

REST

- Vad är API?
- Vad är ett Web API?
- Vad är ett RESTful API?

WEB API

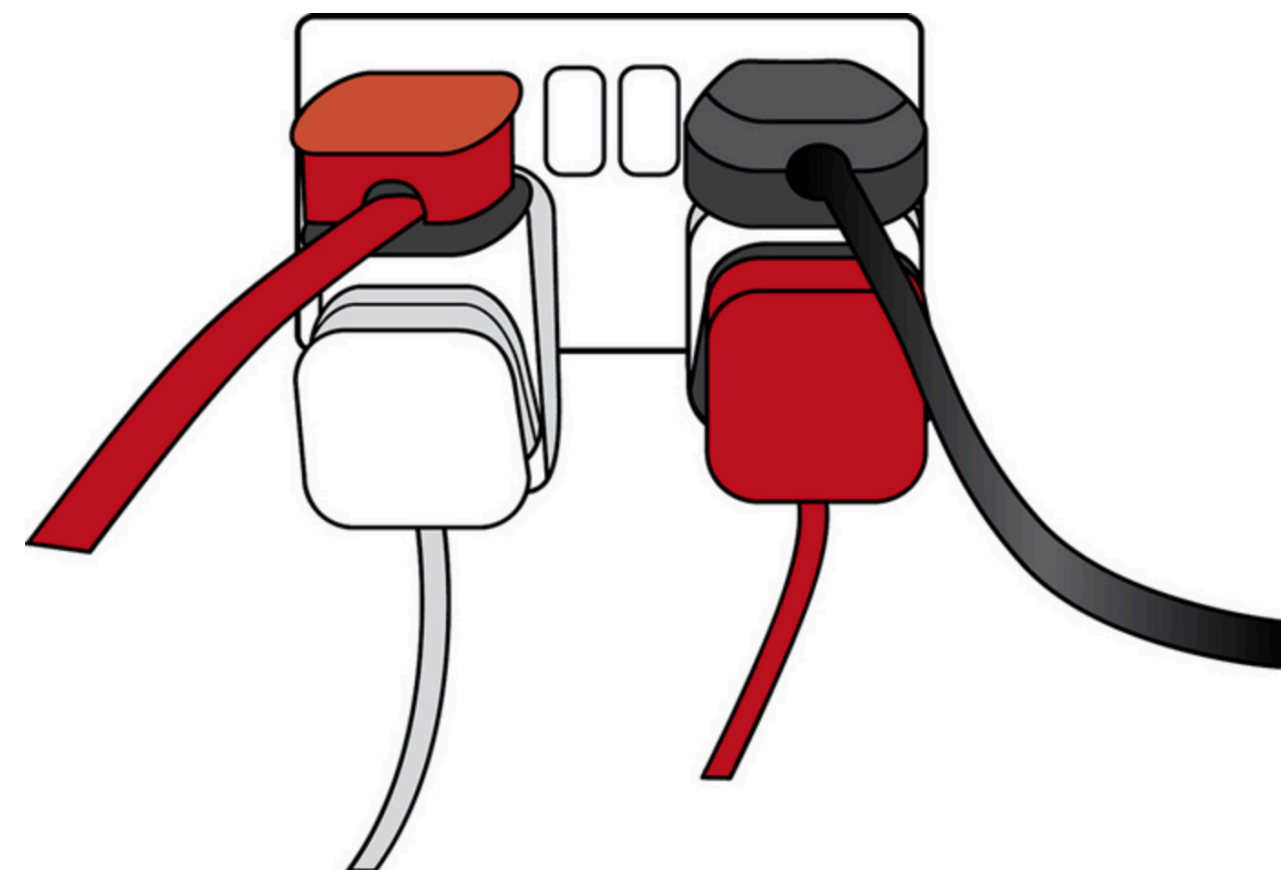
- API: Application Programming Interface
 - Mjukvara som tillåter 2 applikationer att prata med varandra.
 - Ex: När du använder en **mobil** och en app kopplar upp sig mot en **server** så skickas data fram och tillbaka via ett API oftast
 - Data?

