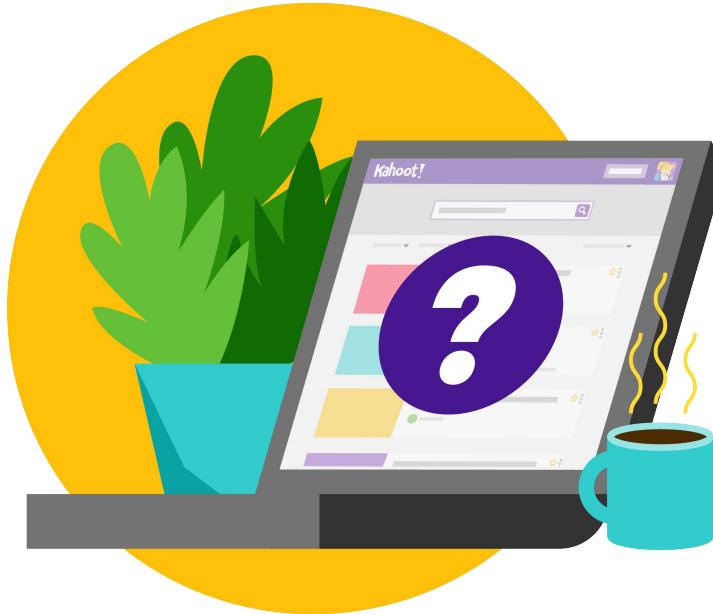


Qualité & Génie Logiciel - Kick-Off

Mathias COUSTÉ
16.01.2020

1

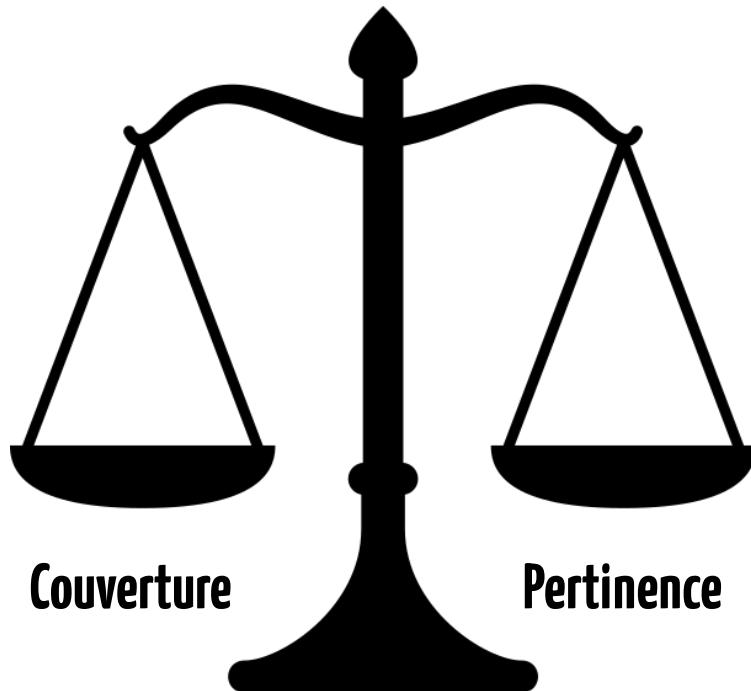


Go Kahoot!

