



# Web Development: Frameworks

*Created by Lasse Jenssen*

(Based on material from Atle Geitung, 2021)

[Home](#)

# Agenda: Using Web Frameworks

- What is a framework?
- Pros and cons: Why use frameworks?
- Types of Web Frameworks
  - Action Based Web Frameworks
  - Component Based Web Framework
- Introduction to Spring Web MVC (later)
- Introduction to Thymeleaf (later)

# Introduction to Web Frameworks

## What is a Framework?

Definition from wiki: <http://en.wikipedia.org/wiki/Framework>

*"A framework is a generic term commonly referring to an essential **supporting structure** which **other things are built on top of**."*

# Introduction to Web Frameworks

## What is a Web Framework?

Definition from wiki: [https://en.wikipedia.org/wiki/Web\\_framework](https://en.wikipedia.org/wiki/Web_framework)

*"A software framework that is designed to **support the development of web applications** including web services, web resources, and web APIs.*

*Web frameworks provide a **standard way to build and deploy** web applications on the World Wide Web, and aim to **automate the overhead associated with common activities** performed in web development."*

# Introduction to Web Frameworks

## Pros:

- Supposed to simplify the development.
- Faster and more robust code (but faster is not always better).
- Common architecture that project/ team/ department follows.
- Less own code to maintain.
- Comunity: Get help or assistance from other users.
- Offers:
  - Easy and flexible way to configure (XML, Annotations, property files etc)
  - Tag library (for instance validating input in Ajax)
  - Class libraries
  - And much more ...

# Introduction to Web Frameworks

## Cons:

- High startup cost: a lot to learn.
- Many frameworks to choose from.
- Framework implementation might not be the best approach for ALL your problems.
- Dependency/ "lock-in".

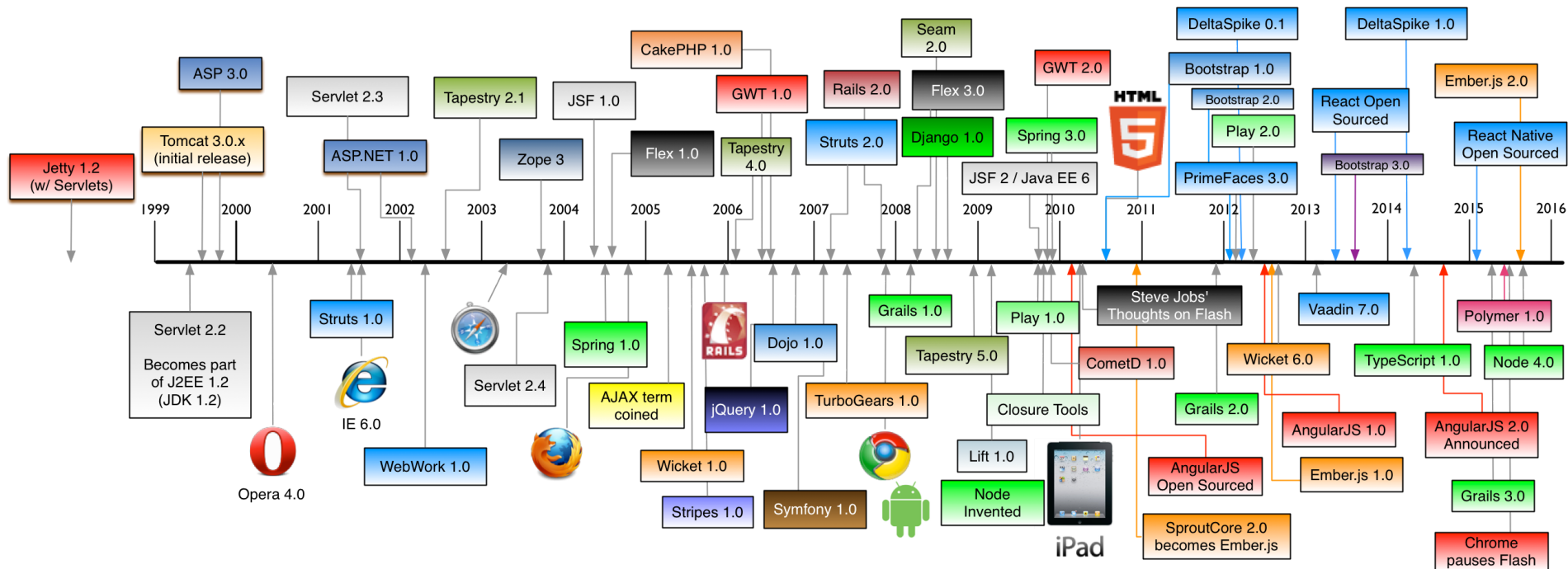
# Types of Web Frameworks

- Types:
  - **Client-side Web Frameworks:** For instance: Angular, Vue, React (you have looked at React earlier).
  - **Server-side Web Frameworks:** Todays agenda.

