Pràctica 1

Si entres al món del "blogging", és probable que hagis sentit parlar de Medium, la plataforma de blocs en línia amb més de 170 milions de lectors. Com a exemple de periodisme social, és la llar d'articles llargs i curts, amb publicacions de periodistes tan professionals com a ciutadans, líders de pensament i acadèmics.

En el nostre cas, l'objectiu d'aquesta pràctica serà proporcionar una API Web per a poder afegir articles en línia de temes tecnològics com per exemple, *Bitcoin*, *Blockchains* o llenguatges de programació com *JavaScript* o *WebAssembly*.

1. Objectius

En la primera part de la pràctica construirem una API Web que implementarà la funcionalitat necessària per afegir un article curt.

2. Funcionament general

Com que encara no implementarem cap "front-end" web, només ens haurem de preocupar en aquesta pràctica d'implementar les interfícies RESTful definides en l'API Web. Específicament, l'API Web servirà per exposar el model de dades relacional implementat amb JPA, juntament amb la implementació d'una part de la lògica de negoci de l'aplicació.

Les aplicacions web de blogging en línia mostren un llistat dels articles més populars. Aquest llistat inclou una **imatge destacada** per cada article. Al costat esquerre de la imatge, apareix en primera instància l'**autor** de l'article, en lletres més grans, el **títol** de l'article i dessota, un **resum del contingut de 20 paraules**. També, s'hi acostuma a afegir la **data de la publicació** i el **nombre total de visualitzacions** per indicar-ne la popularitat (tres mil tres-centes visualitzacions en l'exemple: 3.3k).

Cal destacar que els articles poden ser totalment públics o només disponibles pels usuaris registrats. En el cas de la plataforma Medium, els articles privats s'indiquen amb la icona ✦ com en la Figura 1.

Un exemple, podria ser el següent:

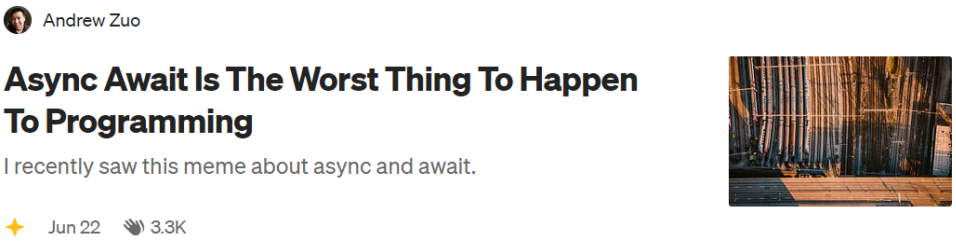


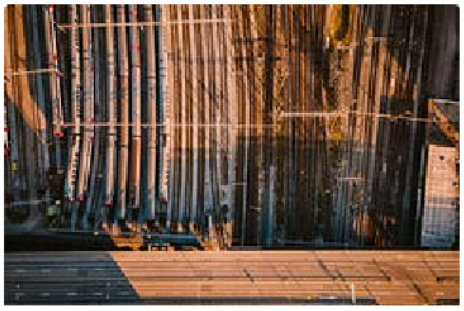
Figura 1. Resum de l'article.

Quan es clica damunt de la imatge o el títol de l'article, típicament s'obre una nova pàgina amb l'article complet. Per facilitar el desenvolupament de la pràctica, l'article seguirà el següent format:

- Damunt de tot, tornarà a aparèixer en mida més gran la **imatge destacada**.
- Seguit del **títol** i el nom de l'**autor** (p. ex., *Andrew Zuo*) i la **data de la publicació** (p. ex., *June 22, 2024*).
- A continuació, s'inclou el **nombre de visualitzacions** (p. ex., 3.3k) i **dos tòpics** per l'article, p. ex., *Web programming, JavaScript*.
- Finalment, s'afegeix el **text curt** de l'article complet ajustat a **500 paraules**.

Un exemple podria ser:

?



Async Await Is The Worst Thing To Happen To Programming

Andrew Zuo

Web programming, JavaScript

3.3K June 22, 2024

It's so good. It captures exactly how I feel about `async` and `await`. If you wait for `async` code your function now has to be `async` as well. And if there's any function relying on that function that code also has to be `async` and on and on and on. Until you reach the top level.

I've always hated this. But there's no better way to do this. It's a necessary evil. Right..? Well, I've recently been coding a lot in Go. And Go does not have `async` and `await`. And this is despite it being famous for its concurrency. And you know what? I really like it.

Figura 2. Visualització sencera de l'article.

Tota aquesta informació s'haurà de representar correctament mitjançant JPA.

