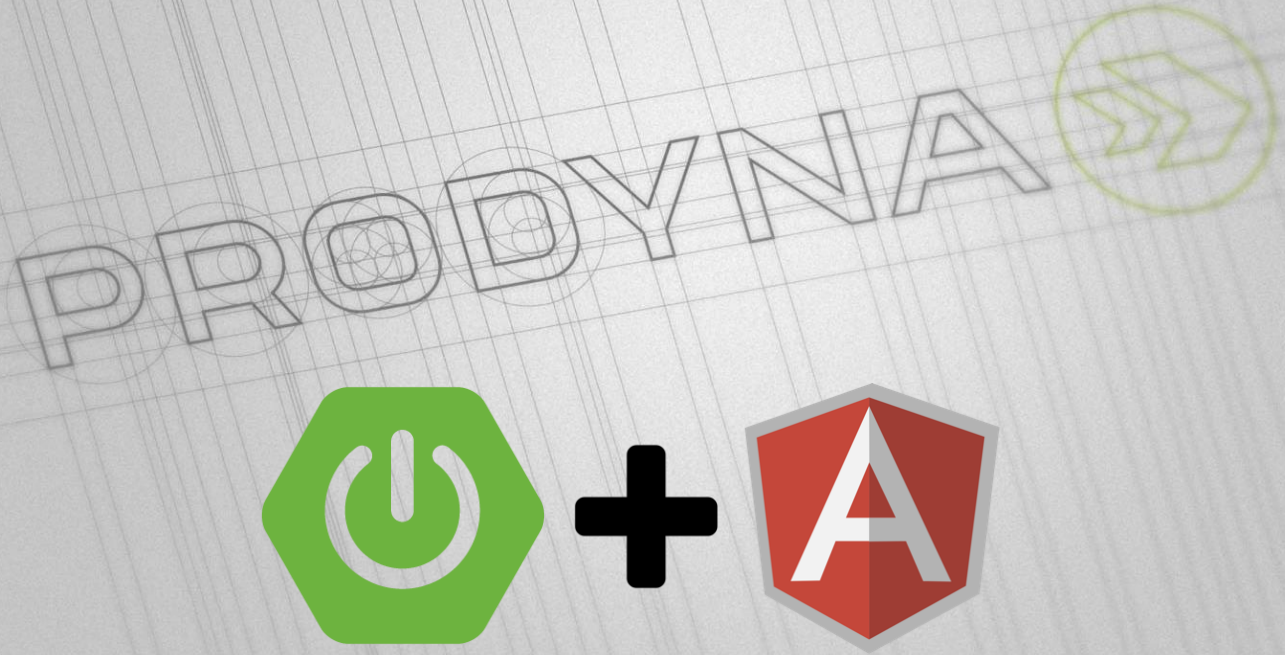
MICROSERVICES WITH SPRING BOOT AND ANGULAR

PRODYNA S.E.

