





Welcome to

## 2. Starting out with Camel

KEA System Integration F2020 10 ECTS

Henrik Kramselund Jereminsen [hkj@zencurity.com](mailto:hkj@zencurity.com) @kramse  

Slides are available as PDF, [kramse@Github](https://github.com/kramse)

2-starting-out-camel-system-integration.tex in the repo [security-courses](#)

# Plan for today



## Starting out with Camel

- Java recap, Java tools needed
- Book chapters and examples
- Running Camel - using Maven

## Exercises

- Book examples from chapters 1 and 2
- Various sites and trying out Camel

# Reading Summary



Camel book chapter 1: Meeting Camel

Camel book chapter 2: Routing with Camel

Browse / skim this:

Not put into the lecture plan, but if the concepts presented today seems hard, lookup on wikipedia:

Java, Java virtual machine,

[https://en.wikipedia.org/wiki/Apache\\_Ant](https://en.wikipedia.org/wiki/Apache_Ant),

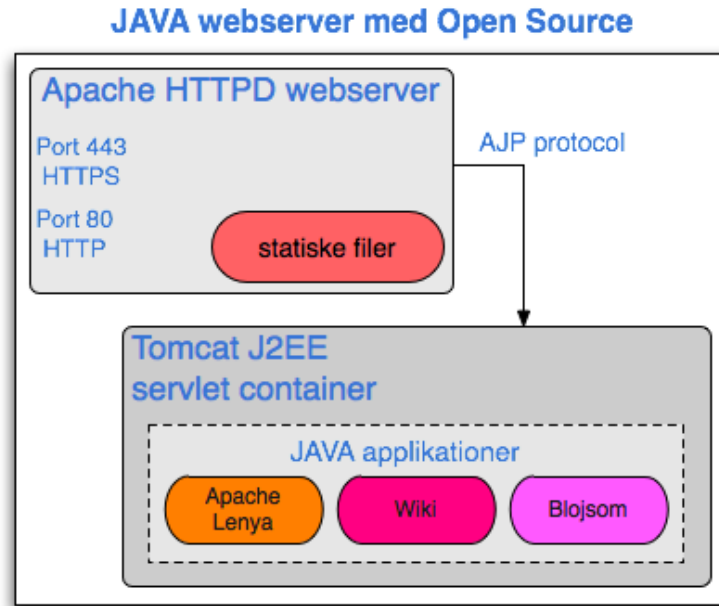
[https://en.wikipedia.org/wiki/Apache\\_Maven](https://en.wikipedia.org/wiki/Apache_Maven)

# Today's Agenda - approximate time plan



- 45m Working with Java, introducing some tools, configure those tools on your machines
- 45m Get the repository from the book, large download, investigate contents
- 10:00 Break 15m
- 45m Chapter 1 presented and discussed
- 45m Get up and running with the book examples
- 11:45 Lunch Break
- 45m Chapter 2 presented and discussed
- 45m Chapter 2 presented and discussed
- 14:00 Break 15m
- 45m Work with small examples, make Camel download from FTP and HTTP

# Java Recap



- Java is a virtual machine environment
- Java Runtime Environment (JRE) / Java Development Kit (JDK)

# Java Environment Variables



- JAVA\_HOME points to the JDK installation path

Typical value:

```
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64/
```

add that to your profile .profile, bashrc or similar on Windows

- We did that last time for running Apache Tomcat
- CLASSPATH the search path for libraries

# Make and install programs from source



```
./configure;make;make install
```

Many open source programmer where distributed as a tar file, tape archive

Most programs would use Makefile and auto tools, as shown above

- configure software - check operating system, features and libraries
- build the software - compile and link
- install software - copy files often putting binaries in /usr/local/bin

# Java Make Tool Ant



Apache Ant is a Java library and command-line tool whose mission is to drive processes described in build files as targets and extension points dependent upon each other. The main known usage of Ant is the build of Java applications. Ant supplies a number of built-in tasks allowing to compile, assemble, test and run Java applications. Ant can also be used effectively to build non Java applications, for instance C or C++ applications. More generally, Ant can be used to pilot any type of process which can be described in terms of targets and tasks.

```
hlk@debian-lab:~$ ant -v
```

```
Apache Ant(TM) version 1.10.5 compiled on August 27 2018
```

```
Trying the default build file: build.xml
```

