HELLO, WORLD!

Development of Modern Web Applications
(with Vaadin)

Lecture 00

vaadin

WELCOME TO THE COURSE!

- Miki
  - Vaadin Expert
  - Ph.D., CSM, CSPO, WTF, OMG
  - Husband, father, picture taker and curling player
- 12 mandatory lectures
  - Mondays and Thursdays from 8:45
- No exam!
  - Hackathon instead – Vaadin Halloween!
    - Or group project
    - Or an outstanding individual app

vaadin}>

WHAT IS THIS?

- Development
  - All about writing and designing software
- Modern
  - Using today's technology
- Web Applications
  - Software accessible through a web browser
- Vaadin
  - Open source framework for web apps in Java
  - Company with the same name

vaadin}>

ARCHITECTURE OF WEB APPLICATIONS & INTRODUCTION TO VAADIN

Development of Modern Web Applications
(with Vaadin)

Lecture 01

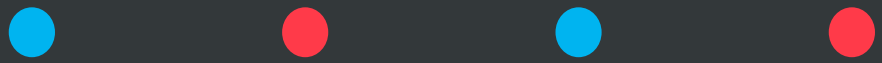
vaadin

OUTLINE

- Desktop applications
- Web applications
- Architecture of web applications
- The Vaadin way
- Development
- Agile methods
- What's next?

DESKTOP APPLICATIONS

What, why, when?



vaadin}>

WHAT

- Any software run with a dedicated executable
  - Code is usually (pre)compiled
  - Efficient execution on target OS
- Distinguished distribution channel
  - Over the net
  - Or pendrives
  - Or discs
  - Or floppies

WHY

Pros

- Execution is local
- Full access to the host machine
  - Peripherials
  - More possibilities
- Can be run on an isolated host machine
  - Less security worries
- Open source

Cons

- Distribution
  - Each and every user must get a copy
  - Each copy must be kept up to date
- Hosts differ in configuration
  - Peripherials
  - Endless number of configurations
  - Operating system matters
- User data is local to the machine
  - Unless exporting is available
- Explicit hardware or software requirements
  - E.g. CPU type, RAM, etc.

WHEN

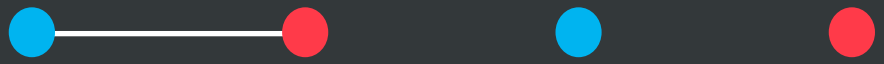
- Performance is essential
- Access to the underlying OS is a must
- Network communication is not
- Strong security concerns
- Compatibility with old systems
- Application is local

SUMMARY

- Desktop apps are still here
  - And likely will be here for another decade(s)
- They are sometimes the best option
  - Not everything needs to be in the interwebz
- This course is not about desktop apps anyway

WEB APPLICATIONS

What this course is about



vaadin}>

WHAT IS WEB APPLICATION?

Any application that uses a web browser as a client. The application can be as simple as a message board or a guest sign-in book on a website, or as complex as a word processor or a spreadsheet.

D. Nations, former About.com Guide

WHAT IS WEB APPLICATION?

A computer software application that is coded in a browser-supported programming language (such as JavaScript), combined with a browser-rendered markup language (like HTML) and reliant on a common web browser to render the application executable.

From Wikipedia

WHAT DOES THAT MEAN?

- Software accessible through a web browser
  - Any web browser
  - Any operating system

vaadin}>

PROS

- Well suited for rapid development
- Problemless distribution
  - All changes affect all users immediately
- No need for client configuration
  - Browser is a part of the OS
  - And usually runs sandboxed
- Remote user access
  - Cloud, cloud everywhere
- Huge level of control
  - Source code
  - User access
  - User actions
  - User data

vaadin}>

CONS

- Web server required to deploy
  - It has to be accessible 24/7
- Compatible web browser required to use
  - Lynx, anyone?
  - ~~IE~~ Edge, anyone?
  - Loss of UX – it runs in a browser
- Privacy
  - Everything can be tracked without user's knowledge
  - Ever wondered who owns **your** data?
- Proprietary
  - Free to use ≠ open source

WHY

- Rapid development
- Easy to get started
- Cheap to develop and maintain
- Can reach thousands in minutes

ARCHITECTURE OF WEB APPS

Some pointers

THE BASIC APPROACH

Presentation	Application	Storage
• Browser • What the user iteracts with • Displays content from and sends requests to the web server	• Web server • Reacts to user actions • Sends content to and receives requests from the web browser • Modifies data according to the application logic	• Stores data • Accessed directly by the web server

vaadin}>

TECHNOLOGY

Presentation	Application	Storage
• Markup languages • Client-side scripting languages	• Web server • Any language that the server speaks	• Database • File system • Document repository

SOME EXAMPLES

Presentation	Application	Storage
• (X)HTML • JavaScript • CSS	• Apache • Tomcat • Jetty • Ruby on Rails • Django • ASP.NET • JSP • Perl • PHP	• PostgreSQL • MySQL • SQLite • SVN

vaadin}>

SOME APPROACHES

- Dumb client, smart server
  - Client only presents the application
  - Server does all the work
- Smart client, dumb server
  - Server only accesses the data
  - Client does all the work
- Combined
  - The most common one

CLIENT SIDE

- Web page is processed inside the browser
  - Meaning: on the client machine
  - DOM – a tree-like structure that holds the source
- HTML5 is here!
  - Support for audio, video, canvas,...
- JavaScript is here!
  - Can modify the web page that is already loaded
  - The assembly language of the web

SERVER SIDE

- Server generates HTML based on the URL
- The way the page is generated is hidden from the client
- The server has little to no idea who the page is generated for

STATELESSNESS IS BLISS?

