Internet Applications Design and Implementation

(Lecture 10 - Recap - Q&A)

MIEI - Integrated Master in Computer Science and Informatics
Specialization block

João Costa Seco (joao.seco@fct.unl.pt)

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

Internet Applications Design and Implementation

RECAP

We focus on the **principles and concepts on the development of Internet applications**.

The syllabus follows an approach based on the fundamentals of software development based on **web and service oriented architectural patterns**, **advanced modularity mechanisms**, **data persistency abstractions**, **good development practices**, **performance concerns**, and **validation techniques**.

Lectures run along **practical assignments** and the **development of a running project** using frameworks, languages, and programming tools for Internet Applications that ensure the safety and compliance of the solution with relation to a specification

Goals: To Know

RECAP

- Essential aspects of **architectural patterns** for inversion of control and software architectures specific for Internet Applications.
- **Principles of the development** of web applications and single page web applications.
- Mechanisms of **specifying and implementing web services** and web service orchestrations.
- **Internal structure** of an Internet **browser** and its **client applications**.
- Principles of **data-centric** and **user-centric development** in the context of Internet applications.
- Main **data abstraction** mechanisms used in Internet applications.
- Major performance **pitfalls** of Internet applications and their workarounds.
- Main specification and implementation mechanisms for **security policies in Internet Applications**.

Goals: To Do

RECAP

- **Use development frameworks** that implement architectural styles for Internet applications.
- **Specify and build** web and cloud **applications** to support thin, flat, and native clients.
- **Specify and build** client **applications with reactive and rich behaviour**.
- **Implement authentication mechanisms** and **specify** the core **security rules** of an Internet Application
- Specify and efficiently use abstraction data layers such as **Object Relational Mappings** in Internet applications.
- **Design and deploy** Internet Applications that are efficient and maintainable.

Internet Applications Design and Implementation

- You now know Component-Based Programming
  - SpringBoot, Spring Web, SpringJPA,
  - React, Redux
  - IFML
- You now know how to develop and test modular software
  - Mock Tools (tomcat with MockMVC),
  - Repositories and Components (MockBean),
  - MicroServices (with Eureka)
- You now know how to build loosely coupled structures (the safest way possible)
  - SpringJPA, FeignClients, OpenAPI (with generators) - Look for new tools that strengthen frontiers

Internet Applications Design and Implementation

- SpringBoot, Hibernate, React, Redux, and OpenAPI are only examples of frameworks and tools that promote architectural patterns and practices.
- Paradigms to look up to (Functional, Functional Reactive, Declarative)
- Trending technologies.
  - For APIs: GraphQL
  - For Cloud: Serveless computing (lambdas)
  - For Architecture: Reactive and Tierless frameworks (WebFlux, Meteo.js, Blitz.js,
  - For Processes and MicroServices: Message- and Event-based architectures (Akka)
- Trending frameworks that you may want to explore:
  - Phoenix-Elixir, Next.js+React Query

Final Test (18 of December)

- MCQ
  - General Software Architecture (Micro-servides)
  - General Client-Side Applications and
  - Security questions, Access Control, Capabilities
- Open Answer Questions
  - Produce tokens between microservices
  - User Stories to IFML
  - React (Components)
  - Redux (Actions and Asynchronous Actions)

Project Presentations

- Many projects will be presented **before** the submission date:
  - Discussions to be scheduled from **16 to 20 of December** (max 6 a day)
  - Presentation and discussion is 30 minutes max
  - Report on the status and choices of:
    - User Stories
    - Architecture (Microservices)
    - Security
    - Demo UI if possible
    - Challenges
  - Bring a powerpoint/canva/keynote/wtv presentation

The End.

Thank you!