

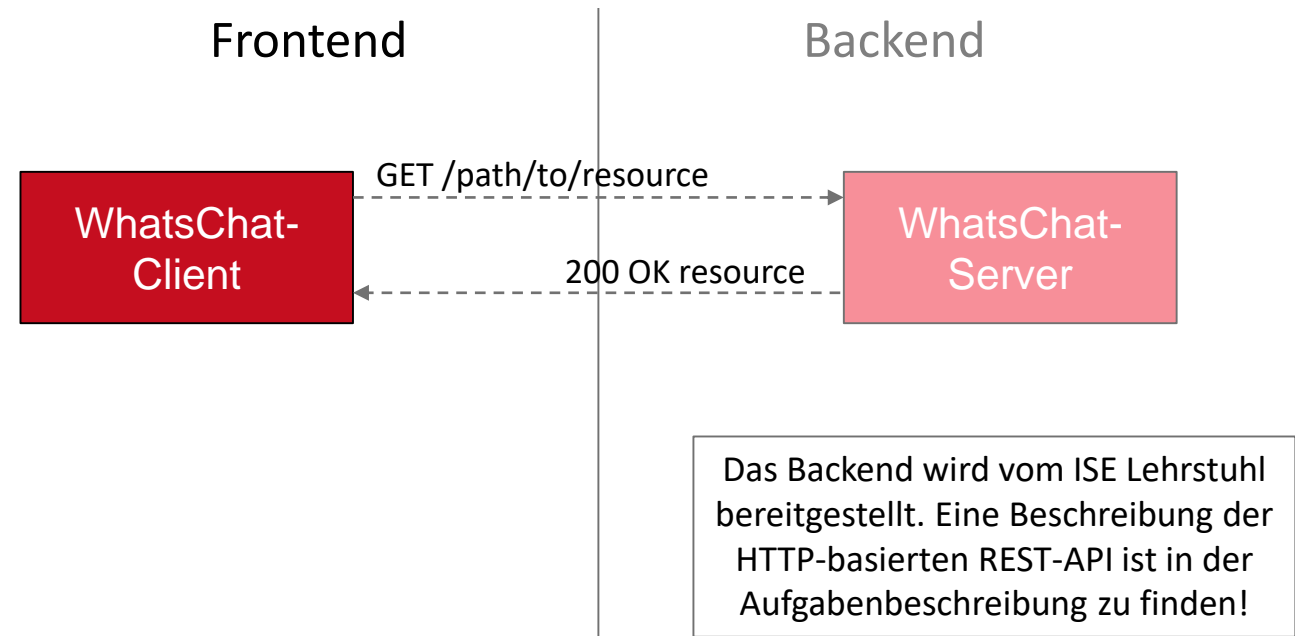
# Übung 1 – Bewertete Programmieraufgabe

# ChatAnwendung: WhatsChat

Sie werden einen Browser-basierten **Client** für eine Chat-Anwendung implementieren!

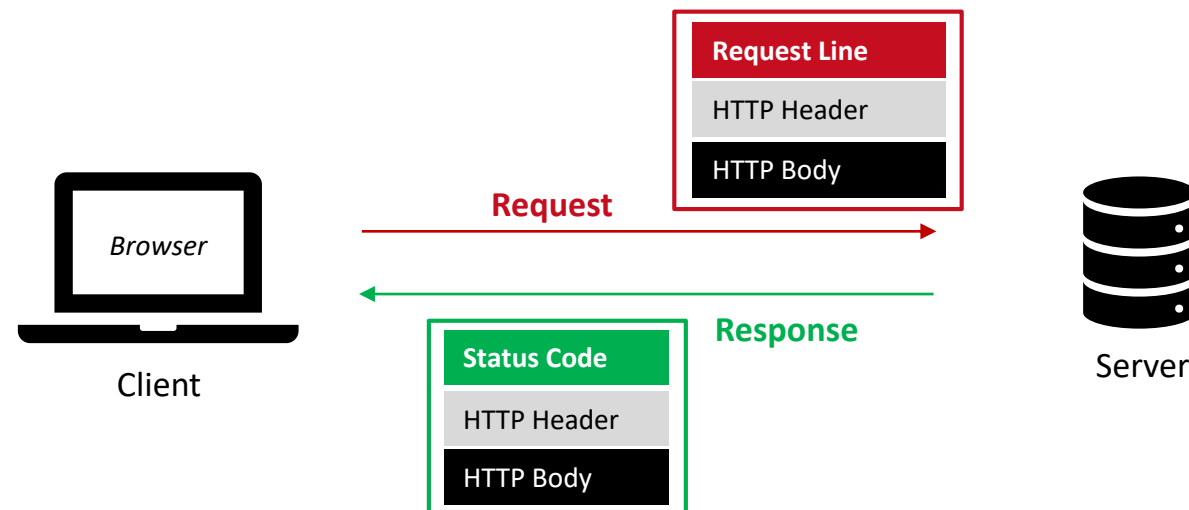
## Technologien (Frontend):