- Apache Ant
- <https://ant.apache.org/>



# Java tools Needed for Camel Maven



Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

```
hlk@debian-lab:~$ mvn -v
```

```
Apache Maven 3.5.4 (1edded0938998edf8bf061f1ceb3cfdeccf443fe; 2018-06-17T20:33:14+02:00)
```

```
Maven home: /home/user/projects/system-integration/apache-maven-3.5.4
```

```
Java version: 11.0.6, vendor: Debian, runtime: /usr/lib/jvm/java-11-openjdk-amd64
```

- Maven - mvn command
- [https://en.wikipedia.org/wiki/Apache\\_Maven](https://en.wikipedia.org/wiki/Apache_Maven)

# Software Requirements - amended!



The following software is required to run the examples:

- JDK 8, I used OpenJDK 11 on Debian 10
- Maven 3.5 or better, I used `apache-maven-3.5.4`
- Apache Camel 2.20.1 or better, I used `apache-camel-2.24.3`
- Apache Camel itself can be downloaded from its official website: <http://camel.apache.org/download.html>
- All the examples can be run using Maven.

# Start Cloning



- Work in groups of two
- First I will show the process, then you
- Only have one laptop downloading, then the other
- Clone Git repository
- Start the maven command, it will download other files etc.

# Running Maven



```
user@Projects:~$ tail -5 .bashrc
```

```
# Add Maven to PATH
```

```
PATH="$HOME/bin:$PATH:$HOME/user/projects/system-integration/apache-maven-3.5.4/bin"
```

```
export PATH
```

```
user@Projects:~$ which mvn
```

```
/home/user/projects/system-integration/apache-maven-3.5.4/bin/mvn
```

- Download Maven and "install it", as shown above I unpacked version 3.5.4 and pointed my PATH to the directory
- Maven uses a `pom.xml` file - project object model (POM)
- This file lists requirements for dependencies etc.
- Since we will be using other versions, lets change this file

# Maven pom changes



To use the later versions of the tools Maven and Camel change this:

```
user@Projects:camelinaction2$ git diff
diff --git a/pom.xml b/pom.xml
index d9b1b160..7d512cd1 100644
--- a/pom.xml
+++ b/pom.xml
@@ -12,7 +12,7 @@
     <inceptionYear>2015</inceptionYear>

     <prerequisites>
-        <maven>3.5.0</maven>
+        <maven>3.5.4</maven>
     </prerequisites>

     <developers>
@@ -58,7 +58,7 @@
         <activemq-version>5.15.7</activemq-version>
         <activemq-karaf-version>5.13.8</activemq-karaf-version>
-        <camel-version>2.25.0</camel-version>
+        <camel-version>2.24.3</camel-version>
```

# Clone the repository from the book



- We need to run examples from the book
- The source code for the examples in this book is available online at GitHub:  
`https://github.com/camelinaction/camelinaction2`
- Clone this repository - about 22Mb
- Should take very little time, 2-5 minutes
- `git clone https://github.com/camelinaction/camelinaction2.git`

# Inside a project - chapter 1



```
hlk@debian-lab:camelinaction2$ ls
```

```
LICENSE      appendixA  chapter2  chapter4  chapter6  chapter8  chapter10  chapter12
chapter14    chapter16  chapter18  chapter20  errata.txt  mvnw.cmd  README.md  chapter1
chapter3     chapter5   chapter7   chapter9   chapter11  chapter13  chapter15  chapter17
chapter19    docs mvnw   pom.xml
```

# Chapter 1: file-copy example



```
hlk@debian-lab:~/projects/system-integration/camelinaction2/chapter1/file-copy$ find data/  
data/  
data/outbox  
data/outbox/message1.xml  
data/inbox  
data/inbox/message1.xml
```

- We want to run the command for Maven to download tools, and *do stuff*
- `mvn compile exec:java`
- This might take some time!
- Note: this is a two step process, so split into `mvn compile` and `exec:java` if you have trouble running



# Success Compile



```
hlk@debian-lab:~/projects/system-integration/camelinaction2/chapter1/file-copy$ mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.camelinaction:chapter1-file-copy >-----
[INFO] Building Camel in Action 2 :: Chapter 1 :: File Copy Example 2.0.0
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-resources-plugin:2.4.3:resources (default-resources) @ chapter1-file-copy ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.6.1:compile (default-compile) @ chapter1-file-copy ---
[INFO] Nothing to compile - all classes are up to date
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.270 s
[INFO] Finished at: 2020-02-17T07:08:02+01:00
[INFO] -----
```

