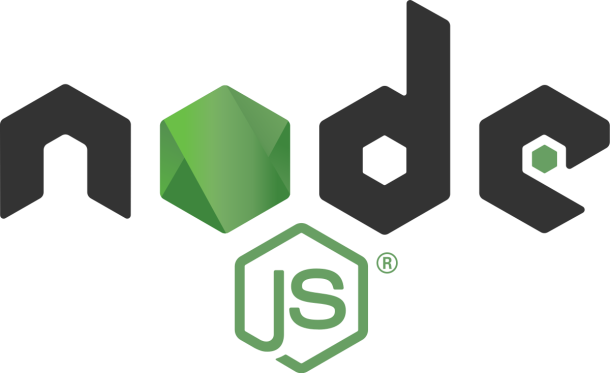
Quizz

 Tehnologii folosite :

* Spring Boot 2.2.6
* Angular 8
* Node JS 12.16.2





***Spring Boot***

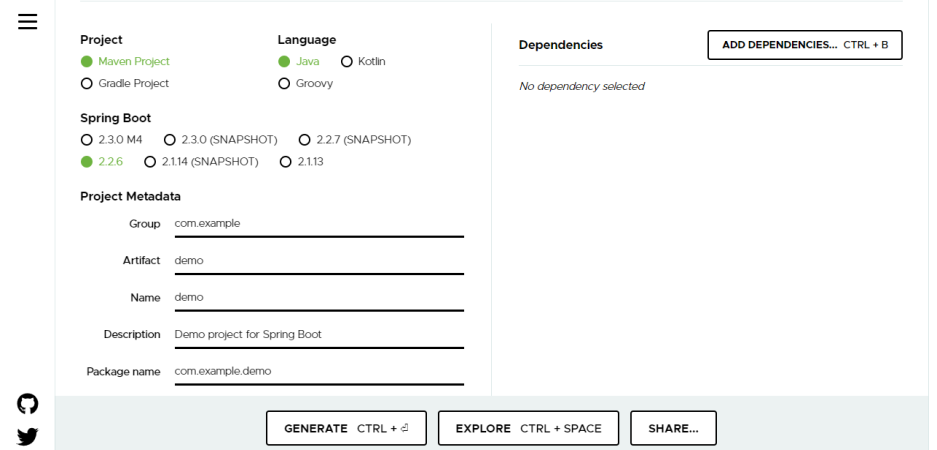
Spring Boot face foarte usoara crearea de aplicatii stand-alone pe baza framework-ului Spring pe care poti sa le pornesti instant.

Acesta simplifica din start foarte mult utilizarea librariilor provenite din alte surse precum si a diverselor configuratii astfel incat sa putem incepe doar cu minimul necesar.

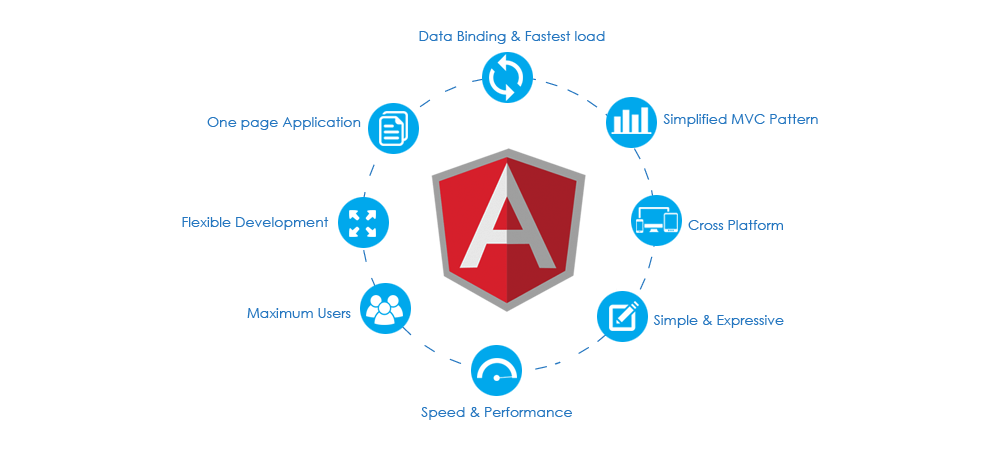
* Spring Framework este o platformă cu [sursă deschisă](https://ro.wikipedia.org/wiki/Surs%C4%83_deschis%C4%83" \o "Sursă deschisă) pentru simplificarea scrierii aplicațiilor în limbajul [Java](https://ro.wikipedia.org/wiki/Java_(limbaj_de_programare)), dar există și o versiune pentru [Platforma .NET](https://ro.wikipedia.org/w/index.php?title=Platforma_.NET&action=edit&redlink=1" \o "Platforma .NET — pagină inexistentă).
* Deși este folosit în principal pentru platforma [Java EE](https://ro.wikipedia.org/w/index.php?title=Java_EE&action=edit&redlink=1), Spring poate fi utilizat pe orice aplicație Java. Este văzut în comunitatea programatorilor ca o alternativă la modelul [Enterpise JavaBeans](https://ro.wikipedia.org/w/index.php?title=Enterpise_JavaBeans&action=edit&redlink=1" \o "Enterpise JavaBeans — pagină inexistentă) (EJB).
* Spring a fost lansat în anul 2002 de către Rod Johnson împreună cu cartea sa, Expert One-on-One: J2EE Design and Development. În iunie 2003 Spring a intrat sub licența Apache 2.0.

