

Ionic 8 – Framework für „hybride apps“

1. Theorie
2. Installation von
 - a. node.js - Laufzeitumgebung (runtime environment) um JavaScript am PC ausführen zu können
 - b. ionic – wird bequem durch node.js installiert (mit NPM)
 - c. angular - wird ebenfalls bequem mit Hilfe von node.js installiert (mit NPM)
 - d. capacitor NICHT installieren, das passiert jeweils mit einem neuen Projekt
 - e. Visual Studio Code als Entwicklungsumgebung

<https://ionicframework.com>

1)Theorie:

Laufen nicht nur im Web, sondern auch auf nativen Geräten in native Apps.

Normalerweise kann man z.B. kein iOS-App erstellen, ohne dabei ein Mac-System bei der Herstellung zu nutzen. Mit Ionic ist das kein Problem, da geht es auch Mac-System.

Das Ergebnis kann man auf den Google Play und den ios App Store stellen und anbieten zum Download.

Ist Open Source. Ist ein

- Frontend-Framework zum Entwickeln von
- Hybriden mobilen Apps mit
 - HTML5 und
 - Angular
- für iOS- und Android-Plattformen. Somit kann man eine Anwendung auf verschiedenen Plattformen nutzen.

Ionic Framework is an open source **UI toolkit** for building performant, high-quality mobile and desktop apps using web technologies — **HTML, CSS, and JavaScript** — with integrations for popular frameworks like Angular, React, and Vue.

Nutzt eine Codebasis, nämlich HTML, JavaScript und CSS.

Ist eine große „library“ für Componets die sind „ready to use“ z.B. „ion-action-sheet“.

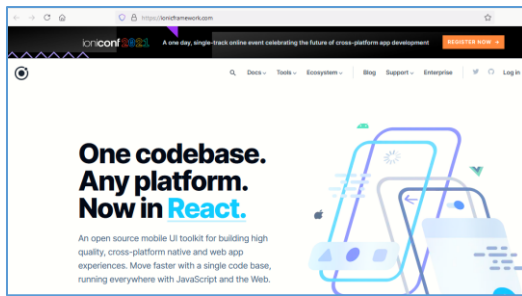
Benötigt „node.js“.

1. Ionic 1 (2013)

- Technologie: Basierte auf AngularJS (Angular 1).

2. Update to the latest version of Ionic 8:

```
npm install @ionic/angular@latest
```



<https://ionicframework.com>

Hat eine sehr gute Dokumentation. Leicht zu finden durch Klick auf die Suche und dann wird es schon vorgeschlagen „docs“.

Ionic Native Plugins: bietet viele Plugins wie z.B. Barcodescanner, Kalender, Kamera, Taschenlampe.

Ionic Plattform:

1. Web Components – damit erzeugt man das, was man am Bildschirm sieht – z.B. <ion-app> oder <ion-button>. Diese Elemente haben bereits ein spezielles Layout, ohne dass man CSS schreiben muss. Es gibt alle UI-Elemente die man nutzt, wie z.B. cards, buttons – sie sind „ready to use“ 😊
 2. Möglichkeit von Zugriffen auf native Elemente wie z.B. Kamera. Diese Ionic Native Schnittstellen nutzen die Ionic eigene Plattform „Capacitor“. Damit sind Zugänge zu den nativen Hardwarekomponenten, z.B. Taschenlampe möglich die mit JavaScript alleine ja nicht möglich wären. Mit „Capacitor“ wird somit ein App erzeugt, das wie ein nativ mobiles App läuft.
- Web app, running as a native mobile app
3. Ionic CLI – (comand line interface) + Angular CLI dient für das Projekt Management und den Build Workflow, dient z.B. für den live-reload

Das Ionic-Framework selbst ist mit dem Tool „stencil“ erstellt worden.