CHRISTOS MANIOS

2018-12-01



CONTENTS

OVERVIEW

ABOUT PRODYNA

WEB SERVICES

MICROSERVICES

SPRING BOOT

ANGULAR

DEMO TIME

PRODYNA: KEY FACTS

POSITIVE ORGANIC GROWTH

FINANCIAL

FY18/19 expected 42 million EUR

Zero venture capital

Employee owned

EMPLOYEES

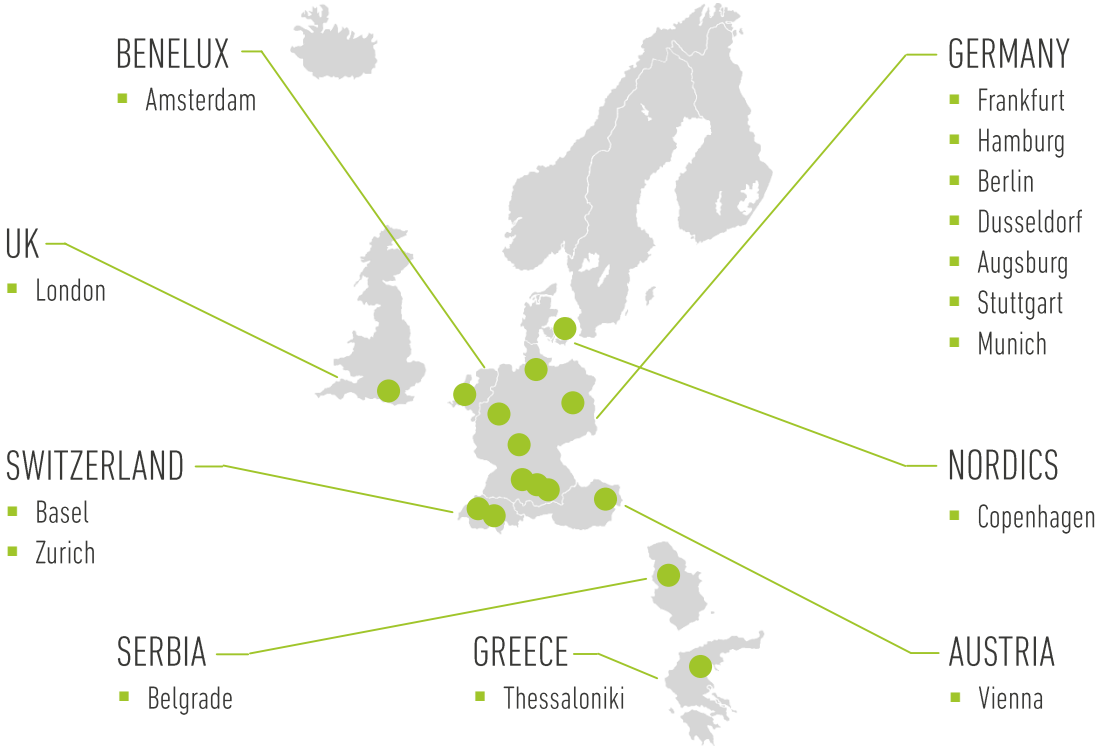
320 employees

Lean back office

Zero freelancers

PRODYNA: WE ARE CLOSE TO YOU

FOCUSSED ON THE MAJOR EUROPEAN REGIONS



SOLUTION PARTNERS FOR DIGITAL BUSINESS

OUR PARTNER PORTFOLIO IS BASED ON THE FIVE MAIN DRIVERS OF DIGITAL BUSINESS

USER EXPERIENCE

LIFERAY
- Portal Server
- Digital Experience
- B2B Platforms

MAGNOLIA
- Enterprise WCMS
- Internet sites
- E-Commerce

ADOBE
- Enterprise WCMS
- Analytics & Marketing
- Experience Cloud

AI AUTOMATION & TOOLS

COGNIGY
- Conversational AI
- Omni-channel
- Natural Language

ITyX
- AI Platform
- Process Automation
- Virtual Agents

STARMIND
- AI based Knowledge Management
- Silo breaker

API

MULESOFT
- API Management
- Enterprise Integration
- ESB

DATASCIENCE & BIG DATA

TABLEAU
- Visual analysis
- Insight generation
- Self service BI

NEO4J
- Graph Database
- For structured data
- Large data volumes

MONGO DB
- NoSQL Database
- For unstructured data
- Large data volumes

STRATEGIC APPLICATION PLATFORMS

RED HAT
- Cloud Platform
- Linux Platforms
- Virtualization

AMAZON AWS
- Cloud Server
- Cloud Storage
- Cloud Services

MICROSOFT
- Azure Cloud
- Office 365
- .NET Platform

KUBERNETES
- Container Orchestration
- Application Scaling
- Deployment Automat.