WEB API

- Du har skapat en databas med Twitterinlägg sparade och vill exponera (sälja) data till företag som kan tänkas vara intresserade.
 - User: Patrik Nygren
 - Time: 2022-01-12, 13:45
 - Title: "DevOps"
 - Content: "Jag gillar devops! #devops"
- Fundera: Hur skulle du representera denna data?
 - Tänk på att det kan finnas flera inlägg
 - Tänk på att den här datan ska skickas över nätet flera gånger => använd lite tecken
 - Ingen data får gå förlorad

WEB API

- Tänk om alla applikationer i världen skickade tog emot data på samma sätt...



WEB API

- Web API
 - Ett Web API exponerar metoder för omvärlden så att
 - Data kan *tas emot*
 - Data kan *sändas*
 - Denna data brukar ha 2 olika format
 - XML (eXtensible Markup Language)
 - JSON (JavaScript Object Notation)
 - Dataformaten XML och JSON är standardiserade (IETF)

JSON

XML

```
<empinfo>
  <employees>
    <employee>
      <name>James Kirk</name>
      <age>40</age>
    </employee>
    <employee>
      <name>Jean-Luc Picard</name>
      <age>45</age>
    </employee>
    <employee>
      <name>Wesley Crusher</name>
      <age>27</age>
    </employee>
  </employees>
</empinfo>
```

JSON

```
{ "empinfo" :
  {
    "employees" : [
      {
        "name" : "James Kirk",
        "age" : 40,
      },
      {
        "name" : "Jean-Luc Picard",
        "age" : 45,
      },
      {
        "name" : "Wesley Crusher",
        "age" : 27,
      }
    ]
  }
}
```

■ XML vs JSON

JSON

- JSON values

- Boolean (true, false)

- Number (1.23, 10)

- String ("hejsan")

- Object ({ })

- "key" : value

- Array ([....])

```
{
  "Avengers": [
    {
      "Name" : "Tony stark",
      "also known as" : "Iron man",
      "Abilities" : [ "Genius", "Billionaire",
                     "Playboy", "Philanthropist" ]
    },
    {
      "Name" : "Peter parker",
      "also known as" : "Spider man",
      "Abilities" : [ "Spider web", "Spidy sense" ]
    }
  ]
}
```

