course Esilv - Angular & Node Js

Advanced Web Features

arnaud.nauwynck@gmail.com

Outline

- Some Advanced features
  - AG-Grid
  - HighCharts
  - Generating (Angular) Client from Swagger
  - Code Scafolding
  - Security: Cookie, Login, OAuth2, JWT, …
  - RxJs, Auto completion, ng-prime
- Final Words

Reminder

5 x CMs - Lessons

5 x TDs - Hands on

1 Project submission .... deadline 31 dec

Reminder …
expectation on project submission for Exam

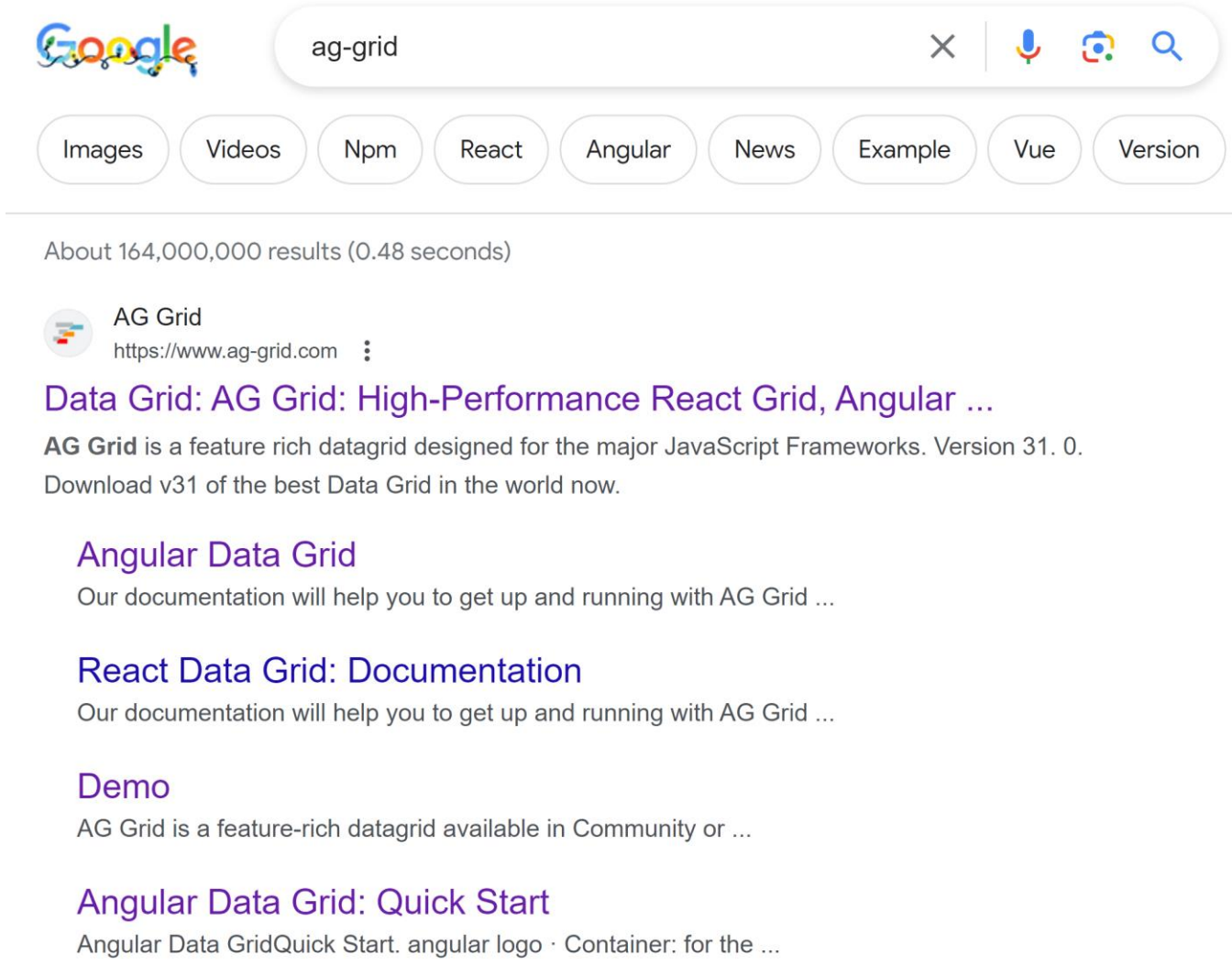
Your project MUST include at least 1 advanced feature

Outline

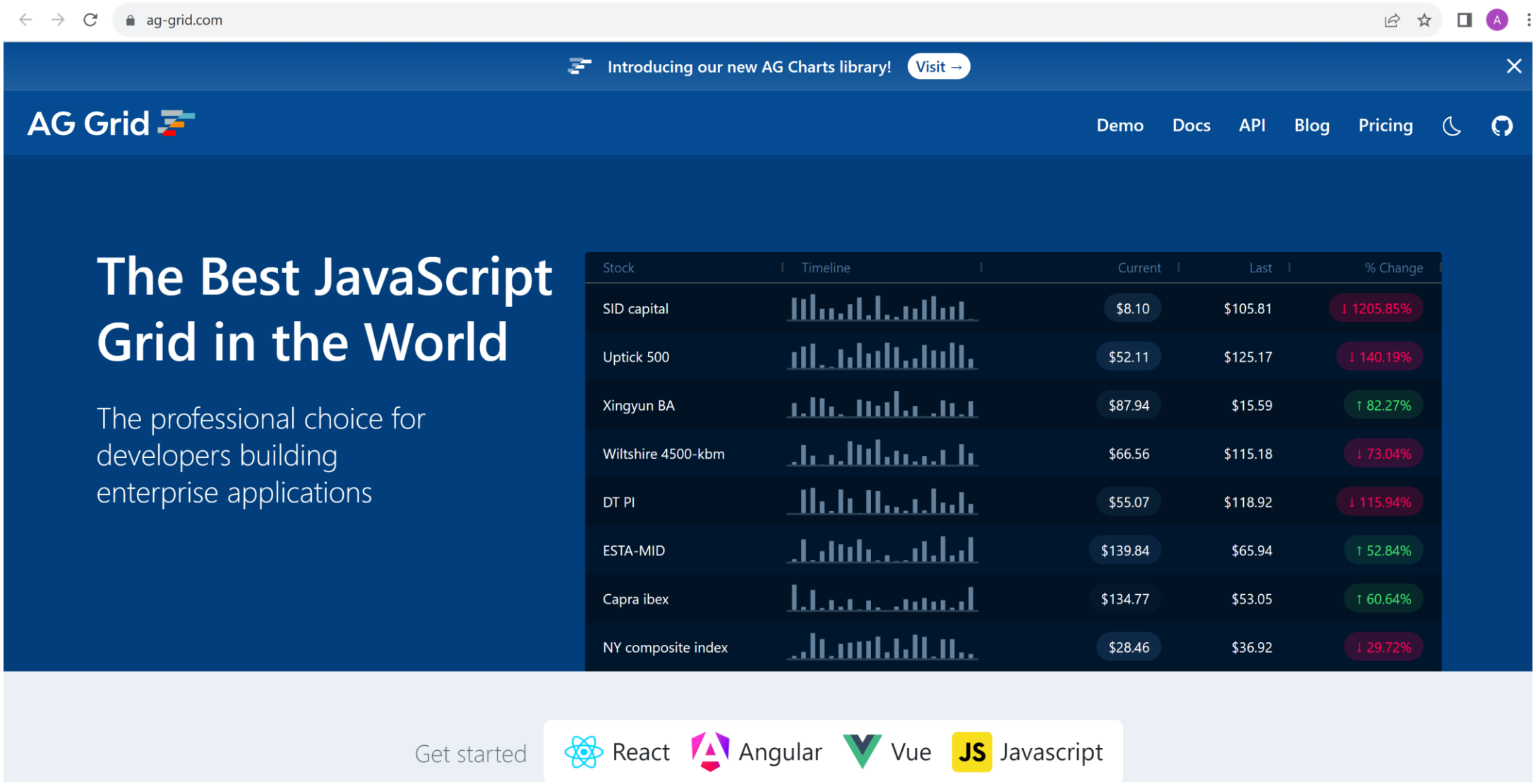
- Some Advanced features
  - AG-Grid
  - HighCharts
  - Generating (Angular) Client from Swagger
  - Code Scafolding
  - Security: Cookie, Login, OAuth2, JWT, …
  - RxJs, Auto completion, ng-prime
- Final Words

https://www.ag-grid.com/

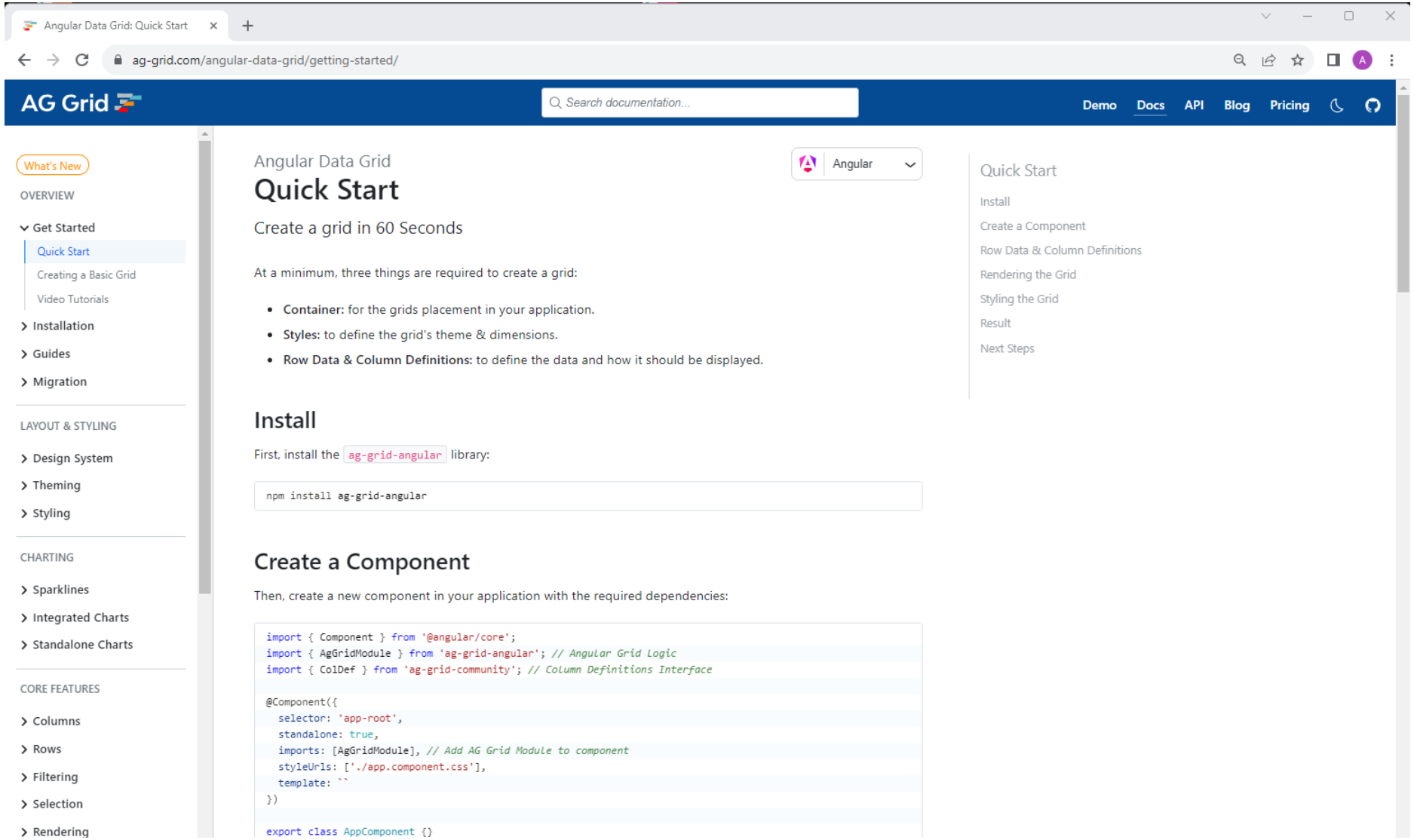
AG-Grid


Google
ag-grid
Images
Videos
Npm
React
Angular
News
Example
Vue
Version
About 164,000,000 results (0.48 seconds)
AG Grid
https://www.ag-grid.com
Data Grid: AG Grid: High-Performance React Grid, Angular ...
AG Grid is a feature rich datagrid designed for the major JavaScript Frameworks. Version 31. 0.
Download v31 of the best Data Grid in the world now.
Angular Data Grid
Our documentation will help you to get up and running with AG Grid ...
React Data Grid: Documentation
Our documentation will help you to get up and running with AG Grid ...
Demo
AG Grid is a feature-rich datagrid available in Community or ...
Angular Data Grid: Quick Start
Angular Data GridQuick Start. angular logo · Container: for the ...

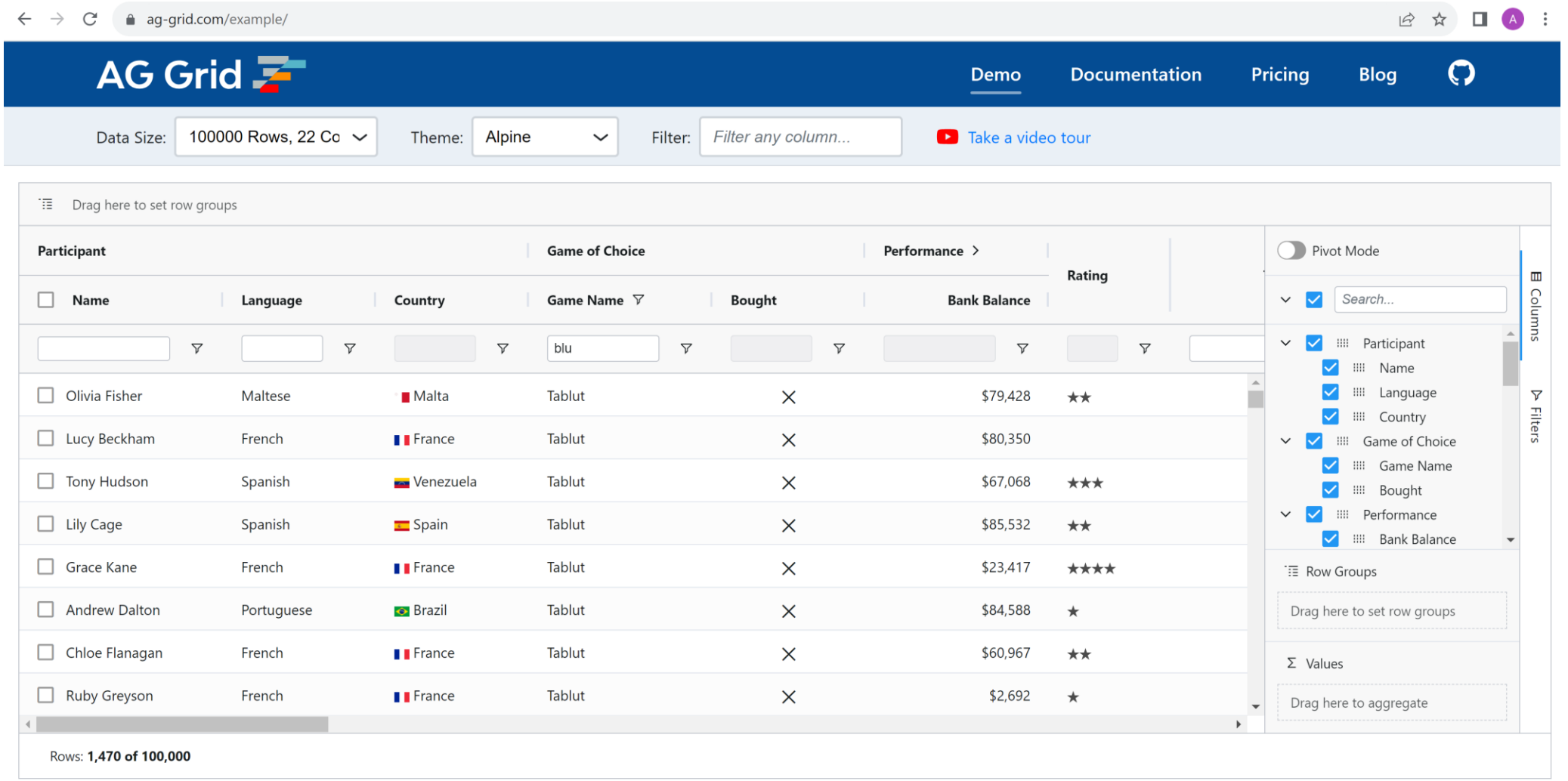
https://ag-grid.com



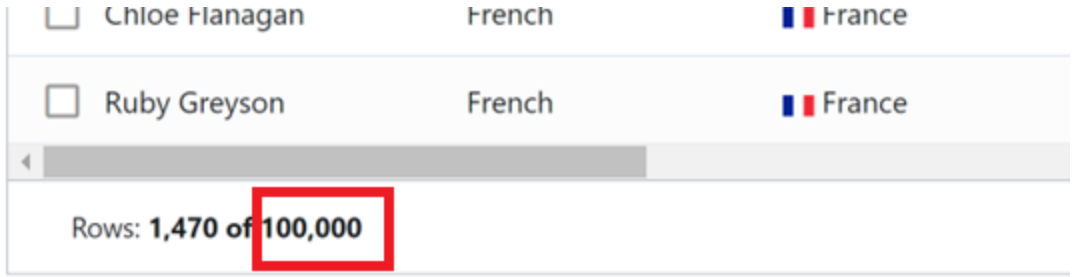
Getting Started



https://ag-grid.com/example



Did you notice? "Rows ... of 100,000"



AG-Grid performances

Naive

create millions of

Use Browser to scroll page

Virtual Scrolling

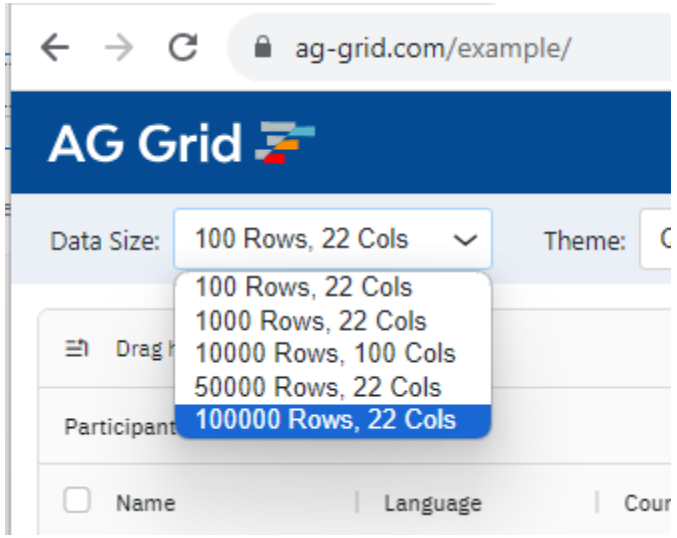
create ~50 only if visible
<div ..translateY(0px)>
<div ..translateY(41px)>
<div ..translateY(82px)>

change "div.y" position when scrolling
Scroll

Your-own-table vs AG-Grid


```
<div *ngFor="let row of rows">
   <div class="row">{{row}}</div>
</div>
```

