Command line command:

Curl localhost:8080/api/games/risk

To use angular material:

Ng add @angular material

Typography: Y

Animation: Y

Notes:

Controller can be use in place of RestController but have to declare a @Requestbody

Formatting:

%%%s%

PrimeNg , ng-zorro = use mainly for enterprise applications

Angular material = mobile application

Create material module to store all angular materials modules to prevent clutter for import and export

Subsequent creation of component neeed to tell which module to add to:

ng g c search --module app.module

Deploying angular application:

ng serve is use for development,

mvn spring-boot:run for development,

ng build is use for deployment (production) this is what vercel does

ng build converts typescript(angular) into javascript(browser) + angular package and put into dist folder called client, need a webserver to run this. Install npm i -g http-server. Go into the dist directory and key http-server. Starts on http port 8081 as port 8080 is taken by backend. Can copy out the client folder and host it in two places: 1. To vercel or to own hosted server where backend is on another server (Cross Origin). 2. Take the angular file and put it into static folder. Then do mvn package in springboot

mvn package for spingboot (generates jar file snapshot), to run Java jar <<snapshot>> and put into target

Same origin deployment

1. Compile angular (ng build)
2. Copy dist/<appname> folder to the static directory of backend springboot (src/resources/static), ruby, express
3. Compile spring-boot (mvn package)

Cross origin deployment

1. Backend hosted in a server with spring-boot
2. Frontend hosted in a separate web server
3. Load angular into user (GET/index.html) from web server
4. Make HTTP request to backend server for access to database/api
5. Localhost 4200 is angular development server, it is get cross origin error because this setup cause it to be automatically cross origin even without a web server
   1. To solve, need to add in a request header to allow cross origin in spring-boot
   2. Enable CORS header globally
      1. Adding a bean @configuration @Bean public WebMVcConfigurer to configure it to add cors header with any links that starts with /api/\* in AppConfig.java

Ways of passing data from components:

1. Traditional event and property binding
2. @ViewChild
   1. Implement afterview init
   2. Subscription
   3. Unscubscribed
   4. Need to account for change detection
3. Using service to pass data
   1. Create subject in service