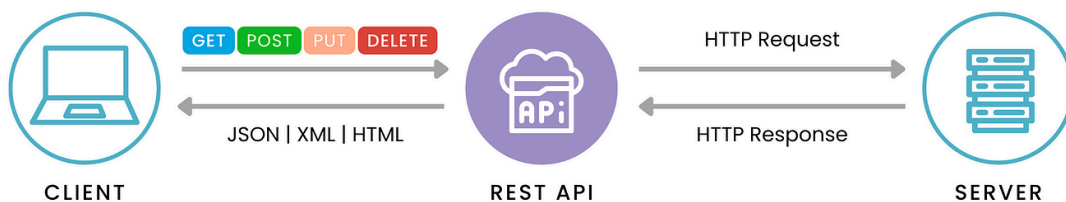
WHAT IS A REST API?



mannhowie.com

Slide 3 - REST

REST API Model



skiplevel.co

- REST -> Operazioni HTTP per dati
 - Operazioni standard di modifica / lettura / cancellazione
- Formati dati
 - XML / JSON

Slide 4 - HelloServer

```
WebServer server(80); // HTTP port: 80
```

```
Serial.begin(115200);

WiFi.begin(ssid, password); //Wi-Fi: connettersi

// Gestione delle diverse pagine con WebServer
server.on("/", handleRoot);
server.on("/inline",handleInline);
server.onNotFound(handleNotFound);

server.begin();
```

Slide 5 - Principi REST

- Risorse come URI
 - `site.com/sensors/temperature`
 - `site.com/actuators/relay1`
- Metodi HTTP
 - GET / POST / PUT
- Rappresentazioni JSON
 - `{"temperature": 25.4, "unit", "C"}`
- Stateless
 - Non salviamo cache