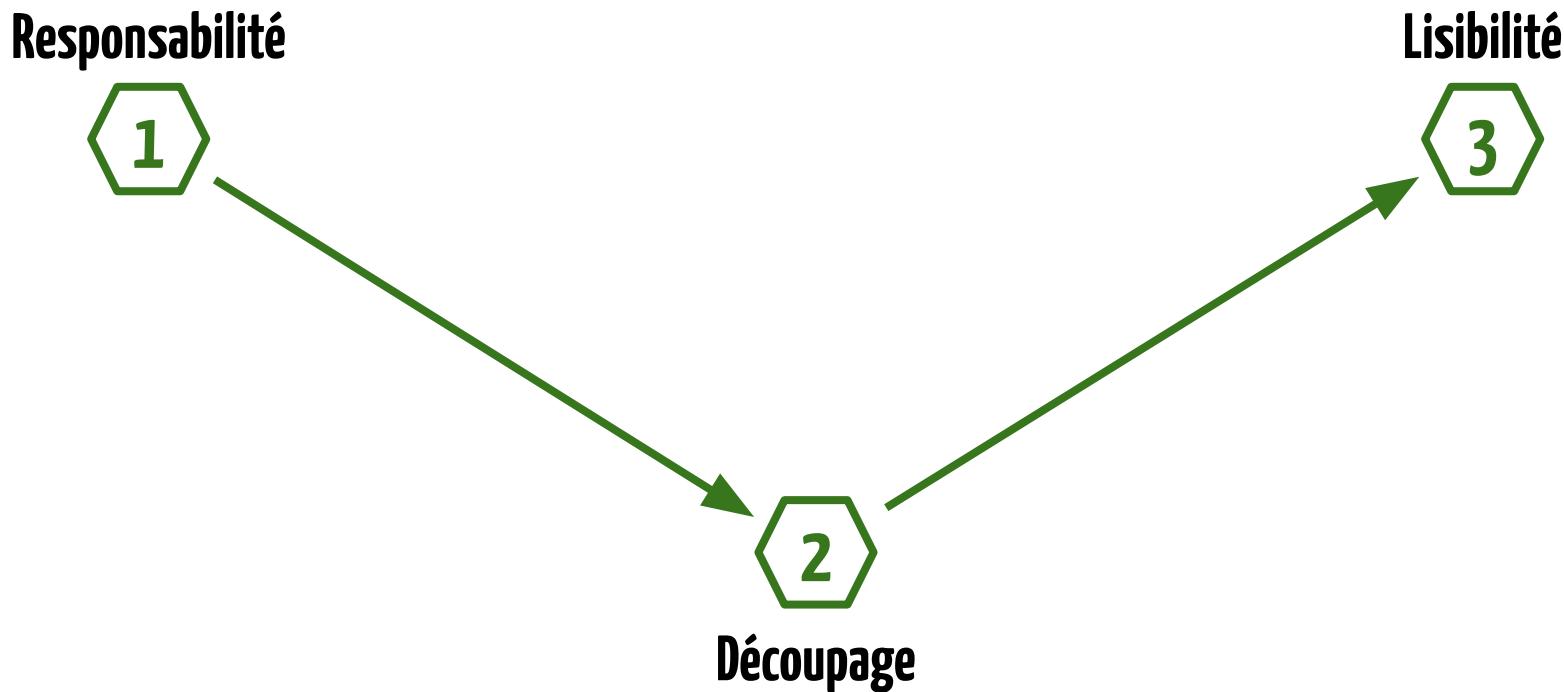
kahoot.it

Rappel PS5

TESTER



Rappel PS5



2



QGL
Quels objectifs?

JSON & interfaces



Objectifs:

- Quelles différences entre “utiliser” et “être utilisé” ?
- Qu'est ce qu'une interface ?
- Qu'est ce que le JSON ?

Qualité de code & métriques



Objectifs:

- Comment mesurer la qualité ?
- Comment interpréter mes mesures?

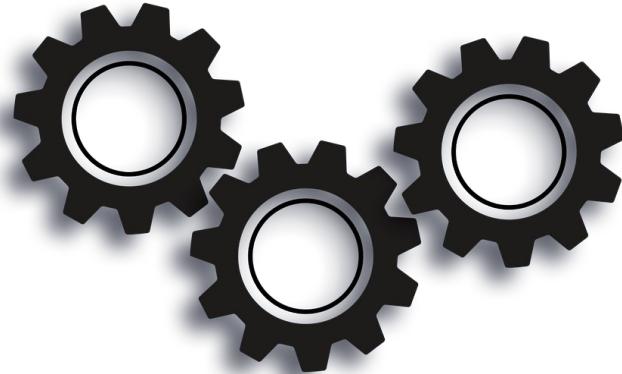
Git branch & Git flow



Objectifs:

- Maîtriser les branches Git
- Comment choisir son modèle de branche ?

Principes SOLID



Objectifs:

- Comprendre les principes SOLID
- Appliquer les principes SOLID

Refactoring



Objectifs:

- Quand et pourquoi restructurer mon code ?
- Comment restructurer mon code ?
- Quels outils dans mon IDE ?

Testing



Objectifs:

- Test de mutations
- Test d'intégration

Comprendre Maven



Objectifs:

- Comment lire un pom.xml ?
- D'où viennent les dépendances ?
- Comment découper un projet en modules ?

