

File Server

- ❑ Java New I/O
- ❑ **Java** SimpleDateFormat

Java New I/O

Grundlegende Dateisystem-Operationen mit `Path` und `Files`

- ❑ `java.nio.file.Path`-Interface [\[Javadoc\]](#)
 - Eine `Path`-Instanz stellt einen Pfad im Dateisystem dar
 - Zeigt auf eine Datei oder ein Verzeichnis
 - Ist absolut oder relativ

- ❑ `java.nio.file.Files`-Klasse [\[Javadoc\]](#)
 - Stellt Methoden zur Dateimanipulation bereit
 - Arbeitet mit `Path`-Objekten

Java New I/O

Anwendungsbeispiel: Statuscode ermitteln

```
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;

public class JavaNioExample {
    /**
     * Determine the status code for an HTTP request
     * of the following form:
     * GET /path HTTP/1.1
     *
     * 200 (OK) for a valid HTTP 1.1 GET request
     * 404 (Not Found) for a non-existing path location
     */
    public int getStatusCode(String request) {
        Path path = Paths.get(request.split(" ")[1]);

        if (!Files.exists(path)) {
            return 404;
        } else {
            return 200;
        }
    }
}
```

Java SimpleDateFormat

Formatieren von `Date`-Objekte

- ❑ `java.text.SimpleDateFormat`-Klasse [\[docs\]](#)
 - Einfaches Beschreiben von Datenformaten
- ❑ `java.text.DateFormat`-Klasse [\[docs\]](#)
 - Formatiert ein `Date`-Objekt mittels einem `Format` (wie `SimpleDateFormat`)

```
import java.util.Date;
import java.text.DateFormat;
import java.text.SimpleDateFormat;

public class SimpleDateFormatExample {
    public String getHttpDate() {
        // Proper date format as per RFC 2616 (HTTP/1.1):
        // https://tools.ietf.org/html/rfc2616#section-3.3.1
        String httpDateFormat = "EEE, dd MMM yyyy HH:mm:ss zzz";
        DateFormat dateFormat = new SimpleDateFormat(httpDateFormat);
        return dateFormat.format(new Date());
    }
}
```