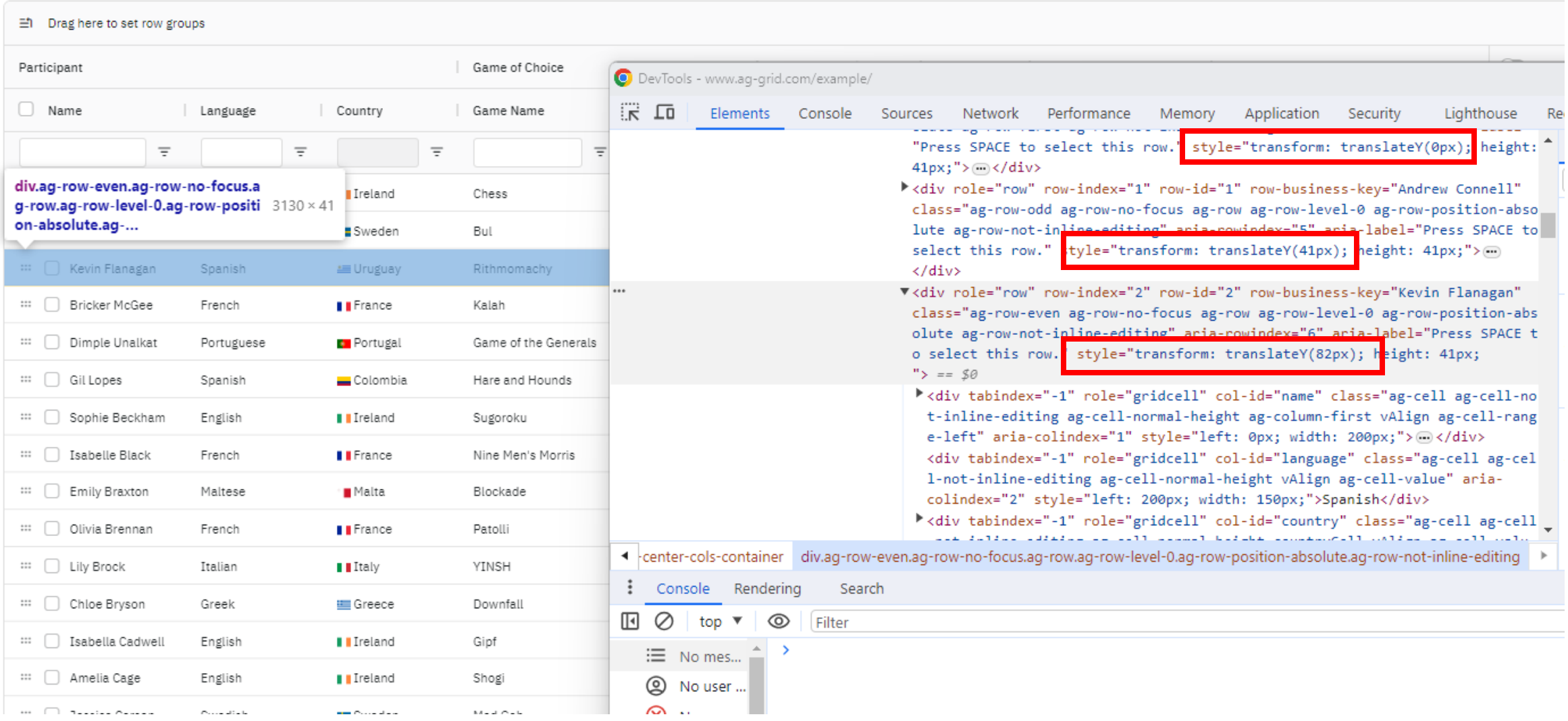
Try online yourself



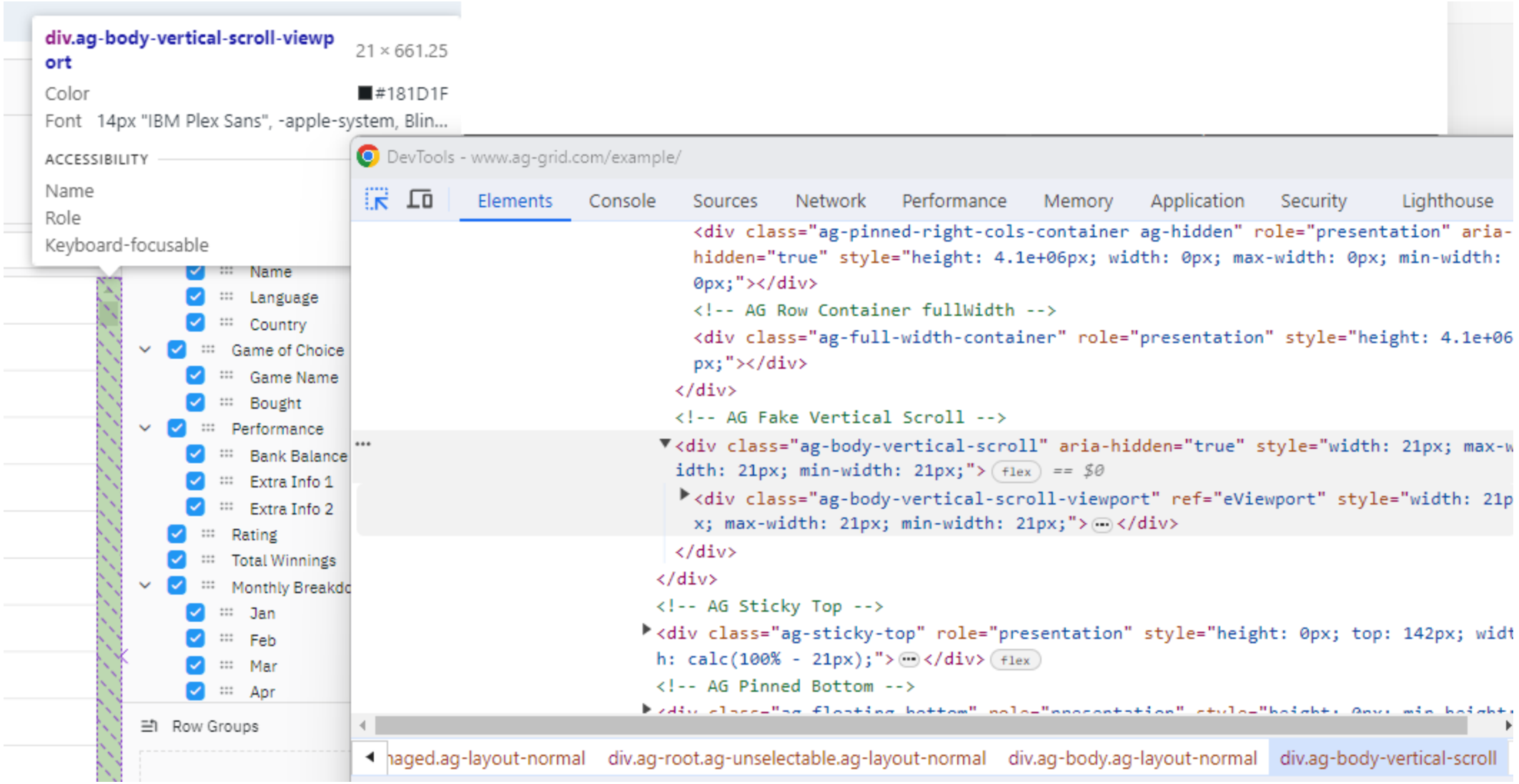
Then open Chrome DevTools

Virtual Scroll

<div "style="transform: translateY(82px); ">



AG-grid virtual-scroll



AG-Grid Features

CORE FEATURES

- Columns
- Rows
- Filtering
- Selection
- Rendering
- Editing
- Client-Side Data
- Scrolling
- Interactivity

ADVANCED FEATURES

- Group & Pivot
- Master Detail
- Accessories
- Server-Side Data
- Import & Export
- Components

Demo Example : Sorting, Export to Excel

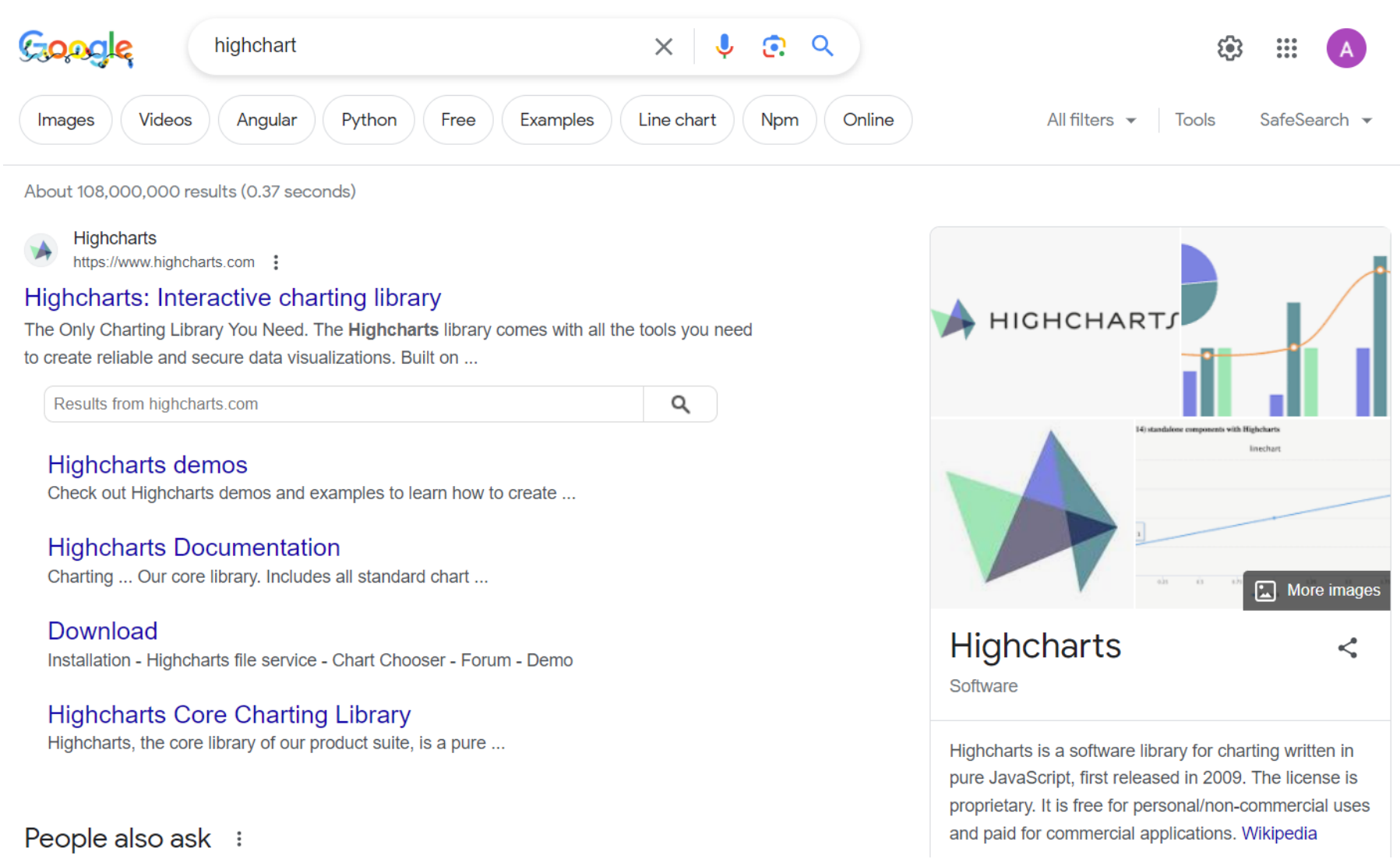
Outline

- Some Advanced features
  - AG-Grid
  - HighCharts
  - Generating (Angular) Client from Swagger
  - Code Scafolding
  - Security: Cookie, Login, OAuth2, JWT, …
  - RxJs, Auto completion, ng-prime
- Final Words

