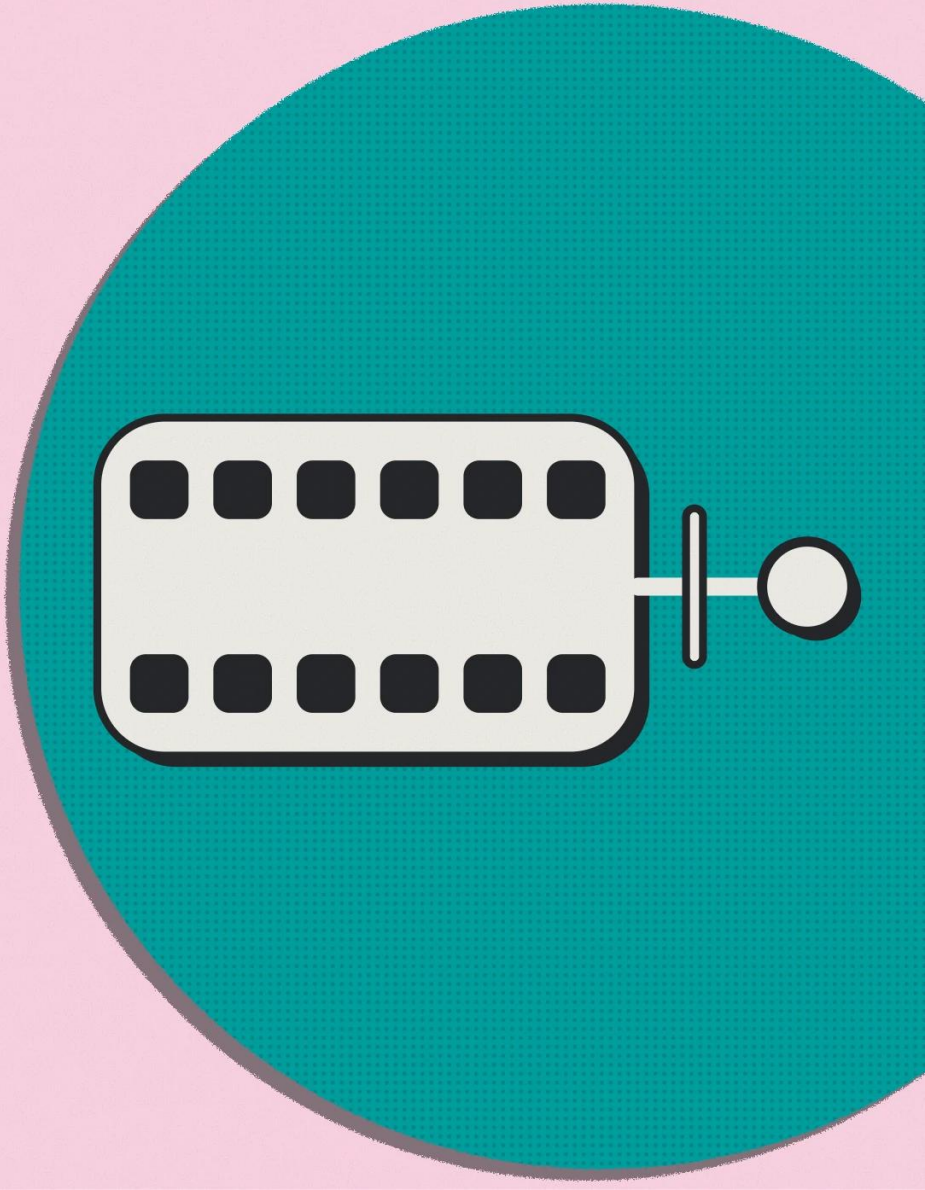
3.04

TEMPS 3 · L'APPLICATION DE GESTION · SÉANCE 1

Les règles métier

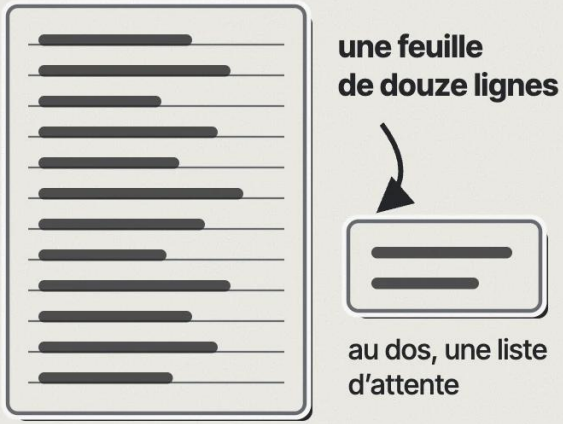
Thibault Six · IFOSUP Wavre · 2026-2027



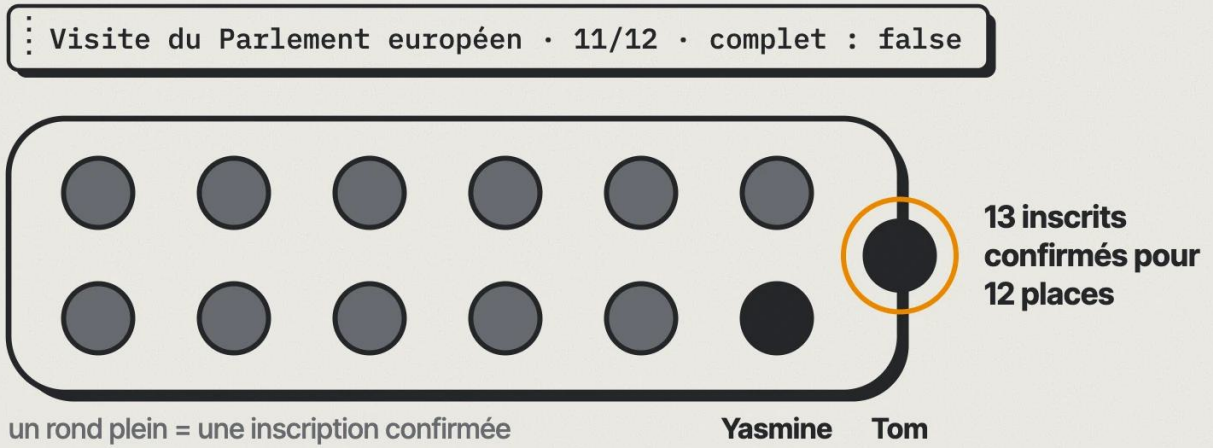
? Le bus a douze places. Que fait l'application si on confirme un treizième inscrit ?

Sans règle métier, l'application confirme un treizième inscrit dans un bus de douze places.

AU SECRÉTARIAT, SUR PAPIER



DANS L'APPLICATION, SANS LA RÈGLE



Une règle métier est une contrainte qui vient du travail du client, pas de la technique.

Elle resterait vraie avec une feuille et un crayon : il faut la demander au client.

À gauche, la feuille du secrétariat s'arrête à douze lignes. À droite, l'application confirme Tom quand même.

? D'un sujet à l'autre, quelles formes prend une règle métier ?

Une règle métier calcule un chiffre, limite un changement d'état ou compte ce qui existe déjà.

Calcul

La règle dit comment obtenir un chiffre.

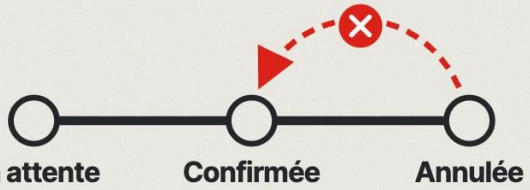
240 € ÷ **inscrits confirmés**

= la part de chaque participant

« Une activité payante coûte 8 € par personne, et 5 € pour un étudiant boursier. »

État

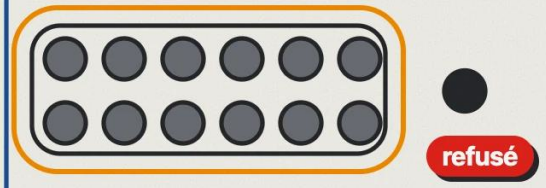
Certains passages d'un état à l'autre sont interdits.



