Internetprogrammering
DD1386

Föreläsning 3

Innehåll

- Det som var kvar från förra föreläsningen, socket
- HTTP-Request
- HTTP-Response
- HTTP-Metoder
- Hantering tillstånd i HTTP-kommunikation
  - Headerfältet Set-cookie
  - Headerfältet Cookie
- PRG(POST/Redirect/GET)

Socket

Socket är gränssnitt mot TCP/IP och UDP/IP och används alltså för att upprätta en anslutning baserat på IP mellan två maskiner

- I java finns följande klasser implementerade
  - `java.net.Socket`
  - `java.net.ServerSocket`
- Klientens port slumpas

Server.java

```
import java.net.*;

import java.io.*;

public class Server{
     public static void main(String[] args) throws Exception{
     ServerSocket serverSckt = new ServerSocket(1234);

          Socket sckt = serverSckt.accept();
          while( (sckt != null){

                    BufferedReader indata = new BufferedReader(
                    new InputStreamReader(sckt.getInputStream()));

                    String text = null;
                    while( (text = indata.readLine()) != null){

                         System.out.println(text);

                    }

               sckt.shutdownInput();

               sckt = serverSckt.accept();

          }

     }

}
```

Client.java

```
import java.net.*;

import java.io.*;

public class Client{
   public static void main(String[] args){

      try{
         Socket sckt = new Socket("share-02.csc.kth.se",1234);

         PrintStream out = new PrintStream(sckt.getOutputStream());

         BufferedReader indata = new BufferedReader(

                                 new InputStreamReader(System.in));

         String text;
         while((text = indata.readLine()) != null)

              out.println(text);

         sckt.shutdownOutput();

      }catch (Exception e){

          System.err.println("Ett fel intraffade!");

      }

  }}
```

HTTP-request exempel

```
GET /~vahid/jag.jpg HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Accept-Language: sv-SE
User-Agent: Mozilla/5.0 (Windows NT 6.1…
Accept-Encoding: gzip, deflate
Host: www.csc.kth.se
Connection: close
```

HTTP-request

- Formatet för en http-request är:
  - Start line: <metod> <sökväg> <HTTP-version> ↲
  - Header 1: <variabel> : <värde> ↲
  - Header 2: <variabel> : <värde> ↲
  - Header 3: <variabel> : <värde> ↲
  - ...
  - Header N: <variabel> : <värde> ↲
  - ↲
  - Payload: Eventuell data...

OBS väldigt viktigt!! Två efterföljande radbrytningar innan payload börjar

HTTP-metoder

Viktiga Metoder	**Beskrivning**
GET	Frågar efter en resurs
POST	Skicka data till servern för att skapa en resurs

Mindre viktiga Metoder	**Beskrivning**
HEAD	Frågar efter information om en resurs
PUT	Skicka data till servern för att uppdatera en resurs
DELETE	Ta bort en resurs
TRACE	När förfrågan måste passera genom en proxy, gateway etc
OPTIONS	Frågar efter tillåtna (tillgängliga) metoder

Viktiga headerfält i en request

- Accept: vilka mime-type kan klienten hantera
- User-Agent: t.ex Mozilla/4.75[en] (Win98)
- Host: serverns domän namn och portnummer
- Cookie: används för identifiering
- Date: när requesten skapades

HTTP-response (binär exempel)