3. Enunciat

Haurem de suportar les funcionalitats següents:

1. Definirem una API REST comuna per tothom i que els grups hauran d'implementar a les seves pràctiques.
2. Cal implementar els serveis REST perquè treballin i retornin JSON. Eines que us poden ser d'interès: Gson, JAXB (exemple d'ús)
3. Fer servir versionat per la sostenibilitat de l'API.
4. Cal implementar un client REST per a poder interaccionar i testejar la vostra pràctica. No esteu obligats a fer servir cap tecnologia per implementar els serveis o clients REST, és més, l'essència d'aquest sistema és la simplicitat, i això serà el que més es valorarà de la vostra solució. Suggeriment PostMan API: https://www.postman.com/. Podeu fer servir TDD de manera simple amb Postam també:

.

3.1 API

Els serveis web exposats relatius als dos recursos principals: **articles** i **usuaris.** Cal destacar que un **usuari pot o no ser l'autor d'un o més articles curts.**

Els dos recursos principals hauran de seguir la següent especificació:

Pels articles:

- **GET /rest/api/v1/article?topic=${topic}&author=${author}**
    - Per defecte, proporciona un llistat JSON amb tots els articles **ordenats per popularitat de forma descendent. La informació dels articles és la que es correspon a la Figura 1.**
    - Els paràmetres "**topic**" i "**author**" són **opcionals**.
    - Si s'especifica el paràmetre "**topic**", només s'inclouran en el llistat els articles del tipus **${topic}**. Es podrà especificar un màxim de dos topics en l'atribut **topic**, i per tant la part de consulta de l'URL podria ser del tipus: **topic=${topic1}&topic=${topic2}&author=${author}**
    - Si s'especifica el paràmetre "**author**", només s'inclouran en el llistat els articles de l'autor **${author}**.
    - El filtratge s'ha de fer mitjançant una consulta a la base de dades. No s'acceptarà com a vàlid retornar el llistat de tots els articles i filtrar-los de forma programàtica amb Java.
- **GET /rest/api/v1/article/${id}**
    - Retorna l'article sencer a partir de l'identificador de l'article, és a dir, **tota la informació indicada en la Figura 2.**
    - Si l'article és privat, només es podrà retornar si l'usuari està registrat.
    - Aquesta operació implica augmentar el nombre de visualitzacions de l'article en +1.
- **DELETE /rest/api/v1/article/${id}**
    - **Opcional!** Esborra l'article amb identificador **${id}** del sistema.
    - Per aquesta operació, cal que l'usuari sigui l'autor de l'article i estigui autentificat.
- **POST/rest/api/v1/article**
    - Permetre afegir un article nou a partir de les dades proporcionades en format JSON/XML.
    - La **data de la publicació** es determinarà de forma automàtica i s'haurà de validar que els **tòpics** escollits siguin vàlids i que l'**usuari** existeixi.
    - Cal retornar el codi **HTTP 201 Created** en cas afirmatiu juntament amb l'identificador de l'article.
    - Per aquesta operació, cal que l'usuari estigui autentificat.

Pels usuaris:

- **GET /rest/api/v1/customer**
    - Llistat JSON dels usuaris.
    - Si l'usuari és autor d'un article, cal indicar que és autor i retornar l'**enllaç a l'últim article que ha publicat** per suportar el principi de **HATEOAS**. Per exemple, en JSON:

```
"links": {
     "article": "/article/12345"
}
```

    - Aquesta crida no pot retornar informació confidencial, p. ex., la contrasenya dels usuaris.
- **GET /rest/api/v1/customer/${id}**
    - Retorna la informació de l'usuari amb identificador ${id}.
    - Aquesta crida no pot retornar informació confidencial, p. ex., la contrasenya d'aquest usuari.
- **PUT /rest/api/v1/customer/${id}**
    - **Opcional!** Modifica les dades del client amb identificador ${id} al sistema amb les dades JSON/XML proporcionades.
    - Per aquesta operació, cal que el client estigui autentificat.

3.2 Autentificació dels usuaris

Per aquells serveis web que requereixen que el client estigui autentificat, s'ha creat una anotació de tipus **Runtime** Java anomenada **@Secured.** Quan anoteu un mètode amb aquesta anotació, el servidor intentarà autentificar a l'usuari amb el mètode d'autenticació d'accés bàsica (en anglès, **HTTP Basic Authentication**), que usa la capçalera HTTP Authorization per passar al servidor les credencials de l'usuari.

Hi ha formes més professionals de realitzar aquesta autentificació i autorització d'operacions REST via JAX-RS. Com a solució simple i sense seguir cap estàndard, podríeu implementar un petit mecanisme d'API Key Authentication. O fer servir altres mecanismes d'autentificació com JWT, etc.