# *Spring initialization (Spring Initializr)*

Cu ajutorul platformei Spring Initializr putem selecta ceea ce vrem sa contina proiectul nostru.



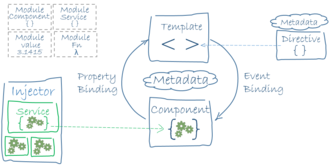
***Angular***



Un singur framework atat pentru desktop cat si pentru mobile.

Angular (numit și "Angular 9" ) este o platformă de [dezvoltare web](https://ro.wikipedia.org/wiki/Dezvoltare_web" \o "Dezvoltare web) [cu sursă deschisă](https://ro.wikipedia.org/wiki/Open-source) bazată pe limbajul [TypeScript](https://ro.wikipedia.org/wiki/TypeScript" \o "TypeScript). Proiectul este dezvoltat de Echipa Angular de la [Google](https://ro.wikipedia.org/wiki/Google) și de o comunitate de utilizatori individuali și companii. Angular este o rescriere completă, de către aceeași echipă, a frameworkului [AngularJS](https://ro.wikipedia.org/w/index.php?title=AngularJS&action=edit&redlink=1" \o "AngularJS — pagină inexistentă).

Inițial, versiunea rescrisă a AngularJS a fost numită "Angular 2" de echipă, însă acest lucru a provocat confuzie printre dezvoltatori. De aceea, echipa a anunțat că "AngularJS" se va referi la versiunile 1.X și "Angular" (fără "JS") la versiunile 2 și ulterioare.

[](https://ro.wikipedia.org/wiki/Fi%C8%99ier:Architecture_of_an_Angular_2_application.png)

*Arhitectura unei aplicații Angular. Principalele blocuri sunt module, componente, template-uri, metadate, legături de date, directive, servicii și injecții de dependențe.*

Angular nu are conceptul de domeniu de vizibilitate (în [engleză](https://ro.wikipedia.org/wiki/Limba_englez%C4%83" \o "Limba engleză) "scope") sau controlere, ci utilizează o ierarhie de componente ca principală caracteristică arhitecturală.

Angular are o sintaxă diferită pentru expresii, ce se concentrează pe "[]" pentru conectarea proprietăților și pe "( )" pentru conectarea evenimentelor

Modularitate - mare parte din funcționalitatea frameworkului a fost mutată în module

Angular recomandă folosirea limbajului [TypeScript](https://ro.wikipedia.org/wiki/TypeScript" \o "TypeScript), ce are următoarele proprietăți:

* [Programare orientată pe obiecte](https://ro.wikipedia.org/wiki/Programare_orientat%C4%83_pe_obiecte) folosind clase
* [Tiparire statică](https://ro.wikipedia.org/w/index.php?title=Tipare_static%C4%83&action=edit&redlink=1)
* [Programare generică](https://ro.wikipedia.org/w/index.php?title=Programare_generic%C4%83&action=edit&redlink=1)

