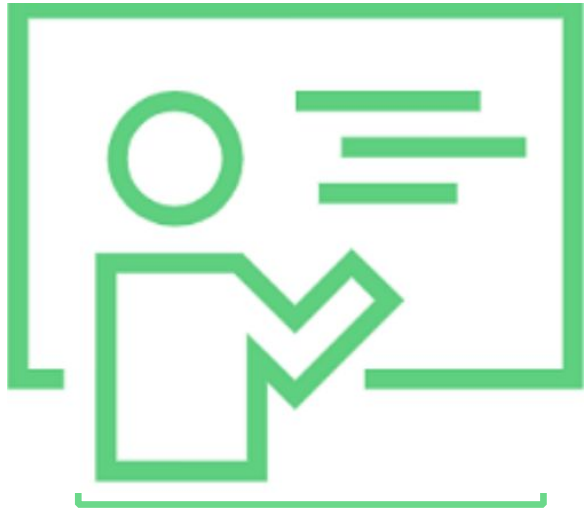




HTTP(S), API:er och web services

Joakim Englund, Systemmentor AB

Repetition från dag 2

Vad behövde vara rätt för att två program skulle kommunicera?

- Transportprotokoll
- IP-adress eller värdnamn
- Port
- Serverprocess som lyssnar
- Gemensam tolkning av datan

Från socket till web service

Klient → HTTP-request → API på servern → HTTP-response → Klient

Dagens mål

- Förstå request, response, endpoint och statuskod
- Skapa och testa ett enkelt API för sensordata
- Beskriva ett endpoint-kontrakt
- Jämföra REST och SOAP

HTTP-request och response

- Request: metod + sökväg + headers + body
- Response: statuskod + headers + body

URL:en hittar resursen

`http://127.0.0.1:8080/api/readings/latest`

Schema: ``http``

Värd: ``127.0.0.1``

Port: ``8080``

Sökväg: ``/api/readings/latest``

HTTP-metoder

<i>Metod</i>	<i>Typisk användning</i>
GET	Läs data
POST	Skicka eller skapa ny data
PUT	Ersätt en känd resurs
DELETE	Ta bort en resurs

https://www.tutorialspoint.com/http/http_requests.htm

API-kontraktet

POST /api/readings

Content-Type: application/json

Body: sensorId + value + unit

- Request: metod + sökväg + headers + body

Svar: 201 eller ett dokumenterat felsvar

- Response: statuskod + headers + body

Statuskoder berättar utfallet

<i>Kod</i>	<i>Betydelse</i>
200	Operationen lyckades
201	En resurs skapades eller data togs emot
400	Requesten är ogiltig
404	Resursen saknas
415	Innehållstypen stöds inte
500	Oväntat serverfel

https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

Vad är REST?

- Står för Representational State Transfer
- Resurser identifieras med URI:er (Uniform Resource Identifier)
- HTTP-metoder används med avsedd betydelse
- Varje request innehåller informationen som behövs
- Representationen kan vara JSON, XML eller något annat

URL - Uniform Resource Locator

REST i praktiken: sensormätningar

POST /api/readings	skapa en ny mätning
GET /api/readings/latest	läs den senaste mätningen
GET /api/readings/temp-01	läs data för en sensor

Exempel på request:

```
POST /api/readings
Content-Type: application/json
{"sensorId":"temp-01","value":21.7,"unit":"C"}
```

Så används REST oftast i ett IoT-system

ESP32/sensor → POST /api/readings → tjänst eller gateway
App/dashboard → GET /api/readings/latest → aktuell mätning
Driftverktyg → GET /api/devices/temp-01 → enhetsstatus

REST passar exempelvis för:

- Konfiguration och diagnostik
- Frågor efter aktuell eller historisk data
- Integration med appar och andra tjänster

Vad är SOAP?

SOAP Envelope

- └ Header
- └ Body
 - └ Operation + XML-data

- XML-baserat meddelandeprotokoll
- Formella kontrakt med WSDL (Web Services Description Language) är vanliga
- Etablerat i många enterprise-integrationer

REST och SOAP

<i>REST-baserat API</i>	<i>SOAP-webbtjänst</i>
Resurser och HTTP-semantik	Operationer i XML-kuvert
JSON vanligt	XML obligatoriskt
Ofta lätt att prova sig fram	Ofta strikt och kontraktsdrivet
OpenAPI vanligt	WSDL vanligt

<https://www.openapis.org/>

HTTP eller HTTPS?

HTTP = läsbar trafik

HTTPS = HTTP skyddat med TLS

HTTPS bidrar med:

- Konfidentialitet
- Dataintegritet
- Verifiering av servers identitet

Demo: API-kontrakt

| Metod och sökväg | Request | Lyckat svar | Vanliga felsvar |

|---|---|---|---|

| `GET /health` | Ingen body | `200` med status | – |

| `GET /api/readings/latest` | Ingen body | `200` med senaste mätning | `404`
om data saknas |

| `POST /api/readings` | JSON-mätning | `201` med sparad mätning | `400`,
`415` |

Demo - Testa utifrån kontraktet

1. Starta tjänsten
2. Testa hälsokontroll (`curl.exe -i http://127.0.0.1:8080/health`)
3. Skicka giltig data (`curl.exe -i http://127.0.0.1:8080/api/readings/latest`)
4. Läs tillbaka data (`curl.exe -i -X POST http://127.0.0.1:8080/api/readings ``
`-H "Content-Type: application/json" ``
`--data '{"sensorId":"temp-01","value":21.7,"unit":"C"}'`)
5. Skicka ogiltig data
6. Kontrollera statuskod och body

Var passar API:t i ett IoT-system?

Sensor → MQTT/buss → gateway eller tjänst → REST-API → app/analys

API direkt på enheten passar när:

- Lokal diagnostik eller styrning behövs
- Klientantal och datamängd är begränsade
- Nät och åtkomst kan kontrolleras

Dagens laboration

- Välj om API:t tar emot eller tillgängliggör sensordata
- Dokumentera minst ett endpoint-kontrakt
- Testa lyckade anrop och felsvar
- Koppla API:t till casets dataflöde