Resumint tot això, tindrem les següents funcionalitats obligatòries a implementar:

De l'API REST:

1. Les crides API REST obligatòries.
2. Versionat en l'API REST.
3. Dades enviades i rebudes en format JSON.
4. L'autentificació de les crides que així ho requereixen.
5. **Ús de codis HTTP adequats als resultats de les operacions de l'API REST.**
6. Proveir un client per l'API REST.
7. Proveir un joc de proves per aquest client.

Nota. Cal que diferencieu els recursos de l'API REST (**article** i **customer**) de les entitats o objectes de domini que necessiteu per poder implementar l'API web.

Ampliacions opcionals:

Sempre i quant la part obligatòria estigui ben dissenyada i desenvolupada, amb els següents punts s'aconseguirà pujar nota:

Àmbit actuació REST:

- Permetre que els serveis REST rebin i retornin XML com a l'alternativa a JSON.
- Cuinar codis i missatges HTTP per la gestió de les excepcions remotes (ex,. 404 Not Found, 403 Forbidden, 201 Created, ...).
- Ampliar l'API REST per incorporar les parts opcionals indicades.
- Preparació de **tests unitaris** automàtics amb Postman (https://learning.postman.com/docs/writing-scripts/test-scripts/)
- Sistema d'autentificació diferents a l'HTTP Basic Authentication.

4. Documentació

Juntament amb el codi font caldrà lliurar un informe en format **PDF** amb la següent estructura:

- **Introducció**. Presentació general de la pràctica i dels objectius.
- **Estructura de la pràctica**. Per exemple, caldrà especificar l'estructura del projecte, dels fitxers, les entitats JPA, entre d'altres.
- **Decisions de disseny**. Quines alternatives heu considerat a l'hora de fer el projecte i perquè heu escollit l'alternativa escollida enfront de les altres. Aquí, entre altres coses, hauríeu d'explicar quin model dels vistos a classe heu fet servir, quines parts opcionals o millores addicionals heu fet i com heu decidit implementar-les. Heu emprat eines i recursos externs?
- **Jocs de proves realitzats**. Quines proves heu fet per assegurar-vos que el vostre projecte funciona correctament en tots els casos possibles. És molt important realitzar un joc de proves extensiu, que cobreixi tots els casos, tant els positius com els negatius, com aquells que puguin produir excepcions d'algun tipus. Per tot això, heu aplicat TDD i/o BDD?
- **Conclusions**. A quines conclusions heu arribat en finalitzar aquest primer projecte.
- **Manual d'instal·lació**. La instal·lació s'ha de reduir als següents passos:
  1. Obrir projecte NetBeans;
  2. Donar-li a deploy de la vostra pràctica;
  3. (Opcionalment) Executar scripts per crear la base de dades;
  4. Passos per executar el client REST i petit joc de proves.

  A més, ha d'haver-hi un usuari de prova anomenat **sob** i contrasenya **sob** per poder provar el correcte funcionament del mecanisme de les revisions o reviews. Si necessiteu forçosament alguna altra acció, expliciteu-la adequadament.

  La **nom de la vostra base de dades ha de ser** com segueix: "**sob_grup_NN**", on "**NN**" és el número del vostre grup de pràctiques del Moodle.

5. Lliurament del projecte

El projecte es lliurarà via Moodle abans de la data màxima d'entrega.

Cal que entregueu:

1. Un arxiu **.zip** anomenat SOB_P1_nom1_nom2.zip, on nom1 i nom2 seran el nom (i cognoms) de cada un dels components del grup que entrega la pràctica.

- Hi ha d'haver una carpeta anomenada **projecte** on hi haurà tots els fitxers que formen el projecte en NetBeans, de manera que es pugui obrir directament en el mateix entorn del laboratori.
- Cal deixar ben clar l'ús dels passos addicionals que siguin necessaris per posar en funcionament la vostra pràctica.

2. Un fitxer anomenat **documentacio.pdf** amb la documentació de la pràctica que segueixi el format explicat en el punt anterior.

6. Avaluació

Cal fer com a mínim les parts obligatòries que us permetran aconseguir fins a un 9.5 de la nota del projecte.

Si implementeu 1 ampliació opcional correctament, la nota que es podrà aconseguir serà fins a un 10 per aquesta pràctica. És preferible tenir 1 ampliació ben desenvolupada que una quantitat superior i que estiguin incorrectes.

Alguns dels aspectes que es valoraran són:

- Funcionalitat correcta del projecte
- Estructuració de la pràctica
- Decisions de disseny preses
- Elegància de la solució proposada a nivell de programació
- Seguretat i extensibilitat de l'aplicatiu
- Compliment dels estàndards i convencions a l'hora de desenvolupar una aplicació web en Java+Netbeans (Java Code Conventions).
- Qualitat de la documentació presentada.

Darrera modificació: dijous, 3 d'octubre 2024, 10:13

◄ Lliurament segona convocatòria

Salta a...

Codi base per la pràctica 1 (JDK 17; GlassFish 6) ►