**29. Angular as a Platform & Closer Look at the CLI**

431. A Closer Look at "ng new"

**Angular CLI:**

უახლესი **CLI**-ს დასაინსტალირებლად, საჭიროა ბრძანება: npm install -g @angular/cli@latest

თუ მაგალითად **CLI**-ში დავწერთ მხოლოდ ng new, იგი შეგვეკითხება პროექტის სახელს.

ნებისმიერი ბრძანების გვერდით შეგვიძლია დავწეროთ **--help**, რომელიც კონკრეტულ ბრძანებაზე ყველა შესაძლო ოფშენს გვაჩვენებს: მაგ.: ng new --help

**მაგალითის განხილვა:**

---dry-run- კონტენტის შექმნის გარეშე გვაჩვენებს **CLI**რა მოქმედებას მოგვთხოვდა.

*დანარჩენ ინფორმაციას ანგულარის დოკუმენტაციაში შეგვიძლია გავეცნოთ...*

**29. Angular as a Platform & Closer Look at the CLI**

433. Understanding the Config Files

**Angular CLI Config Files 1.1:**

ანგულარის აპლიკაციაში არსებული ფაილების განხილვა:

* **.editorconfig** - აღნიშნულ ფაილს ტექსტედიტორები მოიხმარენ, მაგალითად *VSCode* და აწერს ზევიდან ედიტორს თავის პარამეტრებს.
* **.gitignore**- იგი ეუბნება გიტს, თუ რომელი ფაილები არ შეინახოს.
* **.browserslistrc** - იგი გამოიყენება ფროდუქშენ მოდში ბილდისას და ანგულარ CLI-ს ეუბნება, რომელმა ბრაუზერებმა დაუჭირონ მხარი აპლიკაციას.
* **karma.conf.js** - იგი გამოიყენება ტესტინგისთვის.
* **package-lock.json** - იგი გენერირდება*package.json*-ის მიერ.
* **package.json** - იგი ინახავს ე.წ. დეფენდესის, ანუ მესამე პარტიის მიერ მოწოდებულ პაკეჯებს. ანუ, როდესაც npm install-ს გავუშვებთ, იგი დაამოდიფიცირებს *node\_module*-ს იმაზე დამოკიდებულებით, თუ *package.json*-ში რა წერია, და ასევე დააგენერირებს *package-lock.json* ფაილს
* **tsconfig ფაილები** - იგი ინახავს ტაიპსკრიპტის კონფიგურაციას.

**29. Angular as a Platform & Closer Look at the CLI**

433. Understanding the Config Files

**Angular CLI Config Files 1.2:**

**tsconfig.json**-ში მაგალითად შეგვიძლია ჩავამატოთ **compilerOptions**-ში: "strict": true, რომელიც სტრიქტ მოდს ჩართავს.

*დანარჩენს შეგვიძლია ოფიციალურ დოკუმენტაციაში გავეცნოთ...*

ბოლოს კი გვაქვს **tslint.json**, რომელიც ამოწმებს კოდის ხარისხს, ანუ რამდენად მიჰყვები **best practices**.

**29. Angular as a Platform & Closer Look at the CLI**

434. Important CLI Commands

**Angular CLI some commands:**

მაგალითად სხვა პორტის შესაქმნელად, ვიდრე 4200-ა, შეგვიძლია**--port** ჩავამატოთ: ng serve --port 2020

ნებისმიერ ბრძანებას უბრალოდ მივუწეროთ **--help** და იქვე ვნახავ ყველა ფროფერთის...

**29. Angular as a Platform & Closer Look at the CLI**

436. Angular Schematics - An Introduction

**Angular CLI Schematics:**

იგი გვეხმარება საკუთარი ng გენერეიტინგის ქომანდების შექმნაში.

იგი დოკუმენტაციაში შეგვიძლია ვნახოთ.

ასევე არსებობს მესამე მხარის მიერ შექმნილი სქემატიკები, მაგალითად **Angular Material.**იგი ემატება შემდეგი ქომანდით: ng add @angular/material

**29. Angular as a Platform & Closer Look at the CLI**

438. Using Custom "ng generate" Schematics

**Angular CLI Angular Material:**

მაგალითზე, ანგულარ მატერიალის სქემატიკის მუშაობს ნახვა შეგვიძლია შემდეგი ბრძნებით: **ng generate @angular/material:nav main-nav**

რომელიც შექმნის ახალ კომპონენტს და შიგნით ჩააგდებს კონკრეტულ ნავიგაციას.

**29. Angular as a Platform & Closer Look at the CLI**

439. Smooth Updating of Projects with "ng update"

**Angular CLI Updating:**

ანგულარის დააფდეითება შეგვიძლია მარტივი ბრძანებით: ng update

**29. Angular as a Platform & Closer Look at the CLI**

441. Understanding "Differential Loading"

**Angular CLI Differencial Loading:**

ანგულარს გააჩნია აღნიშნული, რომელიც აპლიკაციის ჩამოწერამდე იწერება ეს კოდი და ამოწმებს, რომელ ბრაუზერს იყენებს მომხამრებელი და მისი მიხედვით გზავნის აპლიკაციას, ანუ რაც უფრო ახალია ბრაუზერი, მით უფრო მცირე ზომის კოდია საჭირო.

იგი კონფიგურირდება **.browserlistrc**-ში.

ასევე, უფრო ფართოდ შეგვიძლია ვნახოთ, **polyfills.ts-**შიც.

**29. Angular as a Platform & Closer Look at the CLI**

442. Managing Multiple Projects in One Folder

**Angular CLI რამდენიმე პროექტი ერთ angular.json-ში:**

მის დასაგენერირებლად გვჭირდება **CLI** ბრძანება: ng generate application სახელი

იგი შექმნის ცალკე ფოლდერს **projects**და იქ ჩააგდებს.

**ng serve**-იც შემდეგნაირად შეგვიძლია გამოვიყენოთ: ng serve --project=სახელი

ან **angular.json**-ში **defaultProject**შევცვალოთ.

შესაძლოა აღნიშნული დოკომენტაცია არ მოგვეწონოს. ამისათვის, შეგვიძლაი უბრალოდ, სულ თავიდან დავიწყოთ და შემდეგნაირად შევქმნათ ანგულარის აპლიკაცია:

ng new სახელი --create-application=false

იგი მოგვცემს ფოლდერს, თუმცა **app**ფოლდერი არ იქნება. აი შემდეგ კი გავუშვათ ზემოთ მოცემული ბრძანებები, რაც ცალკე ფოლდერშ შექმნის სხვადასხვა პროექტისთვის.

**29. Angular as a Platform & Closer Look at the CLI**

443. Angular Libraries - An Introduction

**Angular CLI ბიბლიოთეკა:**

იგი სტანდარტული აპლიკაციისგან იმით განსხვავდება, რომ იგი დამოუკიდებლად არ იკითხება, როგორც აპლიკაცია, ამის ნაცვლად, მანდ შექმნილი კოდი საზიარო იქნება ყველა სხვა აპლიკაციისთვის.

მაგალითია**Angula Material,** რომელიც ანგულარის ბიბლითეკას წამოადგენს.