[TypeScript](https://ro.wikipedia.org/wiki/TypeScript) este un supraset al [ECMAScript 6](https://ro.wikipedia.org/w/index.php?title=ECMAScript_6&action=edit&redlink=1" \o "ECMAScript 6 — pagină inexistentă) (ES6), fiind compatibil cu [ECMAScript 5](https://ro.wikipedia.org/w/index.php?title=ECMAScript_5&action=edit&redlink=1" \o "ECMAScript 5 — pagină inexistentă) (i.e.: JavaScript). Angular include și noutățile din [ES6](https://ro.wikipedia.org/w/index.php?title=ECMAScript_6&action=edit&redlink=1):

* [Programare lambda](https://ro.wikipedia.org/w/index.php?title=Programare_lambda&action=edit&redlink=1)
* [Iteratori](https://ro.wikipedia.org/w/index.php?title=Iterator&action=edit&redlink=1)
* bucle For/Of
* generatori după modelul [Python](https://ro.wikipedia.org/wiki/Python)
* [Reflecție](https://ro.wikipedia.org/w/index.php?title=Reflec%C8%9Bie_(programare)&action=edit&redlink=1)
* [Încărcare dinamică](https://ro.wikipedia.org/w/index.php?title=%C3%8Enc%C4%83rcare_dinamic%C4%83&action=edit&redlink=1)
* Compilare asincronă a template-urilor

Callback-uri iterative oferite prin RxJS. RxJS limitează vizibilitatea și posibilitățile de depanare ale stării s state visibility and debugging, dar aceste probleme pot fi rezolvate cu module precum ngReact sau ngrx.

Suportă Angular Universal, o tehnologie ce rulează aplicații Angular pe server

Are propria suită de componente de UI moderne ce merg atât pe mobil cât și pe desktop, numită Angular Material.

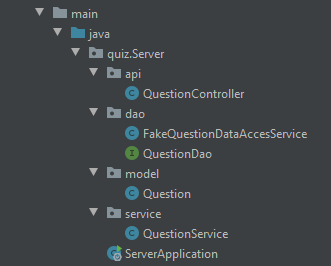
***Node JS***

Node.js este o platforma de server open source gratuita ce poate rula pe diferite sisteme de operare (Windows, Linux, Unix, Mac OS X, etc), ce iti permite sa rulezi Javascript pe server. Ea foloseste programare asincrona.

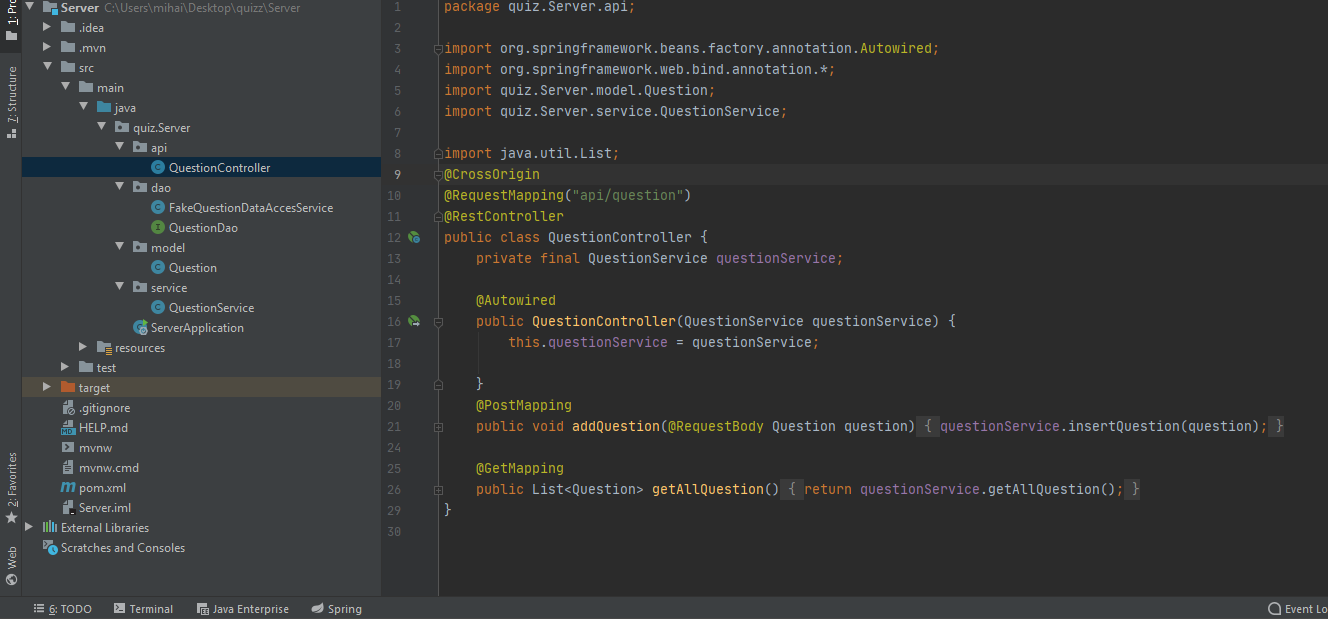
### Ce poate face Node.js?

