

DP2 2024

Acme Software Factory

Repositorio: <https://github.com/DP2-2024-C1-029/Acme-Software-Factory.git>

Miembro:

- Juan José Gómez Borrallo (juagombor@alum.us.es)

Tutor: José González Enríquez
06/03/2024

GRUPO C1.029
Versión 2.0

Índice

Historial de versiones.....	3
Introducción.....	4
Contenido.....	4
D01	4
D02	4
Notice.....	4
TrainingModule	5
TrainingSession.....	6
Developer	6
DeveloperDashboard.....	7
AdministratorDashboard	7
Popular base de datos.	8
Conclusiones	8
D01	8
D02	9
Bibliografía	9

Historial de versiones

Fecha	Versión	Descripción	Entrega
16/02/2024	V1.0	Inicio del documento	D01
06/03/2024	V2.0	Actualización D02	D02

Introducción

A continuación, se procede a explicar un breve contenido sobre el proyecto.

Estamos trabajando con el proyecto “Acme Software Factory” en el cual hemos tenido que crear un repositorio de GitHub como grupo (C1.029), y hemos tenido que montarlo localmente en nuestros equipos para poder desarrollar las tareas que se nos han otorgado tanto como grupo como de manera individual.

En este documento, se van a detallar el porqué de las decisiones utilizadas para la creación de los atributos.

Contenido

D01

En primer lugar, se comenzó con la instalación de los softwares necesarios para desplegar el proyecto. Durante la instalación no hubo ningún error remarcable, se siguió las instrucciones al pie de la letra y se pudo instalar todo correctamente.

Seguidamente, se procedió a crear las dos bases de datos requeridas. Una vez creadas las dos bases de datos, se cargó el proyecto en eclipse y se esperó a que cargara. Cuando finalizó de cargar, se comprobó que no había ningún error y se cerró eclipse. Tras ello, se abrió el create-launcher.bat y se abrió eclipse nuevamente para popular la base de datos. En este paso sí que hubo un error, el cual fue que una vez seleccionabas la herramienta de debug en eclipse, no aparecían los launchers. El error estaba ocasionado por importar el proyecto con las etiquetas “[artifacId]-[version]”. Al quitar la etiqueta de “[version]” se consiguió arreglar el problema.

Con ello se pudo ejecutar el software y empezar las tareas sin problemas.

D02

En este apartado vamos a centrarnos en 3 Clases y en 2 Dashboards. También vamos a hablar sobre cómo he creado los csv para popular datos.

Por aclarar, **no se ha implementado** la relación de training con el rol developer, porque se pide explícitamente en el apartado 7 del D03.

Vamos a empezar por la clase:

Notice.

En primer lugar, hemos definido la clase con las anotaciones @Getter y @Setter para que nos cree automáticamente los get y los set de los atributos de la entidad. Y además, le hemos añadido la anotación @Entity. Se han creado los atributos:

instantiationMoment de tipo Date con los atributos “@Temporal(TemporalType.TIMESTAMP)”, “@Past” y “@NotNull” puesto que es una fecha y tiene que estar en el pasado. Además se entiende que es el momento de creación por lo que no debe ser nulo.

title de Tipo String y con los atributos “@NotBlank” y “@Length(max = 75)” puesto que en los requisitos nos piden que sean notblank y menor a 76 caracteres.

author de Tipo String y con los atributos “@NotBlank” y “@Length(max = 75)” puesto que en los requisitos nos piden que sean notblank y menor a 76 caracteres.

message de Tipo String y con los atributos “@NotBlank” y “@Length(max = 100)” puesto que en los requisitos nos piden que sean notblank y menor a 101 caracteres.

email de Tipo String y con el atributo “@Email”, para que se haga la validación automáticamente y coincida con el formato de un email. También se ha añadido la anotación “@Length(max = 255)” para que no haya ningún atributo sin límites, tal y como han indicado los profesor, siguiendo así sus buenas prácticas

link de Tipo String y con el atributo “@URL”, para que se haga la validación automáticamente y coincida con el formato de una URL. También se ha añadido la anotación “@Length(max = 255)” para que no haya ningún atributo sin límites, tal y como han indicado los profesor, siguiendo así sus buenas prácticas

También se ha creado una función para cuando se implementen los servicios para formatear el nombre como “usuario – Apellido, Nombre” llamado setAuthorFormatter el cual recibe como entrada el nombre de usuario, y nombre y apellidos del autor.