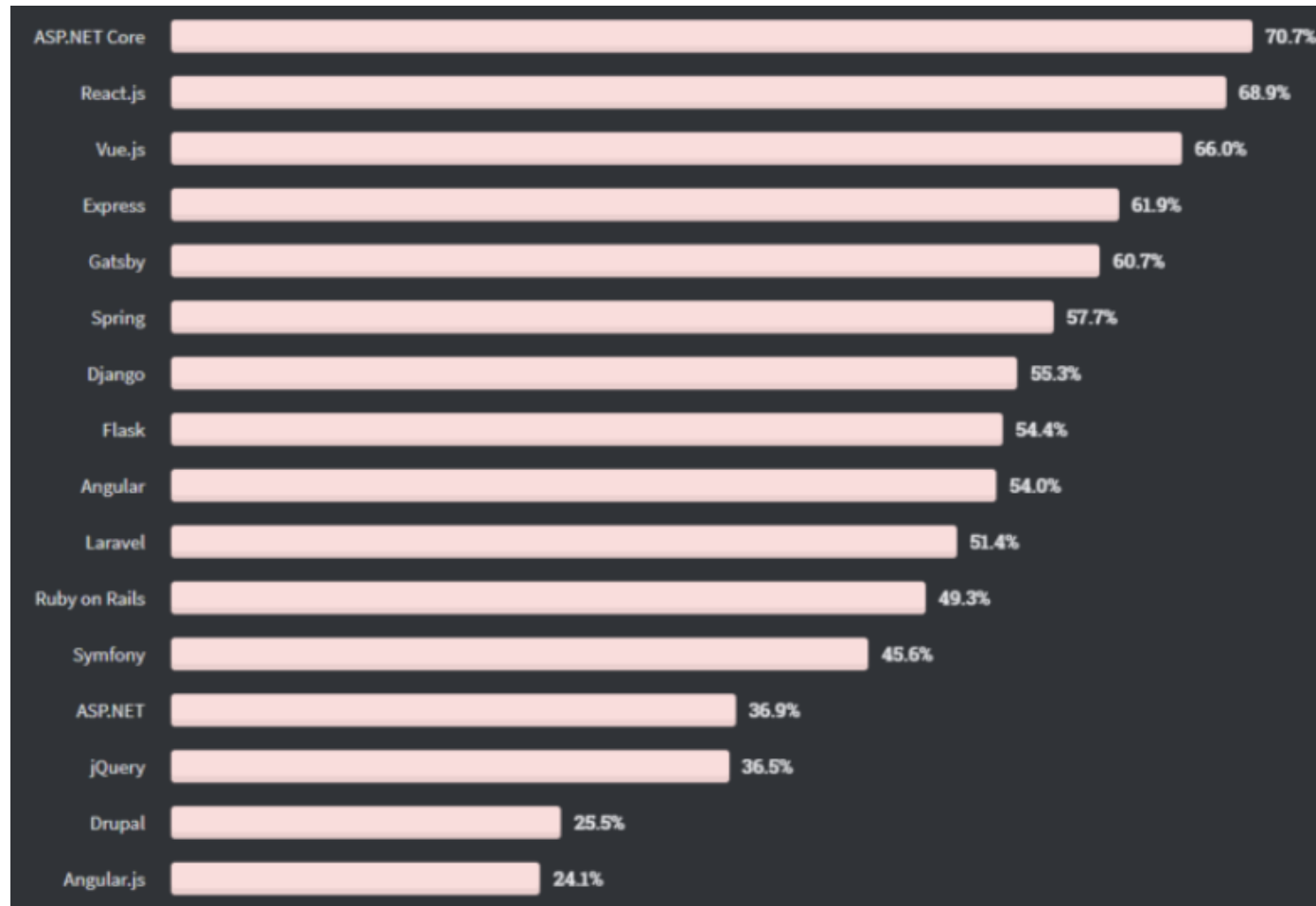
# Types of Web Frameworks

- A common classification of web frameworks is based on the "programming model" being offered the developer.
- Web applications have a core with a programming model based on the **request-response** over a stateless protocol, HTTP (Servlet).
- Types:
  - **Action-based**: Frameworks that exposes the request-response model, linking "actions" up to URLs (requests).
  - **Component-based**: Frameworks that hides the request-response model behind UI components (+events).
  - Also more specialized frameworks, for example RIA (Client-Side: Angular, React, Vue).
- Note! Not always a clear distinction.





# Use of frameworks 2022



<https://www.monocubed.com/blog/most-popular-web-frameworks/>

# Action based Web Frameworks (MVC)

## Examples of action-based frameworks:

- Struts2 (Java)
- Spring Web MVC (Java/ Kotlin/ Groovy)
- PlayFramework (Java / Scala)
- Rails (Ruby)
- Grails (Groovy)



Try Play

Documentation

Get Involved



# The High Velocity Web Framework For Java and Scala



GET STARTED

Try Play  
2.8.17

or [browse all versions](#)

READ THE DOCS

Documentation  
2.8.17

or go to [Java docs](#) or [Scala docs](#)

GET INVOLVED

Join the Discuss  
Play Forum

Join [Play's Discord server](#) and [Stack Overflow](#)

