Rapport.md 17/06/2022

# Rapport test unitaires

Nous avons décidé de faire les tests unitaires sur l'API de notre projet. En effet, c'est sur ce projet que se concentre l'essentiel de l'intelligence du projet (création, modification et suppression des éléments, vérification des données, etc).

L'API est créé en TypeScript et utilise les librairies Express et Sequelize. Nous avons donc choisi ts-jest pour nos tests unitaires, couplé avec le package @jest-mock/express afin de faciliter le test de nos controllers express et faker afin de créer un jeu de données réaliste.

### Qu'est ce qui a été testé ?

Nous avons fait le choix de tester les controlleurs, car il représente la quasi totalité de l'intelligence de l'api. De plus, tester les controlleurs permet de tester indirectement les modèles de la base de données.

Plus spécifiquement, nous avons décidé de tester les controlleurs gérant les utilisateurs et les voyages.

#### Quelles leçons en tirer?

Mettre en place ces tests unitaires nous a permis de mettre en valeur certains défauts de conception dans le code de ces controllers.

Par exemple, nous avons pu remarquer qu'aucune vérification n'était effectuée sur la forme des informations rentrées en base (par exemple, vérifier qu'une adresse mail à un format valide).

Nous avons également remarqué que certains codes d'erreurs renvoyés n'était pas les bons.

### Quels bénéfices en tirer?

L'avantage de jest est qu'il est très facile à intégrer dans une CI. Nous avons donc ajouté une Github Actions pour automatiquement lancer les tests unitaires depuis une CI. Désormais, des tests sont mis en place lorsqu'on ouvre une pull request vers notre branche master. Cela évite de mettre en production du code non fonctionnel qui casserait une fonctionnalité.

