

Eksamen A

I fagene

Programmering og Teknik

2024, juni, kl. 9.00 – 13.00 (14.00)

Alle skriftlige materialer, pc'er, bærbare computere og internet ressourcer er tilladt til eksamen.

Mobiltelefoner og kommunikation med andre personer (inkl. chatbots), bortset fra kommunikation med eksaminator og censor, er forbudt. Dog er det tilladt at bruge GitHub Copilot

Du må ikke gemme dine løsninger på eksterne netværksdrev/værter som GitHub, Facebook o. lign. Overtrædelse af dette vil medføre bortvisning fra eksamen og passende sanktioner vil senere blive tildelt både til afsender/uploader og modtager.

Ved eksamens afslutning skal du lægge din løsning op i Wiseflow.

Eksamen varer 4 timer og efterfølges af 1 times evaluering. I disse tidsrum kan du kun forlade eksamenslokalet med henblik på toilet besøg. Rygning er ikke tilladt.

Ud over disse opgaver, kan du kan blive bedt at besvare nogle ekstra spørgsmål omkring dine svar og eventuelle andre valg.

Kontroller at dette sæt indeholder 3 sider inklusive forsiden.

Introduktion

I denne opgave skal du lave et system der kan administrere legepladser (playgrounds).

| ID | Name | MaxChildren | MinChildAge |
|----|-------------------|-------------|-------------|
| 1 | Millpark | 10 | 5 |
| 2 | Secret Playground | 12 | 4 |
| 3 | Library | 8 | 3 |
| 4 | School | 15 | 7 |

Opgave 1 Model klasse

Lav et projekt i Visual Studio, type Class Library, navn PlayGroundLib

Lav en klasse PlayGround med properties

- Id et tal
- Name en tekst
- MaxChildren et tal
- MinAge et tal

Tilføj en ToString metode der afleverer alle properties i en string

Opgave 2 Repository

Tilføj til projektet fra forrige opgave: En klasse PlayGroundsRepository, der indeholder en liste af PlayGround objekter.

Tilføj mindst 4 PlayGround objekter til listen.

PlayGroundsRepository er ansvarlig for tildeling af Id'er til PlayGround objekter.

PlayGroundsRepository skal have flg metoder

- GetAll() returnerer alle PlayGround objekter
- GetById(id) returnerer PlayGround objekt med det nævnte id, eller *null*.
- Add(playGr) tilføjer et nyt PlayGround objekt, og returnerer objektet
- Update(id, playGr) opdaterer PlayGround objektet med det nævnte id.
Returnerer det opdaterede objekt/.

Opgave 3 REST Controller

Lav et nyt projekt i Visual Studio, type ASP.NET Core Web API.

Tilføj en reference til dit PlayGroundLib.

Tilføj en Controller klasse med metoder svarende til metoderne i PlayGroundsRepository.

Du skal bruge relevante HTTP metoder og statuskoder.

Afprøv din REST service i Swagger.

Opgave 4 Azure & CORS

Publish dit REST API til Azure

Nævn min 2 fordele ved at køre et REST API i Azure i forhold til at køre på localhost.

Tilføj CORS til din REST controller.

Du bestemmer selv, hvordan du vil implementere CORS

Opgave 5 Web-applikation

Du skal lave en web-applikation med HTML, JavaScript, Vue.js, Axios og Bootstrap.

1. Lav en HTML side med en tabel, der viser alle PlayGrounds
2. Giv brugeren mulighed for at tilføje en ny PlayGround
3. Giv brugeren mulighed for at opdatere en PlayGround
4. Du skal pynte din html-side ved at anvende Bootstrap til at style nogle af html-elementerne

Opgave 6 TCP server

Du skal lave en TCP server der skal tjekke om et barn må lege på en legeplads ved hjælp af barnets alder. Det skal så tjekkes op imod den property, der hedder MinChildAge.

Du skal implementere en TCP server der skal lytte på port 7531

Din server skal være concurrent

Serveren skal have en liste med ovenstående objekter

Serveren skal kunne modtage en besked med en alder, og så skal den svare tilbage med en liste af de legepladser hvorpå et barn i den alder må lege. Altså hvor MinChildAge er mindre eller lig med alderen. Den skal aflevere listen i JSON format

F.eks:

| Server modtager: | Server svarer tilbage: |
|------------------|---|
| 3 | [{"ID": 3, "Name": "Library", "MaxChildren": 8, "MinChildAge": 3}] |
| 5 | [{"ID": 1, "Name": "Millpark", "MaxChildren": 10, "MinChildAge": 5}, {"ID": 2, "Name": "Secret Playground", "MaxChildren": 12, "MinChildAge": 4}, {"ID": 3, "Name": "Library", "MaxChildren": 8, "MinChildAge": 3}] |

Du skal teste at serveren virker med socket-test

Opgave 7 Database

Du skal lave en ny PlayGroundsRepository klasse der skal bruge en database i stedet for en liste.

Du skal oprette en PlayGrounds database-tabel med relevante kolonner

Det er databasens ansvar at tildele nye ID'er til PlayGround objekter.

Du skal kun lave ændringen i din PlayGroundLib. Ikke i din REST service.

Opgave 8 Test af Repository

Lav Unit test af dit nye repository. Din test skal have en god "code coverage"