- HTTP is request-response protocol
  - Server generates the response and forgets
- All the fun in web apps is adding state to HTTP
  - Cookies
  - Sessions

AJAX

- Asynchronous
  - Background communication
  - Both ways (to and from)
  - Without leaving currently displayed page
    - Without changing its URL
  - Does not have to be asynchronous
- JavaScript
  - Can access the DOM of the page to update it
- HXMLHttpRequest – XHR
  - Technique used to send HTTP(S) requests to server and receive the responses by the script
  - Response is most often JSON, but can be XMLor text

vaadin}>

THE VAADIN WAY

Develop your app – Vaadin will make it work

vaadin}>

HOW IT LOOKS

Presentation	Application	Storage
• GWT	• Java • Static resources	• Vaadin Data Model

REQUIREMENTS

- Deployment
  - Java web server
    - Like Tomcat
  - Or Java portal
    - Deeper relationship with Liferay
  - Web browser
    - With JS support
- Development
  - Java
    - With all supported technologies on the server side
      - Including Spring or Java EE
  - Some good IDE recommended
    - Plugins for Eclipse and IntelliJ IDEA
  - Build system recommended
    - Support for Maven and Gradle
    - Eclipse plugin is Maven-based, yay!

vaadin}>

TEST DRIVE – ECLIPSE

- Get Eclipse for Java EE developers
- Install Vaadin plugin
  - https://vaadin.com/eclipse
  - No need to install Vaadin Designer (yet)
- File → New → Vaadin 7 project (Maven)
  - Standalone application
- Run as → Maven build… → `jetty:run`

TEST DRIVE – MAVEN

```
mvn -B archetype:generate
    -DarchetypeGroupId=com.vaadin
    -DarchetypeArtifactId=
       vaadin-archetype-application
    -DarchetypeVersion=7.7.0
    -DgroupId=org.vaadin.miki
    -DartifactId=shoutbox
    -Dversion=0.1-SNAPSHOT

cd shoutbox
mvn package jetty:run
                    # YAY!
```

vaadin}>

ABOUT THE CODE

- `public class HelloUI extends UI {`
  - UI is a base class for any application
- `protected void init(VaadinRequest r) {`
  - This is where the execution *starts*
  - Each request ends up in this method
- `setContent(layout);`
  - Sets the contents of the UI
  - Without it, the components would not be visible in the browser
    - Lecture 02 → Components
    - Lecture 04 → Layouts and navigation

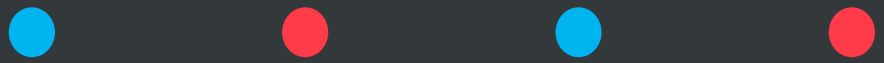
vaadin}>

IMPRESSIONS

- (Almost) like a desktop Java app
  - Events
  - Listeners
  - Components
- Not a lot of webapp-specific stuff
  - At least not at the beginning ☺

DEVELOPMENT

Things to consider



WEB APPS ARE SOFTWARE

- Do the right things
  - Clear specifications
  - User expectations
- Do things right
  - Testing
  - Reliability

vaadin}>

ADDITIONAL CONCERNS

- Availability
  - Any time
  - Any place
- Accessibility
  - Any device
  - Different users
  - Standard compliance
- Scalability
  - Stress testing
  - Stable growth
- Security
  - All users are affected

vaadin}>

CHALLENGES

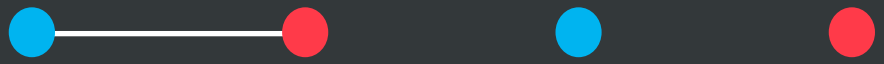
- How to test cross-browserness?
- How to test user interface?
- How to perform stress tests?
- How to anticipate growth?
- How to make money out of the app?

FRAMEWORKS

- Support code reuse
- Reduce programming effort
  - Focus on application logic
  - Magic!
- Every major language has at least one
  - Ruby on Rails
  - Django
  - Vaadin

AGILE METHODS

Amazingly well suited for web app development



vaadin}>

WHY

- Rapid development in short cycles
  - Incremental design
  - Stakeholders involved in the process
- Easy to track progress
  - Short cycles = small increments
- User feedback after each cycle
  - Demo results to the stakeholders

AGILE MANIFESTO

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more

Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

vaadin}>

WHAT'S NEXT?

Summary and overview of future lectures

vaadin}>

LESSONS OF TODAY (HOPEFULLY)

- Desktop applications
  - What are they?
- Web applications
  - What are they?
  - How they differ from desktop apps?
  - What is the common architecture of web apps?
  - What technology can be used?
- Vaadin
  - What is it?

NEXT LECTURES

2. Overview of components
   - Basic API, UX, demo!
3. Events and data binding
   - Basic API, best practices, demo!
4. Styling, theming and layouts / Navigation
   - Valo theme, tips, demo!
   - Basic API, demo!

DEMO?

- Shoutbox app
- Maybe more, depending on lecture topics

THE END

SUGGESTIONS?
QUESTIONS?

miki@vaadin.com
t: @mikiolsz

vaadin