2.04

TEMPS 2 · LE TAMAGOTCHI TROUÉ · SÉANCE 1

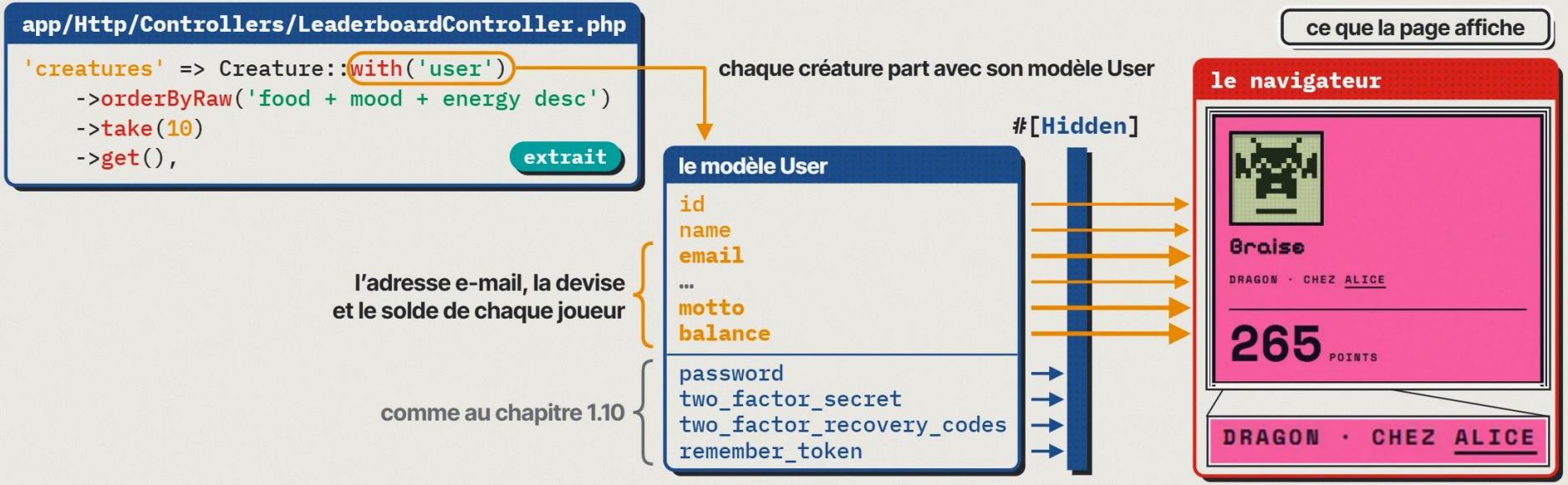
Les props publiques

Thibault Six · IFOSUP Wavre · 2026-2027



? Sur le podium du classement, que voit-on de chaque joueur ?

La page n'affiche que le nom ; le navigateur reçoit le `User`, sauf les colonnes de `#[Hidden]`.

