

Tools

- ❑ Java
- ❑ Text Editor
- ❑ Git
- ❑ Telnet
- ❑ Integrated Development Environment (IDE)

Java

Installieren

Für Programmieraufgaben wird wie in der Vorlesung Java verwendet.

- ❑ Windows & macOS:
 - [Java SE Development Kit 8](#) von Oracle herunterladen und installieren
- ❑ Ubuntu & Debian:
 - OpenJDK 8 via Paketmanager installieren
- ❑ Die Version von `javac` und `openjdk` sollte `1.8.0_111` oder neuer sein. Das lässt sich mit folgenden Befehlen überprüfen:

```
sudo apt update && sudo apt install openjdk-8-jdk
```

```
javac -version
```

```
java -version
```

Java

Kompilieren und ausführen

- Aus dem Quelltext wird zunächst eine Byte-Code-Datei erzeugt:

```
javac Server.java FileRequestHandler.java
```

- Die `Server.class` wird dann mit folgendem Befehl ausgeführt:

```
~/dev/file-server$ java Server 8080  
Server started on port 8080.
```

Text Editor

Sublime Text, Atom, Visual Studio Code, VIM

- ❑ Sublime Text (Jon Skinner)
 - Closed Source, kostenpflichtig (aber ohne Einschränkungen *testbar*)
 - bewährt, sehr schnell, leider kein gutes Git Package
- ❑ Atom (GitHub)
 - Open Source, kostenlos
 - nicht sehr schnell, sehr gute Git-Integration
- ❑ Visual Studio Code (Microsoft)
 - Open Source, kostenlos
 - sehr neu, schnell, gute Git-Integration, gute \LaTeX integration
- ❑ VIM (Bram Moolenaar)
 - Open Source, kostenlos
 - extrem schnell, extrem steile Lernkurve

Integrated Development Environment (IDE)

Eclipse, IntelliJ IDEA

Neben Texteditoren werden beim Schreiben von Java-Code häufig auch IDEs benutzt. Sie bieten Werkzeuge wie eine mächtige Autovervollständigung und integrierte Debugger, die für fortgeschrittene Entwickler oft unverzichtbar sind.

- ❑ [Eclipse](#) (Eclipse Foundation)
 - Open Source, kostenlos
- ❑ [IntelliJ IDEA](#) (Jet Brains)
 - Community Edition: Open Source, kostenlos
 - Ultimate Edition: Closed Source, kostenpflichtig (als Student kostenlos)

Git

Sourcecodeverwaltung

Git ist ein Programm zur Versionierung von Source Code. Es ist ein essenzielles Werkzeug beim kollaborativen Programmieren. Wir verwenden Git für die Abgabe der Belege!

- ❑ Installation unter Ubuntu/Debian:

```
~$ sudo apt install git
```

- ❑ Unter Windows kann man [Github Desktop](#) benutzen. Dies installiert sowohl eine GUI, als auch eine vorkonfigurierte Posh. Diese funktioniert so wie Git im Linux-Terminal.
- ❑ Alternativen (inklusive Terminal-only Versionen und Editor integration) gibt es unter [Git SCM](#).

Git

Essenzielles Git für die Belege

Wir stellen für jede Programmieraufgabe ein Repository auf unserem [GitLab](#) bereit. Dort müsst ihr euch mit euren Uni-Login anmelden und bekommt dann Zugriff auf den Code.

Das Abgabeverfahren läuft folgendermaßen:

1. Forkt das Repository über GitLab (siehe jeweiliger Beleg).
2. Cloned das Repository auf euer System

```
~/code/$ git clone  
https://git.webis.de/username/netspeak-requests.git
```

3. Schreibt das Programm und Committed die Änderungen

```
~/code/netspeak-requests/$ git commit -am "Wohoo, it's finally  
working"
```

4. Übernimmt die Commits in das Remote Repository

```
~/code/netspeak-requests/$ git push
```

Überprüfen von HTTP-Anfragen

Telnet, Curl und Kitty

Das Öffnen der meisten URLs im Browser kommt einem GET-Request gleich. Einfache Tests sind also auch im Browser möglich. Ausführlicher geht das mit Tools wie Telnet, Curl und Kitty.

- Telnet-Session starten und HTTP-Anfrage absetzen

```
~$ telnet localhost 8080
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET / HTTP/1.1
Method handle not implemented.
Connection closed by foreign host.
```

`localhost` ist ein reservierter Domainname für den Rechner, den ein Nutzer gerade bedient und wird in aller Regel vom Betriebssystem zur IP `127.0.0.1` (IPv4) bzw. `:::1` (IPv6) aufgelöst.

Telnet

Telnet unter Windows

Der Telnet-Client ist unter Windows installiert, aber standardmäßig deaktiviert. So lässt er sich aktivieren:

1. Start → Systemsteuerung
2. Programme
3. Windows-Funktionen aktivieren oder deaktivieren
4. Telnet-Client aktivieren
5. OK

Auf manchen Systemen werden beim Eingeben der Anfrage keine Zeichen im Konsolenfenster sichtbar. Das Programm funktioniert trotzdem – es gibt lediglich kein Feedback über die gemachten Eingaben.

Alternativ kann unter Windows auch [Kitty](#) benutzt werden.

Handle-Methode

Ausschnitt aus `FileRequestHandler.java`

Die `handle`-Methode schreibt eine Antwort in das `response`-Objekt, die der Server schließlich ausgibt.

```
public void handle(String request, OutputStream response)
throws IOException {
    response.write("handle method not implemented.".getBytes());
    response.write(NEW_LINE.getBytes());
    // instead of using NEW_LINE, you can also use:
    // response.write("handle method not implemented.\n".getBytes());
}
```

Bemerkungen

Java Classpath

- Das `classpath`-Argument ist eine Möglichkeit anzugeben, wo die Java Virtual Machine nach Dateien suchen soll, die vom Programm benötigte Klassen definieren
- Dabei kann ein Verzeichnis relativ zum aktuellen Arbeitsverzeichnis angegeben werden. Folgender Aufruf würde dafür sorgen, dass die JVM auch im Unterverzeichnis `lib` sucht:

```
~/dev/file-server$ java -classpath lib/* Server 8080
```

- Befinden sich die Dateien in einem Java Archive (JAR), können diese direkt angegeben werden:

```
~/dev/file-server$ java -classpath lib/library.jar Server 8080
```