- HTTP
  - Kommunikation mit dem Backend
- HTML/CSS
  - Strukturierung und Darstellung der Inhalte
- JavaScript
  - Nutzerinteraktion und Manipulation der HTML Seite

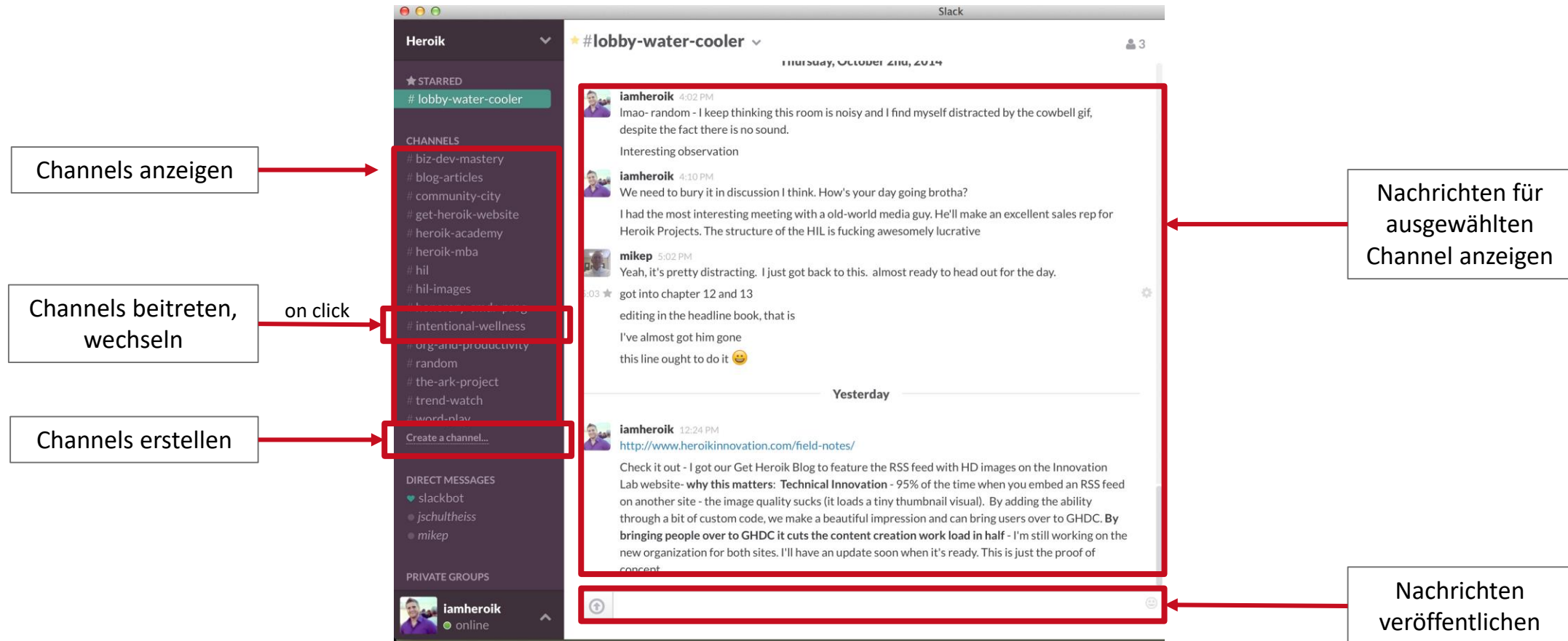


## Recap: HTTP Requests and Responses

- Typically, an HTTP client establishes a TCP connection to a particular port on a host (port 80 by default)
- An HTTP server *listening* on that port waits for the client to send a request message
- Upon receiving the request, the server sends back a *status* line, such as "HTTP/1.1 200 OK", and optionally the body of which is perhaps the requested resource, an error message or some other information.