 struts.apache.org

PrivatTOREADBouvetTietoEVRYOracleCloudVimondtmuxEritecGmailYouTubeMaps

STRUTS

HOME▼SUPPORT▼DOCUMENTATION▼CONTRIBUTING▼

  
The Apache Software Foundation  
<http://www.apache.org/>

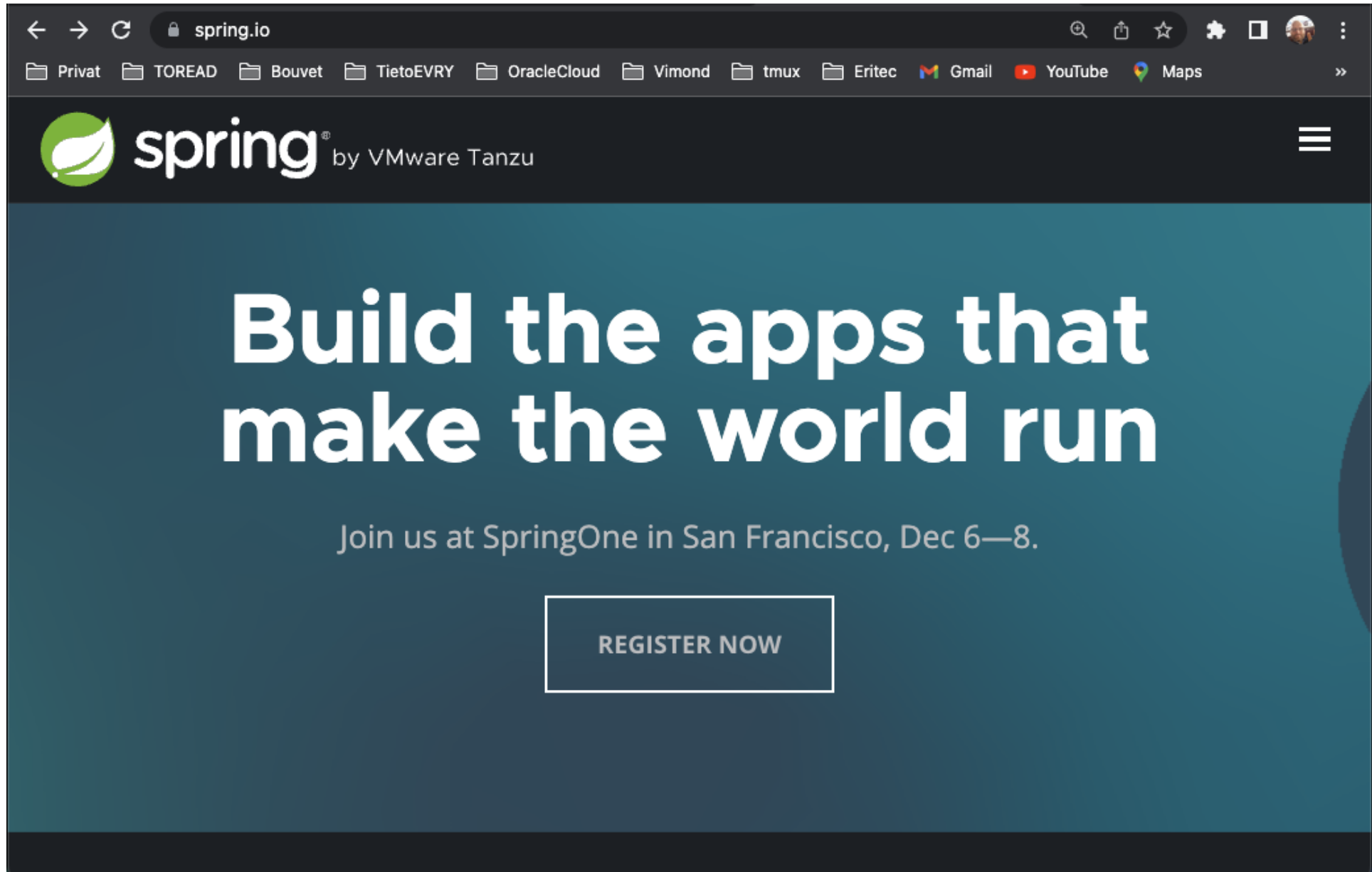


# Apache Struts

Apache Struts is a free, open-source, MVC framework for creating elegant, modern Java web applications. It favors convention over configuration, is extensible using a plugin architecture, and ships with plugins to support REST, AJAX and JSON.

 Download

 Technology Primer



# Component Based Web Frameworks

## **Examples of component-based frameworks:**

- Vaadin (Java) : We'll have a quick look at this.
- GWT (Java) : GWT is a development toolkit for building and optimizing complex browser-based applications.
- ASP.NET WebForms (MVVM): a part of the ASP.NET web application framework

← → ↻

vaadin.com

🔖

☆

⚙

🖥

👤

⋮

Privat TOREAD Bouvet TietoEVRY OracleCloud Vimond tmux Eritec Gmail YouTube Maps Translate News Fag

vaadin}>

Platform ▾ Solutions ▾ Company ▾ Developers ▾ Resources ▾ Pricing