Automatisation avec Travis



Travis CI

Objectifs:

- Qu'est-ce que l'automatisation ?
- Que dois-je automatiser ?
- Comment faire fonctionner Travis à mon projet ?

3



Comment réussir QGL ?

Assiduité et constance



Ne restez pas bloqués !

Google

slack

Boulot & Fair-play

On bosse dur !

On favorise l'entraide à la guerre entre équipes !

On laisse personne derrière !

4



Quelle notation ?

Livrailles hebdomadaires

- Livraison automatisée (utilisation d'un tag)
- Tous les mardis
- 50 points / 200 points

Soutenance

- Présentation & démo de votre projet
- A mi-semestre
- 40 points / 200 points

Rapport de projet

- Livraison automatisée (utilisation d'un tag)
- A la fin du semestre
- 20 points / 200 points

Livraison finale

- Livraison automatisée (utilisation d'un tag)
- A la fin du semestre
- 40 points / 200 points

Partiel

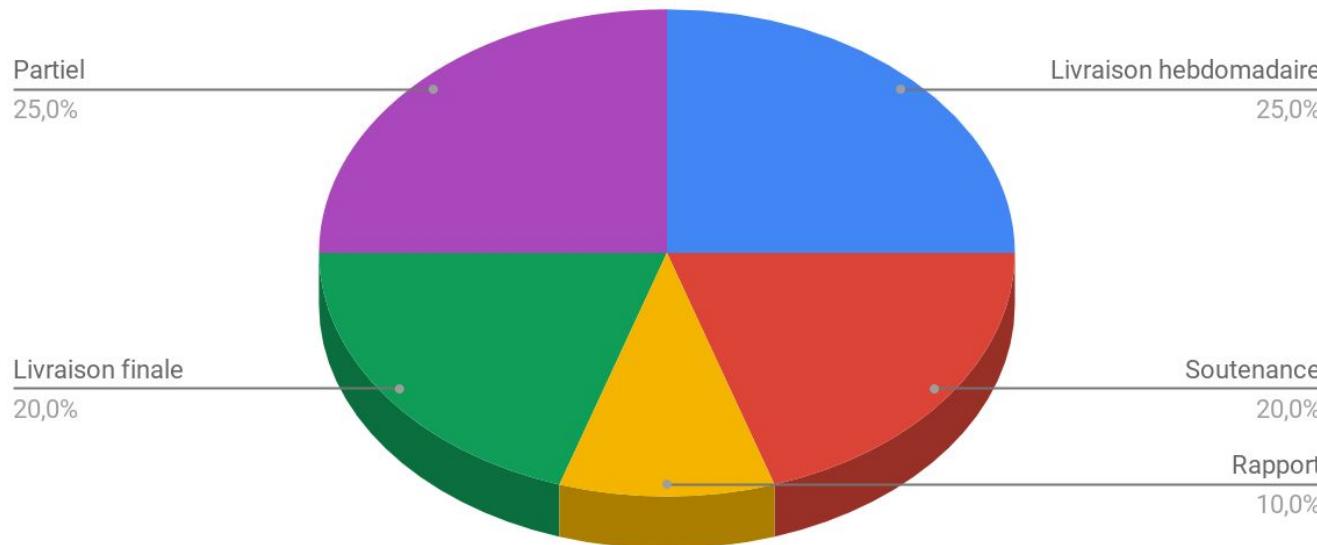
- Partiel surveillé
- A la fin du semestre
- 50 points / 200 points

Classement projet

- Classement des groupes
- +20 points bonus

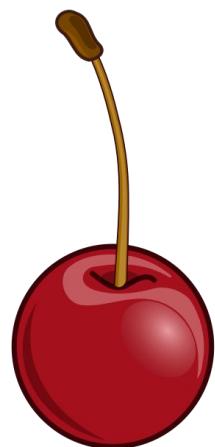
Pour résumer

Répartition de la notation



Championnat

+10%



Q&A



Project Kick-Off

Mathias COUSTÉ
16.01.2020

1



Projet fil-rouge

Projet fil-rouge



1 seul projet pour tout le semestre



Projet en groupe



2 Compétition inter-équipes (apprentis / non-apprentis)



Application des concepts enseignés

Par équipe



Le sujet



2



Règles du jeu

Modes de jeu

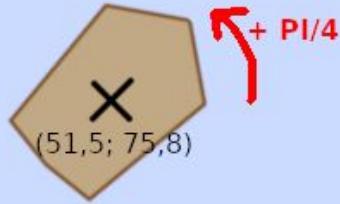


Régates



Batailles navales

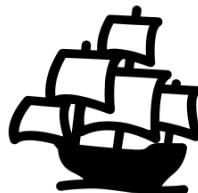
La mer



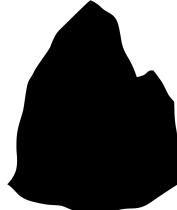
X
Origin(0;0)