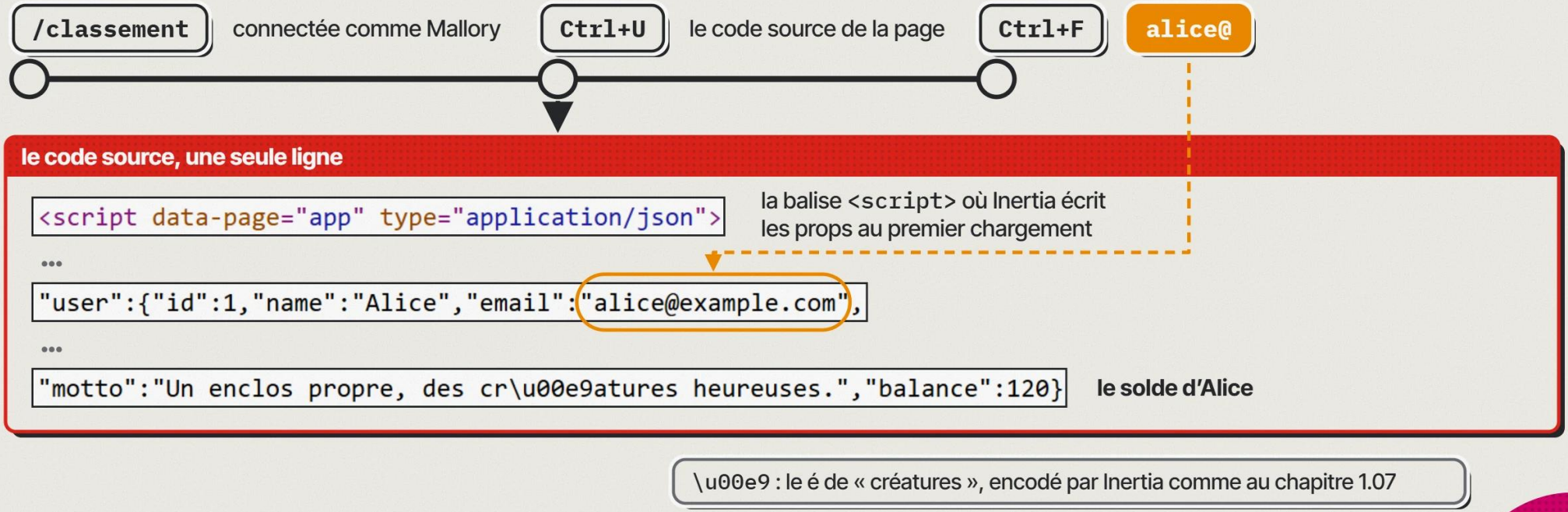


Ce que Mallory obtient : l'adresse e-mail et le solde de tous les joueurs du classement, sans attaque.

La page montre « chez Alice » ; les props emportent aussi son adresse et son solde. Le développeur n'a regardé que la page.

? La page n'affiche pas l'adresse d'Alice. Où Mallory peut-elle quand même la lire ?

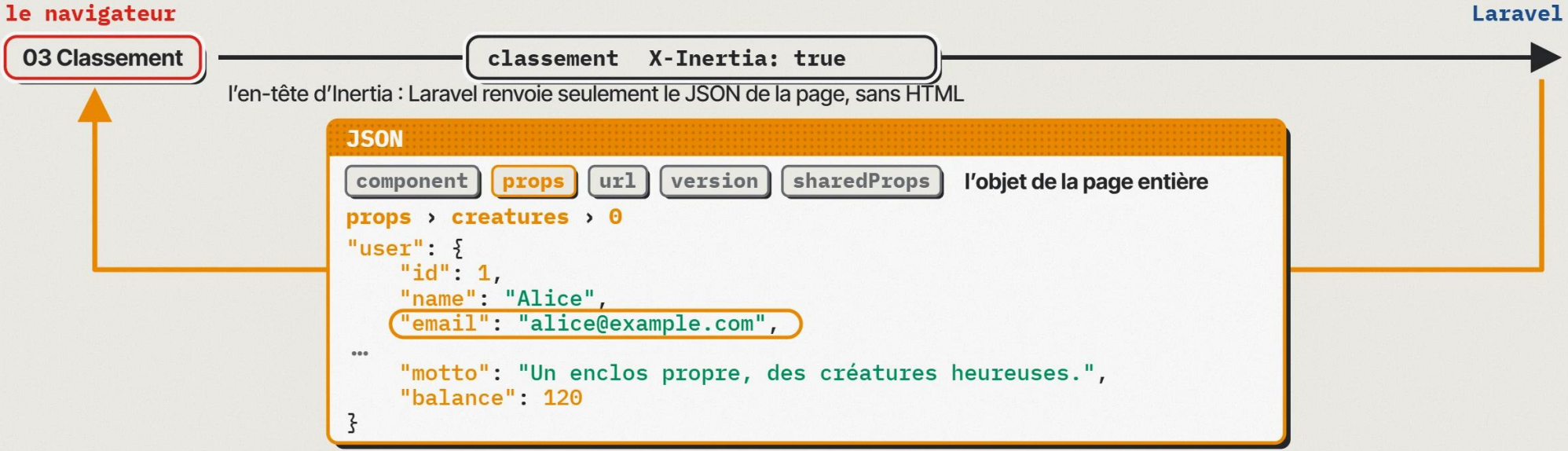
Au premier chargement, les props sont écrites dans le HTML : le code source les montre.



