



packaging in jar

- Der `.` (Punkt) am Ende sagt aus, dass alle Dateien im aktuellen Verzeichniss angewendet werden.
- Im obigen Befehl ist der `-C` Mit dem Flag wird das Verzeichnis angegeben, aus dem der Inhalt der JAR-Datei entnommen werden soll. Das `.` (Punkt) nach dem `-C` Flag gibt das aktuelle Verzeichnis an. Das `.` (Punkt) nach dem `-C` Das Flag wird auch verwendet, um den Ausgangspunkt für die Glaserstellung anzugeben.

Das `--verbose` Mit der Option werden detaillierte Informationen zum Jar-Erstellungsprozess angezeigt.

Schließlich die `--main-class` Option wird verwendet, um die Hauptklasse für die JAR-Datei anzugeben, die sein sollte App in diesem Fall.

```
>> java  
  
jar --create --file Hello.jar -C . . --verbose --main-class de.iad.MainApp .
```

- mit folgenden Befehl führen wir jar aus

```
>> powershell  
  
.\build.ps1  
cd .\out\  
jar -c -f my-app.jar -v --main-class MainApp .  
java -jar .\my-app.jar
```

```
>> powershell  
  
jar -c -f MyApp.jar -v --main-class MainApp .
```

```
>> powershell
```

```
java -jar C:\dev\MyApp.jar --argument1 --argument2 .
```

packages anlegen

- ein java package ist eine Verzeichnisstruktur, in der Quelltext-Dateien und class-Dateien für den compiler hinterlegt sind.

classPath

- dieser ist eine Liste von Verzeichnissen & JAR-Dateien getrennt durch Semikolon
- unter Windows

```
>> powershell  
  
java -d output --class-path ".;C:\libs\*;C:\vendor\my-packages"
```

generelle Aufruf

- unter Windows

```
>> powershell  
  
jar --create --file <filename-of-jar> --main-class <full-qualified-name-of-main-class>
```