HTTP/1.1 200 OK↲

Date: Sun, 21 Jan 2018 15:27:34 GMT↲

Server: Apache↲

Accept-Ranges: bytes↲

Content-Length: 2946↲

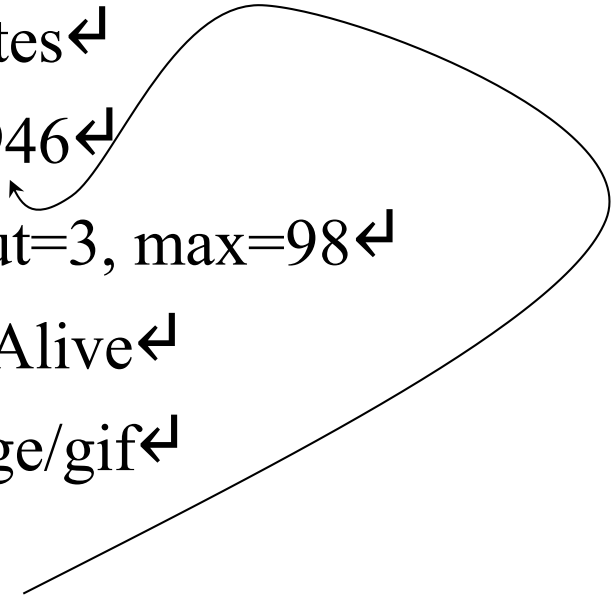
Keep-Alive: timeout=3, max=98↲

Connection: Keep-Alive↲

Content-Type: image/gif↲

↲

Här kommer att 2946 bytes data finnas



HTTP-response

- Formatet för en http-response är:
  - Start line: <http-version> <statuskod> <OK/Error>↲
  - header: <variabel> : <värde> ↲
  - …
  - header: <variabel> : <värde> ↲
  - ↲
  - payload: data...

OBS väldigt viktigt!! Två efterföljande radbrytningar innan payload börjar

- I header:n bör alltid ***Content-Type*** ingå för att webbläsaren ska kunna skilja t.ex text från binär information, etc. Anges i MIME-formatet type/subtype (t.ex, *text/html, image/jpeg*) och specificerar innehållet i *payload*.

Viktiga headerfält i en response

- Content-Type: mimetype
- Content-Length: storlek av data
- Server: info om server, t.ex: Apache
- Set-Cookie: skickar kaka
- Date: datum då servern skapade responsen

Hantering av tillstånd

- HTTP är tillståndslöst protokoll, d.v.s. det är designad så att det finns inga förutsättningar om att servern ska komma ihåg något om tidigare förfrågningar (requester).

- Med hjälp av cookies kan servern spara information om tidigare tillstånd:
  - Headern `Set-Cookie` i responsen

  - Headern `Cookie` i request

Cookies

- En cookie har ett namn och ett värde och sparas i webbläsarens minne.
- Servern definierar cookie:n i response header:n
  - Set-Cookie: <namn> = <värde>↲ (obligatorisk)
  - expires = <datum>↲ (frivillig)
  - domain = <domän>↲ (frivillig)
  - path = <bibliotek>↲ (frivillig)
- exempel: `Set-Cookie: JSESSIONID=751668B6904B; Path=/`↲

- Om klienten vid ett senare tillfälle försöker komma åt samma domän/sökväg skickas cookie:n med i request-header:n. – *"Cookie: JSESSIONID=751668B6904B;"*

Response med set-cookie

```
HTTP/1.1 200 OK↵
Content-type: text/html↵
Set-Cookie: status = morgonpigg↵
Set-Cookie:likes = 1038;path=/img↵
↵
[Payload]
```

Request med Cookie

```
GET /step1 HTTP/1.1
Host: u-shell.csc.kth.se
Cookie: vardagar = morgonpigg;
  likes = 1038
```

Format

- Set-Cookie: <name>= <value> ⟵ raderas när man stänger av webbläsaren
- Set-Cookie: <name>=<value>; Expires=<date>
- Set-Cookie: <name>=<value>; Max-Age=<non-zero-digit>
- Set-Cookie: <name>=<value>; Domain=<domain-value>
- Set-Cookie: <name>=<value>; Path=<path-value>

  …

Exempel:

- Set-Cookie: id= 47w123
- Set-Cookie: id= 47w123; Max-Age=3600
- Set-Cookie: id= 47w123; Expires=Fri, 21 Jan 2022 09:00:00 GMT;
- Set-Cookie: id= 47w123; Domain=kth.se

Exempel

Vi går igenom Form.java och lägger till Set-Cookie för den.

Statuskod

Typ av status	Beskrivning
1xx: Information	Kommunikation i protokoll-nivå
2xx: Success (framgång)	Förfrågan accepterades med framgång
3xx: Omdrigering	Klienten behöver utföra ytterligare handling
4xx: Klientfel	Ett fel inträffade p.g.a. klienten
5xx: Serverfel	Ett fel träffade p.g.a servern

Form.java

1. Läs request
2. Kolla om det är GET, POST eller annan metod
3. Om GET
    1. Om cookie inte finns i requesten svara med formulär för att kunna skicka in förnamnet.
    2. Om cookie finns hämta data från minnet och avgör nästa rutt.