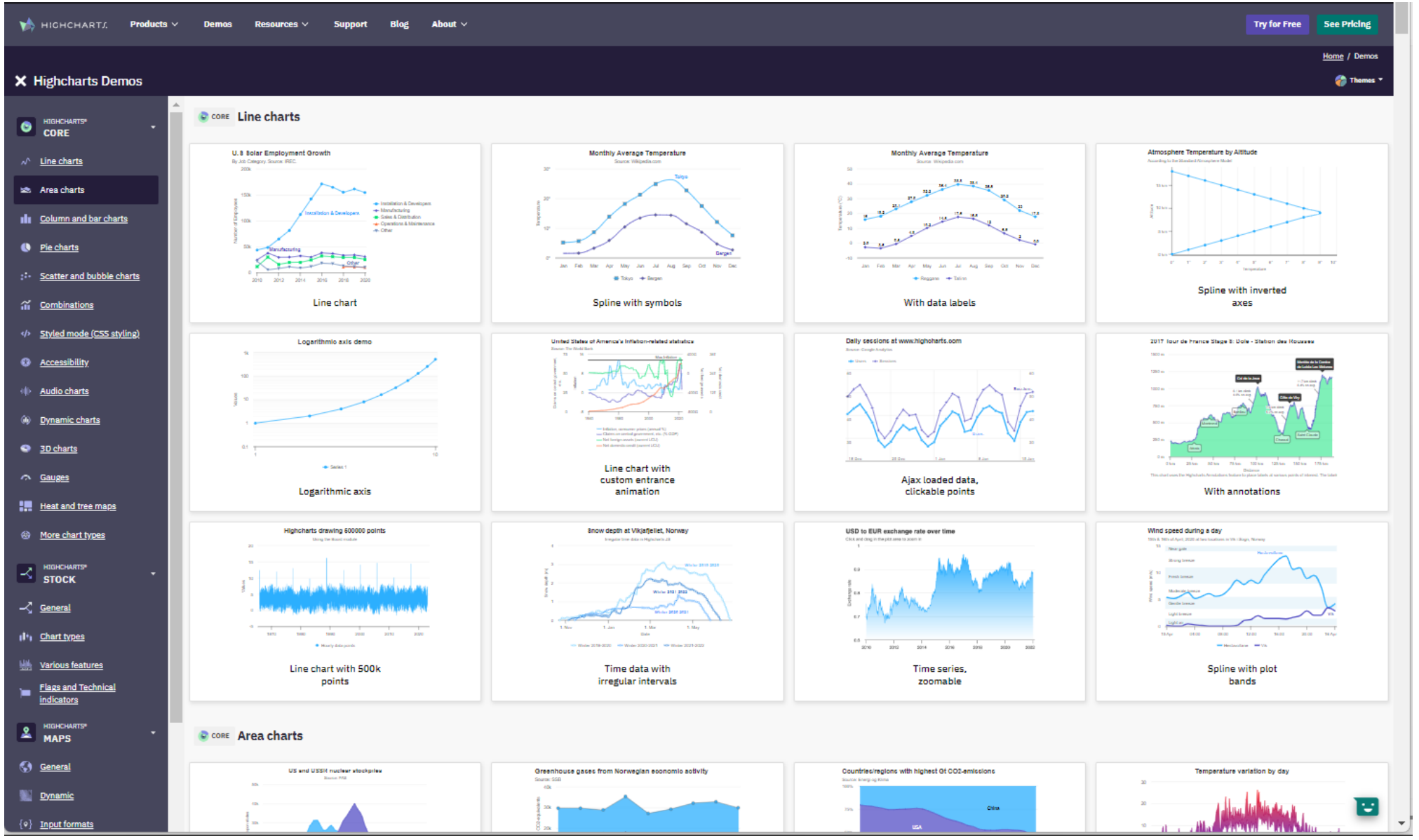
HighCharts

https://www.highcharts.com

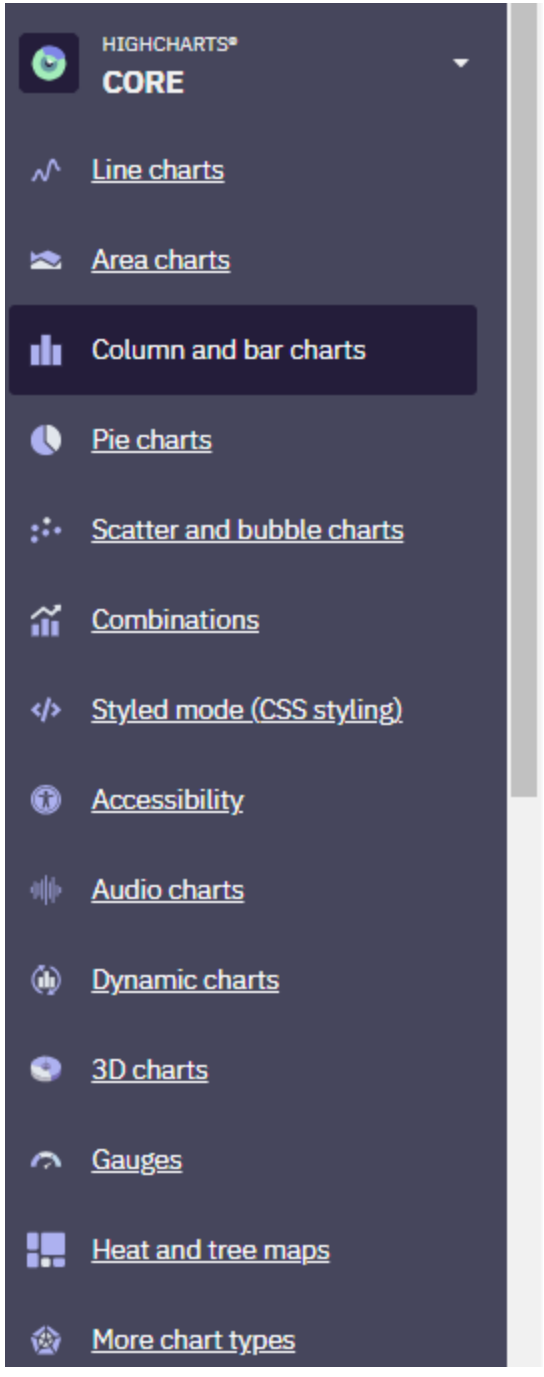
google HighCharts

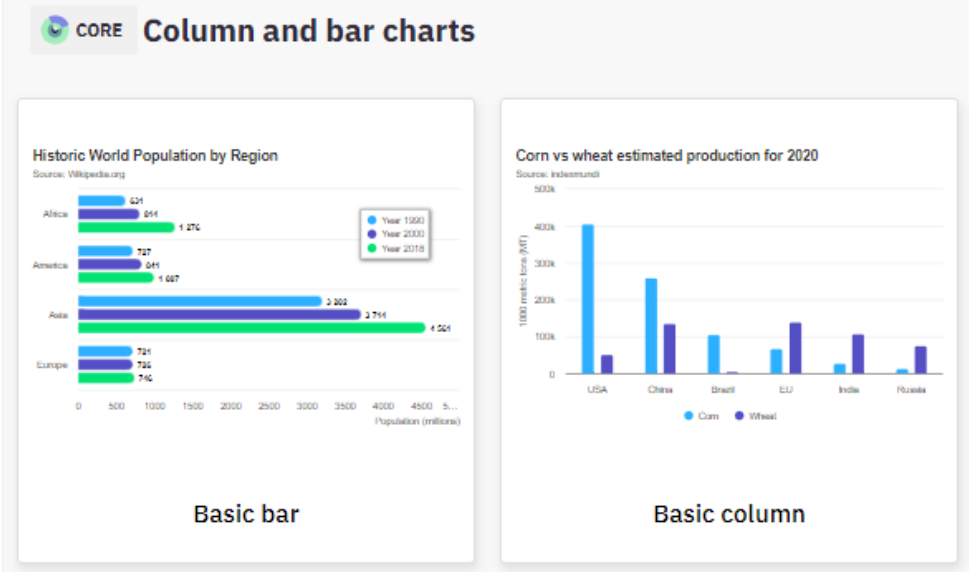


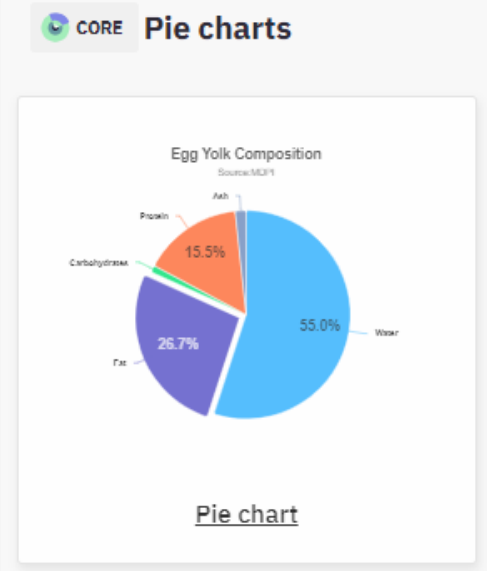
https://www.highcharts.com/demo

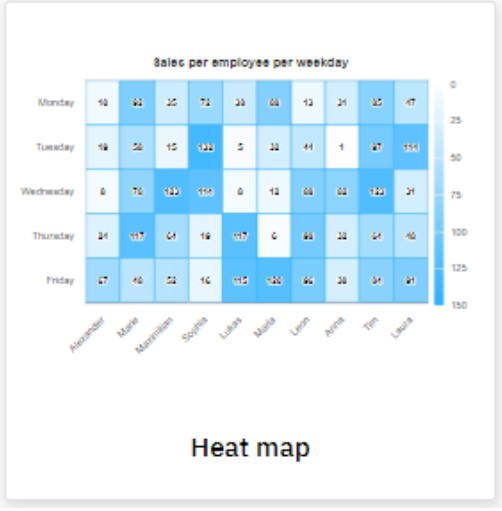


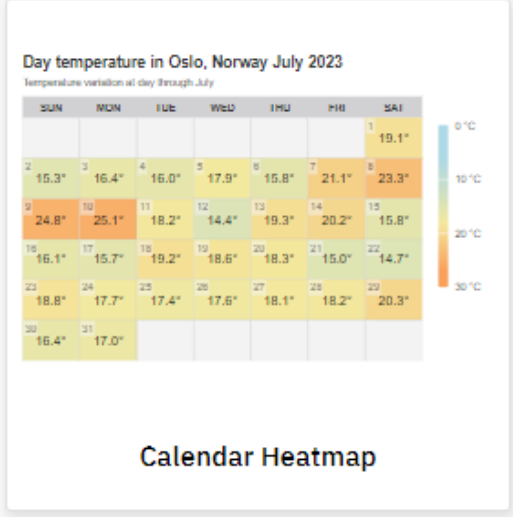
HighChart Features

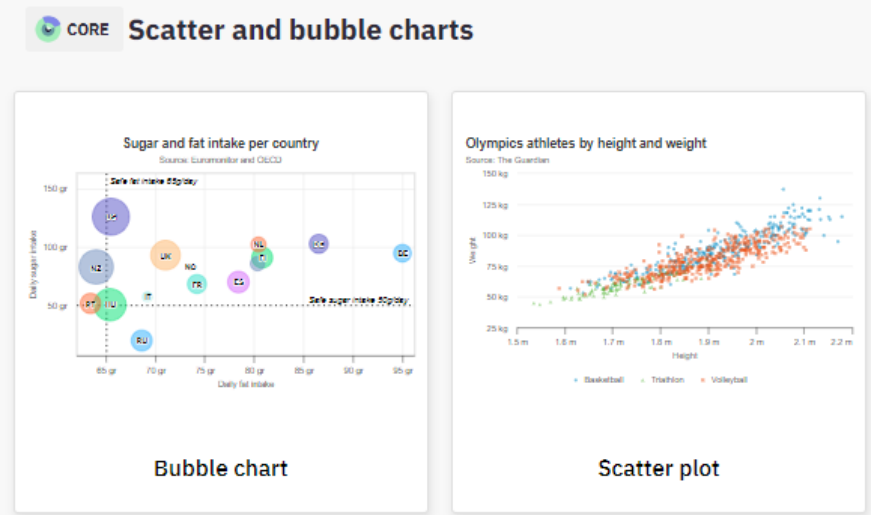




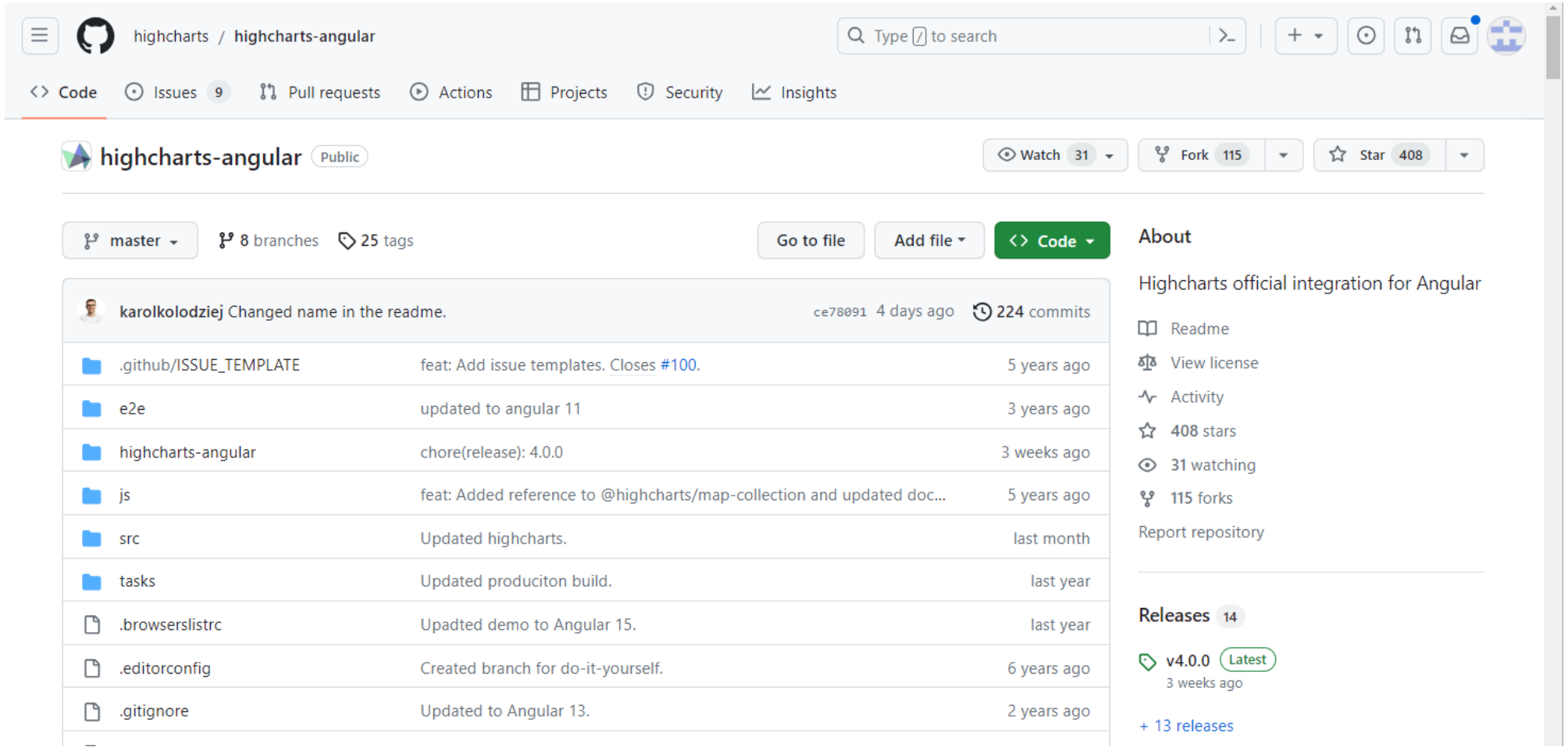








https://github.com/highcharts/highcharts-angular



Getting Started
npm install -s highcharts-angular

Installing

Get package from NPM in your Angular app:

```
npm install highcharts-angular --save
```

In your app.module.ts add the HighchartsChartModule:

```
...
import { HighchartsChartModule } from 'highcharts-angular';

@NgModule({
  imports: [
    ...
    HighchartsChartModule
```

https://github.com/highcharts/highcharts-angular#hello-world-demo

Hello world demo

To create a simple demo start with installing.

Next for `app.component.ts` 's HTML template use:

```
<highcharts-chart
  [Highcharts]="Highcharts"
  [options]="chartOptions"

  style="width: 100%; height: 400px; display: block;"
></highcharts-chart>
```

and export variables:

```
export class AppComponent {
  Highcharts: typeof Highcharts = Highcharts;
  chartOptions: Highcharts.Options = {
    series: [{
      data: [1, 2, 3],
      type: 'line'
    }]
  };
  ...
```

Notice: Alternatives To HighCharts

HTML <SVG>
HTML canvas

builtin features, but too low-level to use yourself

d3js also low-level and very complex
(redundant with Angular *ngFor="let .. of ... ")

chart.js
& MANY others libraries

Outline

- Some Advanced features
  - AG-Grid
  - HighCharts
  - Generating (Angular) Client from Swagger
  - Code Scafolding
  - Security: Cookie, Login, OAuth2, JWT, …
  - RxJs, Auto completion, ng-prime
- Final Words

Sharing (Partially) Codes / Same Technology for Backend and Frontend ?

Node
(reuse V8 !)

Chrome V8
JavaScript Engine

Chrome WebBrower

Edge
(reuse V8 !)

your-javascript-file-1.js
your-javascript-file-2.js

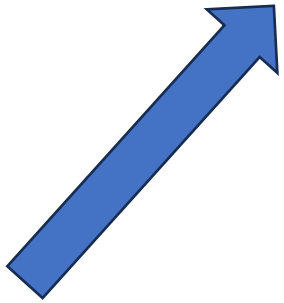
Different Code ANYWAY !!

because Http Client != Http Server
submit a Request ! = process a Request

DTO classes can be fully shared

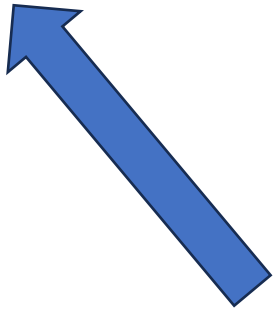
All Node-Express route methods must be translated to HttpClient

Backend != FrontEnd



Long running Application Servers
run on Linux
in a DataCenter (OnPremise, AWS, Azure, ..)

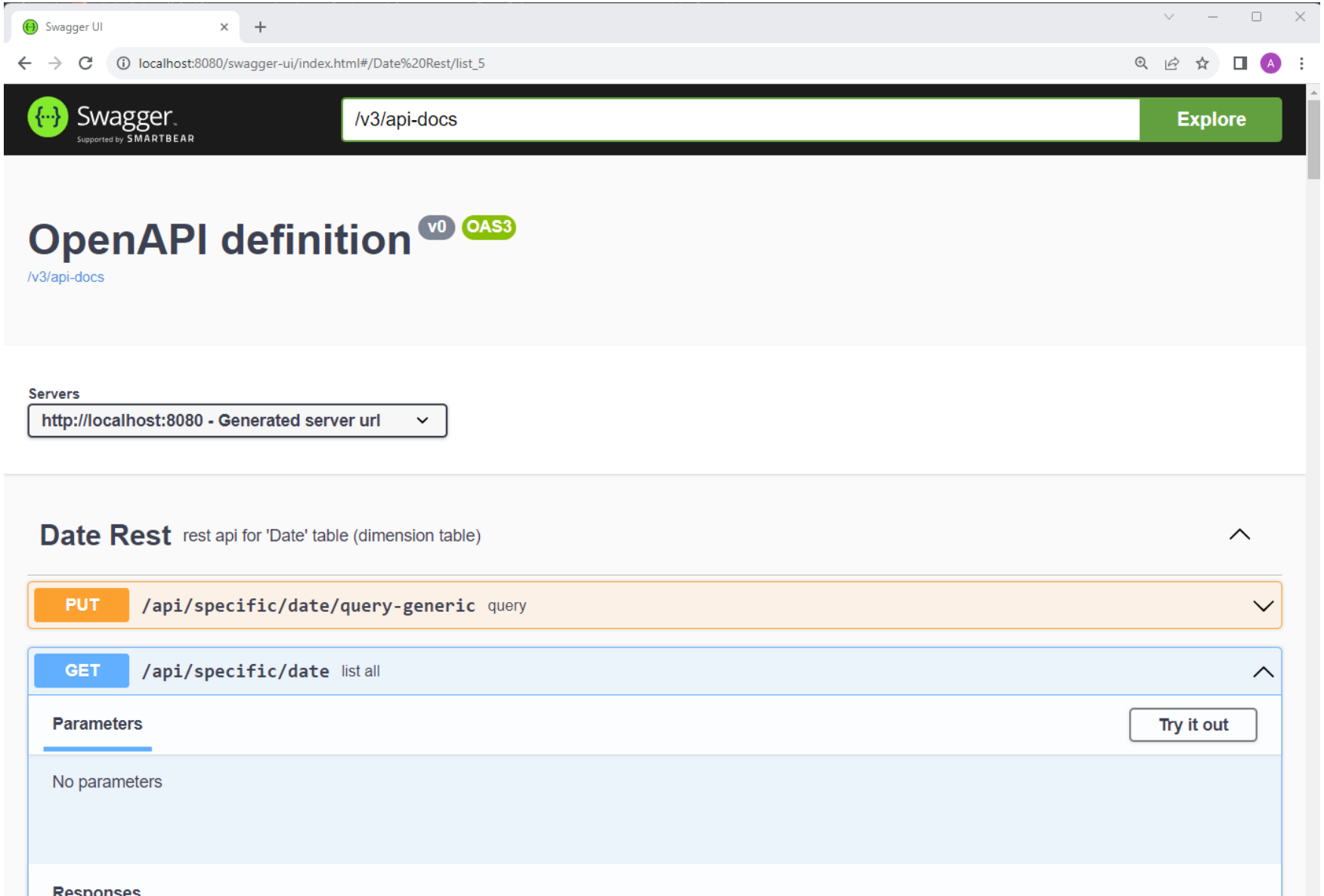
Most Probably in Java
Because ... !!



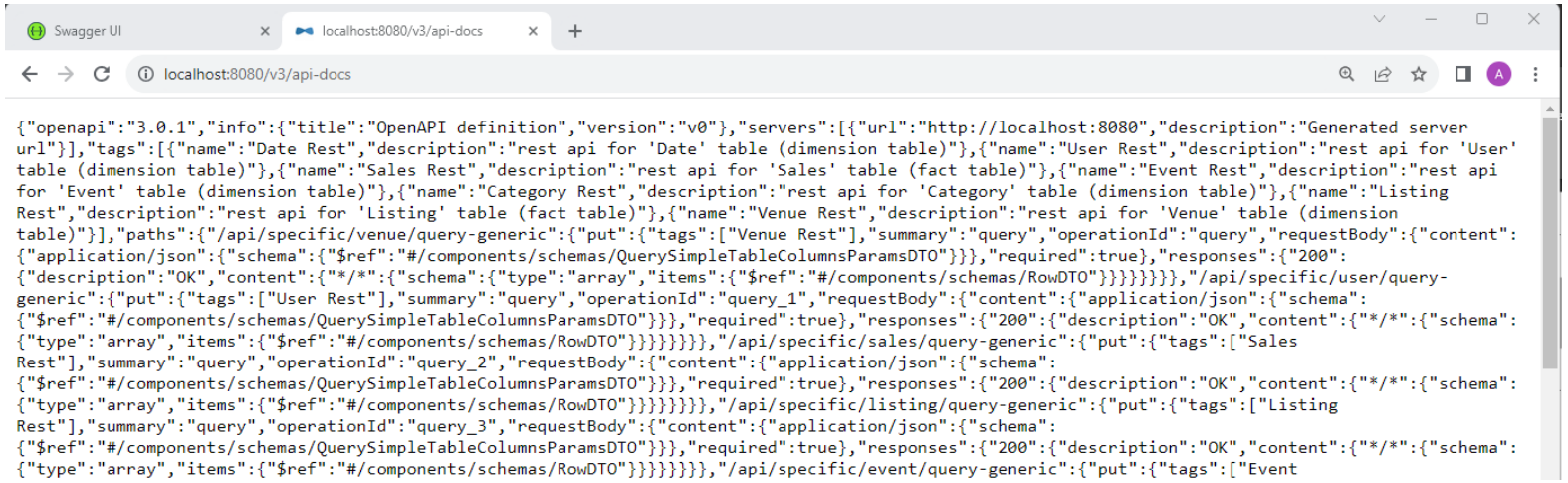
Short interactive browsing
run on your Windows Laptops
or Android/iPhone SmartPhones

On Chrome / FireFox / Edge /
in JavaScript
(because you have NO choice) !!

Reminder OpenApi - swagger-ui.html



Open Api description in json : /v3/api-docs



NOT to be read by Humans !!!!

