FullStack #WebDevelopment Bootcamp

Course Week 1 - Introduction

Introduction

- ***Name**: Antonio Marasescu-Duran*
  - *Full-Stack Software Engineer at Msg*
  - *Currently working at work on a project using Angular, Spring Kotlin, Corda Blockchain*
  - *Side-Projects include: Node.js, AWS, Terraform, Ethereum Blockchain*
  - *Other interests: Gaming, D&D*

- ***Contact**:*
  - *Email: antonio.marasescuduran@gmail.com*
  - *Linkedin: https://www.linkedin.com/in/antonio-marasescu-duran-953aa2177/*

Introduction

- ***Name**: Luchian Alex-Sebastian*
  - *Web Developer at Msg*
  - *Current work: commerce website using Angular and SAP technologies (Hybris and Spartacus Storefront)*
  - *Side-Projects include: AWS, SAP UI5, SEO, Java*
  - *Other interests: mountain biking, climbing, nature*

- ***Contact**:*
  - *Email: contact@alexluchian.com*
  - *Linkedin: https://www.linkedin.com/in/alex-luchian-b157a1172/*

Course Content

The course will last usually 2-hours over a period of 9 weeks.

- *Week 1: Introduction and Basic Setup*
- *Week 2: Introduction to TypeScript, CSS, GIT and Web Development Concepts*
- *Week 3: Backend: API Startup, Nest.js, OpenAPI*
- *Week 4: Backend: Database Connection, ORM*
- *Week 5: Backend: API Authentication and Authorization*
- *Week 6: Frontend: UI Startup, Reactive Form*
- *Week 7: Frontend: Authentication*
- *Week 8: Frontend: Admin Application Authorization and Shared Component Library*
- *Week 9: Assignment, Open Discussions and Feedback.*

What can you expect?

- *By the end of the course you will have a fully functional web application composed of:*
  - *An API which will connect to a Database and expose a set of documented endpoints*
  - *An UI which will connect and authenticate to the API*

- *You will learn about the following concepts:*
  - *TypeScript, JavaScript, CSS, HTTP, Promises, Git*
  - *Dependency Injection, Client-Server Architecture*
  - *ORM, DTO, Mono-repository, OpenAPI documentation*
  - *Authentication Strategies (Basic, Json Web Token (JWT))*
  - *Single Page Applications, Observables, Angular*