<https://www.youtube.com/watch?v=r2ga-iXS5i4>

<https://ionicframework.com/docs/native/>

Ionic erzeugt „cross-kompilierte mobile Apps“

Die Grundidee ist es, native mobile Apps für die unterschiedlichen mobilen Betriebssysteme auf Grundlage **EINER Quellcodebasis** zu generieren. Dazu werden Cross-Compiler eingesetzt, die den Quellcode ohne weitere Zwischenschicht für unterschiedliche Betriebssysteme cross-kompilieren, sodass eine native mobile App entsteht. Entwicklungswerkzeuge dafür sind Xamarin (iOS), React Native und Apache Cordova (Capacitor) mit Ionic.

Ionic und Angular ist eine perfekte Kombination. Man muss aber nicht Angular nutzen. Man kann auch Vanilla JS oder andere Frameworks verwenden (z.B. React), oder sogar ohne Framework arbeiten.

Info:

- Zuerst erzeugt man in ionic die Darstellung (user interface) z.B. einen Button
- Dann den Code, damit dieser Button reagiert z.B. einen Event erstellen, damit auf einen Klick darauf etwas passiert
- Als 3. Punkt kann man mit Hilfe von CAPACITOR (bzw. früher cordova) als API eine Kamera starten

2)Installation: Nötige Programme zum Entwickeln:

Info: <https://www.youtube.com/watch?v=gplRGpkOIUY>

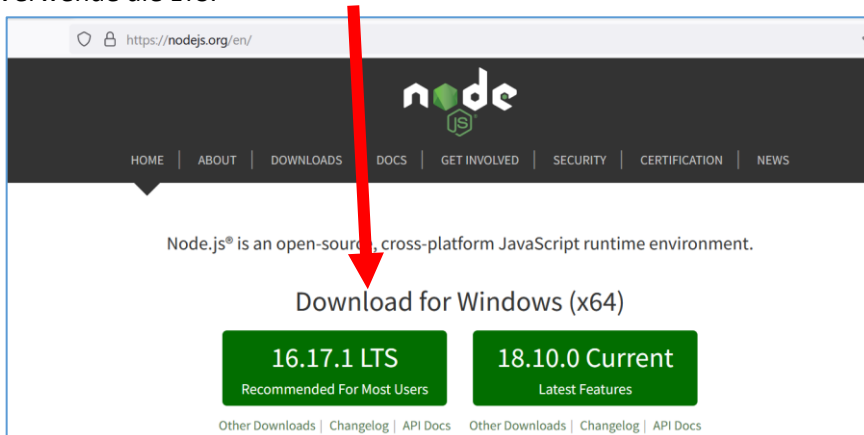
Node.js mit dem „Node Package Manager“ (NPM) zum Installieren und Nutzen von Ionic

a. <http://nodejs.org>

Wir benötigen node.js aus 2 Gründen:

1. zur Verwendung des Package-Managers – zum Managen der dependencies
2. zur korrekten Ausführung von Ionic, welches mit node.js läuft

Verwende die LTS.



Eingabeaufforderung öffnen:

Nach der Installation von Node.js öffne die Eingabeaufforderung mit der Suche von Windows und der Eingabe von „Eingabeaufforderung“ oder „cmd“.

Es soll die aktuelle Version von „node“ angezeigt werden.

Daher gib ein

- `node -v`

```
C:\Users\Hallo>node -v
v16.17.1
```

Ergebnis: Version 16.3.0

Somit ist auch der Package-Manager installiert worden. Dessen Version sieht man mit