- ★ Monde “infini”
- ★ Différente à chaque partie
- ★ Contient des éléments positionnés dans un repère orthonormé et orientés

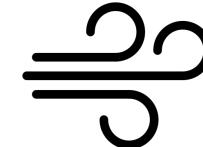
Bateaux



Récifs



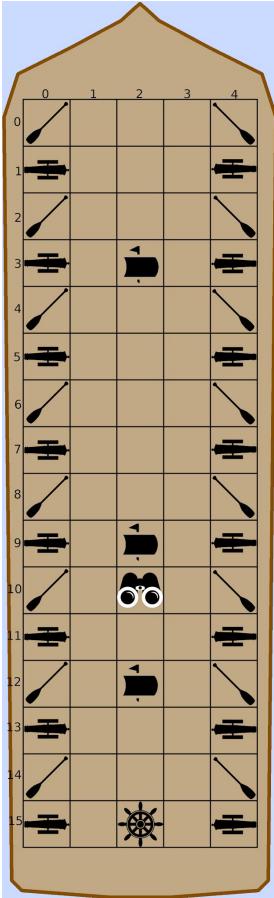
Vent



Courants



Le bateau



- ★ 1 seul par équipe
- ★ Positionné dans la mer
- ★ Représenté par une grille donnée
- ★ Contient des marins positionnés dans la grille
- ★ Contient des éléments positionnés dans la grille



Canon



Vigie



Rame



Gouvernail



Voile

Marins & actions



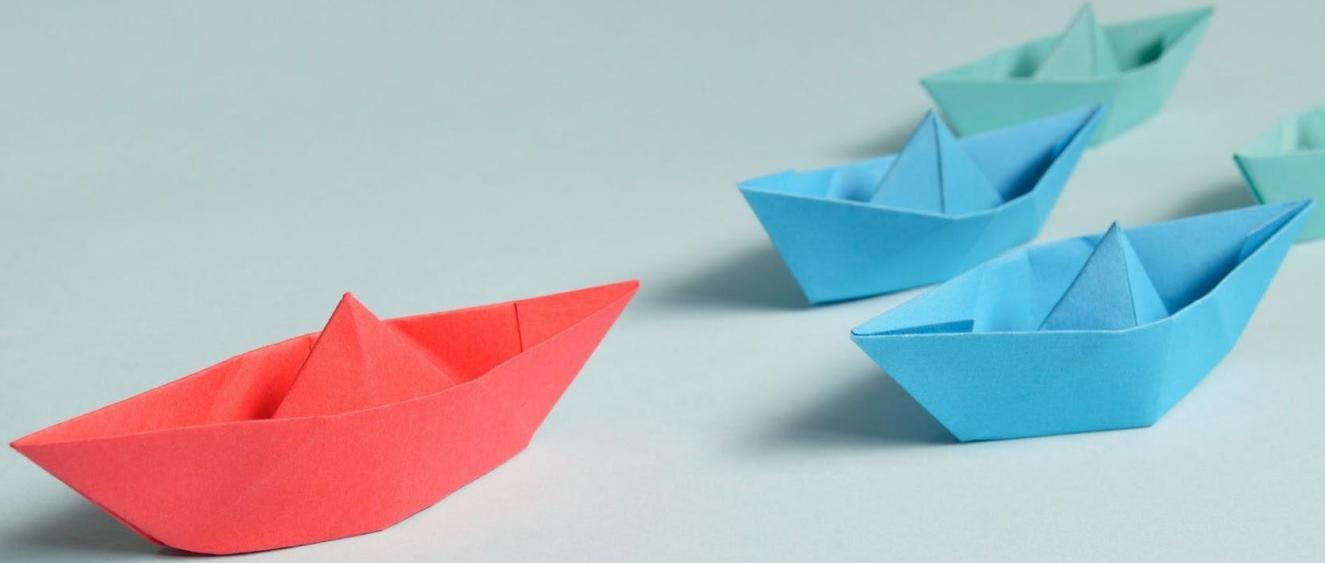
Two black silhouettes of human footprints, one above the other, pointing to the right.	Se déplacer	A black silhouette of a single-bladed oar or paddle.	Ramer
A black silhouette of a sailboat's hull and a single mast with a triangular sail.	Hisser la voile Affaler la voile	A black silhouette of a ship's steering wheel.	Orienter bateau
A black silhouette of a pair of binoculars.	Surveiller	A black silhouette of a naval gun or cannon barrel.	Viser Charger Tirer

Attention !