TECHNOLOGIES & TOOLS

SOME OF OUR BUILDING BLOCKS FOR SUCCESS

Systems - Apache Tomcat Oracle Xg / RAC Liferay SOLR Adobe CQ Lucene IBM WebSEAL Amazon AWS magnolia JBoss ESB talend ESB inubit ESB VMWare RedHat Linux FirstSpirit CMS Neo4J

Libraries & Frameworks - Vert.X 3 netty.io Hazelcast Apache Wicket Framework Spring IO Spring Boot Spring Framework Spring others ehCache SSL SSO Vaadin UI Apache Axis SAP JCO JCR JSP Junit

Languages - Java Javascript Typescript HTML / CSS

Tools - Atlassian Bamboo Jenkins Atlassian Bitbucket Atlassian Confluence Checkstyle Docker Eclipse Findbugs Gatling Git GitLab Gliffy Gradle maven Atlassian Jira Atlassian Jira Agile Nexus PMD Postman API-Console Sonar

PRODYNA REFERENCE CUSTOMERS

SELECTION FROM VARIOUS SEGMENTS

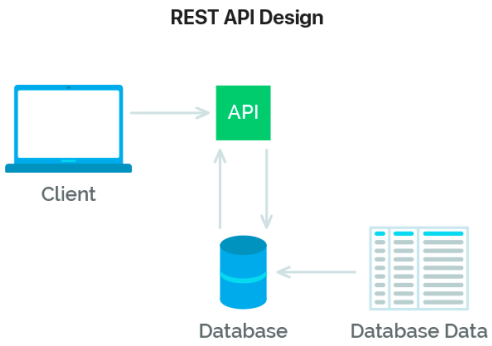
BOSCH
MERCK
Deutsche Bank
Lufthansa
BMW
BASF
The Chemical Company
COMMERZBANK
Thomas Cook
VOLKSWAGEN
AKTIENGESELLSCHAFT
KION
GROUP
BNP PARIBAS
BKW
Audi
SCHOTT
glass made of ideas
SOCIETE GENERALE
AIRBUS
vodafone
Telefónica
Allianz
ESPRIT

WEB SERVICES

KEY FEATURES

Types

- SOAP
  - WS Specification: WSDL (web service description language)
  - Communication: XML
- REST
  - WS Specification: Swagger or RAML(not standardised yet)
  - Communication: JSON, XML or any other common MIME type



WEB SERVICES IN EVERYDAY LIFE

DO YOU KNOW ? ...

Worldwide

- Google APIs (Search, Maps, Gmail, Drive, ...)
- Facebook (Messenger, Instagram, ...)
- Netflix
- Imdb

Greece

- Gsis.gr (SOAP)
- Diavgeia (REST)

RESTFUL APIS

HTTP METHODS AS SIMPLE AS IT GETS

GET	/movies	Get list of movies
GET	/movies/:id	Find a movie by its ID
POST	/movies	Create a new movie
PUT	/movies	Update an existing movie
DELETE	/movies	Delete an existing movie

SPRING BOOT

ABOUT

Spring Boot is the starting point for building all Spring-based applications. Spring Boot is designed to get you up and running as quickly as possible, with minimal upfront configuration of Spring.

- Get started in seconds using Spring Initializr
- Build anything - REST API, WebSocket, Web, Streaming, Tasks, and more
- Simplified Security
- Rich support for SQL and NoSQL
- Embedded runtime support - Tomcat, Jetty, and Undertow
- Production-ready features such as tracing, metrics and health status

ANGULAR

FEATURES

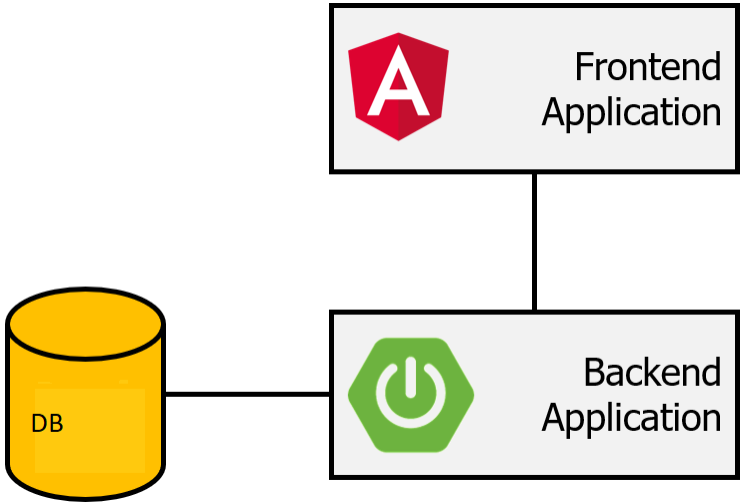
- MVC Progressive single page web apps
- Written in Typescript (superset of Javascript)
- i18n
- Speed (transpiler, linter, minifier)
- Code splitting
- Testing
- Accessibility