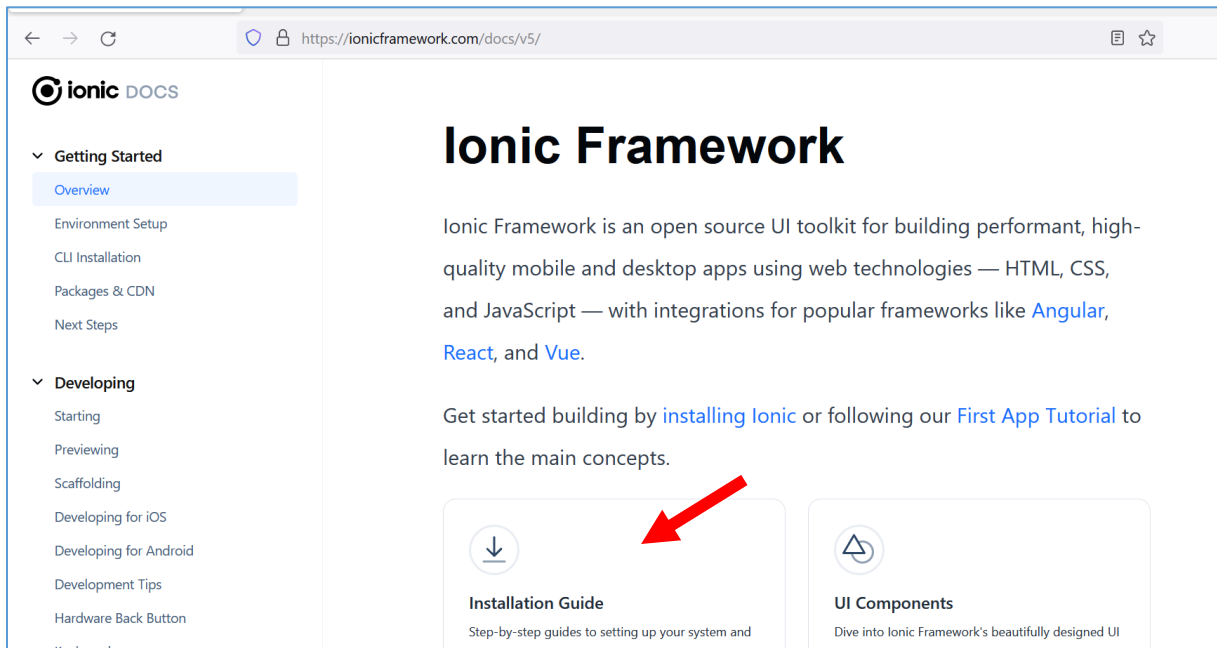
- `npm -v`

```
C:\Users\Hallo>npm -v  
8.15.0
```

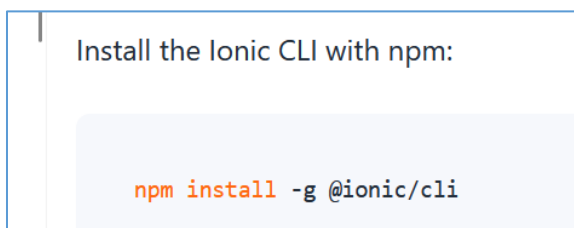
b. Ionic installieren

```
npm install -g @ionic/cli
```

Allgemein: Info siehe auch auf <https://ionicframework.com/> und dann „guide“ und „getting started“ und auf „Installation Guide“



Info: Ionic Apps werden durch die „ionic command line utility (CLI)“ entwickelt, die Capacitor (seit v6 nicht mehr Cordova) verwendet, um eine native App zu erstellen („build and deploy“).



Ebenfalls in der Eingabeaufforderung. Zum Installieren eines Packages gib ein: npm install und dann den Namen nämlich

```
npm install -g @ionic/cli
```