# Success Execute Java - shortened for slide



```
hlk@debian-lab:~/projects/system-integration/camelinaction2/chapter1/file-copy$ mvn exec:java
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.camelinaction:chapter1-file-copy >-----
[INFO] Building Camel in Action 2 :: Chapter 1 :: File Copy Example 2.0.0
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- exec-maven-plugin:1.2.1:java (default-cli) @ chapter1-file-copy ---
[ion.FileCopierWithCamel.main()] DefaultCamelContext INFO Apache Camel 2.24.3 (CamelContext: camel-1) is starting
[ion.FileCopierWithCamel.main()] FileEndpoint INFO Using default memory based idempotent repository with cache max size: 10
[ion.FileCopierWithCamel.main()] DefaultCamelContext INFO Route: route1 started and consuming from: file://data/inbox?noop=true
[ion.FileCopierWithCamel.main()] DefaultCamelContext INFO Total 1 routes, of which 1 are started
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 11.908 s
[INFO] Finished at: 2020-02-17T07:11:18+01:00
[INFO] -----
```

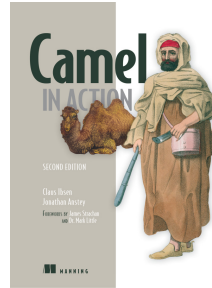
# Success Execute Java - new files



```
$ find data/  
data/  
data/outbox  
data/outbox/message1.xml  
data/outbox/message2.txt  
data/inbox  
data/inbox/message1.xml  
data/inbox/message2.txt
```

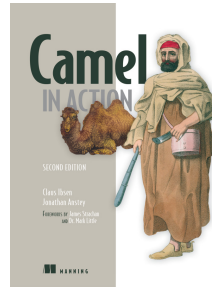
- Try adding a new file using editor, and re-run  
`echo "some data" > data/inbox/message2.txt`

# Chapter 1



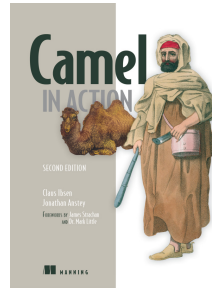
- Meeting Camel
- Introducing Camel:  
What is Camel?, Why use Camel?
- Getting started:  
Getting Camel, Your first Camel ride, Camel's message model, Message, Exchange
- Camel's architecture:  
Architecture from 10,000 feet, Camel concepts
- Your first Camel ride, revisited

# Chapter 2



- Routing with Camel
- Introducing Rider Auto Parts
- Understanding endpoints  
Consuming from an FTP endpoint, Sending to a JMS endpoint
- Creating routes in Java  
Using RouteBuilder, Using the Java DSL
- Defining routes in XML  
Bean injection and Spring, The XML DSL, Using Camel and Spring

## Chapter 2, cont



- Endpoints revisited  
Sending to dynamic endpoints, Using property placeholders in endpoint URIs,  
Using raw values in endpoint URIs, Referencing registry beans in endpoint URIs
- Routing and EIPs  
Using a content-based router, Using message filters, Using multicasting,  
Using recipient lists, Using the wireTap method
- Summary and best practices

# File Transfer Pattern Revisited



- Unix has a saying, *Everything is a file*
- With modern computers and networks, we have URLs and can access files remotely
- Example URLs, `http://www.example.com/index.html`,  
`smb://workgroup;user:password@server/share/folder/file.txt`

# Camel Examples



We can try using Camel

- Downloading files from FTP or HTTP sites
- Put them in a local directory
- Work with small examples, make Camel download from FTP and HTTP
- Hint:
- <https://camel.apache.org/components/latest/ftp-component.html>
- <https://camel.apache.org/components/latest/http-component.html>



# Download from FTP or HTTP



- What are some differences between these?
- Which one do you prefer as programmers?
- Which one is easier to implement in the network  
Hint: FTP uses dynamic ports - look it up
- Running a service, consider if you are providing data - which would you choose  
Hint: security history of FTP

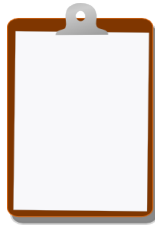
## Actual example



- Go to the web page of the RFC editor <https://www.rfc-editor.org/retrieve/bulk/>
- Read about the possibilities for downloading in bulk
- Pros and cons of using rsync, FTP, HTTP, tgz/zip
- Using Camel or rsync, rsync described at <https://www.rfc-editor.org/retrieve/rsync/>

**Lesson in system integration, lots of different possibilities exist ☺**

## For Next Time



Think about the subjects from this time, write down questions

Check the plan for chapters to read in the books

Visit web sites and download papers if needed

Retry the exercises to get more confident using the tools