TrainingModule

code de Tipo String y con los atributos “@NotBlank”, “@Pattern(regex = “[A-Z]{1,3}-[0-9]{3}”)” y “@Column(unique = true)” puesto que en los requisitos nos piden que sean notblank y único y con el patrón “[A-Z]{1,3}-[0-9]{3}”.

creationMoment de tipo Date con los atributos “@Temporal(TemporalType.TIMESTAMP)”, “@NotNull”, “@Past” puesto que es una fecha y tiene que estar en el pasado, y no tiene sentido que si se crea la entidad, tenga la fecha de creación nula, por lo que se ha decidido poner la restricción.

details de Tipo String y con los atributos “@NotBlank” y “@Length(max = 100)” puesto que en los requisitos nos piden que sean notblank y menor a 101 caracteres.

difficultyLevel de tipo Difficulty level, la cual es una clase enumerada nueva que se ha creado y puede tomar los valores (“Basic”, “Intermediate”, “Advanced”), y con la anotación @NotNull, para que no pueda ser nulo y se tenga que coger los valores definidos.

updateMoment de tipo Date con los atributos “@Temporal(TemporalType.TIMESTAMP)” y “@PastOrPresent” puesto que es una fecha. Se ha elegido tomar la anotación @PastOrPresent puesto que se ha observado el proyecto starter Acme-Jobs, y el atributo “moment” está definido con esa anotación.

link de Tipo String y con el atributo “@URL”, para que se haga la validación automáticamente y coincida con el formato de una URL, y tiene que ser opcional. También se ha añadido la anotación “@Lenght(max = 255)” para que no haya ningún atributo sin límites, tal y como han indicado los profesor, siguiendo así sus buenas prácticas

estimatedTotalTime de tipo Integer, puesto que voy a medir los minutos, y con la anotación “@NotNull”, para que por defecto pueda tomar el valor 0.

También se ha creado una relación ManyToOne con developer (nos hará falta para el D03). Se han añadido las anotaciones “@NotNull @Valid @ManyToOne(optional = false)” puesto que se tiene que validar a nivel de Framework, Application y Base de datos.

TrainingSession

code de Tipo String y con los atributos “@NotBlank”, “@Pattern(regex = “TS-[A-Z]{1,3}-[0-9]{3}”)” y “@Column(unique = true)” puesto que en los requisitos nos piden que sean notblank y único y con el patrón “TS-[A-Z]{1,3}-[0-9]{3}”.

Para la definición del periodo he creado dos atributos de Tipo Date llamados startTime y endTime ambos con la anotación “@Temporal(TemporalType.TIMESTAMP)”. No se le ha puesto más atributos de tipo temporal, porque simplemente me sirven para calcular un periodo, actualmente no es necesario considerar lo que pasa antes o después de las fechas del periodo. Otro atributo que sí se le ha añadido es el @NotNull, porque no tiene sentido que no exista una fecha, tiene que existir sí o sí.

location de Tipo String y con los atributos “@NotBlank” y “@Lenght(max = 75)” puesto que en los requisitos nos piden que sean notblank y menor a 76 caracteres.

instructor de Tipo String y con los atributos “@NotBlank” y “@Lenght(max = 75)” puesto que en los requisitos nos piden que sean notblank y menor a 76 caracteres.

contactEmail de Tipo String y con los atributos “@Email” y “@NotBlank”, para que se haga la validación automáticamente y coincida con el formato de un email, y además, la anotación notblank puesto que tiene que ser mandatory. También se ha añadido la anotación “@Lenght(max = 255)” para que no haya ningún atributo sin límites, tal y como han indicado los profesor, siguiendo así sus buenas prácticas

furtherInformationLink de Tipo String y con el atributo “@URL”, para que se haga la validación automáticamente y coincida con el formato de una URL, y tiene que ser opcional. También se ha añadido la anotación “@Lenght(max = 255)” para que no haya ningún atributo sin límites, tal y como han indicado los profesor, siguiendo así sus buenas prácticas

También se ha creado una relación ManyToOne con training module. Se han añadido las anotaciones “@NotNull @Valid @ManyToOne(optional = false)” puesto que se tiene que validar a nivel de Framework, Application y Base de datos.