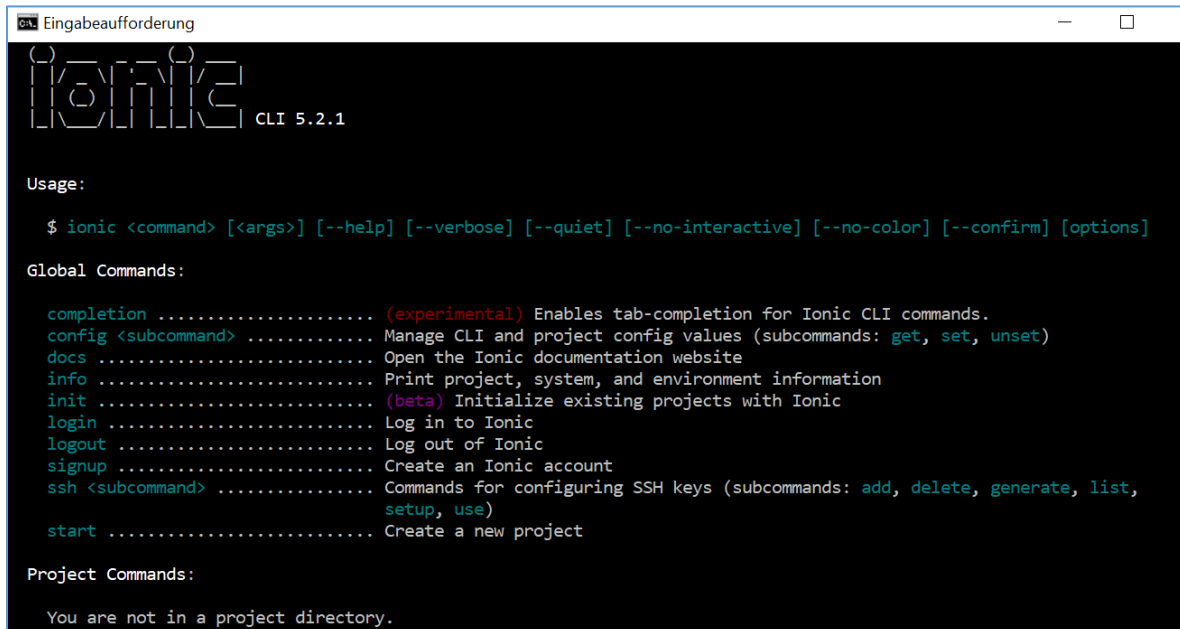
```
PS C:\2526ionic> npm install -g @ionic/cli
```

Das –g erzeugt eine globale Installation, die nicht nur in dem gerade vorhandenen Ordner gültig ist. kann 1-2 Minuten dauern.

Gib danach nur ein

- ionic

und es öffnen sich die Befehle von der ionic CLI – das dient nur der INFO



```
Eingabeaufforderung
ionic
ionic CLI 5.2.1

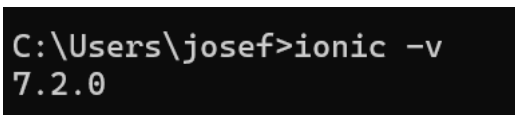
Usage:
$ ionic <command> [<args>] [--help] [--verbose] [--quiet] [--no-interactive] [--no-color] [--confirm] [options]

Global Commands:
completion ..... (experimental) Enables tab-completion for Ionic CLI commands.
config <subcommand> ..... Manage CLI and project config values (subcommands: get, set, unset)
docs ..... Open the Ionic documentation website
info ..... Print project, system, and environment information
init ..... (beta) Initialize existing projects with Ionic
login ..... Log in to Ionic
logout ..... Log out of Ionic
signup ..... Create an Ionic account
ssh <subcommand> ..... Commands for configuring SSH keys (subcommands: add, delete, generate, list,
setup, use)
start ..... Create a new project

Project Commands:
You are not in a project directory.
```

Zur Kontrolle der Version gib ein

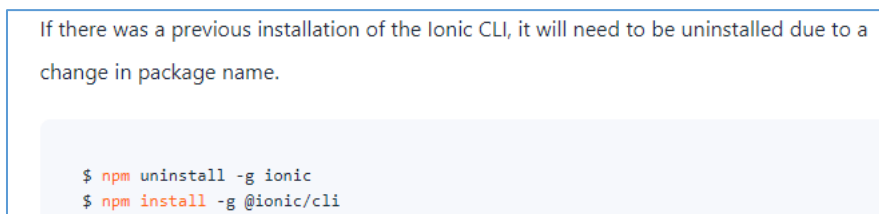
ionic -v



```
C:\Users\josef>ionic -v
7.2.0
```

Bei Update: wenn bereits eine Version von ionic auf dem Laptop vorhanden ist, sollte man nicht einfach updaten, sondern mittels „cmd“ die alte Version löschen und dann die neue aufspielen:

Siehe auch <https://ionicframework.com/docs/intro/cli>



```
If there was a previous installation of the Ionic CLI, it will need to be uninstalled due to a
change in package name.

$ npm uninstall -g ionic
$ npm install -g @ionic/cli
```

Kopiere diesen Code nacheinander. Natürlich ohne dem „\$“ – dieser ist für Apple.

