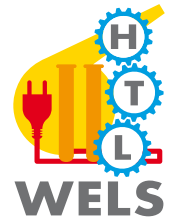


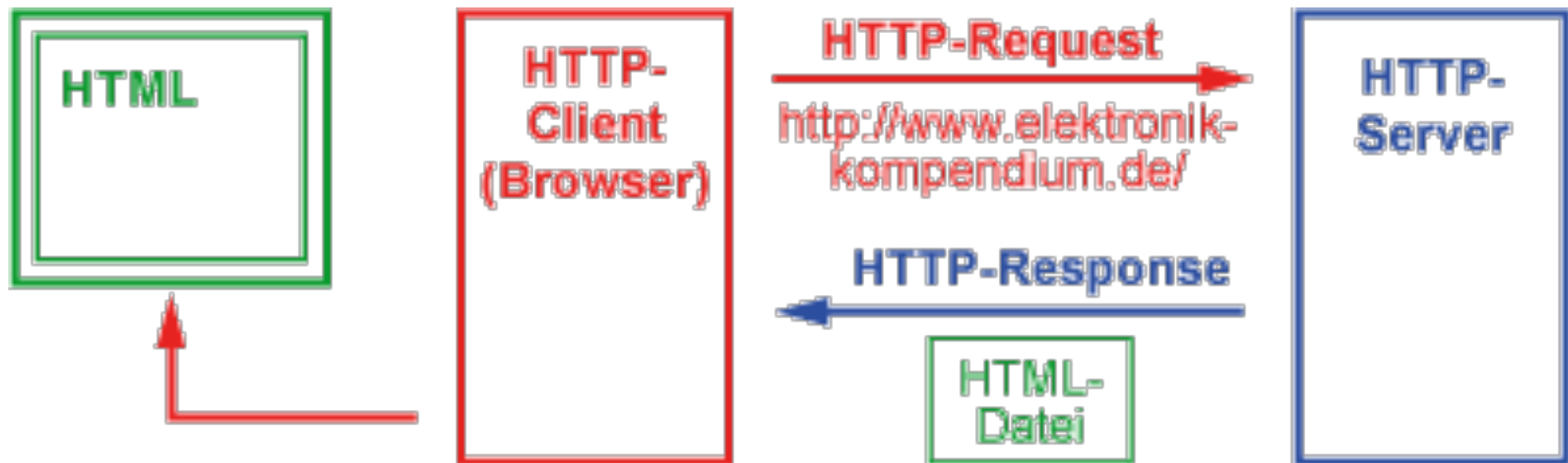


HTTP / WEBSERVER



➤ HTTP Client/Server

1. Server wartet auf eingehende HTTP Anfragen
2. Client erzeugt einen URL: `http://...`
3. Client versucht, TCP Verbindungen zum Server aufzubauen
4. Server akzeptiert Verbindungswunsch
5. Client schickt Nachricht (HTTP-Anfrage) und fordert Ressource mit spezifizierter URL an
6. Server verarbeitet Anfrage (generiert u.U. HTML Seite)
7. Server schickt Rückantwort (HTTP-Antwort) mit Ressource oder Fehlercode
8. Client verarbeitet Antwort
9. Client und/oder Server schließen TCP Verbindung



➤ HTTP Request

Methode	URL	Version	CR	LF
---------	-----	---------	----	----

Parametername	:	Wert	CR	LF
---------------	---	------	----	----

⋮

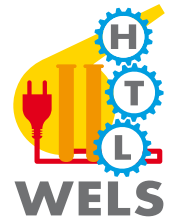
Parametername	:	Wert	CR	LF
---------------	---	------	----	----

CR	LF
----	----

Sendedaten bei Post-Methode



HTTP / WEBSERVER



Connection: close

Trotz HTTP/1.1 soll die Verbindung nach dem Reply geschlossen werden

User-agent: xxx

Gibt den Namen des verwendeten Browsers und seine Versionsnummer an

Accept: text/html...

Gibt an, welche Dateiformate der Browser als Antwort akzeptiert

Accept-language: de

Gibt die bevorzugte Sprache des Client an. Der Server kann darauf mit einer Seite in der richtigen Sprache reagieren

Cookie: xxxx

Der Client hatte zuvor ein Cookie vom Server erhalten. Er sendet es jetzt mit jedem weiteren Request an diesen Server mit. So kann der Server einzelne Requests immer sicher einem bestimmten Kunden zuordnen, z.B. für einen Warenkorb.