« Une inscription annulée ne se confirme plus. »

Capacité ou unicité

La règle compte ce qui existe déjà avant d'accepter une ligne de plus.



« Pas plus de douze inscrits confirmés dans le bus. »

« Un étudiant ne s'inscrit qu'une fois à la même activité. »

Le Tamagotchi du temps 2 en avait déjà : le solde ne descend jamais sous zéro, une jauge ne dépasse pas 100.

? Le statut est bien formé, le secrétariat a le droit. Pourquoi refuser Tom ?

Pour classer un refus, demandez de quoi dépend la réponse : la valeur, la personne ou la base.

	Règle de format	Droit	Règle métier
Exemple	Le statut envoyé est `pending`, `confirmed` ou `cancelled`.	Seul le secrétariat confirme une inscription.	Une activité de douze places n'accepte pas de treizième inscrit confirmé.
La réponse dépend de	**la valeur envoyée, seule**	**la personne connectée**	**les données déjà en base**
Où elle s'écrit	la Form Request	la policy, appelée par `Gate::authorize()`	une méthode du modèle ou de l'enum, appelée dans une transaction
Ce que voit l'utilisateur	une erreur en français sous le champ	la page 403 « Accès refusé »	un message en français qui explique le refus

Ni la Form Request ni la policy ne peuvent voir que douze inscriptions sont déjà confirmées.

? Pour chaque contrainte, de quoi dépend la réponse ?

À vous : classez ces cinq contraintes, et donnez sa famille à chaque règle métier.

LA CONTRAINTE	format	droit	règle métier	famille
1 La date d'une activité est une date valide.	☐	☐	☐	----------
2 Un étudiant ne voit que ses propres inscriptions.	☐	☐	☐	----------
3 Un étudiant ne s'inscrit qu'une fois à la même activité.	☐	☐	☐	----------
4 Une inscription confirmée peut encore être annulée, une inscription annulée ne revient plus.	☐	☐	☐	----------
5 Une conférence gratuite pour les étudiants coûte 10 € aux visiteurs extérieurs.	☐	☐	☐	----------

Une case par ligne. Pour une règle métier, écrivez sa famille : calcul, état, capacité ou unicité.

? « Il ne faut pas trop d'inscrits. » Pourquoi ne peut-on pas en tirer un test ?

Dans le rapport, R1 tient en une phrase et deux exemples chiffrés, un accepté et un refusé.

	MAL ÉCRIT	BIEN ÉCRIT
LA RÈGLE	Il ne faut pas trop d'inscrits à une activité.	**R1. Une activité n'a jamais plus d'inscriptions confirmées que de places.**
accepté	Quand il reste de la place, ça passe.	Activité de 12 places, 11 inscriptions confirmées : le secrétariat confirme la suivante, l'activité compte 12 inscriptions confirmées.
refusé	Sinon, on bloque.	Activité de 12 places, 12 inscriptions confirmées : la confirmation suivante est refusée avec le message « L'activité est complète, les 12 places sont prises. », l'inscription reste en attente.

Onze et douze, pas trois et cinquante : une règle se trompe d'abord d'une unité.

À gauche, « trop » et « de la place » sans chiffre. À droite, chaque exemple devient un test, presque mot pour mot.

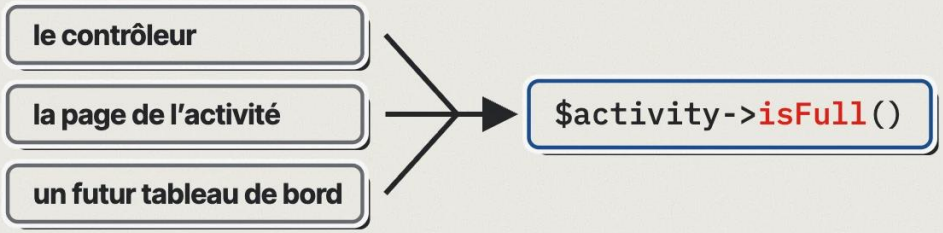
? Où écrire « êtes-vous complète ? » pour que tout le code pose la même question ?