Form.java

4. Om POST:
    1. Läs cookie och hämta tillhörande data från minnet
    2. Om förnamnet/efternamnet är null svara med formuläret för att kunna skicka in förnamn/efternamn
    3. Om bara förnamn/efternamnet är null svara med final-formuläret samt se till att ogiltigtgöra cookien för klienten samt ta bort alla tillhörande data till den klienten
5. Om annan metod: ignorera
6. stäng av förbindelse

Form.java

Koden finns i Canvas

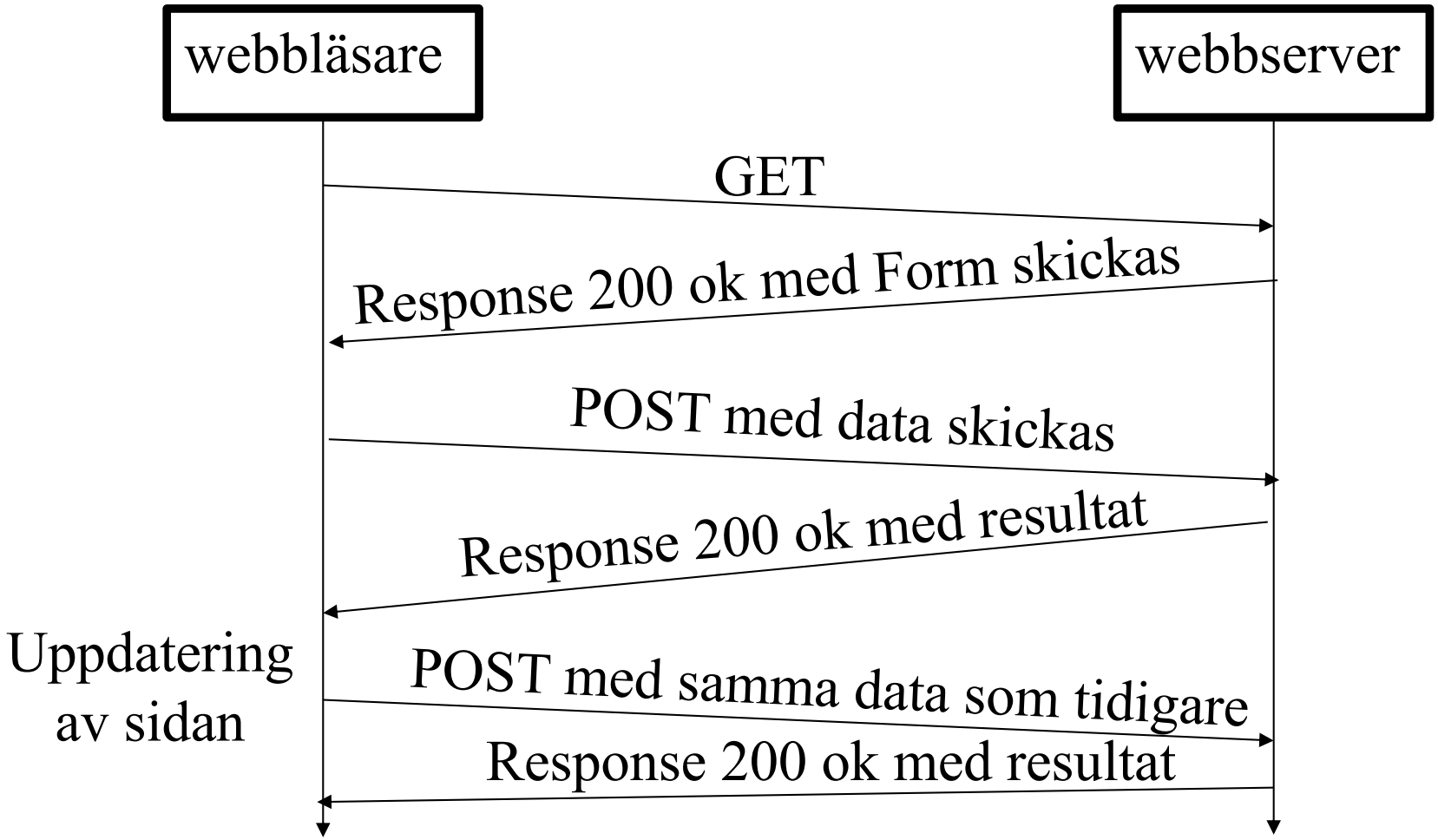
PRG (Post/Redirekt/Get)

- Designmönster
- Vilket problem löser PRG?

  Svar: Högre säkerhet vid användning av POST-request
- Hur?

  Jämför exemplen No_RPG.java och PRG.java

No_PRG.java i bild



PRG.java i bild