Contact us

Get started

Login


# The modern web application platform for Java

Vaadin is a web app **development platform for Java**. It helps you **build reliable web apps with great UX** faster than before.

Latest: 23.2.3 | [All releases →](#)

Get started →

Book a demo →







» Get started

 Docs

➤ Make GWT Better

[↔ Terms](#)

Download

JD Java Doc



●  
○  
○

# Component Based Web Frameworks

Also some Component-based **Desktop** Frameworks:

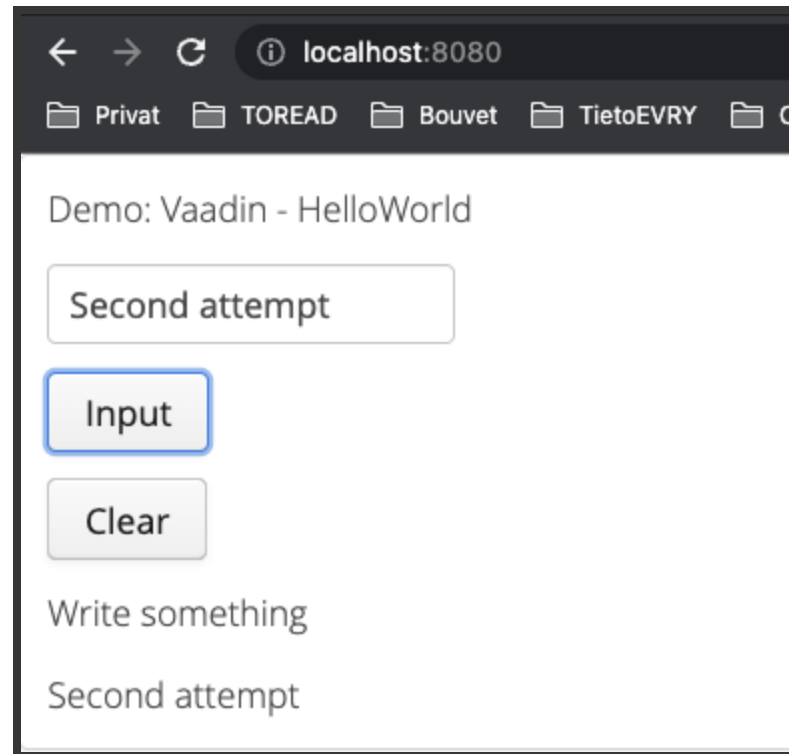
- **Java Swing**: a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) - an API for providing a graphical user interface (GUI) for Java programs.
- **JavaFX**: (<https://openjfx.io/>) - an open source, next generation client application platform for desktop, mobile and embedded systems built on Java.
- **Java SWT** (<https://www.eclipse.org/swt/>): an open source widget toolkit for Java designed to provide efficient, portable access to the user-interface facilities of the operating systems on which it is implemented.
- **Qt** - (C++, Python, etc)