Developer

degree de Tipo String y con los atributos “@NotBlank” y “@Lenght(max = 75)” puesto que en los requisitos nos piden que sean notblank y menor a 76 caracteres.

specialisation de Tipo String y con los atributos “@NotBlank” y “@Lenght(max = 100)” puesto que en los requisitos nos piden que sean notblank y menor a 101 caracteres.

listOfSkills de Tipo String y con los atributos “@NotBlank” y “@Lenght(max = 100)” puesto que en los requisitos nos piden que sean notblank y menor a 101 caracteres.

email de Tipo String y con los atributos “@Email” y “@NotBlank”, para que se haga la validación automáticamente y coincida con el formato de un email, y además, he considerado que como no te dice que es opcional, tiene que ser obligatorio. También se ha añadido la anotación “@Lenght(max = 255)” para que no haya ningún atributo sin límites, tal y como han indicado los profesor, siguiendo así sus buenas prácticas

link de Tipo String y con el atributo “@URL”, para que se haga la validación automáticamente y coincida con el formato de una URL, y tiene que ser opcional. También se ha añadido la anotación “@Lenght(max = 255)” para que no haya ningún atributo sin límites, tal y como han indicado los profesor, siguiendo así sus buenas prácticas

DeveloperDashboard

Que contiene totalTrainingModuleWithUpdateMoment y totalNumberOfTrainingSessionsWithLink que son de tipo **int** puesto que podemos definirlo con valor 0 y luego, tenemos:

Double	averageTrainingModuleTime;
Double	deviationTrainingModuleTime;
double	minimumTrainingModuleTime;
double	maximumTrainingModuleTime;

Que se ha decidió tomar de tipo **Double**, puesto que, en este caso, pueden ser nulos, por ejemplo si tenemos 0 valores y calculamos la media entre 0 valores, no se podría. Para los tipos como “Min” o “Max” como se pueden inicializar como 0 se ponen como double

AdministratorDashboard

```
int            totalNumberOfPrincipalsWithAdministrator;  
int            totalNumberOfPrincipalsWithManager;  
int            totalNumberOfPrincipalsWithDeveloper;  
int            totalNumberOfPrincipalsWithSponsor;  
int            totalNumberOfPrincipalsWithAuditor;  
int            totalNumberOfPrincipalsWithClient;
```

Se ha decidido nombrar el atributo como tipo **int** puesto que podemos tomar por defecto el valor 0

```
Double         ratioOfNoticesWithEmailAndLink;  
Double         ratioOfCriticalObjectives;  
Double         ratioOfNonCriticalObjectives;  
Double         averageValueInTheRisks;  
double        minValueInTheRisks;  
double        maxValueInTheRisks;  
Double         deviationValueInTheRisks;
```

Al igual que antes, se ha decidió tomar de tipo Double, puesto que, en este caso, pueden ser nulos, por ejemplo, si tenemos 0 valores y calculamos la media entre 0 valores, no se podría. Para los tipos como “Min” o “Max” como se pueden inicializar como 0 se ponen como double

Popular base de datos.

Para popular la base de datos, se ha seguido los consejos dados en clase, y cuando se han creado los CSV, se han generado datos para probar valores mínimos, valores máximos, también se ha probado que no se pudiesen insertar valores, comprobando así las restricciones puestas a la hora de desarrollar el código, etc...

Conclusiones

D01

Se ha podido instalar todo el software de manera correcta, comprobando que todo se ejecutaba correctamente y que la base de datos estaba poblada. Tras solucionar el error mencionado, se ha podido ejecutar todo.

Gracias a configurar el proyecto, he podido aprender a montar un proyecto software real, y a enfrentarme errores con los que nunca me he enfrentado. He podido aprender un poco más al desarrollo de mis habilidades.

Además, para este entregable individual (D01) no se ha precisado de analizar ningún requisito, ya que únicamente hay que hacer documentación o insertar un texto en la portada, el cual no requiere de análisis ninguno.

D02

La creación de las clases, y las anotaciones otorgadas a sus atributos, ha sido tomada con la intención óptima para el desarrollo del proyecto. En sus correspondientes apartados se han explicado por qué se han tomado esas decisiones a la hora de colocar las anotaciones a los atributos, tomando en cuenta siempre todo el feedback que dan los profesores para desarrollar nuestro proyecto.

Además, para popular la base de datos, se ha seguido las buenas prácticas dadas por los profesores para rellenar la base de datos con todas las posibilidades y probar así el funcionamiento de nuestro software.

Bibliografía

Intencionadamente en blanco