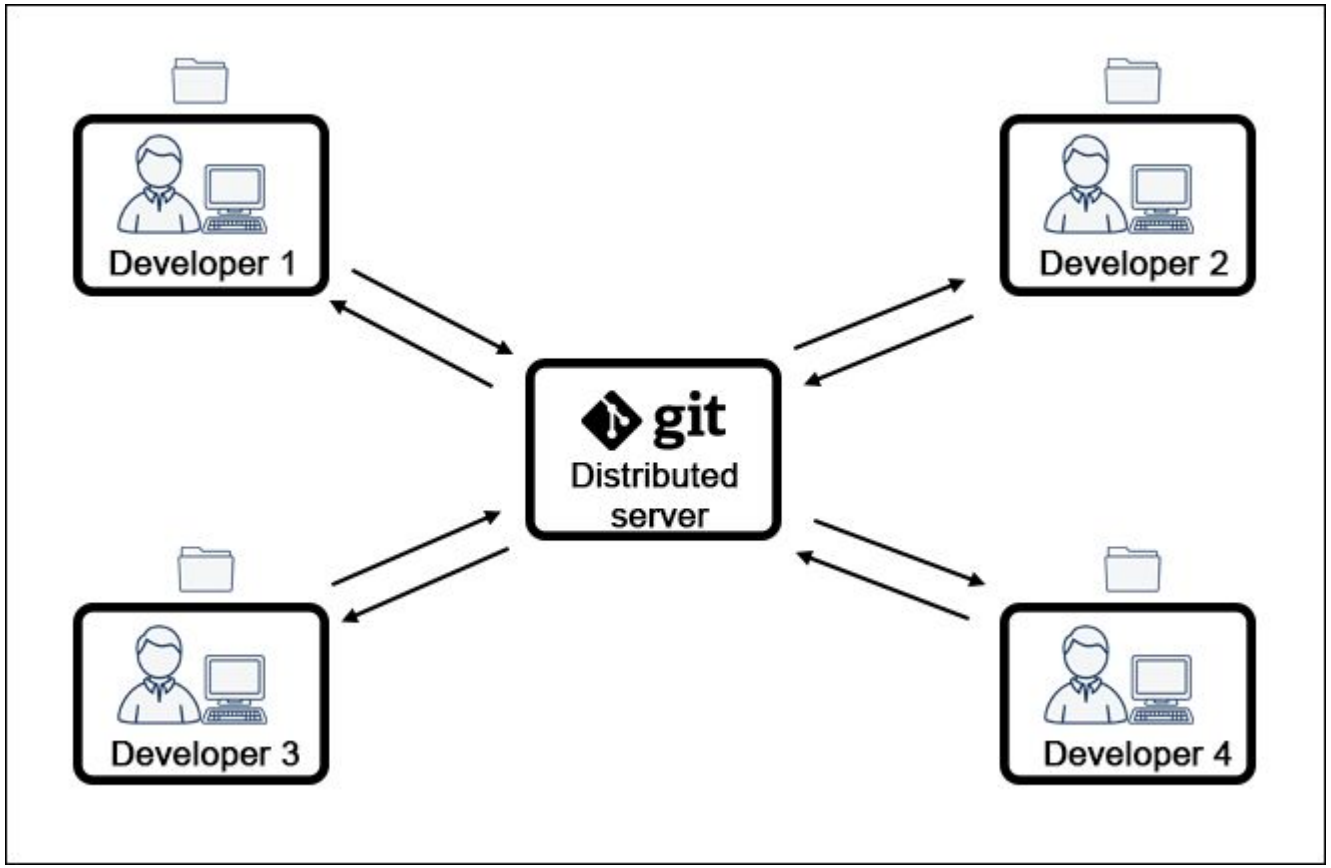
What do you expect?

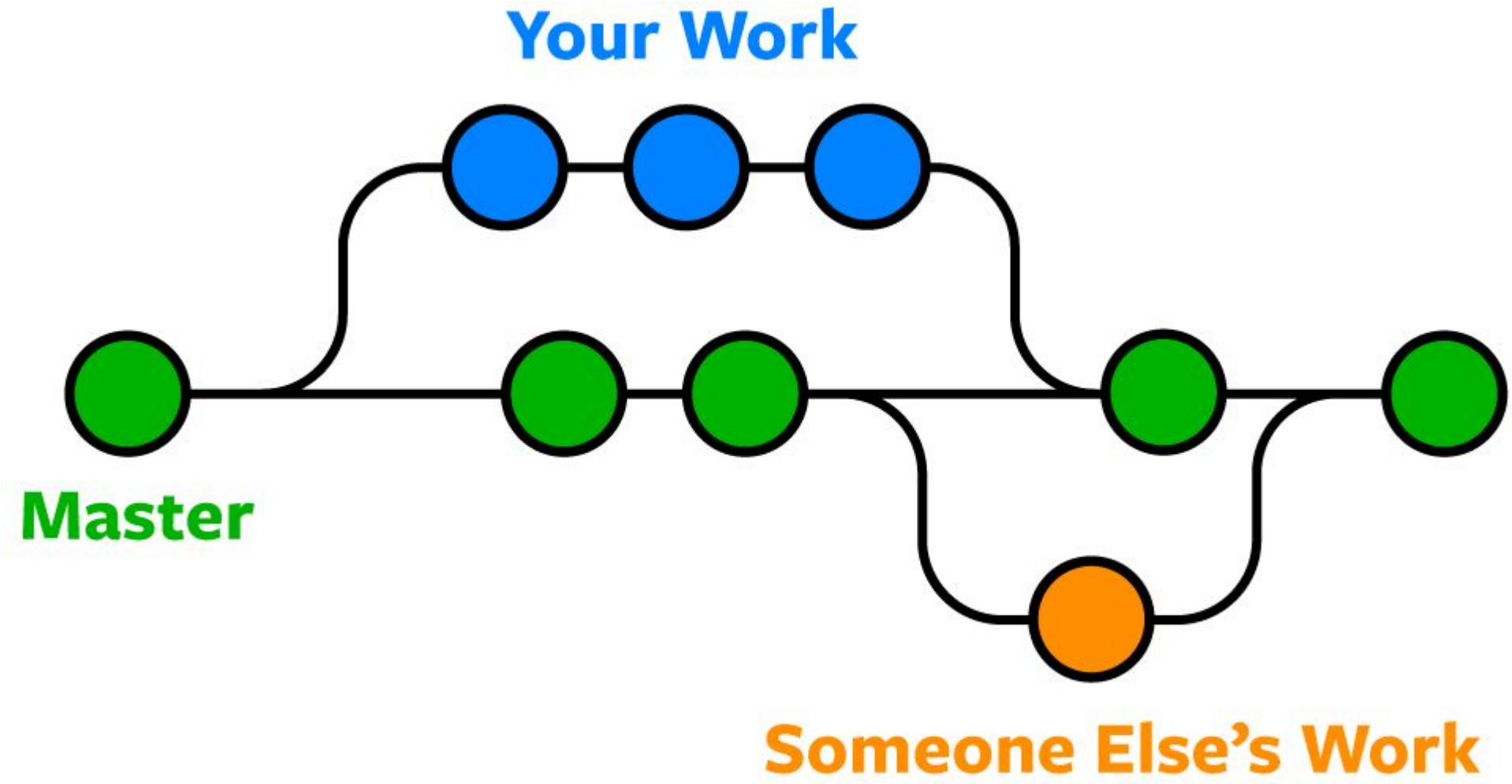
What are your expectations from this course?