DEMO TIME

LET'S CREATE A MICROSERVICE

Technology Stack

- Spring Boot (backend)
- Angular (UI - frontend)



STEP 1

VERIFY DEVELOPMENT ENVIRONMENT

2 min

Open a shell or Git Bash and run:

```
$ node --version
v10.13.0

$ npm version
{ npm: '6.4.1', ...

$ npm install -g @angular/cli

$ ng --version
Angular CLI: 7.1.0

$ java -version
java version "1.8.0_161"
```

STEP 2

CREATE THE MYSQL SCHEMA AND TABLE

Connect to your database using MySQL workbench and run:

2 min

```
CREATE SCHEMA `kariera` DEFAULT CHARACTER SET utf8 ;

CREATE TABLE `kariera`.`student` (
    `id` BIGINT NOT NULL AUTO_INCREMENT,
    `name` VARCHAR(255) NULL,
    `surname` VARCHAR(255) NULL,
    PRIMARY KEY (`id`))
    ENGINE = InnoDB
    DEFAULT CHARACTER SET = utf8;
```

STEP 3

DOWNLOAD REQUIRED PROJECTS

Open a shell or Git Bash and run to clone your projects:

2 min

```
git clone https://github.com/manios/kariera2018-spring

git clone https://github.com/manios/kariera2018-angular
```

STEP 4

OPEN AND RUN ANGULAR APP

1 min

Open kariera2018-angular project in Visual Studio Code and then run on a cli:

```
npm install
```

After it finishes

```
ng serve --open
```

This will run and open your browser at http://localhost:4200

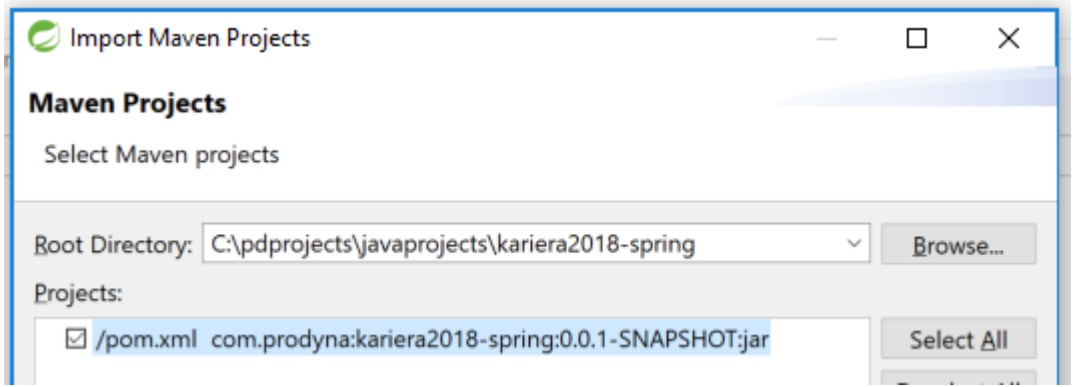
STEP 5

OPEN SPRING BOOT APP

Open Spring Boot app with Eclipse STS:

1 min

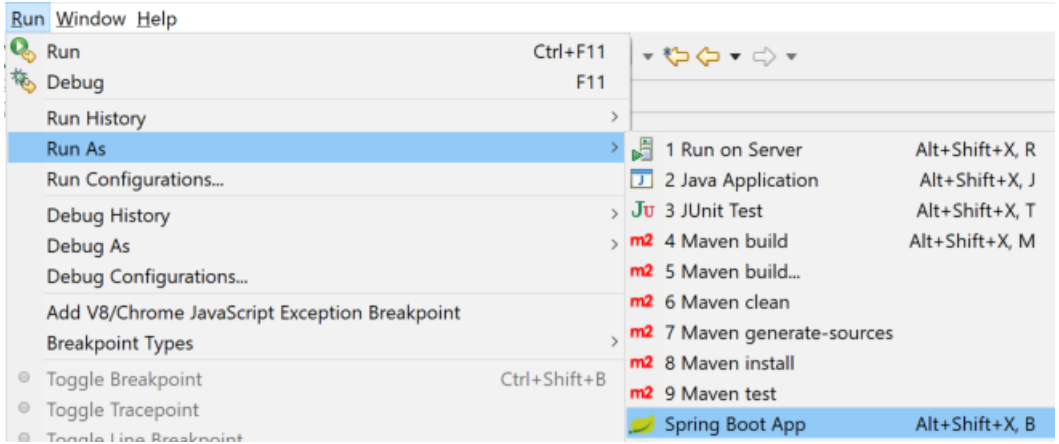
1. File > Import > Maven > Existing Maven Projects
2. Select kariera2018-spring
3. Select pom.xml
4. Press Finish



STEP 6

RUN SPRING BOOT APP

On menu:



1 min

Verify in your browser that http://localhost:8081/students/hello returns HTTP 200 with a JSON response.

STEP 7

IMPLEMENT ADDSTUDENT() IN ANGULAR

5 min

In AddStudentComponent implement the method which will create a new student:

```
addStudent(name: String, surname: String) {
    this.isLoading = true;

    let myStudent = new Student(name, surname);

    this.studentService
        .addStudent(myStudent)
        .subscribe(
            st => {

                this.isLoading = false;

                this.router.navigate(['/']);
            },
            error => {
                this.errors = error.json().errors;
                this.isLoading = false;
            }
        );
}
```

STEP 8

IMPLEMENT ADDSTUDENT() IN ANGULAR (PART 2)

In StudentService implement the method which will create a new student by sending a POST request to the REST API:

2 min

```
addStudent(s: Student): Observable<ActionStatus> {
    return this.http.post<ActionStatus>(ADD_STUDENTS_URL, s, httpOptions);
}
```

STEP 9

IMPLEMENT SEARCH STUDENT IN SPRING REPOSITORY

In StudentRepository interface add a method signature which will search students by their name or surname:

2 min

```
public interface StudentRepository extends CrudRepository<Student, Long> {
    List<Student> findByNameIsLikeOrSurnameIsLike(String name, String surname);
}
```

STEP 10

IMPLEMENT SEARCH STUDENT IN SPRING BOOT SERVICE

In StudentServiceImpl implement the method which will search students by their name or surname:

5 min

```
@Override
public List<StudentDTO> searchStudents(final String term) {

    final String searchTerm = "%" + term + "%";

    final List<Student> st = this.studentRepository
                        .findByNameIsLikeOrSurnameIsLike(searchTerm, searchTerm);

    return this.toDTO(st);
}
```

STEP 11

IMPLEMENT SEARCH STUDENT IN SPRING BOOT CONTROLLER

In StudentServiceImpl implement the method which will search students by their name or surname:

5 min

```
@GetMapping("/search/{term}")
public @ResponseBody List<StudentDTO> searchStudents(@PathVariable(value = "term") String searchTerm) {

    if (StringUtils.isEmpty(searchTerm)) {
        return new ArrayList<StudentDTO>();
    }

    return this.studentService.searchStudents(searchTerm);

}
```

STEP 12

TEST SEARCH STUDENTS USING CURL

2 min

Open a shell or Git Bash and execute:

```
curl "http://localhost:8081/students/search/dor"
```

This will return results like:

```
[
    {
        "id": 1,
        "name": "Doris",
        "surname": "Day"
    },
    {
        "id": 3,
        "name": "Leles",
        "surname": "Pandoroglou"
    }
]
```

THANK YOU FOR YOUR PATIENCE

WE ARE HIRING

FOR MORE INFORMATION VISIT PRODYNA.COM

Come and talk with us in our Devs Kariera booth!

github.com/manios

+30 2311 821503

Facebook/prodyna

info@prodyna.com

@PRODYNASE