La règle s'écrit une fois, avec le nom du métier, dans le modèle ou dans l'enum.

R1 · LE MODÈLE RÉPOND À LA QUESTION

app/Models/Activity.php

```php
public function isFull(): bool
{
    return $this->confirmedCount() >= $this->capacity;
}
```



R2 · L'ENUM CONNAÎT LES PASSAGES PERMIS

app/Enums/RegistrationStatus.php

```php
return match ($this) {
    self::Pending => [self::Confirmed, self::Cancelled],
    self::Confirmed => [self::Cancelled],
    self::Cancelled => [],
};
```

Tinker affiche les passages permis :

```
En attente -> Confirmée, Annulée
Confirmée -> Annulée
Annulée -> rien
```

R1 compte les lignes en base : elle vit dans le modèle. R2 ne dépend que du statut : elle vit dans l'enum.

? Le format est bon, le droit aussi : où le refus de Tom se décide-t-il ?

Le contrôleur vérifie le format, puis le droit, puis les règles, dans une transaction.



app/Http/Controllers/RegistrationController.php

```php
// R1: never more confirmed registrations than places.
if ($next === RegistrationStatus::Confirmed && $activity->isFull()) {
    throw ValidationException::withMessages([
        'status' => "L'activité est complète, les {$activity->capacity} places sont prises.",
    ]);
}
```

`RegistrationPolicy` renvoie `$user->is_staff`, et rien sur les places.

Sinon, le secrétariat verrait la page 403 « Accès refusé » quand le bus est plein.

Le refus lève une `ValidationException` : la transaction est annulée, le message s'affiche sous le bouton « Confirmer ».

? Sans les deux if du contrôleur, lequel des deux tests échoue ?

Chaque exemple devient un test, et le test du cas refusé échoue tant que la règle manque.

1. Les tests d'abord, sans les deux `if`

```
$ php artisan test --filter=R1
 FAIL  Tests\Feature\RegistrationRulesTest
 ✓ R1 accepted: 11 confirmed out of 12 places, the next one is confirmed   0.44s
 × R1 refused: 12 confirmed out of 12 places, the next one stays pending   0.05s
 Session is missing expected key [errors].
 Tests:    1 failed, 1 passed (8 assertions)
```

2. Les deux `if` ajoutés, on relance

```
$ php artisan test --filter=R1
 PASS  Tests\Feature\RegistrationRulesTest
 ✓ R1 accepted: 11 confirmed out of 12 places, the next one is confirmed   0.45s
 ✓ R1 refused: 12 confirmed out of 12 places, the next one stays pending   0.04s
 Tests:    2 passed (11 assertions)
```

Les deux tests relisent la base à la fin : un refus qui affiche le bon message mais confirme quand même serait pris.

Le test accepté passe déjà, puisque l'application accepte tout. Seul le test refusé prouve que la règle existe.

? Avant l'entretien avec le client, que préparez-vous pour votre règle ?

Avec votre groupe, écrivez la R1 de votre sujet et ses deux exemples avant l'entretien.

AVEC VOTRE GROUPE, POUR VOTRE SUJET

1. Relisez la R1 de la fiche de votre sujet.
2. Classez-la : calcul, état, capacité ou unicité.
3. Nommez la donnée que l'application compte ou relit en base avant d'accepter.
4. Écrivez deux exemples chiffrés à la frontière : un accepté, un refusé.

À L'ENTRETIEN AVEC LE CLIENT

« Quelle est votre règle, et donnez-moi un cas où elle refuse ? »

O1 : un test par exemple de la règle, et le cas refusé montré à la présentation.

Le cas refusé deviendra un test, puis la démonstration de votre présentation.