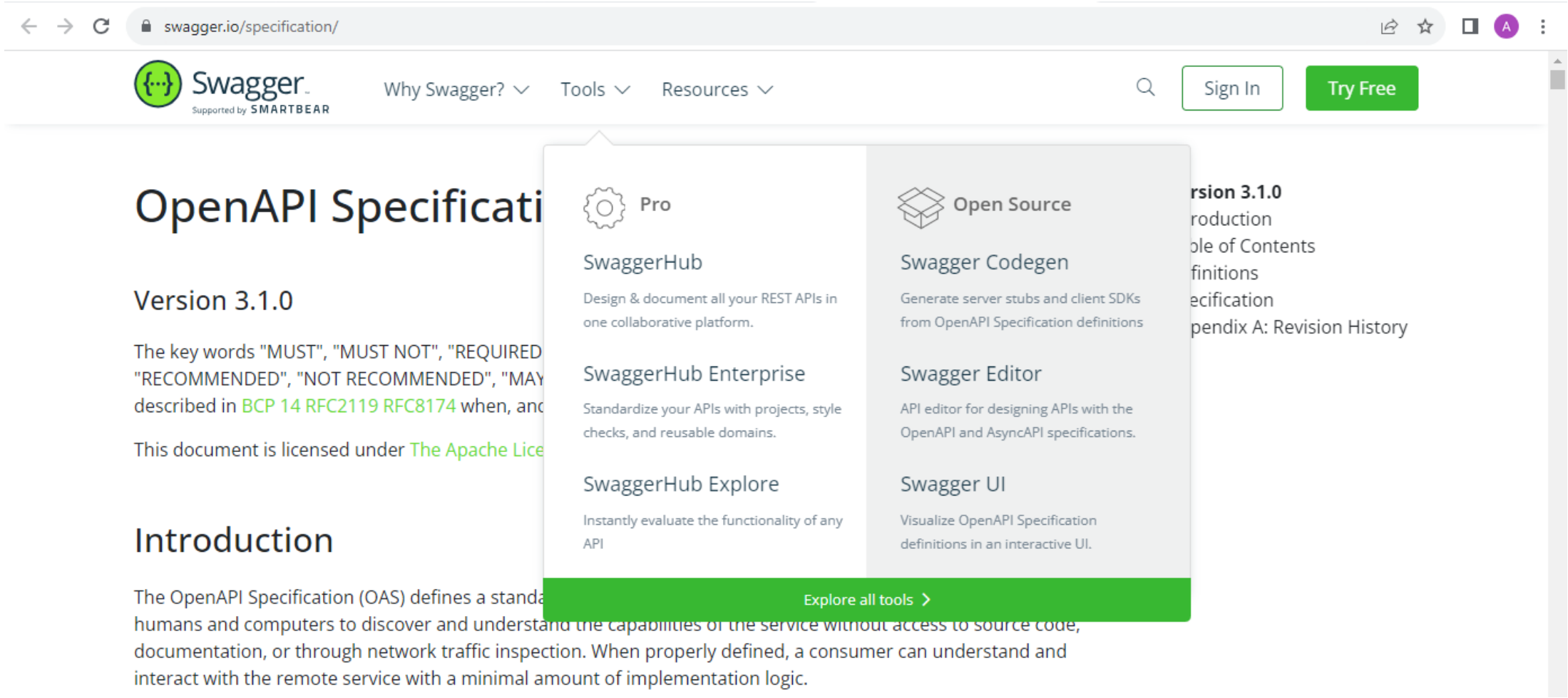
either

=> in swagger-ui.html

OR

=> for Client Code Generators

https://swagger.io
Swagger Codegen



Swagger CodeGen demo (maven plugin)

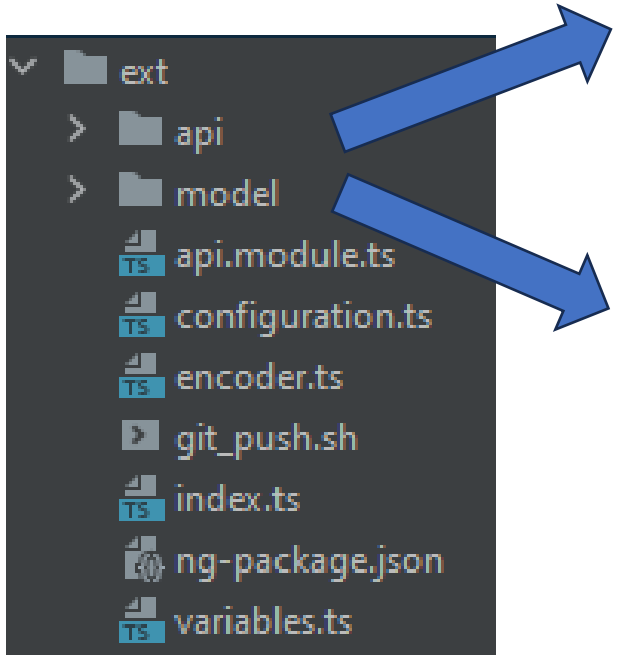
```xml
<plugin>
    <groupId>io.swagger.codegen.v3</groupId>
    <artifactId>swagger-codegen-maven-plugin</artifactId>
    <version>3.0.47</version>
    <configuration>
        <inputSpec>http://localhost:8080/v3/api-docs</inputSpec>
        <language>typescript-angular</language>
        <output>${basedir}/target/generated-typescript-angular16</output>
        <configOptions>
            <ngVersion>16</ngVersion>
        </configOptions>
    </configuration>
    <executions>
        <execution>
            <id>generate-swagger-typescript-angular-16</id>
            <phase>generate-sources</phase>
            <goals>
                <goal>generate</goal>
            </goals>
        </execution>
    </executions>
</plugin>
```

Swagger CodeGen Demo

$ mvn swagger-codegen:generate

```
$ rm -rf ../ui/src/app/ext/*; rm -rf target/generated-typescript-angular16; mvn -Pswagger-gen swagger-codegen:generate; cp -rf target/generat
ed-typescript-angular16/* ../ui/src/app/ext/
[INFO] Scanning for projects...
[INFO]
[INFO] -------------------< fr.an.test.spark-db-app:server >-------------------
[INFO] Building server 1.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- swagger-codegen-maven-plugin:3.0.47:generate (default-cli) @ server ---
[WARNING] Output directory does not exist, or is inaccessible. No file (.swagger-codegen-ignore) will be evaluated.
[INFO] writing file C:\arn\devPerso\test-spark-rest-server\server\target\generated-typescript-angular16\model\aliasedExprTextDTO.ts
[INFO] writing file C:\arn\devPerso\test-spark-rest-server\server\target\generated-typescript-angular16\model\arrayTypeDTO.ts
[INFO] writing file C:\arn\devPerso\test-spark-rest-server\server\target\generated-typescript-angular16\model\binaryOpExprDTO.ts
[INFO] writing file C:\arn\devPerso\test-spark-rest-server\server\target\generated-typescript-angular16\model\binaryTypeDTO.ts
[INFO] writing file C:\arn\devPerso\test-spark-rest-server\server\target\generated-typescript-angular16\model\booleanTypeDTO.ts
[INFO] writing file C:\arn\devPerso\test-spark-rest-server\server\target\generated-typescript-angular16\model\byteTypeDTO.ts
```

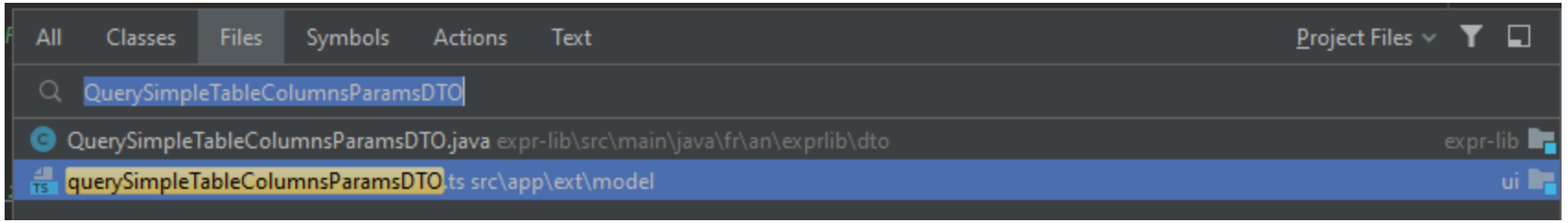
Generated code for Angular-Typescript



contains all Services for http endpoints
(= for @RestController in backend)
each PUT/POST/GET/DELETE
=> correspond to a typescript method

contains all interfaces describing datatypes
(= for class *DTO in backend)
each JSON request body / response type
=> correspond to a typescript interface

The front-end code is ALWAYS IN-SYNC with backend code!




```
/**
 * OpenAPI definition
 * No description provided (generated by Swagger Codegen https://github.com/swagger-api/swagger-codegen)
 *
 * OpenAPI spec version: v0
 *

 * NOTE: This class is auto generated by the swagger code generator program.
 * https://github.com/swagger-api/swagger-codegen.git
 * Do not edit the class manually.
 */
```

Example CodeGen

in Java (SpringBoot) backend → Generated TypeScript for Angular

Rest Controller class

```java
@RestController
@RequestMapping(path="/api/dispatch-table/{tableName}")
public class DispatchGenericQueryRestController {
```

→

```typescript
@Injectable()
export class DispatchGenericQueryRestControllerService {

    protected basePath : string  = 'http://localhost:8080';
    public defaultHeaders : HttpHeaders  = new HttpHeaders();
    public configuration : Configuration  = new Configuration();
```

http PUT method endpoint

```java
@PutMapping("/query-simple-cols")
@Operation(summary = "query simple columns")
public List<RowDTO> query(
        @PathVariable("tableName") String tableName,
        @RequestBody QuerySimpleTableColumnsParamsDTO req) {
```

→

```typescript
public query7(body: QuerySimpleTableColumnsParamsDTO, tableName: string,
              observe?: 'body', reportProgress?: boolean
              ): Observable<Array<RowDTO>>;
```

(cf next slide for detail implementation)

DTO type for request body

```java
@Data
public class QuerySimpleTableColumnsParamsDTO {
    public List<String> cols;
    public int limitCount;
}
```

→

```typescript
export interface QuerySimpleTableColumnsParamsDTO {
    cols?: Array<string>;
    limitCount?: number;
}
```

(glory details : Typescript HttpClient generated code)

```
public query7(body: QuerySimpleTableColumnsParamsDTO, tableName: string, observe?: 'body', reportProgress?: boolean): Observable<Array<RowDTO>>;
public query7(body: QuerySimpleTableColumnsParamsDTO, tableName: string, observe?: 'response', reportProgress?: boolean): Observable<HttpResponse<Array<RowDTO>>>;
public query7(body: QuerySimpleTableColumnsParamsDTO, tableName: string, observe?: 'events', reportProgress?: boolean): Observable<HttpEvent<Array<RowDTO>>>;
no usages
public query7(body: QuerySimpleTableColumnsParamsDTO, tableName: string, observe: any = 'body', reportProgress: boolean = false ): Observable<any> {
    if (body === null || body === undefined) {
        throw new Error( message: 'Required parameter body was null or undefined when calling query7.');
    }
    if (tableName === null || tableName === undefined) {
        throw new Error( message: 'Required parameter tableName was null or undefined when calling query7.');
    }
    let headers :HttpHeaders  = this.defaultHeaders;
    // to determine the Accept header
    let httpHeaderAccepts: string[] = [
        'application/json'
    ];
    const httpHeaderAcceptSelected: string | undefined = this.configuration.selectHeaderAccept(httpHeaderAccepts);
    if (httpHeaderAcceptSelected != undefined) {
        headers = headers.set('Accept', httpHeaderAcceptSelected);
    }
    // to determine the Content-Type header
    const consumes: string[] = [
        'application/json'
    ];
    const httpContentTypeSelected: string | undefined = this.configuration.selectHeaderContentType(consumes);
    if (httpContentTypeSelected != undefined) {
        headers = headers.set('Content-Type', httpContentTypeSelected);
    }
    return this.httpClient.request<Array<RowDTO>>( method: 'put', url: `${this.basePath}/api/dispatch-table/${encodeURIComponent(String(tableName))}/query-simple-cols`,
        options: {
            body: body,
            withCredentials: this.configuration.withCredentials,
            headers: headers,
            observe: observe,
            reportProgress: reportProgress
        }
    );
}
```

Outline

- Some Advanced features
  - AG-Grid
  - HighCharts
  - Generating (Angular) Client from Swagger
  - Code Scafolding
  - Security: Cookie, Login, OAuth2, JWT, …
  - RxJs, Auto completion, ng-prime
- Final Words

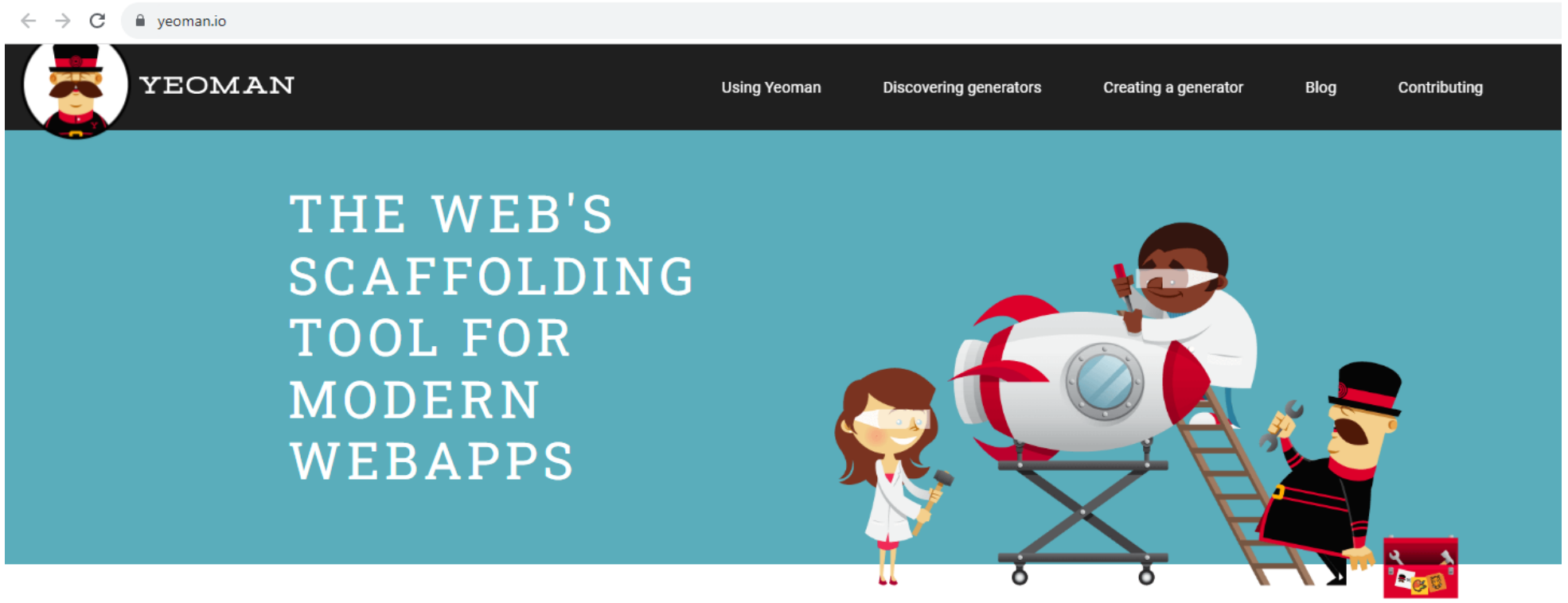
Code Scafolding

$ ng new

$ ng g c = $ ng generate component

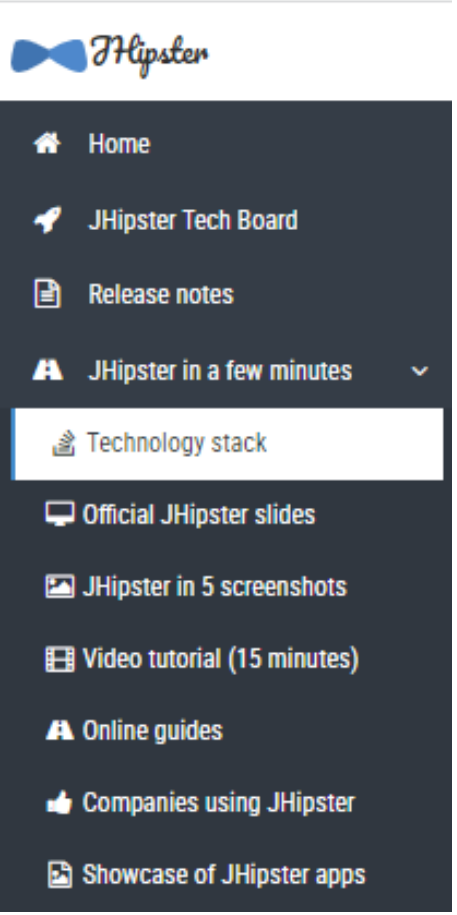
Your first scafolding tools !
a.k.a Code Generator

More Scafolding.. Yeoman https://yeoman.io



More Scafolding … JHipster https://jhipster.tech

JHipster in Few minutes > Technology Stack



Technology stack

Technology stack on the client side

Single Web page application:

- Angular or React or Vue
- Responsive Web Design with Twitter Bootstrap
- HTML5 Boilerplate
- Compatible with modern browsers (Chrome, FireFox, Microsoft Edge...)
- Full internationalization support
- Optional Sass support for CSS design
- Optional WebSocket support with Spring Websocket

With the great development workflow:

- Installation of new JavaScript libraries with NPM
- Build, optimization and live reload with Webpack
- Testing with Jest and Protractor

And what if a single Web page application isn't enough for your needs?

- Support for the Thymeleaf template engine, to generate Web pages on the server side

Technology stack on the server side

A complete Spring application:

- Spring Boot for application configuration
- Maven or Gradle configuration for building, testing and running th
- "development" and "production" profiles (both for Maven and Gra
- Spring Security
- Spring MVC REST + Jackson
- Optional WebSocket support with Spring Websocket
- Spring Data JPA + Bean Validation
- Database updates with Liquibase
- Elasticsearch support if you want to have search capabilities on t
- MongoDB and Couchbase support if you'd rather use a documer
- Cassandra support if you'd rather use a column-oriented NoSQL
- Kafka and Pulsar support if you want to use a publish-subscribe

JHipster Slides - Front End Technologies

https://www.jhipster.tech/presentation/#/

Client-side technologies

YEOMAN, WEBPACK, ANGULAR, REACT, BOOTSTRAP

NPM

- Fast and reliable dependency management
- Used to install and run all client-side tools

```
npm install -g generator-jhipster
```

Yeoman

YEOMAN PROVIDES APPLICATION *GENERATORS*

- Hundreds of generators are available
- Mostly geared toward JavaScript front-end applications
- The top-rated generators have excellent quality