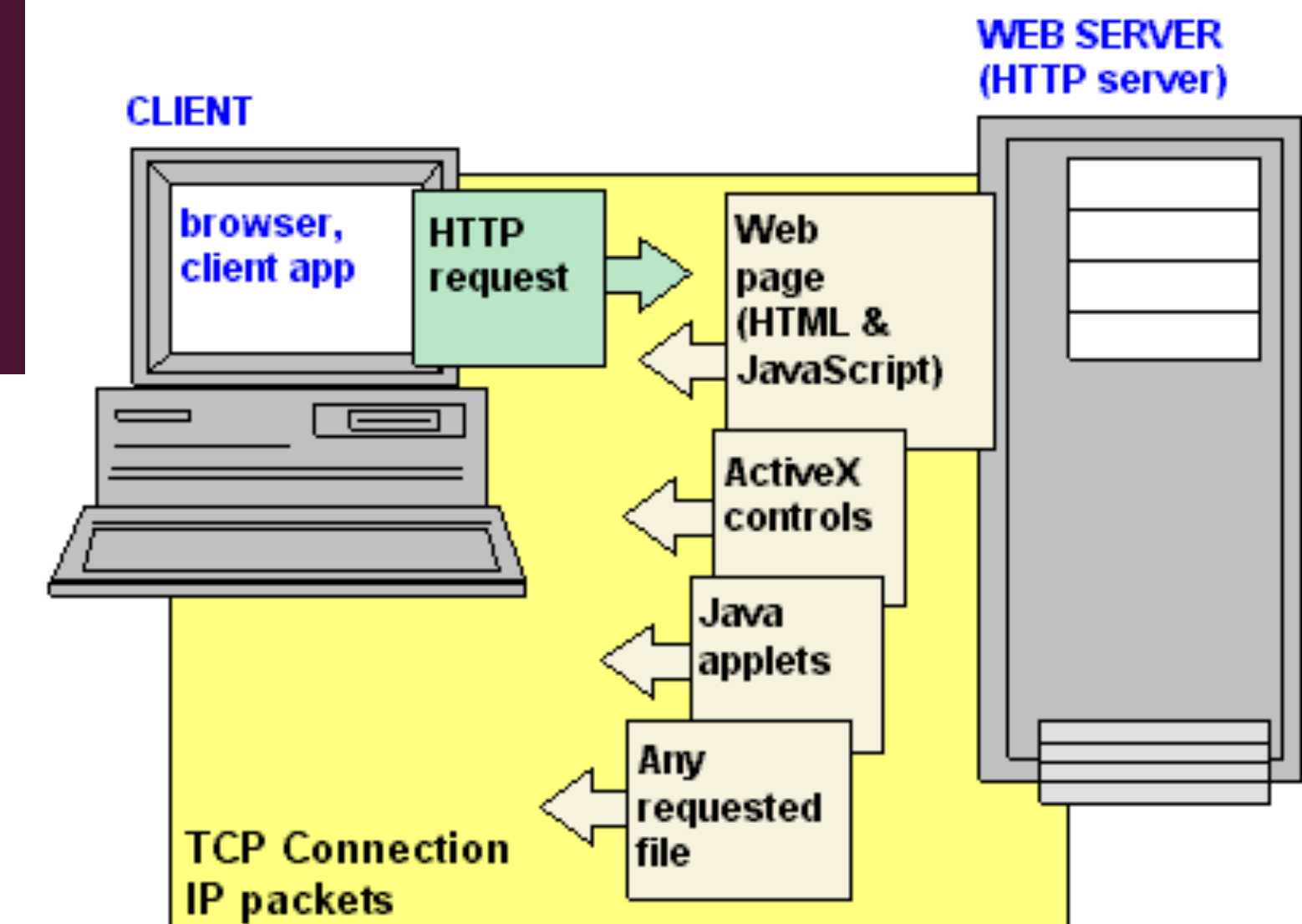
ÖVNING

- Övningar – Föreläsning 7, 1
- 15 min

REST

- Men nu har vi bara talat om standardisera dataformatet 🤔
- Finns det ett standardiserat sätt att få tag på datan?
 - Ska dom skapa en TCP uppkoppling via IP-adress?
 - <https://www.javatpoint.com/socket-programming>
 - Hur skulle ni göra?

REST



- Det finns ett protokoll som kallas HTTP/S (ni kanske känner igen det?)
 - Demo: Webbläsare, HTML, CSS, JavaScript
- Eftersom HTTP protokollet finns på alla datorer, kan man kanske återanvända det?
- Och istället för HTML, CSS, JavaScript kan vi skicka JSON/XML :-)