* Node.js poate genera continut dinamic pe pagina.
* Poate crea, deschide, citi, scrie, sterge si inchide fisiere pe server.
* Node.js poate colecta date din formular.
* Poate adauga, sterge, modifica date intr-o baza de date.

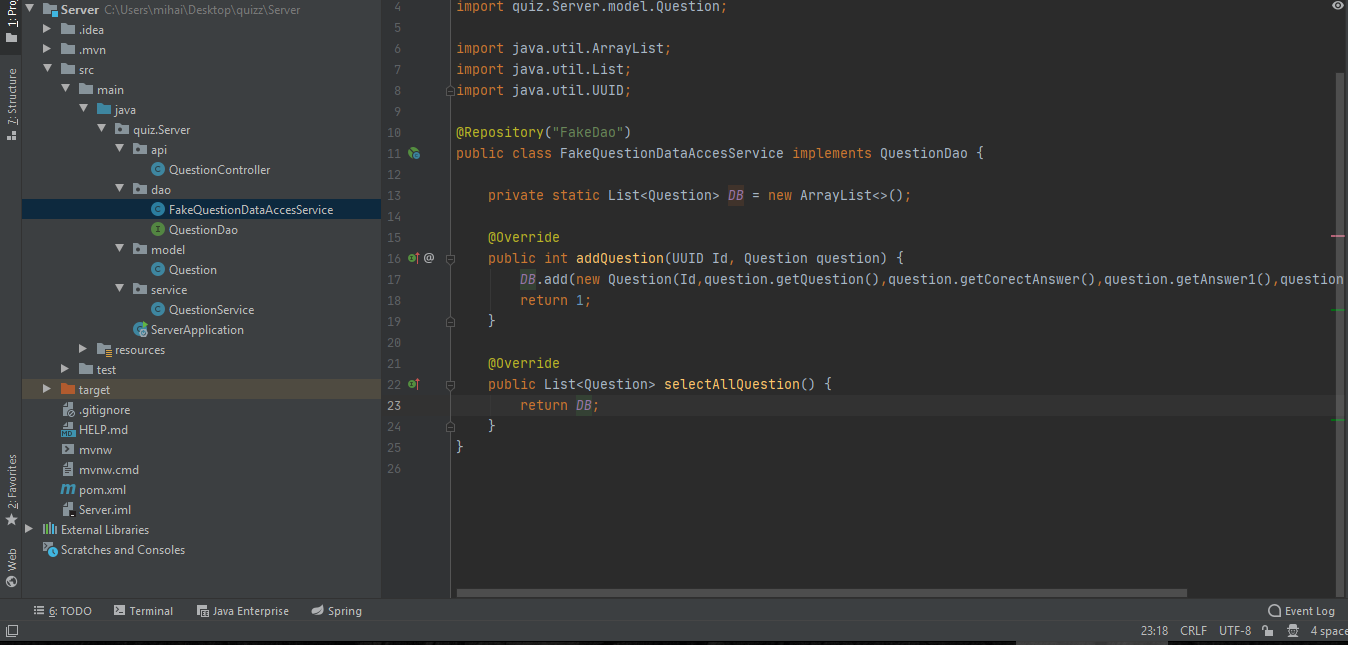
***Arhitectura Spring***



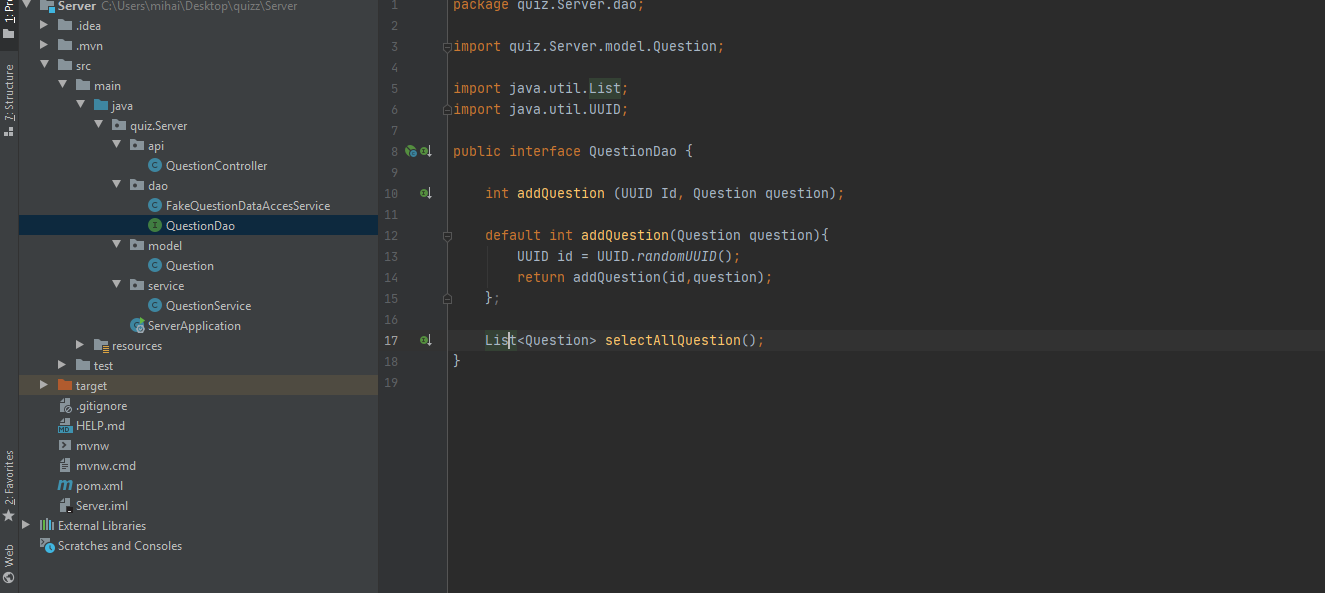
*Question Controller*



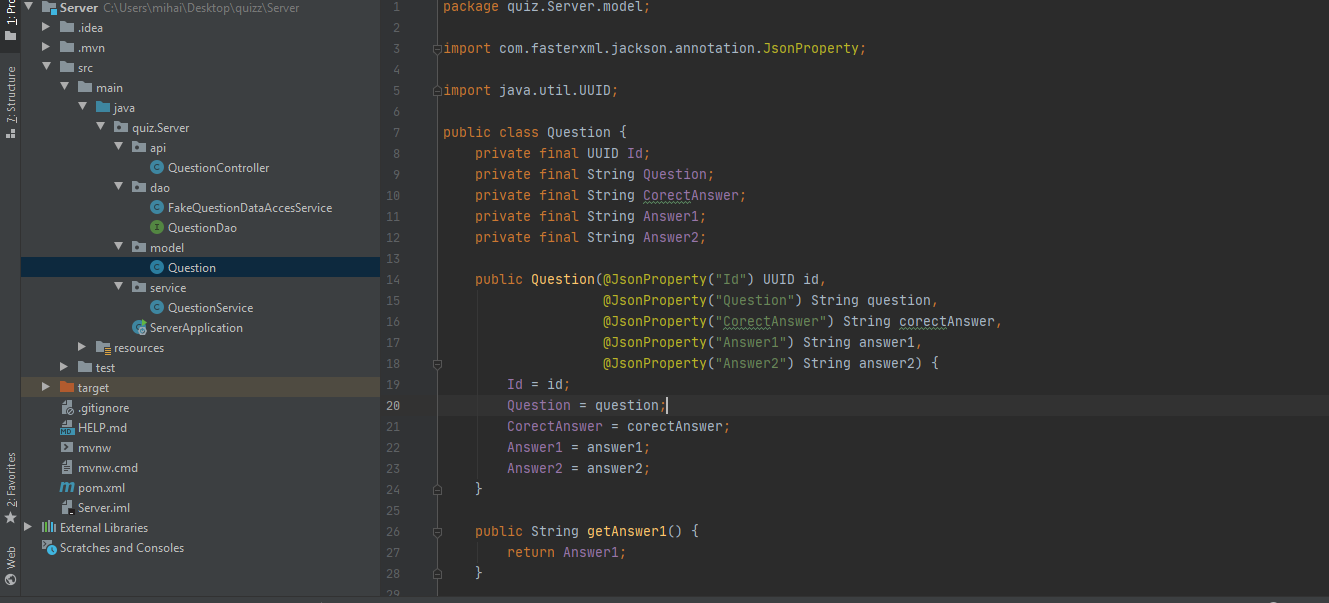
*FakeQuestionDataAccesService*



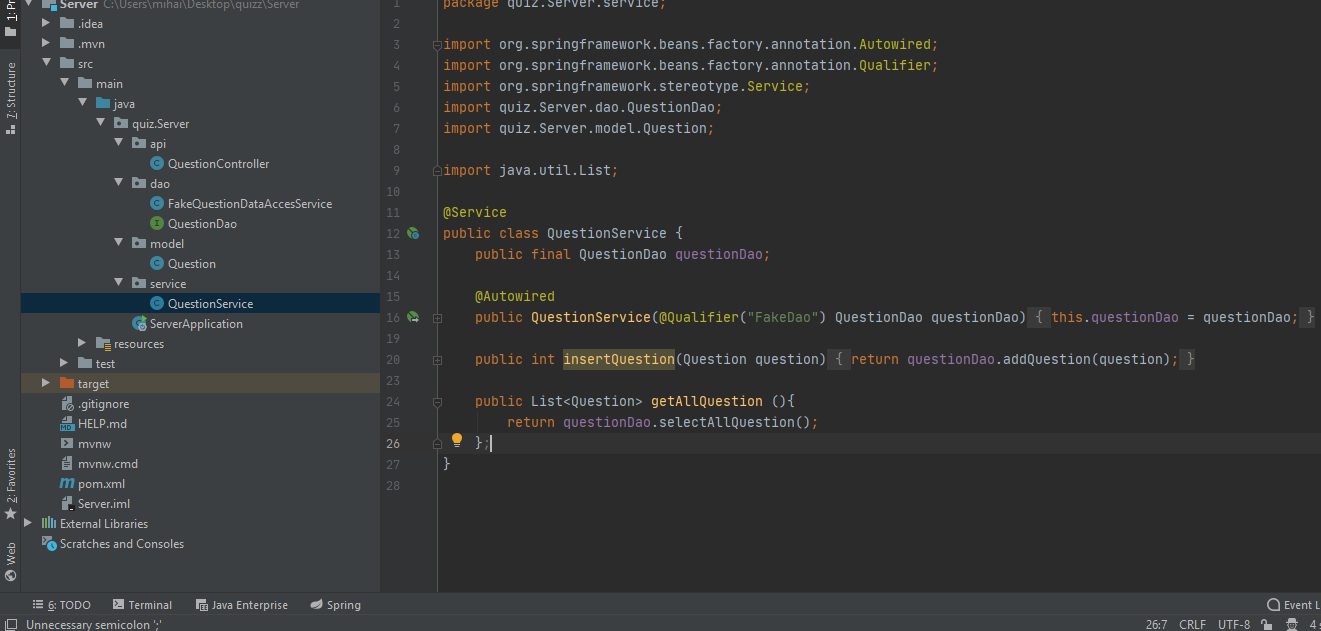
*QuestionDao*



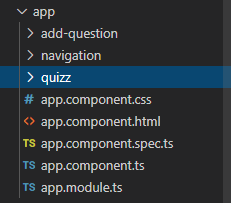
*Question*



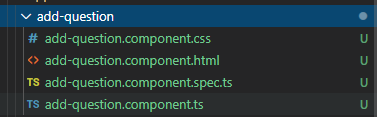
*QuestionService*



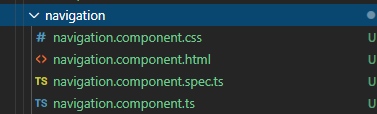
***Arhitectura Angular***



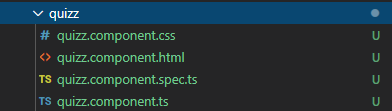
*Add-question Component*



*Navigation Component*



*Quizz Component*



***Comunicarea Client Server***

Modelul client-server este o structură sau arhitectură aplicație distribuită care partajează procesarea între furnizorii de servicii numiți [servere](https://ro.wikipedia.org/wiki/Server" \o "Server) și elementele care solicită servicii, numite [clienți](https://ro.wikipedia.org/w/index.php?title=Client&action=edit&redlink=1" \o "Client — pagină inexistentă). Clienții și serverele comunică printr-o [rețea de calculatoare](https://ro.wikipedia.org/wiki/Re%C8%9Bea_de_calculatoare" \o "Rețea de calculatoare), de obicei prin [Internet](https://ro.wikipedia.org/wiki/Internet), având suporturi hardware diferite, dar pot rula și pe același sistem fizic. Un server (fizic) rulează unul sau mai multe programe server, care partajează resursele existente cu clienții. Clientul nu partajează niciuna dintre resursele proprii, ci apelează la resursele serverului prin funcțiile server. Clienții inițiază comunicația cu serverele și așteaptă mesajele acestora. Pentru menținerea legăturii între cei doi, indiferent de pauzele care intervin, se folosește conceptul de [sesiune](https://ro.wikipedia.org/w/index.php?title=Sesiune&action=edit&redlink=1" \o "Sesiune — pagină inexistentă), care de obicei este limitată în timp.

Hypertext Transfer Protocol (HTTP) este metoda cea mai des utilizată pentru accesarea informațiilor în [Internet](https://ro.wikipedia.org/wiki/Internet) care sunt păstrate pe servere [World Wide Web](https://ro.wikipedia.org/wiki/World_Wide_Web) (WWW). Protocolul HTTP este un protocol de tip text, fiind protocolul "implicit" al WWW. Adică, dacă un [URL](https://ro.wikipedia.org/wiki/URL) nu conține partea de protocol, aceasta se consideră ca fiind http. HTTP presupune că pe calculatorul destinație rulează un program care înțelegele protocolul. Fișierul trimis la destinație poate fi un document [HTML](https://ro.wikipedia.org/wiki/HTML) (abreviație de la HyperText Markup Language), un fișier grafic, de sunet, animație sau video, de asemenea un program executabil pe [server-ul](https://ro.wikipedia.org/wiki/Server) respectiv sau și un editor de text. După clasificarea după modelul de referință OSI, protocolul HTTP este un protocol de nivel aplicație. Realizarea și evoluția sa este coordonată de către [World Wide Web Consortium](https://ro.wikipedia.org/wiki/World_Wide_Web_Consortium) (W3C).

HTTP oferă o tehnică de comunicare prin care paginile web se pot transmite de la un computer aflat la distanță spre propriul computer. Dacă se apelează un link sau o adresă de web cum ar fi [http://www.example.com](http://www.example.com/), atunci se cere calculatorului host să afișeze o pagină web (index.html sau altele). În prima fază numele (adresa) www.example.com este convertit de protocolul DNS într-o adresă IP. Urmează transferul prin protocolul TCP pe portul standard 80 al serverului HTTP, ca răspuns la cererea HTTP-GET. Informații suplimentare ca de ex. indicații pentru browser, limba dorită ș.a. se pot adăuga în header-ul (antetul) pachetului HTTP. În urma cererii HTTP-GET urmează din partea serverului răspunsul cu datele cerute, ca de ex.: pagini în (X)HTML, cu fișiere atașate ca imagini, fișiere de stil (CSS), scripturi (Javascript), dar pot fi și pagini generate dinamic (SSI, JSP, PHP și ASP.NET). Dacă dintr-un anumit motiv informațiile nu pot fi transmise, atunci serverul trimite înapoi un mesaj de eroare. Modul exact de desfășurare a acestei acțiuni (cerere și răspuns) este stabilit în specificațiile HTTP.