### Rapport de couverture du code par les tests

Rapport.md 17/06/2022

File    \$ Stmts   \$ Branch   \$ Funcs   \$ Lines   Uncovered Line #s							
Controllers	File	,	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
Controllers	All files		80.82	44.73	63.15	82.04	
Trip.ts							
User.ts							30-33,46,70,87-90,105,142,162-164
Core							
Database.ts		i i		46.87	68.18		
FileManagement.ts	Database.ts	i	88.88	60	100		11,14,17
Logger.ts	FileManagemer	nt.ts	75	75	57.14	73.68	
Models		i	65.51	21.42	60	65.51	
FileMetadata.ts		i	100	100	100	100	
LogbookEntry.ts	Day.ts	i	100	100	100	100	
Path.ts	FileMetadata.	ts	100	100	100	100	į
Point.ts	LogbookEntry.	ts	100	100	100	100	į
Spent		<u> </u>	100	100	100	100	
Step.ts	Point.ts			100	100	100	
TodoEntry.ts	Spent.ts	i	100	100	100	100	į
Travelers.ts	Step.ts	ii_	100	100	100	100	
Trip.ts	TodoEntry.ts	i	100	100	100	100	
User.ts	Travelers.ts	į.	100	100	100	100	
types/models	Trip.ts	i	100	100	100	100	
Common.ts	User.ts	i	100	100	100	100	
File.ts 64.7 16.66 14.28 68.75 29,36,46,52,60 Spent.ts 68.42 20 14.28 72.22 26,33,42,48,55 Todo.ts 64.7 22.22 14.28 64.7 27,34,42,48,55 Toto.ts 72.72 60 40 80 33,40  types/utils 100 100 100 100 100 Credentials.ts 100 100 100 100 100 Visibility.ts 100 100 100 100 100 Visibility.ts 100 100 100 100 100 utils 45 18.75 57.14 47.36 Hash.ts 83.33 75 75 83.33 20 Token.ts 38.23 0 33.33 40.62 12,31-59,67-75  Test Suites: 1 falled, 1 passed, 2 total Tests: 2 falled, 20 passed, 22 total Snapshots: 0 total Time: 16.599 s Ran all test suites.  All files  NAIM Summers INNEE 44.77% Exercises Now. For the deep revises block. File: 1 Statements 1 1 1 Exercises 1 1 Exercises 1 1 1 Exercises 1 1 1 Exercises 1 1 1 1 Exercises 2 1 E	types/models	į.	66.66	25	18.51	69.84	
Spent.ts	Common.ts	i i	50	100	0	50	10
Todo.ts	File.ts	ĺ	64.7	16.66		68.75	29,36,46,52,60
Trip.ts	Spent.ts		68.42			72.22	
types/utils	Todo.ts		64.7	22.22	14.28	64.7	27,34,42,48,55-56
Credenttals.ts	Trip.ts		72.72	60		80	33,40
Enum.ts							
Visibility.ts		S					
### ### ### ### ### ### ### ### ### ##							
Hash.ts		5					
Token.ts 38.23 0 33.33 40.62 12,31-59,67-75							
Test Suites: 1 failed, 1 passed, 2 total Tests: 2 failed, 20 passed, 22 total Snapshots: 0 total Time: 16.599 s Ran all test suites.  All files 80.82% Statements 303/212 44,73% Branches 51/214 63,15% Functions 68/95 82,04% Lines 303/482  Press nor / to go to the next uncovered block, b, p or k for the previous block Filter:							
Tests: 2 failed, 20 passed, 22 total Snapshots: 0 total Time: 16.599 s Ran all test suites.  All files 80.82% Statements 933/432 44,73% Branches 92/434 63,15% Functions 98/96 82,04% Lines 929/432 Press n or / to go to the next uncovered block, b, p or k for the previous block Filter:   File -   Statements • Branches • Functions • Lines • Lines •   Lines • Statements •   Description • Statements •   Description	Token.ts		38.23	0	33.33	40.62	12,31-59,67-75
Tests: 2 failed, 20 passed, 22 total Snapshots: 0 total Time: 16.599 s Ran all test suites.  All files 80.82% Statements 333/42 44.73% Branches \$1/234 63.15% Functions 88/99 82.04% Lines \$28/482  Press n or / to go to the next uncovered block, b, p or k for the previous block.  Filter:   Statements : 0 Branches : 0 Functions : 0 Lines : 0							
Tests: 2 failed, 20 passed, 22 total Snapshots: 0 total Time: 16.599 s Ran all test suites.  All files  80.82% Statements 333/42 44.73% Branches 32/34 63.15% Functions 88/99 82.04% Lines 328/482  Press n or / to go to the next uncovered block, b, p or k for the previous block.  Filter:   Statements : 0 Branches : 0 Functions : 0 Functions : 0 Lines : 0	T 1 6 11						
Snapshots: 0 total Time: 16.599 s Ran all test suites.  All files 80.8296 Statements 333/432 44.7396 Branches \$1/234 63.1596 Functions 88/99 82.0496 Lines \$289/482  Press n or j to go to the next uncovered block, b, p or k for the previous block. Filter:  File -   Statements : Branches : Finctions : Functions : Lines : D	Test Sultes: 1	Talled,	1 passed	, 2 total	,		
Time: 16.599 s Ran all test suites.  All files 80.8296 Statements 333/412 44.7396 Branches \$1/254 63.1596 Functions 86/95 82.0496 Lines \$287/481  Press n or f to go to the next uncovered block, b, p or k for the previous block.  Filter.  File - • Statements • Branches • Functions • • Lines • • Lines • •			20 passe	a, 22 tota			
Ran all test suites.  All files  80.82% Statements 333/412 44,73% Branches 53/214 63,15% Functions 166/95 82,04% Lines 329/481  Press n or / to go to the next uncovered block, b, p or k for the previous block  Filter:  File -   Statements  Branches  Finchis  Functions  Lines  Lines   Lines   Lines   Lines   Descriptions   Lines  Descriptions  Descrip							
## All files  ### 80.82% Statements 333/412							
90.82% Statements 333/412 44,73% Branches 31/318 63.15% Functions 46/95 82.04% Lines 325/482  Press n or j to go to the next uncovered block, b, p or k for the previous block.  Filter:  File •   Statements • Statements • Branches • Functions • English • Branches • Functions • • Lines • • English • • • • • • • • • • • • • • • • • • •	kan all test si	ittes.					
Press n or / to go to the next uncovered block, b, p or k for the previous block.  Filter:  File -   Statements   Statements   Branches   Find   Find   Lines   Lines   Lines    Lines     Lines     Lines      Lines							
File • • Statements • • Branches • • Functions • • Lines • •				196 Lines 329/401			
	Press n or j to go to the next uncovered block	k, b, p or k for the previo	us block.				
	r mGI.						
controllers 85.47% 100/117 73.91% 17/23 76.92% 10/13 86.95% 100/115	File ▲		Statements	÷	Branches		
76 2104 5076 46 0704 15/22 60 1004 15/22 75 2404 55/72	controllers				100/117		

File •	<b>\$</b>	Statements ÷	\$	Branches ‡	<b>\$</b>	Functions ÷	<b>\$</b>	Lines ‡	<b>\$</b>
controllers		85.47%	100/117	73.91%	17/23	76.92%	10/13	86.95%	100/115
core		76.31%	58/76	46.87%	15/32	68.18%	15/22	75.34%	55/73
models		100%	104/104	100%	0/0	100%	21/21	100%	104/104
types/models		66.66%	44/66	25%	9/36	18.51%	5/27	69.84%	44/63
types/utils		100%	9/9	100%	7/7	100%	5/5	100%	8/8
utils		45%	18/40	18.75%	3/16	57.14%	4/7	47.36%	18/38

## Code source

#### Code source de l'api

Les tests sont disponible dans le dossier <u>tests</u>.