# WhatsChat Funktionalität



# Nachrichten und Zeitstempel

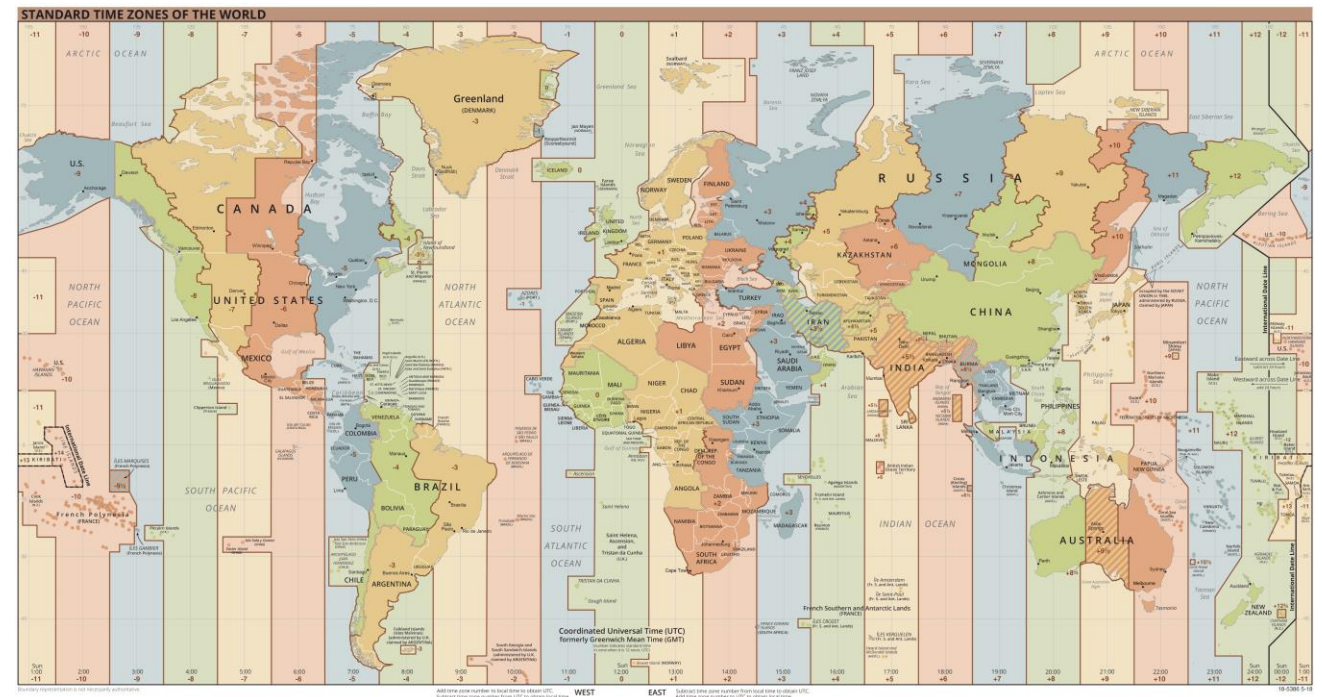
- Beim Betreten eines Channels sollen die jüngsten 10 Nachrichten im Verlauf angezeigt werden. Die Logik dafür ist Server-seitig implementiert (siehe API Dokumentation).
- Sobald der Nutzer den Channel betreten hat, soll der Nachrichtenverlauf regelmäßig mit dem Backend aktualisiert werden, d.h. der aktuell angezeigte Verlauf soll um die jüngsten Nachrichten ergänzt werden. Der entsprechenden GET Anfrage muss dafür der Zeitstempel der letzten im Verlauf sichtbaren Nachricht als Query-Parameter in der URL beigefügt werden.
- Zeitstempel werden ausschließlich Server-seitig generiert.

# Umrechnen: UTC → MEST

ACHTUNG: Server-seitig wird die *Coordinated Universal Time* (UTC) verwendet!

→ Zeitstempel müssen Client-seitig in Lokalzeit *Middle European Summer Time* (MEST) umgerechnet werden bevor sie angezeigt werden!

UTC	(0)
MET	(UTC + 1)
MEST	(UTC + 2)



<https://www.calculator.net/time-zone-calculator.html>

# Werkzeuge

- Browser (Chrome, Firefox,...)
- Integrierte Entwicklungsumgebung oder Editor: Eclipse, IntelliJ, VSCode
- Postman: APIs Testen
- Lookups:
  - Hilfestellungen und Tutorials im Bereich Webentwicklung und Webdesign: <https://www.w3schools.com/>
  - jQuery API: <https://api.jquery.com/>

# Demo



## Organisatorisches: Access Token

- Zur Authentifizierung gegenüber dem Backend erhält jede Gruppe ein Access-Token per Email.

Key → **X-Group-Token: vdfsoriafnns** ← Value

- Das Access-Token muss bei jedem Request im Header mitgegeben werden. Falls nicht, wird ein 403 Status Code zurückgegeben.

## Organisatorisches: Artefakte zur Abgabe

- Eine **\*.zip Datei**, die den kompletten Code beinhaltet (drei Dateien: \*.html, \*.css, \*.js)
- Eine **Präsentation in PDF Format**, die die Dokumentation und Beschreibung des Codes beinhaltet, das heißt:
  - Eine Beschreibung und Abbildung (z.B. lesbare Screenshots) der Codeauszüge, die benötigt waren, um die Anforderungen umzusetzen.
  - Screenshots der Browser-Darstellung des WhatsChat Clients. Die Umsetzung jeder geforderten funktionalen Anforderungen soll erkennbar sein.
  - Beschreibung und/oder Kennzeichnung der Beiträge der einzelnen Gruppenmitglieder zur Bearbeitung der Programmieraufgabe (wer hat was gemacht?).
  - Das Deckblatt der Präsentation muss die Namen und Matrikelnummern der beteiligten Gruppenmitglieder beinhalten.

## Organisatorisches: Abgabefrist

- Die Artefakte zur Abgabe müssen bis **Sonntag, den 16.06.2019, bis 23:55** eingereicht werden (3½ Wochen).
- Laden Sie dafür bitte die entsprechenden Dateien im ISIS hoch.
- Es genügt, wenn ein Gruppenmitglied die Artefakte hochläd.
- Nach Verstreichen der Frist werden keine Abgaben mehr zugelassen!!!

# Fragen?