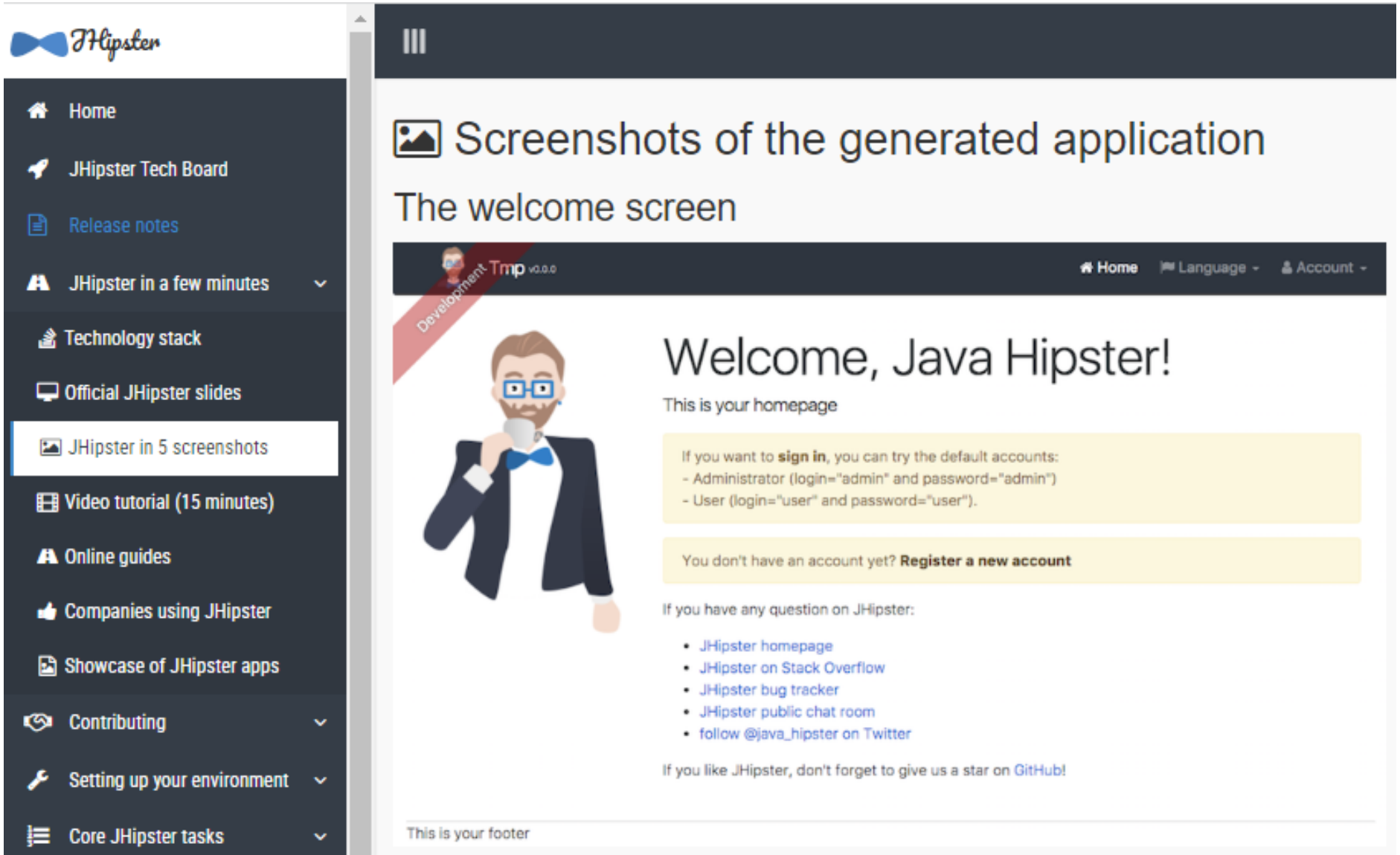
```
jhipster
```

Angular and React

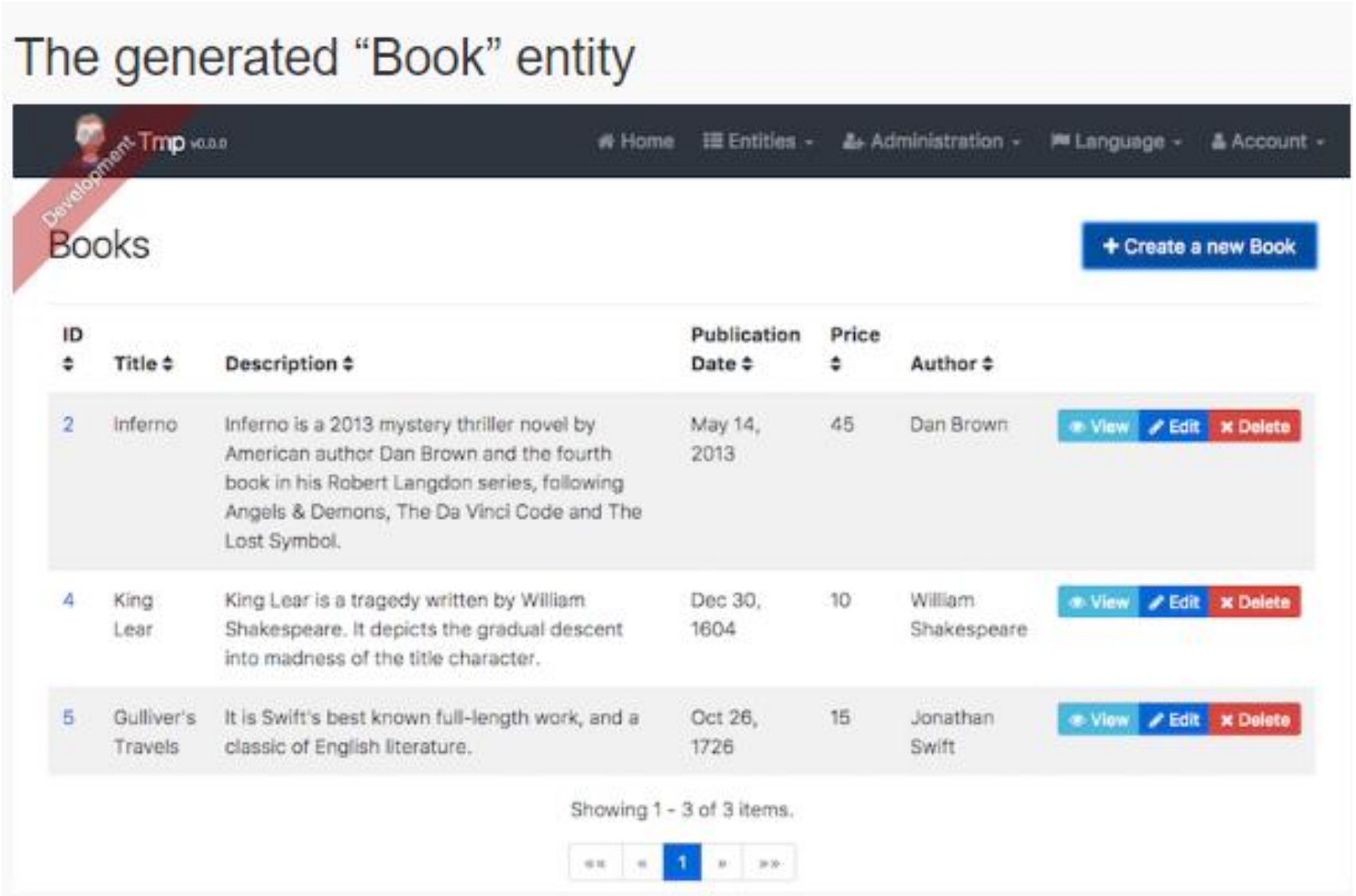
THE 2 MOST POPULAR JAVASCRIPT FRAMEWORKS

- Both Angular and React are supported by JHipster
- Powerful & easy to learn
- Data binding, form validation, i18n... all out of the box

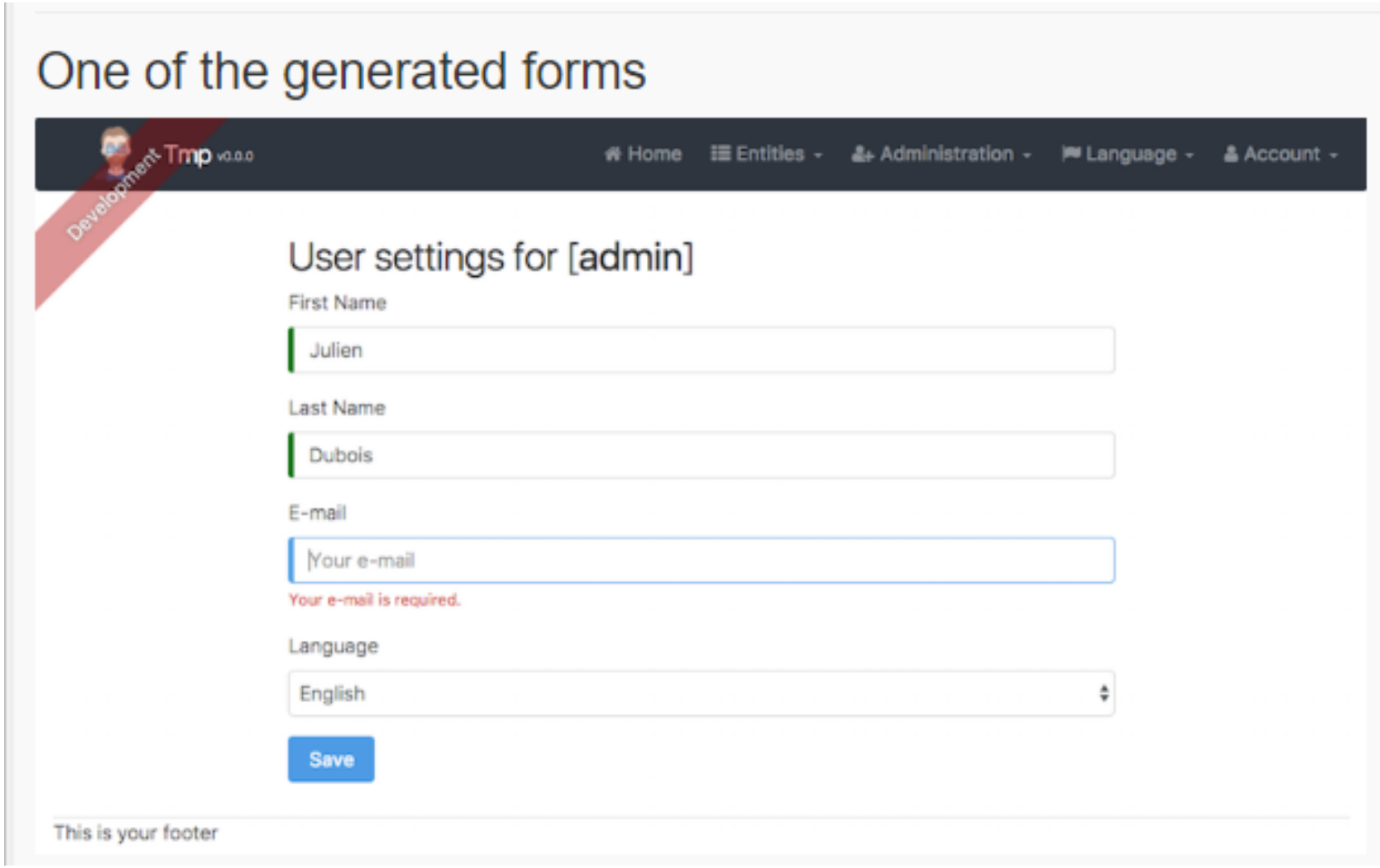
JHispter in 5 Screenshots: 1/5



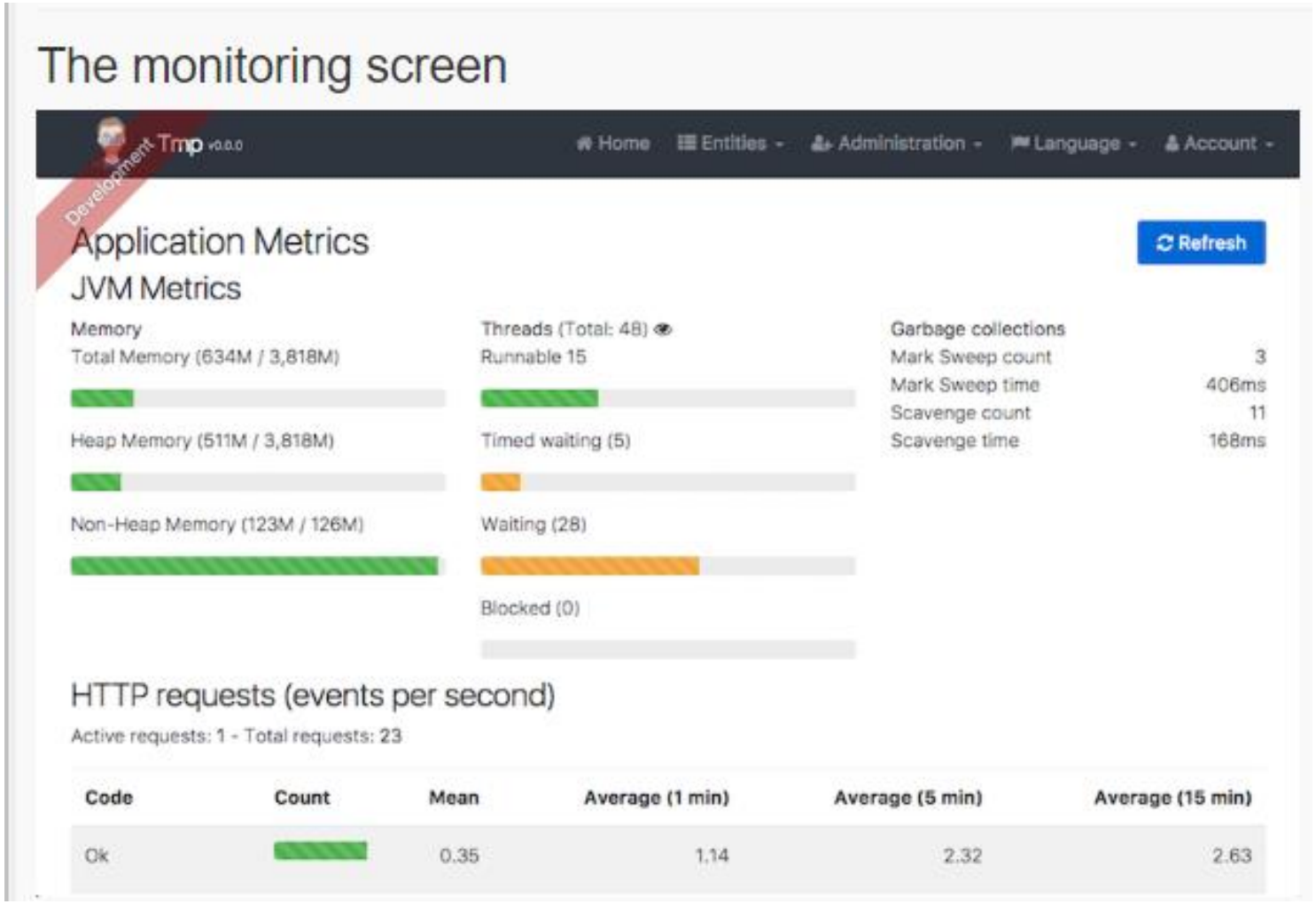
JHispter in 5 Screenshots: 2/5



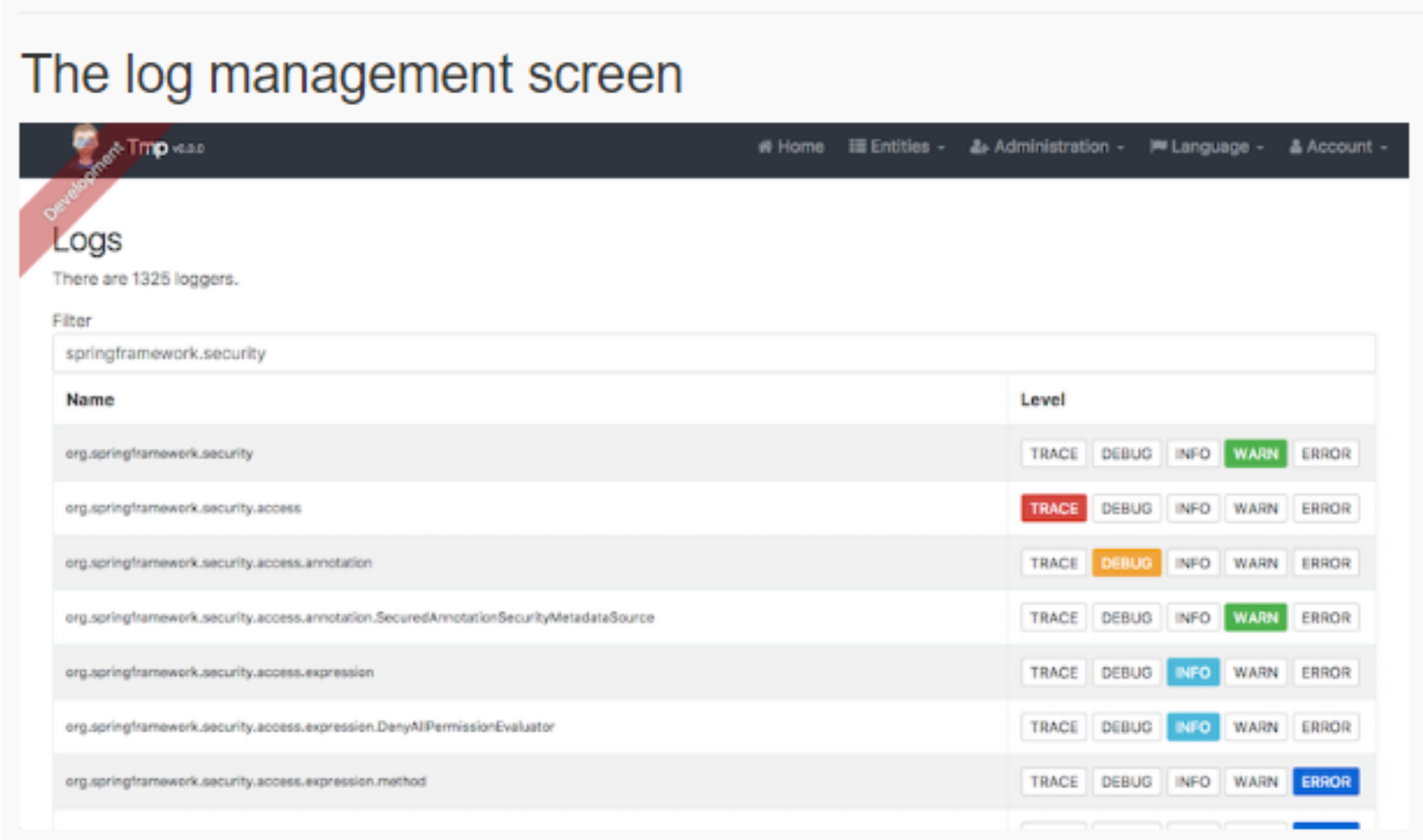
JHispter in 5 Screenshots: 3/5



JHispter in 5 Screenshots: 4/5



JHispter in 5 Screenshots: 5/5



JHipster Demo
(case 1/ with local installation)

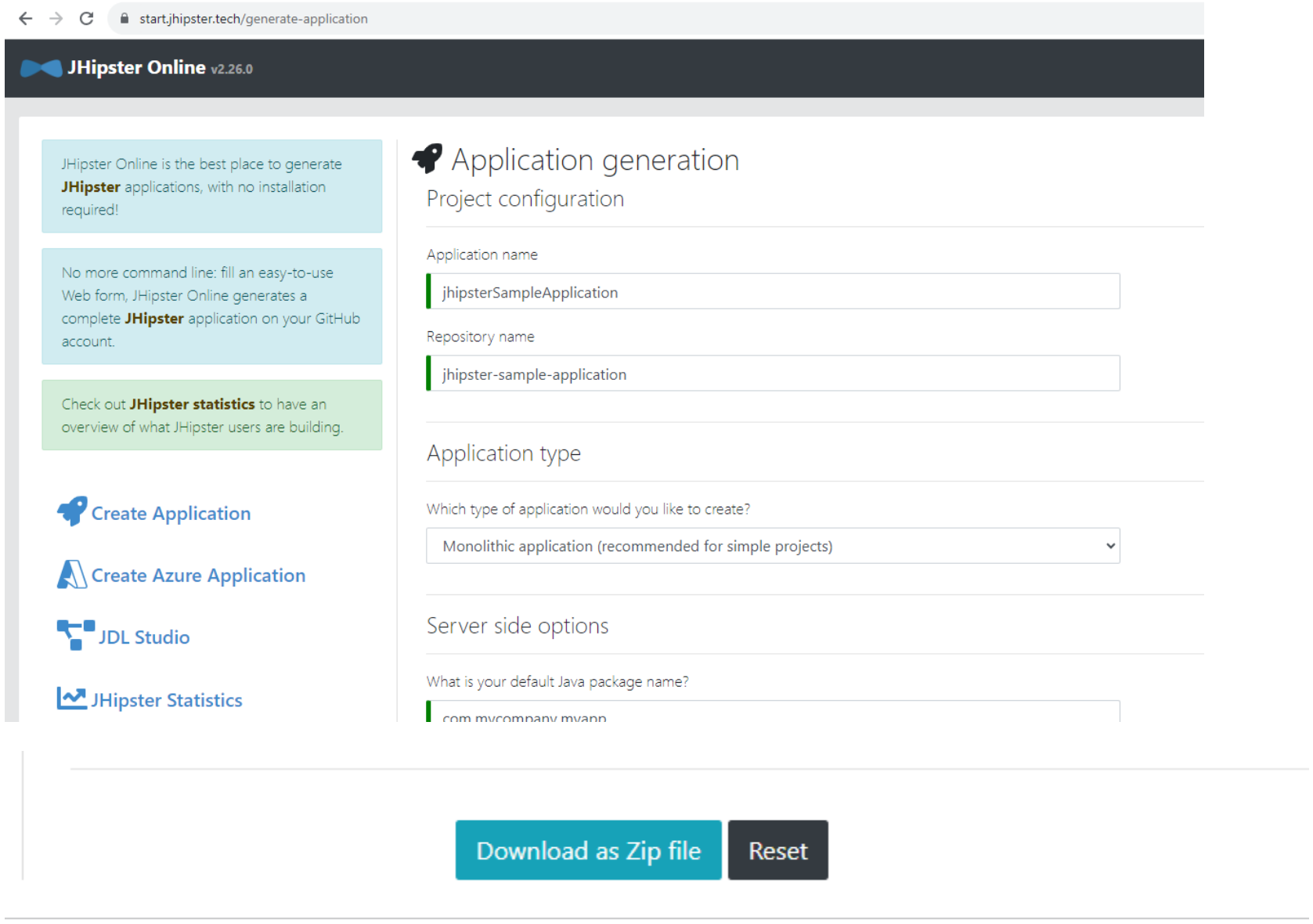
$ npm install -g generator-jhipster

$ jhipster
.... prompt....

