

MASARYKOVA UNIVERZITA  
FAKULTA INFORMATIKY



# Thesis title

MASTER'S THESIS

Dávid Šimanský

Brno, Autumn 2013

## Declaration

Hereby I declare, that this paper is my original authorial work, which I have worked out by my own. All sources, references and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Dávid Šimanský

**Advisor:** Mgr. Marek Grác, PhD.

## Acknowledgement

My thanks go to ...

## **Abstract**

This thesis is about ...

## Keywords

GitHub, Thesis, Key, Words, Specific, ...

## Contents

1	<b>Introduction</b>	2
2	<b>Technologies</b>	3
2.1	<i>Apache Camel</i>	3
2.2	<i>Java NIO</i>	4
2.3	<i>Undertow</i>	4
2.4	<i>Undertow - web server</i>	5
3	<b>Analysis and Design</b>	6
4	<b>Implementation</b>	7
5	<b>Conclusion</b>	8

## 1 Introduction

## 2 Technologies

This chapter introduces technologies that are used for implementation of camel-undertow component. Every sub-chapter consists of description and typical uses cases, where the specific technology excels.

### 2.1 Apache Camel

Apache Camel is open source rule-based and mediation framework implemented in Java. The core of the framework is formed around the theory of EIPs<sup>1</sup> by Gregor Hohpe and Bobby Wolf. It creates base layer in integration efforts across various applications, e.g. in stand alone routing, communication of web services, enterprise messaging solutions or full integration platforms (also known as ESB<sup>2</sup>) like Apache ServiceMix, JBoss Fuse or Fuse Service Works. It aims to be lightweight, easy to adopt and extend for developers, no complex class hierarchy or APIs rather emphasising the focus on integration tasks.

The idea behind Camel is to get the maximum potential from the theory of EIPs and to efficiently minimize the lines of source code needed to implement integration scenarios. Therefore a convention over configuration approach is used to describe the task in declarative way by domain-specific language (DSL). The Camel's DSL creates common way for developers to integrate the applications which is easy to learn and afterwards apply, regardless of transport protocols, delivery format, payload encoding or endpoints connectors.

The format of DSL varies by the preference or experience of individual. It is not bound to Java language only, whatever developer likes Java, XML, Groovy, Ruby or even Scala.

The next key feature, that supports wide adoption of the framework in integration world, is modularity. The Camel can be easily extended to consume or to produce data to endpoint. Following the structure given by the framework and extending core classes developers are able to provide solution to whatever unique system you could imagine. Out of the box it comes with handful of components to start with, called camel-core including bean, file, log, mock. On top of that, there are many more developed by Apache community and third-parties<sup>3</sup>. The most common integration scenarios can

---

1. EIP - Enterprise Integration Pattern  
2. ESB - Enterprise Service Bus  
3. [link to components list](#)



be server by already existing components to integrate JMS, web services (SOAP or REST), database connections, filesystem resources or mobile push services.

//TODO extend camel part

## 2.2 Java NIO

Java NIO represents New IO, an alternative implementation to the standard IO API used in the Java language. The key differences include reading data from channels into buffers or vice versa, whenever the standard API use streams of bytes or chars. The other are non-blocking IO instead of typical blocking IO, when thread is block until read or write is completed and selectors.

Non-blocking approach means that the execution thread is not blocked, therefore is not waiting for completion of read or write operation instead it redirects it to channel and buffer and continues to operate. Afterwards if the data are available in the buffer even partially, the thread process it. The write operation acts the same way, data are stored in buffered and written to channel, in the meantime the executing thread can handle some other operation.

The idea is that a single thread can manage multiple input and output operations. The thread is not blocked by reading from one stream, nevertheless do something useful during meantime.

The selector is a kind of object monitoring multiple channels for events to further extend capability of single thread.

## 2.3 Undertow

Undertow is a flexible performant web server written in Java, providing both blocking and non-blocking API's based on NIO.

Undertow has a composition based architecture that allows you to build a web server by combining small single purpose handlers. The gives you the flexibility to choose between a full Java EE servlet 3.1 container, or a low level non-blocking handler, to anything in between.

Undertow is designed to be fully embeddable, with easy to use fluent builder APIs. Undertow's lifecycle is completely controlled by the embedding application.

Undertow is sponsored by JBoss and is the default web server in the Wildfly Application Server.

### 3 Analysis and Design

## 4 Implementation

## 5 Conclusion

## List of Figures

## List of Tables

## References