Alte Version vorher uninstall:

npm uninstall -g ionic

dann neu installieren wie oben gezeigt

c. Angular installieren – wenn es noch nicht installiert wurde

Dort gib den Code ein, um die Angular zu installieren:

```
npm install -g @angular/cli
```

Achtung: cli klein geschrieben

```
C:\Users\Hallo>npm install -g @angular/cli
```

d. Capacitor wird hier noch nicht installiert, sondern jeweils direkt mit einem neuen Projekt in ionic

e. mit einer IDE arbeiten (Integrated Development Environment)

IDE installieren Beispiel Microsoft Visual Studio Code

Der Visual Studio Code ist sehr gut für Ionic-Entwicklungen <https://code.visualstudio.com/>

(Download für Windows). Man kann dort auf einige Erweiterungen zurückgreifen.

Was kann „Visualstudio“?

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

Was ist eine IDE?

Die integrierte Entwicklungsumgebung, abgekürzt IDE (Integrated Development Environment), sammelt unter einer gemeinsamen Oberfläche die wichtigsten Tools für das Erstellen von Software. Da IDEs dem Programmierer bzw. Software-Entwickler viel Komfort bei der Arbeit bieten und wichtige Arbeitsschritte zusammenfassen, sind sie in vielen Bereichen zu unentbehrlichen Hilfsmitteln geworden. Sie nehmen dem Entwickler häufig wiederkehrende Arbeiten ab und entlasten ihn von formalen Tätigkeiten wie die Verwaltung von Arbeitsergebnissen oder Programmversionen. Dem Softwareentwickler fällt es dank der IDEs wesentlich leichter, sich auf seine eigentliche Arbeit, das Programmieren, zu konzentrieren.

Die wichtigsten Bestandteile einer integrierten Entwicklungsumgebung sind:

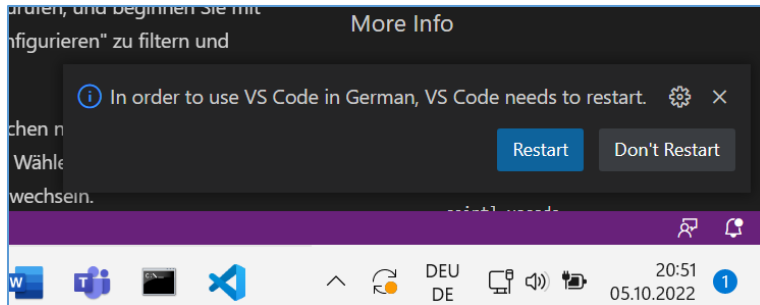
Editor mit Quelltextformatierung und Syntaxhervorhebung, Compiler, Interpreter, Linker, Debugger, Werkzeuge zur Erstellung grafischer Benutzeroberflächen (GUI-Builder), Versionsverwaltungen.

Man sollte auch unter „view“ die Extensions wählen und z.B. folgendes installieren:

- german language pack

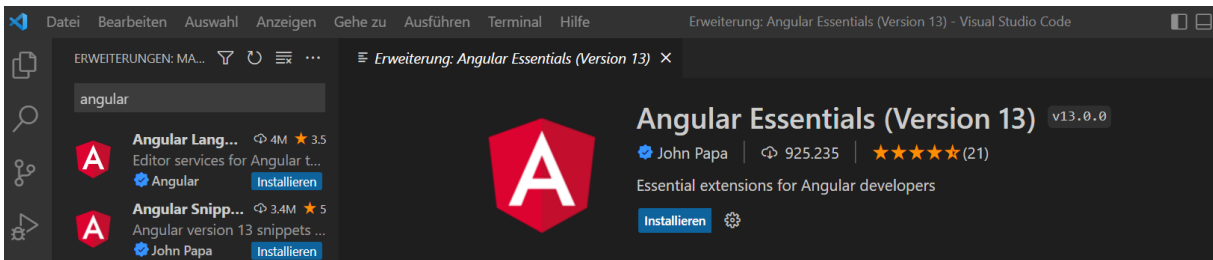


Achtung: restart das Programm:



- angular essentials

rechts unten dann „aktivieren“



Klicke in der linken Navigation wieder zurück auf: Explorer