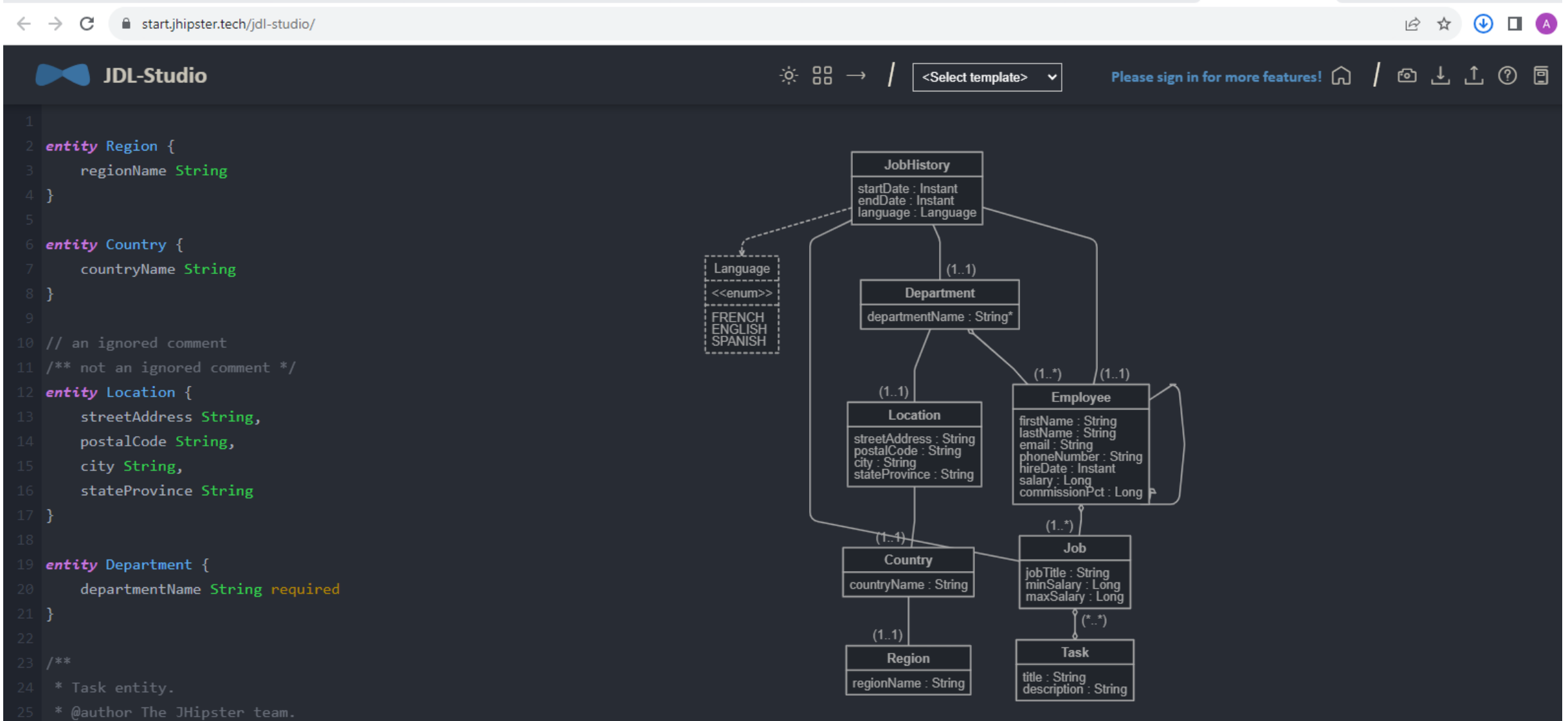
$ jhipster add ...
$ jhipster jdl <<your-file>>.jdl

case 2 / with JHipster Online

https://start.jhipster.tech/generate-application



https://start.jhipster.tech/jdl-studio



Step 1/ Compile backend
$ mvn package

```
$ mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] ----------< com.mycompany.myapp:jhipster-sample-application >----------
[INFO] Building Jhipster Sample Application 0.0.1-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO] The original artifact has been renamed to C:\web\demo-jhipster\jhipsterSam
.jar.original
[INFO]
[INFO] --- modernizer-maven-plugin:2.7.0:modernizer (modernizer) @ jhipster-sampl
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  6.574 s
[INFO] Finished at: 2023-12-02T18:08:06+01:00
[INFO] ------------------------------------------------------------------------
```

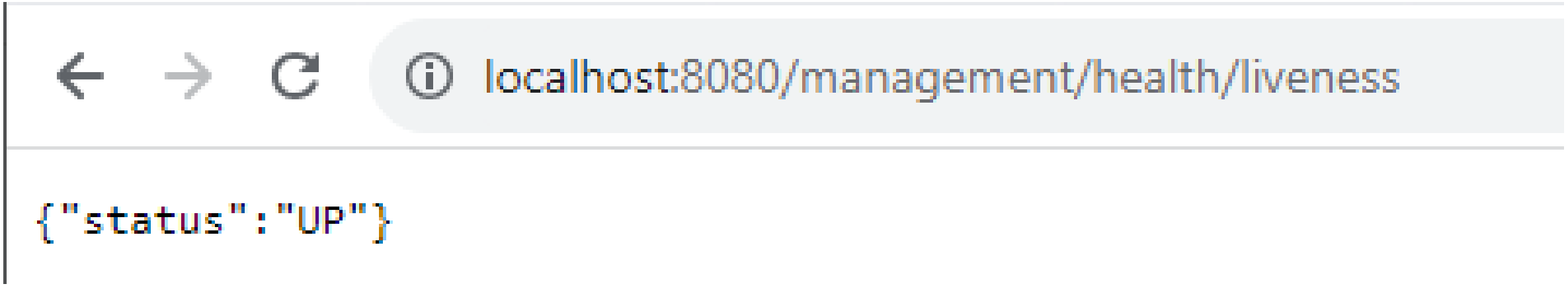
Step 2/ Run backend
$ mvn spring-boot:run -Pdev

```
$ mvn spring-boot:run -Pdev
[INFO] Scanning for projects...
[INFO]
[INFO] ----------< com.mycompany.myapp:jhipster-sample-application >----------
[INFO] Building Jhipster Sample Application 0.0.1-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] >>> spring-boot-maven-plugin:3.1.5:run (default-cli) > test-compile @ jhipster-sample-application >>>
```

```
2023-12-02T18:13:20.679+01:00  INFO 13436 --- [  restartedMain] c.m.myapp.JhipsterSampleApplicationApp   : Started
n 6.889 seconds (process running for 7.296)
2023-12-02T18:13:20.684+01:00  INFO 13436 --- [  restartedMain] c.m.myapp.JhipsterSampleApplicationApp   :
----------------------------------------------------------
        Application 'jhipsterSampleApplication' is running! Access URLs:
        Local:          http://localhost:8080/
        External:       http://192.168.0.10:8080/
        Profile(s):     [dev, api-docs]
----------------------------------------------------------
```

Step 3/ Test Backend
http://localhost:8080/management/health

```
$ curl  http://localhost:8080/management/health/liveness
{"status":"UP"}
```



Step 4/ compile FrontEnd
$ npm install
$ npm start

```
$ npm install
[                    ] \ idealTree:jhipsterSampleApplication: sill idealTree buildDeps
```

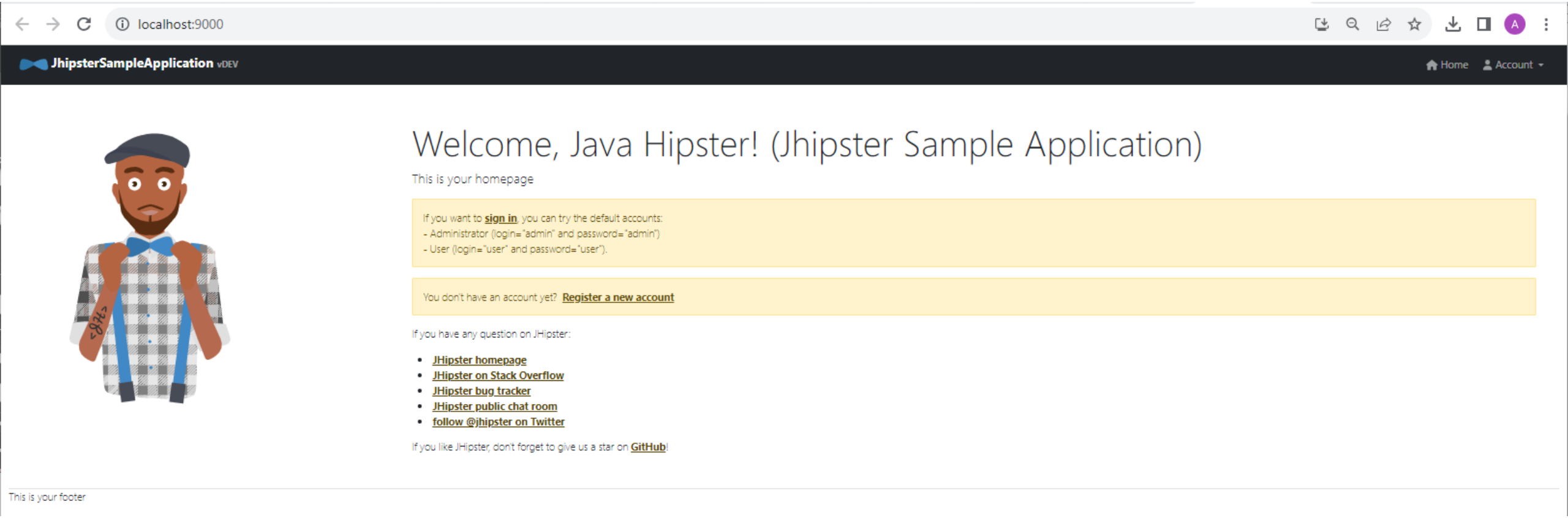
```
added 2162 packages, and audited 2163 packages in 4m

306 packages are looking for funding
  run `npm fund` for details

8 moderate severity vulnerabilities
```

Step 5/ Start the FrontEnd
$ npm start

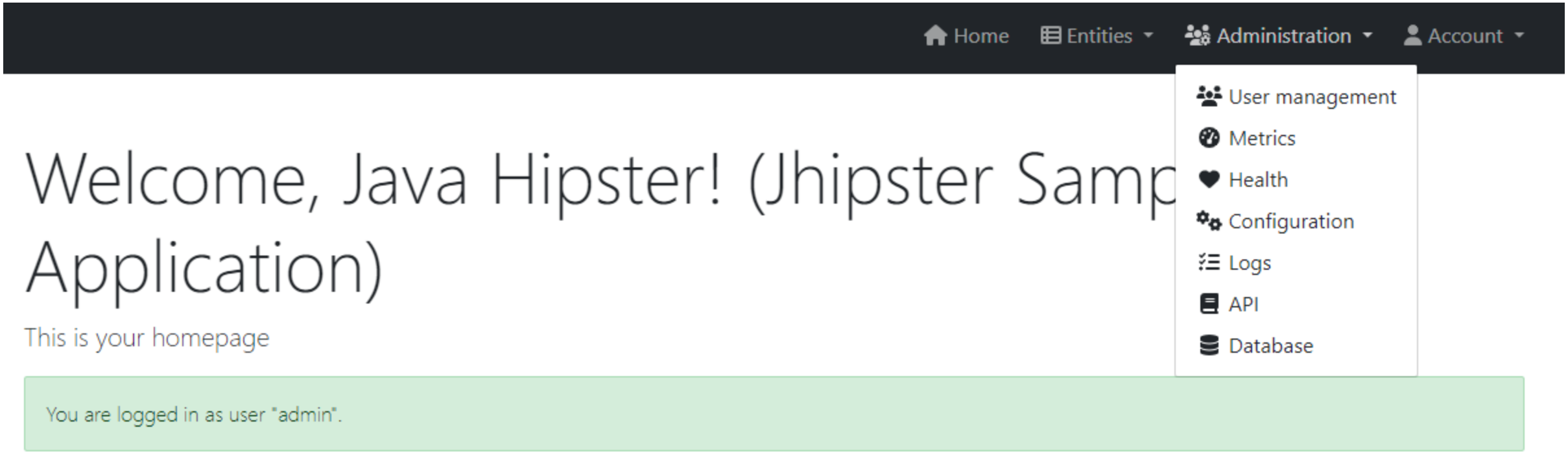
```
$ npm start

> jhipster-sample-application@0.0.1-SNAPSHOT start
> ng serve --hmr
```

```
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **


√ Compiled successfully.
Notifications are disabled
Reason: DisabledByGroupPolicy Please make sure that the app id is set correctly.
Command Line: C:\web\demo-jhipster\jhipsterSampleApplication\node_modules\node-notifier\vendor\snoreToast\snoretoa
otifierPipe-25256b7e-89fb-4008-9ccb-d78e197f65e4 -p C:\web\demo-jhipster\jhipsterSampleApplication\webpack\logo-jh
" -t "Jhipster Sample Application"
[Browsersync] Proxying: http://localhost:4200
[Browsersync] Access URLs:
 ------------------------------------
       Local: http://localhost:9000
    External: http://192.168.0.10:9000
 ------------------------------------
          UI: http://localhost:3001
 UI External: http://localhost:3001
 ------------------------------------
```

Step 6/ open Browser http://localhost:4200

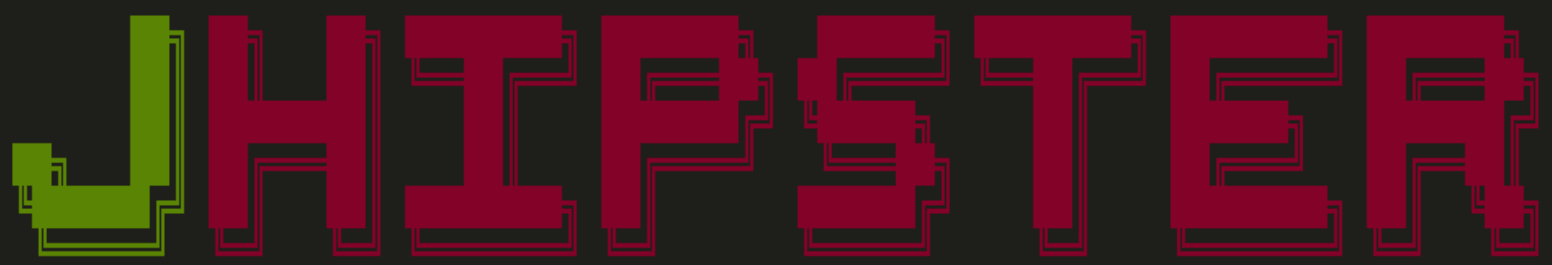


Step 7/ login as "admin/admin" or "user/user"



Step 8/ import jdl … recompile & restart
$ jhipster jdl ${jdlFile}

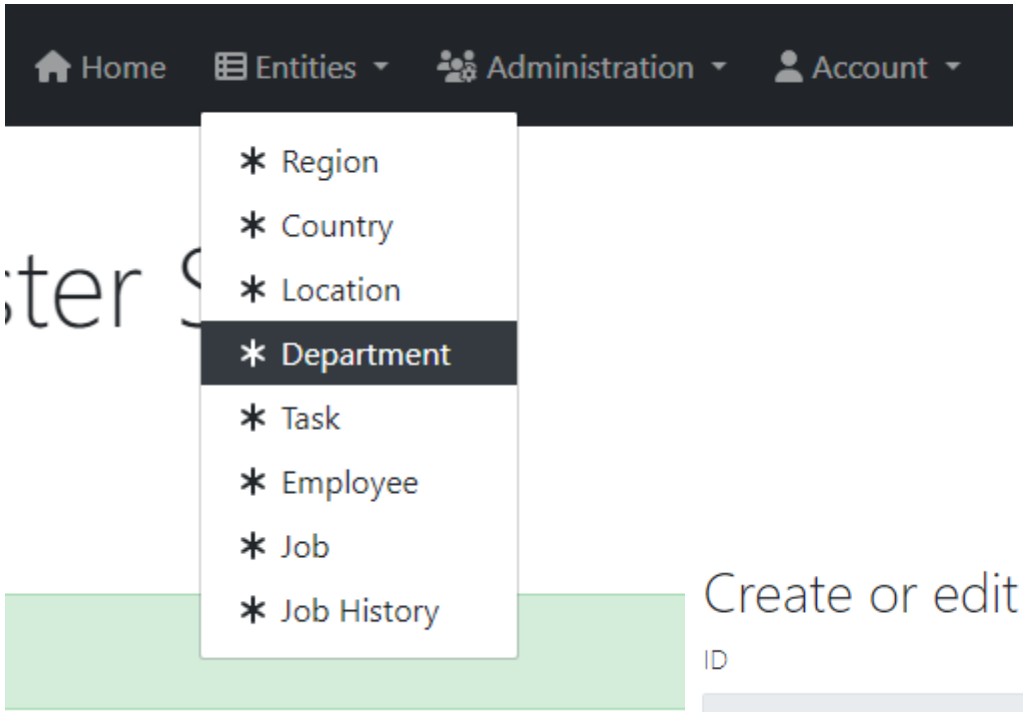
```
$ jhipster jdl ../jhipster-jdl.jdl
WARNING! Since JHipster v8, the jhipster command will not use the locally installed generator-jhipster.
    If you want to execute the locally installed generator-jhipster, run: npx jhipster
```