Authorization: xxx

Sobald der Server für eine Seite einen Benutzernamen und Passwort verlangt, werden diese mit jedem folgenden Request an diesen Server mitgeschickt.

➤ Response

Version	Statusnr	Statustext	CR	LF
---------	----------	------------	----	----

Parametername	:	Wert	CR	LF
---------------	---	------	----	----

⋮

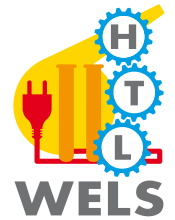
Parametername	:	Wert	CR	LF
---------------	---	------	----	----

CR	LF
----	----

Daten des abgerufenen Dokuments, z.B. HTML-Seite, GIF-Bild...
--



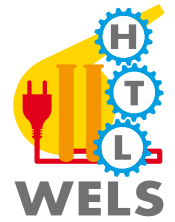
HTTP / WEBSERVER



200 OK	Alles klar, die angeforderte Datei ist im Paket enthalten
301 Moved Permanently	Objekt wurde verschoben. Die neue Adresse steht im Parameterfeld "Location:". Der Client lädt diese automatisch
400 Bad Request	Der Server hat den HTTP-Request nicht verstanden
401 Authorization Required	Der Zugriff auf die Seite erfordert die Freischaltung über einen Benutzernamen und ein Passwort. Der Browser erfragt dies vom Anwender und sendet den HTTP-Request mit diesen Angaben erneut zum Server. Jeder weitere ab diesem Zeitpunkt zum gleichen Server gesendete HTTP-Request enthält die Autorisierungsdaten ebenfalls
404 Not Found	Das angeforderte Dokument existiert nicht



HTTP / WEBSERVER



Request:

```
get /index.php HTTP/1.1
```

```
Host: localhost:8888
```

```
...
```

Response:

```
HTTP/1.1 200 OK
```

```
Date: Thu, 12 Nov 2015 07:27:35 GMT
```

```
Server: Apache
```

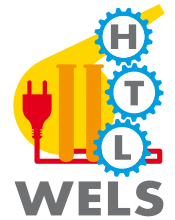
```
X-Powered-By: PHP/5.6.10
```

```
Content-Length: 1850
```

```
Content-Type: text/html; charset=UTF-8
```



HTTP / WEBSERVER



➤ Test mit Telnet:

```
telnet www.htl-wels.at 80
```

```
Trying 10.34.57.251...
```

```
Connected to www.htl-wels.at.
```

```
Escape character is '^['.
```

```
GET / HTTP/1.1
```

```
HOST: www.htl-wels.at
```

HTTP Request

```
HTTP/1.1 302 Found
```

```
Date: Mon, 12 Nov 2018 10:33:31 GMT
```

```
Server: Apache/2.4.18 (Ubuntu)
```

```
Location: https://www.htl-wels.at/
```

```
Content-Length: 289
```

```
Content-Type: text/html; charset=iso-8859-1
```

HTTP Response

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
```

```
<html><head>
```

```
<title>302 Found</title>
```

```
</head><body>
```

```
<h1>Found</h1>
```

```
<p>The document has moved <a href="https://www.htl-wels.at/">here</a>.</p>
```

```
<hr>
```

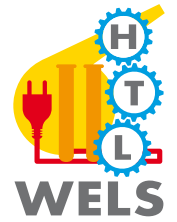
```
<address>Apache/2.4.18 (Ubuntu) Server at www.htl-wels.at Port 80</address>
```

```
</body></html>
```

```
Connection closed by foreign host.
```



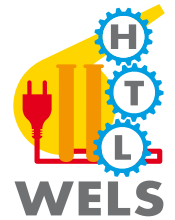

AUFGABE



➤ Aufgabe **WebServer**



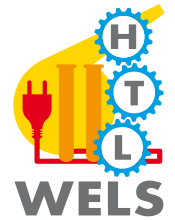
SPEZIELLE NETZWERKKLASSEN



- URL
- URLConnection
- InetAddress (Inet4Address + Inet6Address)
- NetworkInterface



QUELLEN



- <https://docs.oracle.com/javase/tutorial/networking/sockets/index.html>
- <http://phoenix.goucher.edu/~kelliher/s2011/cs325/>
- HERDT Buch Java 8 Fortgeschrittene Programmierung