Trois morceaux de la même ligne du code source : la balise d'Inertia, puis l'adresse et le solde d'Alice.

? Mallory clique sur « 03 Classement ». Que reçoit son navigateur, cette fois ?

En naviguant, Laravel renvoie seulement le JSON de la page : l'adresse d'Alice y est encore.



Les cinq créatures du seeder sont au classement : les adresses d'Alice, de Bob et de Mallory s'y lisent, avec leurs soldes : `"balance": 120`, `0.3` et `5`.

Mallory relit la même donnée, envoyée cette fois seule, sans HTML : elle la déplie dans l'onglet « Aperçu ».

? La page n'utilise que quelques clés. Comment n'envoyer que celles-là, comme au chapitre 1.07 ?

Pour chaque créature, `map()` construit un tableau ; `only()` n'y garde que les clés lues.

app/Http/Controllers/LeaderboardController.php

```
->get()
->map(fn (Creature $creature) => [
    ...$creature->only('id', 'name', 'species', 'food', 'mood', 'energy'),
    'user' => $creature->user->only('id', 'name'),
]),
```

extrait

les trois points déballent le tableau d'only()

CE QUE LA PAGE LIT

Leaderboard.vue

creature

id · name · species · food · mood · energy

creature.user

id · name

la page se sert de `creature.user.id` pour surligner vos propres créatures

au chapitre 1.07 : `map()` pour la liste des projets, `only()` pour la fiche d'un projet

`with('user')` reste : il charge les joueurs en une requête, au lieu d'une requête par créature

Chaque clé du tableau répond à une clé que la page lit : ni plus, ni moins. Rien ne change dans la page.

? La page ne change pas. Comment prouver que l'adresse d'Alice ne part plus ?

Le test de la faille passe au vert, et du joueur, le JSON ne garde plus que `id` et `name`.

AVANT

```
  × the leaderboard only sends what it displays                    0.86s
  Expecting '[{"id":1,"user_id":1,"name":"…:50}}]' not to contain 'alice@example.com'.
```

APRÈS

```
   PASS  Tests\Feature\Flaws\Flaw04PublicPropsTest
  ✓ the leaderboard only sends what it displays

  Tests:    1 passed (12 assertions)
```

~~2 743~~

↓

1 036

caractères dans la balise `<script data-page>`

JSON — la nouvelle réponse classement

```
"user": {
    "id": 1,
    "name": "Alice"
}
```

Le test refuse `email` ou `balance`, et rien d'autre : relisez aussi le JSON.

La page ne prouve rien, puisqu'elle ne change pas : la preuve, c'est le test vert et le JSON relu.

? Qu'est-ce qui doit être vrai à la fin ?

À vous, quatre minutes : corrigez le classement, faites passer le test, recopiez la règle.

À FAIRE

1. Construisez le tableau avec `map()` et `only()`.
2. `php artisan test` : de 18 tests rouges à 17.
3. Rouvrez « 03 Classement » depuis « 01 Mon tamagotchi », et relisez la réponse classement dans l'onglet Réseau.

À VÉRIFIER

4 min

- [ ] Chaque objet `user` ne contient plus que `id` et `name`.
- [ ] La page affiche exactement le même podium.

DANS VOTRE CARNET DES FAILLES

Toute personne lit les props envoyées à son navigateur : les données des autres joueurs partent dans un tableau construit, avec les seuls champs nécessaires, jamais dans un modèle entier.

Trois étapes à gauche, deux vérifications à droite ; la règle va dans votre carnet des failles, mot pour mot.