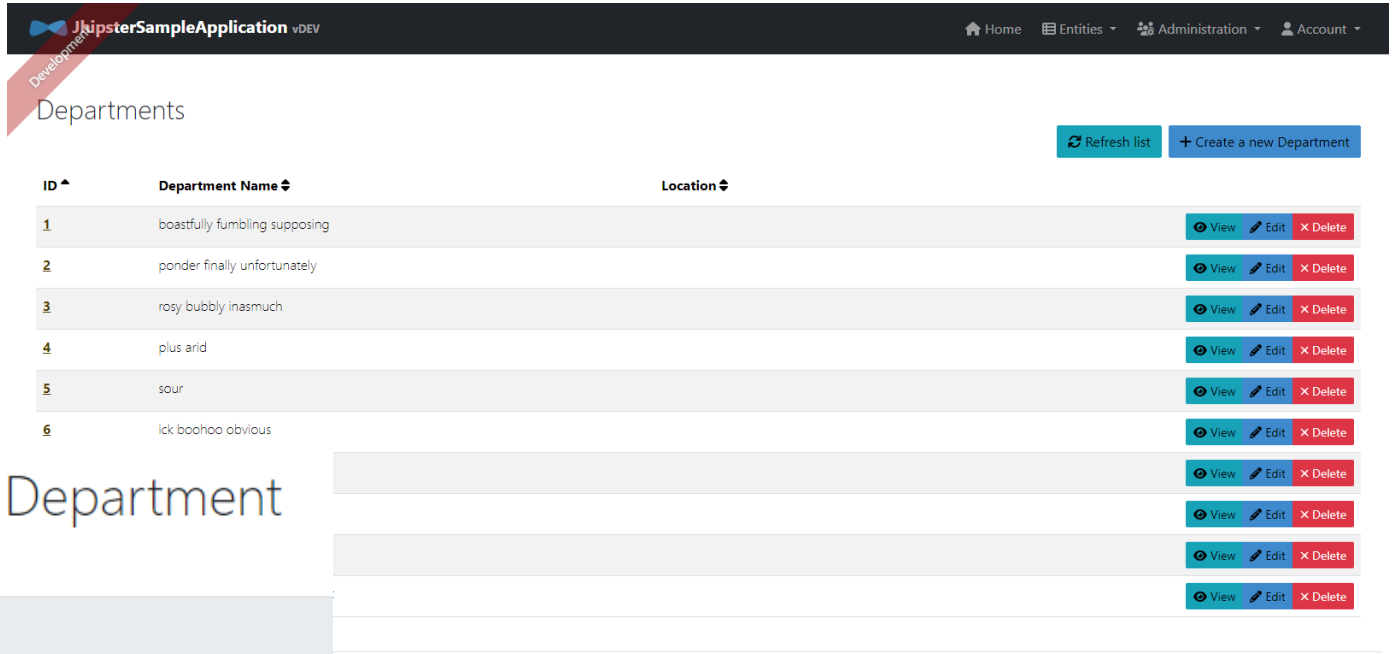


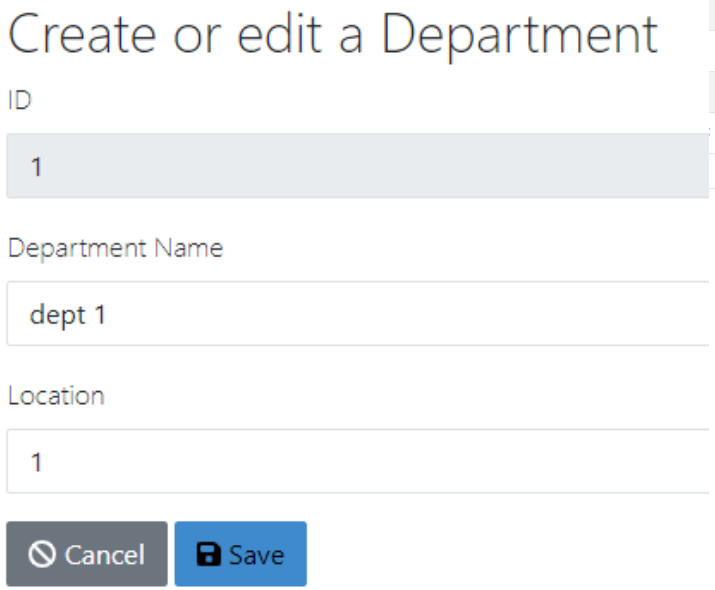

```
                    https://www.jhipster.tech
Welcome to JHipster v8.0.0

INFO! Generating jdls ../jhipster-jdl.jdl
INFO! Generating entities Region Country Location Department Task Employee Job JobHistory
```

```
create src\test\java\com\mycompany\myapp\web\rest\LocationResourceIT.java
create src\test\java\com\mycompany\myapp\domain\EmployeeTestSamples.java
create src\test\java\com\mycompany\myapp\domain\JobTest.java
create src\main\webapp\app\entities\enumerations\language.model.ts
create src\main\webapp\app\entities\region\service\region.service.ts
create src\main\webapp\app\entities\country\update\country-update.component.spec.ts
create src\main\webapp\app\entities\region\region.model.ts
create src\main\webapp\app\entities\country\detail\country-detail.component.ts
```

Step 9/ navigate Entities Table - View - EditForm







Outline

- Some Advanced features
  - AG-Grid
  - HighCharts
  - Code Scafolding
  - Generating (Angular) Client from Swagger
  - Security: Cookie, Login, OAuth2, JWT, …
  - RxJs, Auto completion, ng-prime
- Final Words

Security … should not be done half way

Leave it to Experts

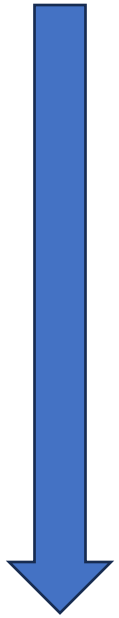
Security Basics

Encryption

Confidentiality
No Man-In-The-Middle

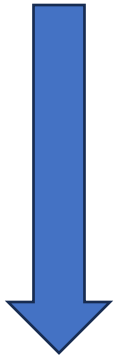
httpS://
certificate Trust

Authentication



user
password

Delegated
Authentication

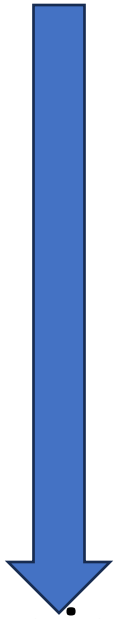


OAuth2
flows

Authenticated
Session

JWT
token
(Cookie)

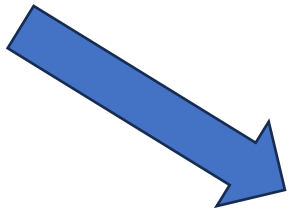
Authorization



relation in code
Resource
-> Permission
-> Group -> Users

Http Headers
"Authorization=Basic <user:pass>"
"Authorization=Bearer <token>"

$ curl -v **-u user:pass** http://localhost:8080



password in Base64 (~clear text) on Http Header

Demo "Basic <Base64(user:password)>"

```
$ curl -v -u user:user http://localhost:8080/
*   Trying [::1]:8080...
* Connected to localhost (::1) port 8080
* Server auth using Basic with user 'user'
> GET / HTTP/1.1
> Host: localhost:8080
> Authorization: Basic dXNlcjp1c2Vy
> User-Agent: curl/8.4.0
> Accept: */*
>
< HTTP/1.1 200 OK
```

```
$ echo dXNlcjp1c2Vy | base64 -d
user:user
```

Demo: Un-authenticated

```
$ curl -v http://localhost:8080/management/liveness
*   Trying [::1]:8080...
* Connected to localhost (::1) port 8080
> GET /management/liveness HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/8.4.0
> Accept: */*
>
< HTTP/1.1 401 Unauthorized
```

Demo ... 401
(no password, bad password, bad user, or unauthorized)

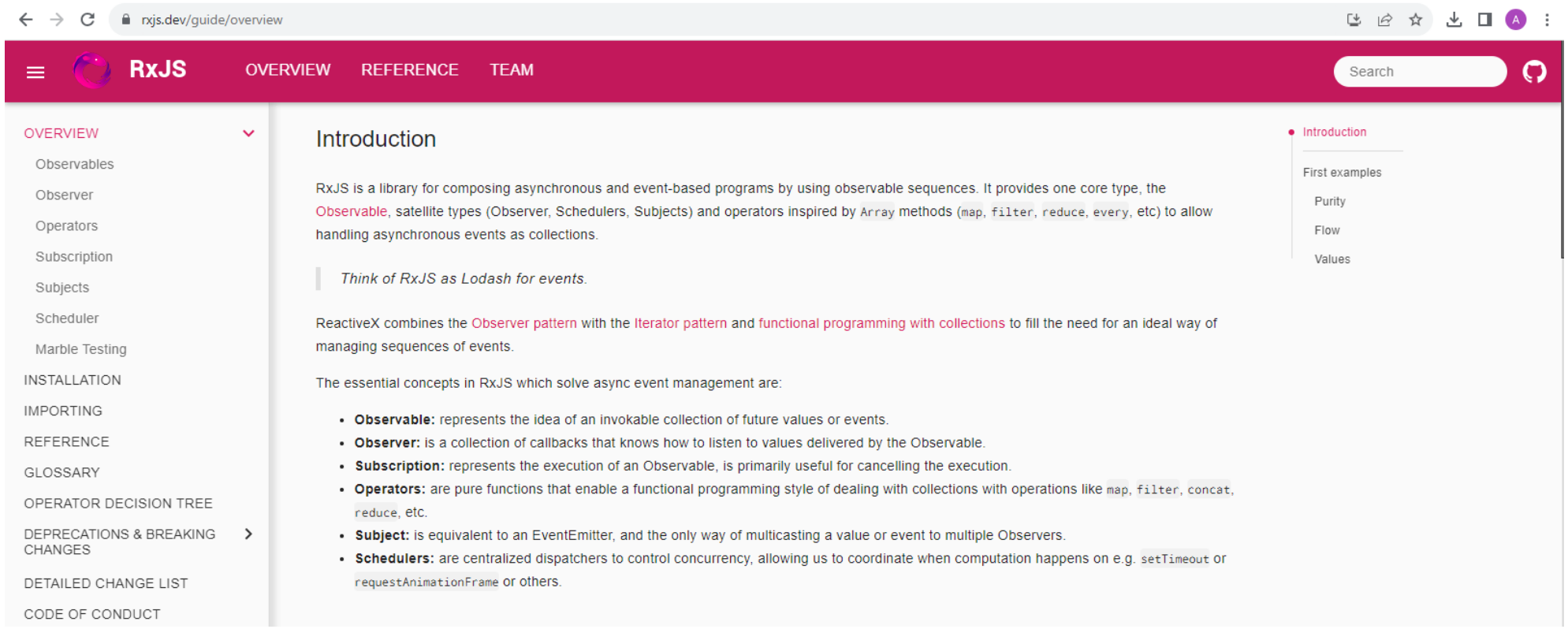
```
$ curl -v -u user:pass http://localhost:8080/management/liveness
*   Trying [::1]:8080...
* Connected to localhost (::1) port 8080
* Server auth using Basic with user 'user'
> GET /management/liveness HTTP/1.1
> Host: localhost:8080
> Authorization: Basic dXNlcjpwYXNz
> User-Agent: curl/8.4.0
> Accept: */*
>
< HTTP/1.1 401 Unauthorized
```

Demo of Login in Jhipster
(if time ok ?)

Outline

- Some Advanced features
  - AG-Grid
  - HighCharts
  - Code Scafolding
  - Generating (Angular) Client from Swagger
  - Security: Cookie, Login, OAuth2, JWT, …
  - RxJs, Auto completion, ng-prime
- Final Words

RxJs : Asynchronous programming with Observable & Subscribe, is ~~Complex~~ Powerful



Example RxJs Operator: map()

rxjs.dev/api/operators/map

RxJS OVERVIEW REFERENCE TEAM

- OVERVIEW
- INSTALLATION
- IMPORTING
- REFERENCE
- GLOSSARY
- OPERATOR DECISION TREE
- DEPRECATIONS & BREAKING CHANGES
- DETAILED CHANGE LIST
- CODE OF CONDUCT

stable (v7.8.1)

API / rxjs/operators

map

FUNCTION STABLE OPERATOR

Applies a given `project` function to each value emitted by the source Observable, and emits the resulting values as an Observable.

```
map<T, R>(project: (value: T, index: number) => R, thisArg?: any): OperatorFunction<T, R>
```

Parameters

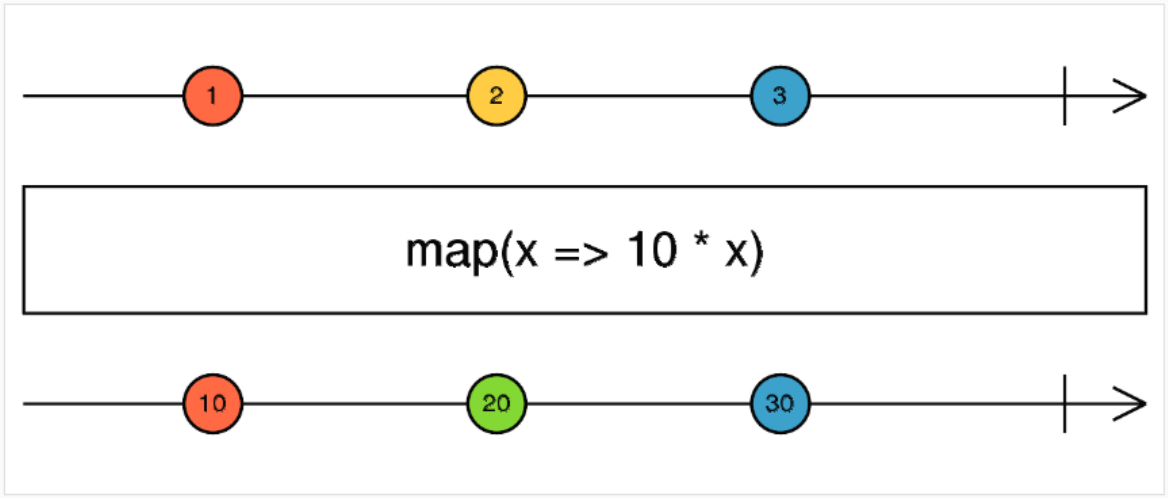
`project`	`(value: T, index: number) => R`	The function to apply to each `value` emitted by the source Observable. The `index` parameter is the number `i` for the i-th emission that has happened since the subscription, starting from the number `0`.
`thisArg`	`any`	Optional. Default is `undefined`. An optional argument to define what `this` is in the `project` function.

Returns

`OperatorFunction<T, R>`: A function that returns an Observable that emits the values from the source Observable transformed by the given `project` function.

Description

Like Array.prototype.map(), it passes each source value through a transformation function to get corresponding output values.



map

Description

Example

See Also

Sample usage:

"unchecked JSON" DTO interface
-map->
type-safe "Model" class

Some Object-Oriented Reasons to re-define class on client-side for DTO interface

```ts
export interface RowDTO {
  cols?: Array<any>;
}
```

```ts
export class RowModel {
  private readonly cols: any[]; // immutable
  1 usage
  public constructor(src: RowDTO) {
    this.cols = src?.cols || []; // sanity check input
  }

  no usages
  col(idx: number): string { return this.cols[idx]; }
  no usages
  strCol(idx: number): string { return <string> this.cols[idx]; }
}
```

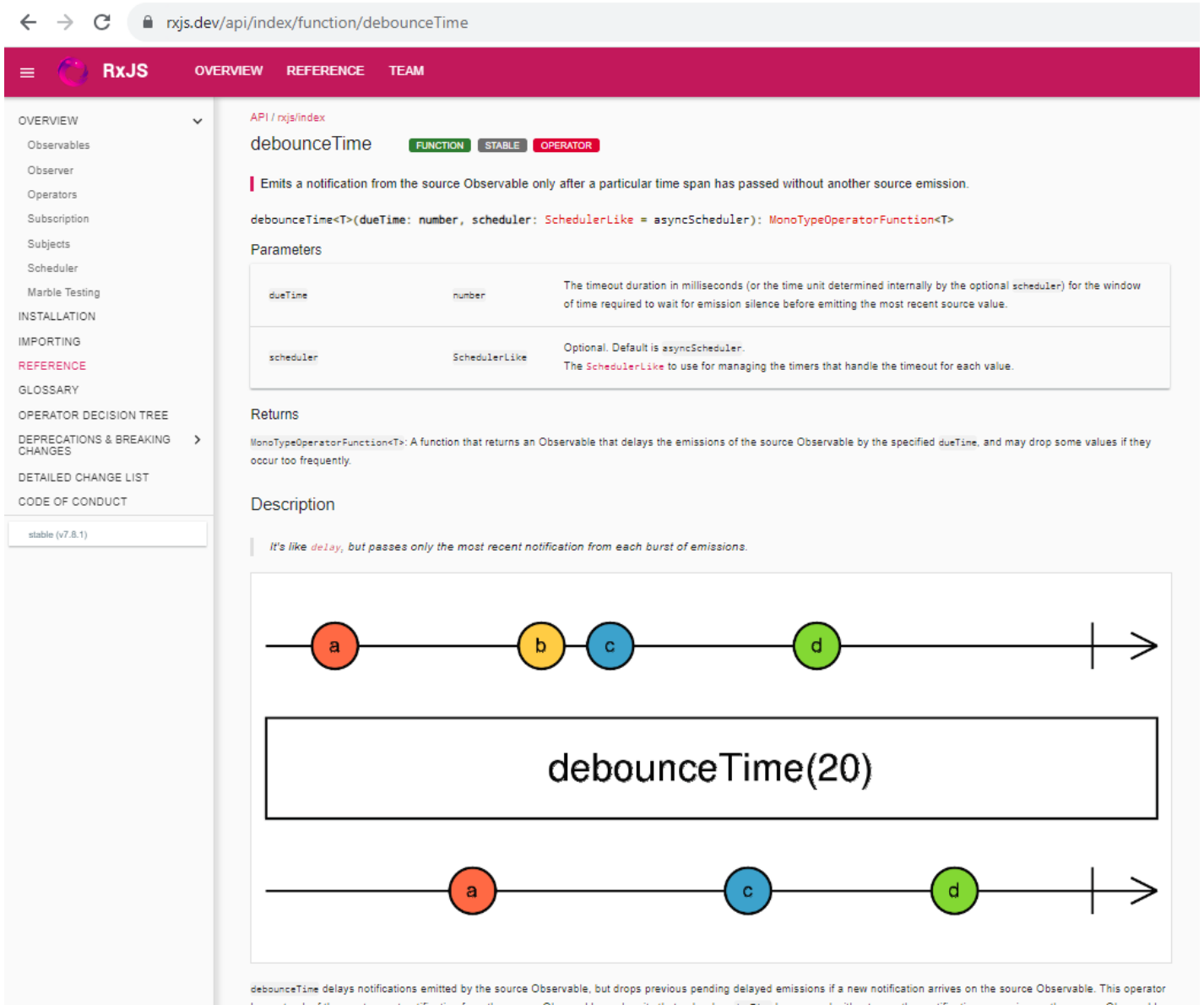
```ts
const json: RowDTO[] = [
  { cols: [ 1, 'test' ]},
  { cols: [ 2, 'test' ]}
];
```

```ts
const model: RowModel[] = [
  new RowModel(json[0]),
  new RowModel(json[1]),
];
const model2 :RowModel[]  = json.map(x :RowDTO  => new RowModel(x));
```

Calling RxJs: .pipe(map(x => tranform(x)))

```
query(req: QuerySimpleTableColumnsParamsDTO, tableName: string): Observable<RowModel[]> {
  return this.restService.query7(req, tableName).pipe(
    map( project: (dto: RowDTO[]) => dto!.map(element : RowDTO  => new RowModel(element)))
  );
}
```

Other Example RxJs operator: debounceTime(millis)



example debounceTime() ?