## **Demo: HelloWorldUI (Vaadin)**

- Intro application with a few simple components (Label, TextField, Button).
- We are NOT going to code with Vaadin.  
(Code in this section: Only to show the concept).

*Code: demo-component-based-vaadin.zip (see course overview).*

## Demo: HelloWorldUI (Vaadin)



# Demo: HelloWorldUI (Vaadin)

*src/main/java/no/hvl/dat152/helloworld/HelloWorldUI.java*

```
1  @Theme("mytheme")
2  public class HelloWorldUI extends UI {
3
4      private static final long serialVersionUID = 1L;
5
6      @Override
7      protected void init(VaadinRequest vaadinRequest) {
8          Panel contentPane = new Panel();
9          VerticalLayout layout = new VerticalLayout();
10
11          Label lblHeading = new Label();
12          helloworldLabel.setValue("Demo: Vaadin - HelloWorld");
13          layout.addComponent(lblHeading);
14          ...
15          contentPane.setContent(layout);
16          setContent(contentPane);
17      }
18
19      @WebServlet(urlPatterns = "/*", name = "HelloWorldUIServlet", asyncSupporte
20      @VaadinServletConfiguration(ui = HelloWorldUI.class, productionMode = false
21      public static class HelloWorldUIServlet extends VaadinServlet {
22          private static final long serialVersionUID = 1L;
23      }
24 }
```

# Demo: HelloWorldUI (Vaadin)

*src/main/java/no/hvl/dat152/helloworld/HelloWorldUI.java*

```
1 @Override
2 protected void init(VaadinRequest vaadinRequest) {
3     Panel contentPane = new Panel();
4     VerticalLayout layout = new VerticalLayout();
5
6     Label lblHeading = new Label();
7     lblHeading.setValue("Demo: Vaadin - HelloWorld");
8
9     TextField textField = new TextField();
10    textField.setValue("Write something");
11
12    Button okButton = new Button("Input");
13    ...
14
15    Button clearButton = new Button("Clear");
16    ...
17
18    layout.addComponent(lblHeading);
19    layout.addComponent(textField);
20    layout.addComponent(okButton);
21    layout.addComponent(clearButton);
22
23    contentPane.setContent(layout);
24    setContent(contentPane);
```

# Demo: HelloWorldUI (Vaadin)

*src/main/java/no/hvl/dat152/helloworld/HelloWorldUI.java*

```
1 @Override
2 protected void init(VaadinRequest vaadinRequest) {
3     ...
4     Button okButton = new Button("Input");
5     okButton.addClickListener(event -> {
6         Label newLabel = new Label();
7         newLabel.setValue(textField.getValue());
8         layout.addComponent(newLabel);
9     });
10
11     Button clearButton = new Button("Clear");
12     clearButton.addClickListener(event -> {
13         layout.removeAllComponents();
14         layout.addComponent(lblHeading);
15         layout.addComponent(textField);
16         layout.addComponent(okButton);
17         layout.addComponent(clearButton);
18     });
19
20     layout.addComponent(lblHeading);
21     ...
22     contentPane.setContent(layout);
23     setContent(contentPane);
24 }
```

## Demo: HelloWorldUI (Vaadin)

Let's look at the code in Eclipse, and run.

*Code: demo-component-based-vaadin.zip (see course overview).*



# Summary: Web Development: Frameworks

## Where are we now?

- Frameworks are there to assist in our daily programming life.
- A lot of frameworks to choose from.
- Most often the project or company choose the frameworks to be used.

*Next*

# **Web Services: The Spring Framework**

Home