Project Setup

- *Install Git: https://git-scm.com/downloads*
  - *What is Git and why do we need it?*
- *Install Node.js: https://nodejs.org/en/*
  - *What is Node.js?*
  - *Open a git bash and run the following command: `npm –version`*
- *Install Postgres: https://www.postgresql.org/download/windows/*
  - *Why do we need it?*
- *Install Webstorm: https://www.jetbrains.com/webstorm/download/#section=windows*
- *Install Postman: https://www.postman.com/downloads/*

Git: An Introduction




Your Work
Master
Someone Else's Work

Git: An Introduction

Common Git Commands:

- **git clone <some-repository-url>**
- **git fetch** // retrieve information about the latest changes in the repository
- **git pull** // retrieve the actual changes for your current branch
- **git commit -m "Message about what changes you introduced"** // commit locally your changes
- **git push origin master** // push to the remote repository on the "master" branch your commits
- **git checkout <branch_name>** // navigate to another branch in the repository

Git: Common Workflows

Start work on a new project and work only on master branch

1. git clone <some-repository>
2. Open the project folder and code something
3. git commit -m “I have fixed the project”
4. git push origin master

Start work on an existing project and work only on “fix_application” branch

1. git fetch
2. git checkout fix_application
3. git pull
4. Code something
5. git commit -m “I have fixed the app”
6. git push
7. Create a Pull/Merge Request
8. Request a Code Review from another developer
9. Is Approved ? => Merge your changes into master

Node.js & NPM

Node.js = a cross-platform (windows/linux/etc) open-source server environment

- you use it to develop any kind of server application (e.g.: An API to retrieve the latest pokemons)

NPM = a software registry & package manager for JavaScript (an TypeScript) libraries

- here you will find any kind of useful libraries you might need to implement your application
- when you initialize your project folder with npm you will:
  - run “npm init”
  - this will create a **“package.json”** => this will mark what libraries you will install for your application
  - run “npm install” or “npm ci” (ci = clean install)
  - this will create a **“package-lock.json”** => this will mark what version of your libraries you will be using (so different developers use the same version)

More on this later in the practical session

Links

Course Content + Presentations
https://github.com/msg-utcn/course-content

Discord
https://discord.gg/29rVN692cm