Doing Auto-completion -> trigger http Call(s) but NOT for each char typed

```
yourAngularField$ : Observable<string> =  ....

yourAngularField$
    .pipe(debounce(300))
    // ... more rxjs operators needed (cf next slide)
    .subscribe(x => {
        // handle field change, example for auto-completion
    });
```

... cf Also .distinctUntilChanged() and switchMap() !!

https://stackblitz.com/edit/autocomplete-suggestions-seach-with-debouncing?file=src%2Fapp%2Fapp.component.ts

```
this.bookForm.controls["searchtext"].valueChanges
  .pipe(
    debounceTime(1000),
    distinctUntilChanged((curr, prev) => {
      return curr.toLowerCase() === prev.toLowerCase();
    }),
    switchMap(text => {
      this.searchstring.push(text);
      if (
        this.searchstring[this.searchstring.length - 1] !==
        this.searchstring[this.searchstring.length - 2]
      ) {
        return this.bookservice.getBookBySearchText(text);
      } else {
        return [];
      }
    })
  )
  .subscribe(res => {
    this.books = res;
  });
```

Outline

- Some Advanced features
  - AG-Grid
  - HighCharts
  - Code Scafolding
  - Generating (Angular) Client from Swagger
  - Security: Cookie, Login, OAuth2, JWT, …
  - RxJs, Auto completion, ng-prime
- Final Words

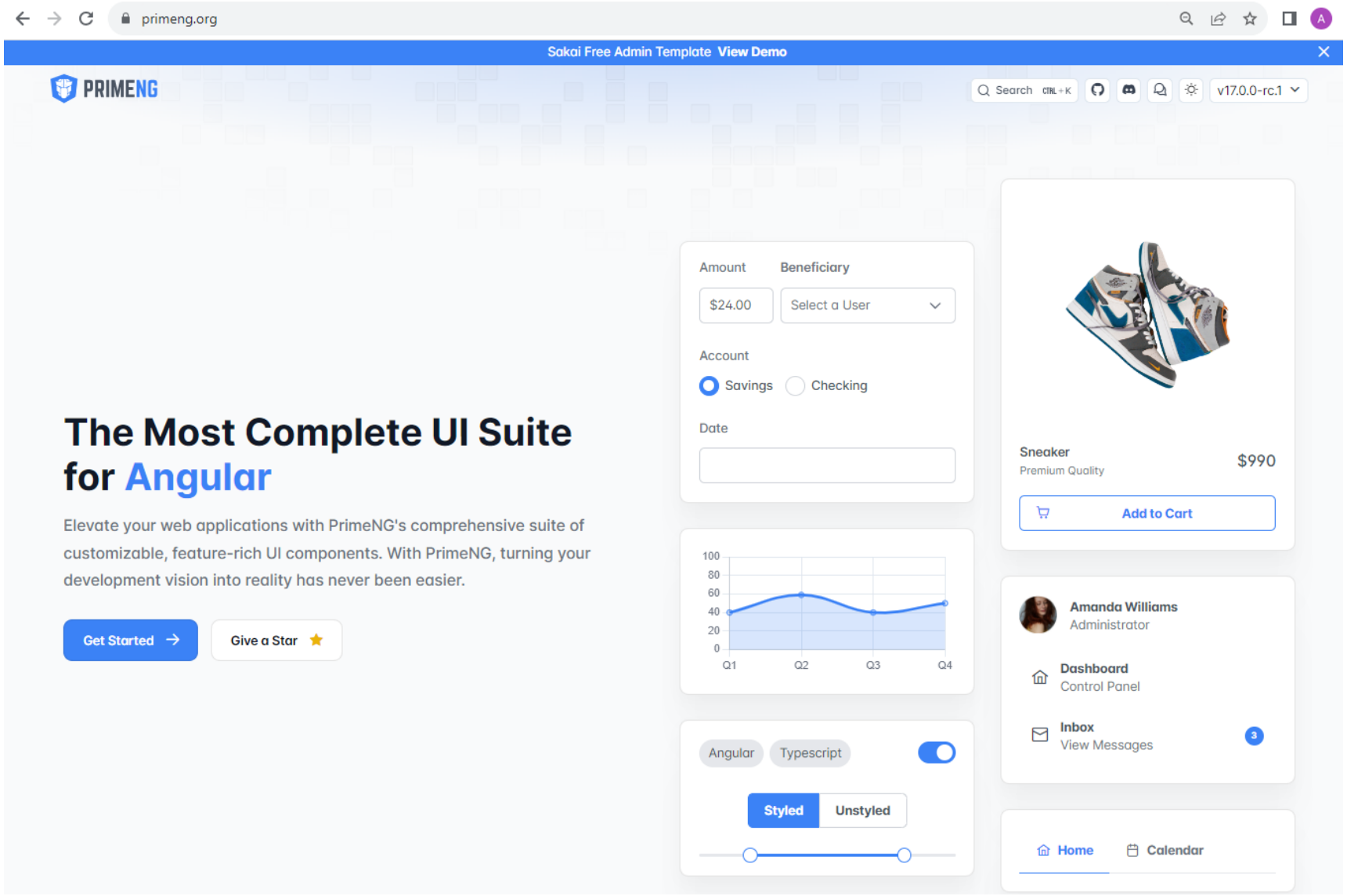
NG-Prime

Doing Rich web components is Difficult (and not funny)

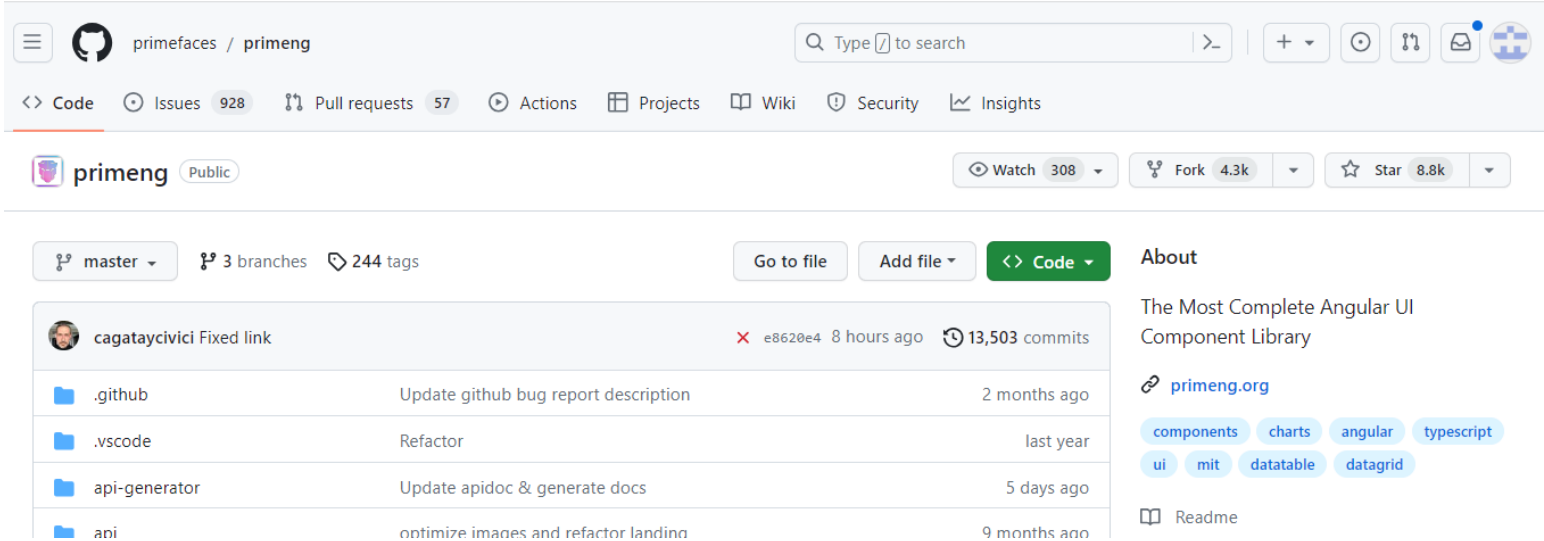
Leave it to Experts !!

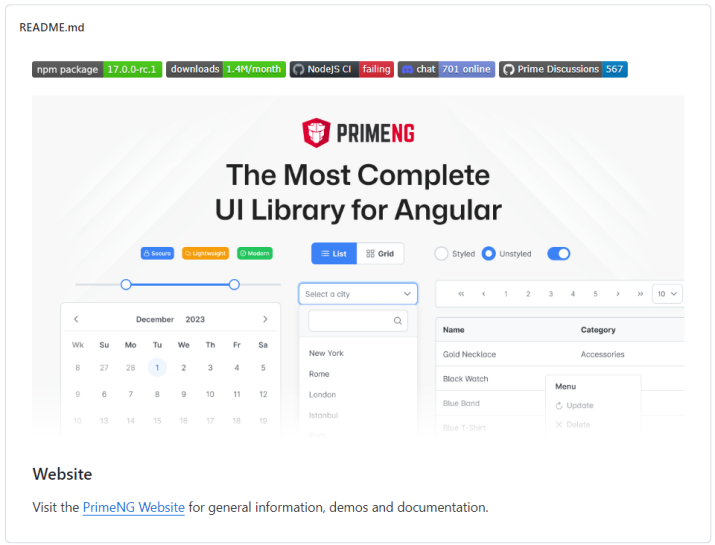
Re-use code, do not re-invent the Wheel

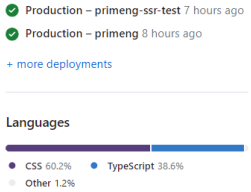
https://primeng.org



https://github.com/primefaces/primeng







(alternatives to) Prime-NG

ng-bootstrap components (but very low level, not so rich behaviors)

tons of npm libraries ... but you have a patchwork of CSS styles

Google Material Design WebComponent library

your own company already has some super cool toolkit ?

Outline

- Some Advanced features
  - AG-Grid
  - HighCharts
  - Code Scafolding
  - Generating (Angular) Client from Swagger
  - Security: Cookie, Login, OAuth2, JWT, …
  - RxJs, Auto completion, ng-prime

- Final Words

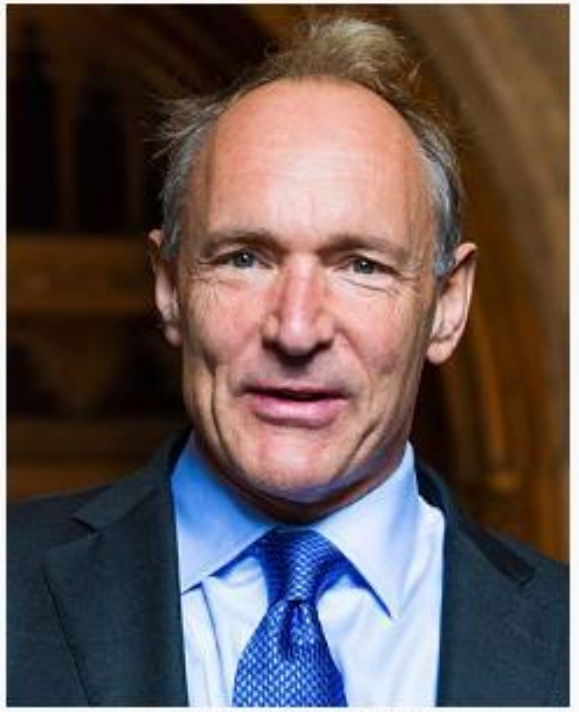
Final Words

Thank you Tim

Sir

Tim Berners-Lee

OM KBE FRS FREng FRSA DFBCS RDI



Berners-Lee in 2014

Born	Timothy John Berners-Lee 8 June 1955 (age 68) London, England
Other names	TimBL TBL
Education	The Queen's College, Oxford (BA)
Known for	Invention of the World Wide Web

Render to Cesar what belongs to Cesar

Brendan Eich (/ˈaɪk/; born July 4, 1961)[1] is an American computer programmer and technology executive. He created the JavaScript programming language and co-founded the Mozilla project, the Mozilla

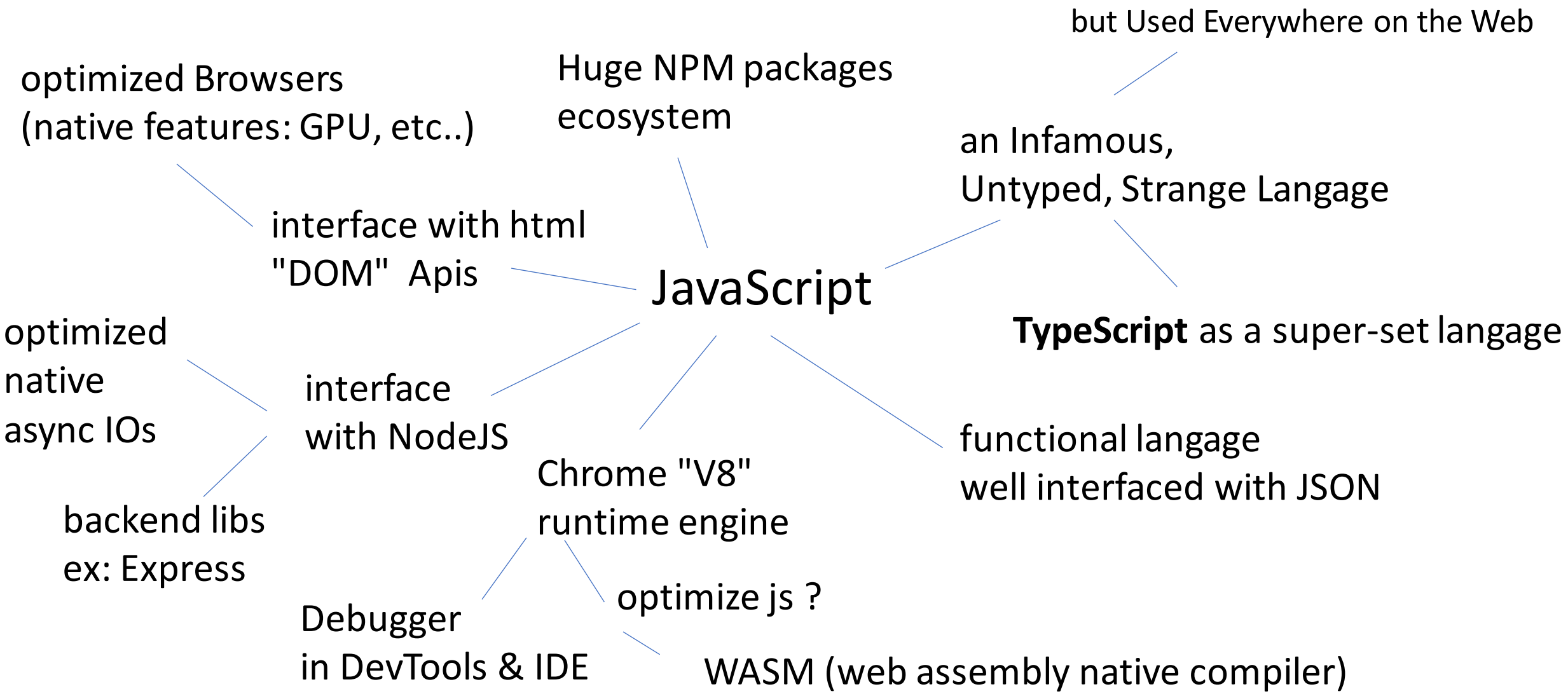
Brendan Eich

Eich in 2012

Born	4 July 1961 (age 62) Pittsburgh, Pennsylvania, US
Alma mater	University of Illinois Urbana-Champaign Santa Clara University
Website	brendaneich.com

Web Fundamental

JavaScript MindMap



TypeScript - JavaScript Schizofrenia Troller

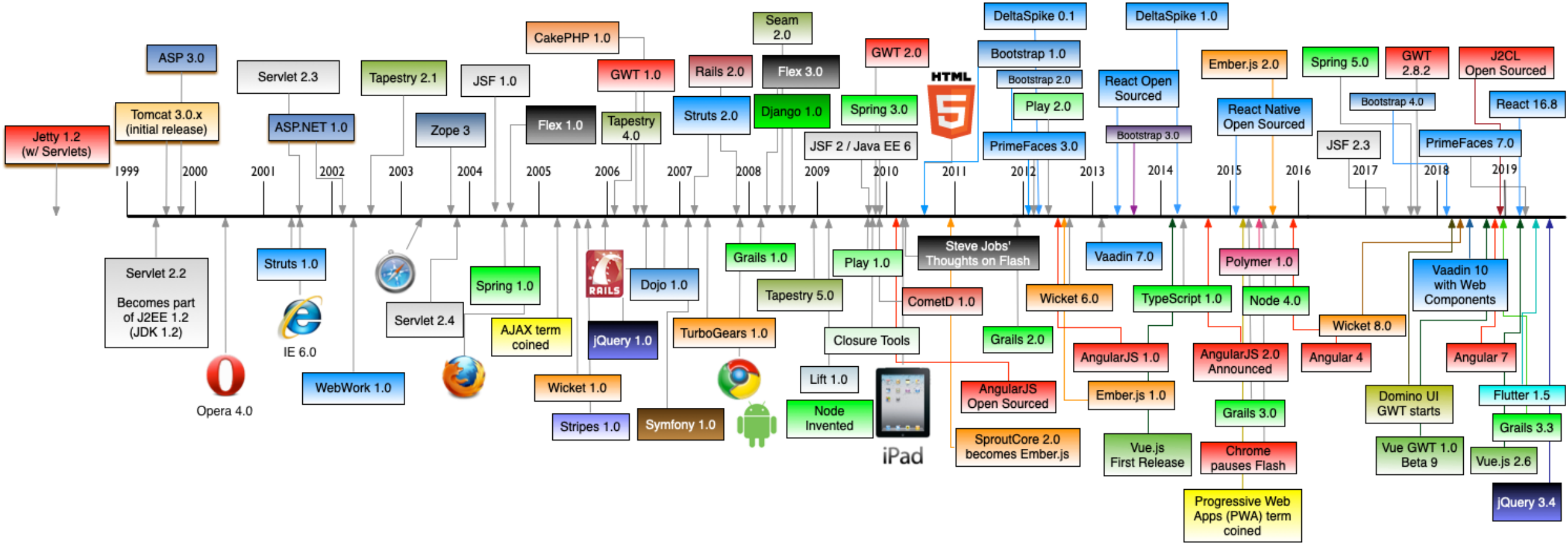
Backend - Frontend
Schizofrenia Troller

Back-end Developper

Front-end Developper

Full-Stack Developper

Web Frameworks … 1999 to 2020



Web Frameworks - 2023 and Future ?

Conclusion

Frameworks change every year

The Web will live long : Html + Javascript + Css

Questions

arnaud.nauwynck@gmail.com