REST

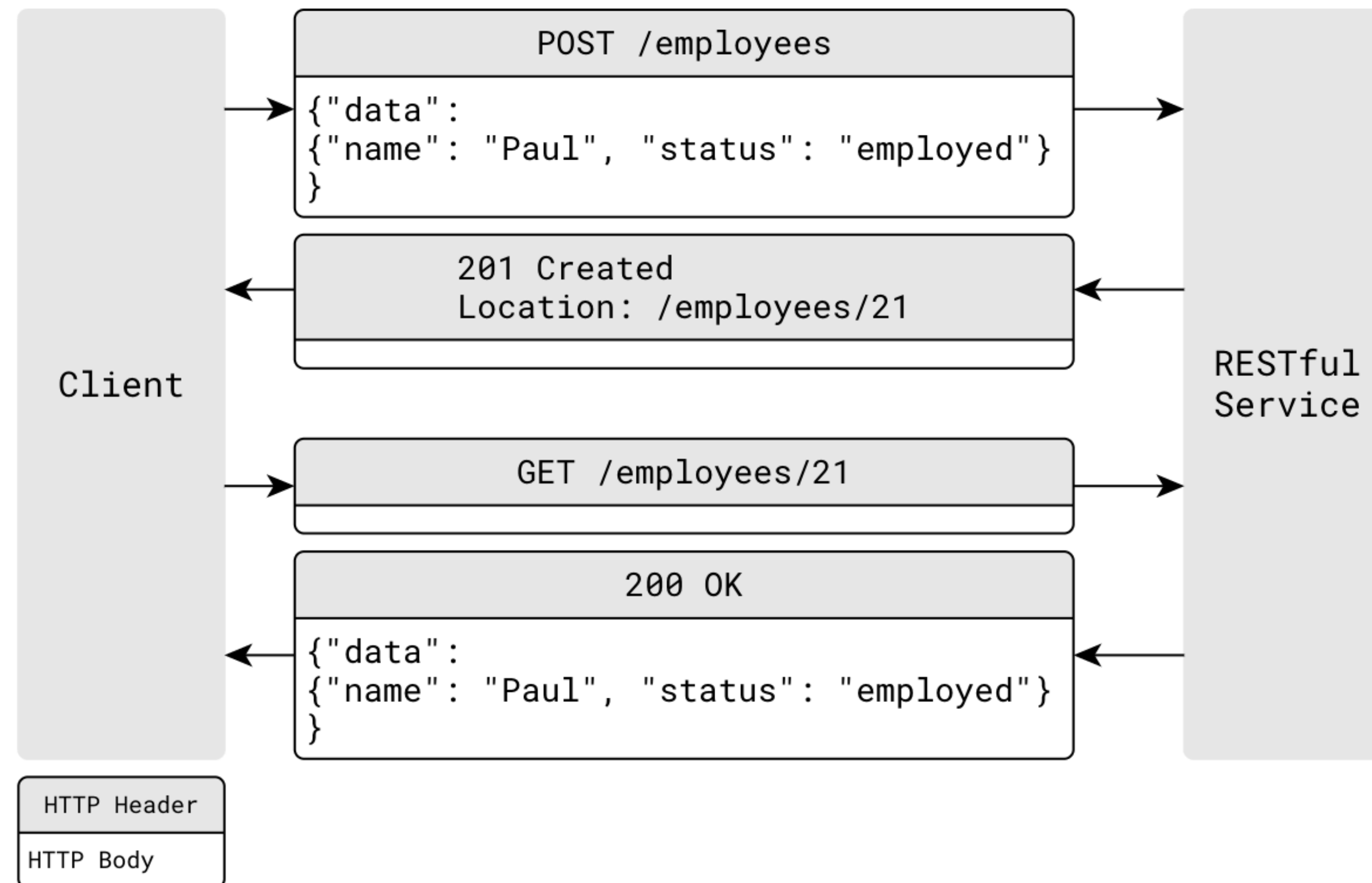
- REST = Representational State Transfer (Roy Fielding i en doktorsavhandling)
- "Resurser" är identifierade av en URL: **`http://myapi.org/v1/students/1`**
 - **`=> { "id": 1, "name": "Gorgy", "active": true, "courses": ["linux", "java", "sql", "agile"] }`**
- "Resurser" görs tillgängliga genom olika HTTP metoder
 - GET (används för att hämta HTML dokument)
 - PUT
 - POST (används för att skicka formulärdata i webbläsare)
 - DELETE
- Sammanfattningsvis: REST topprider HTTP protokollet!

REST

- REST är "Stateless" eftersom inget "tillstånd" behövs mellan de olika anropen
- Samma request (GET, POST, PUT, DELETE) kan göras flera gånger (idempotent)

REST - EXEMPEL

■ Exempel POST/GET



REST - EXEMPEL

■ Exempel PUT/GET