Specs mouvantes...

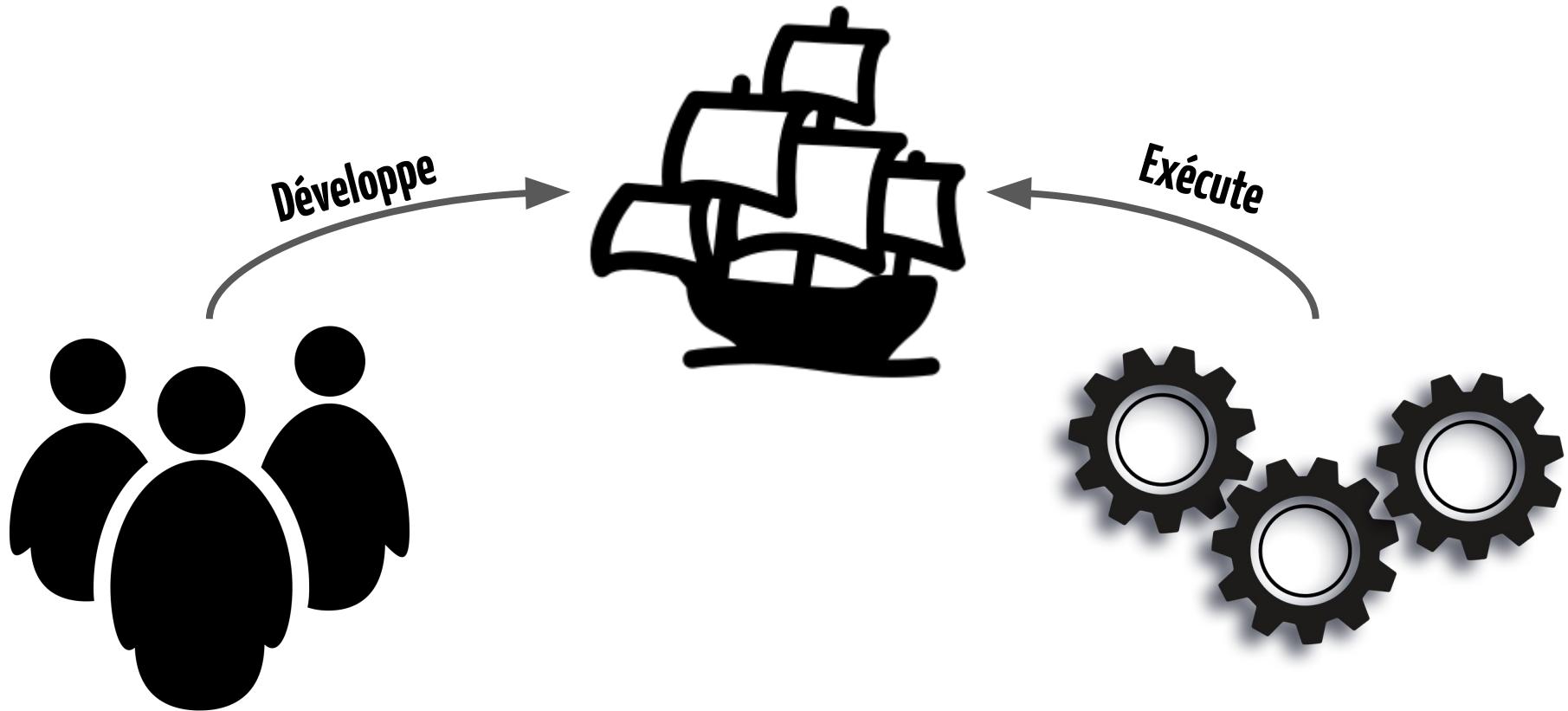
Q&A

3

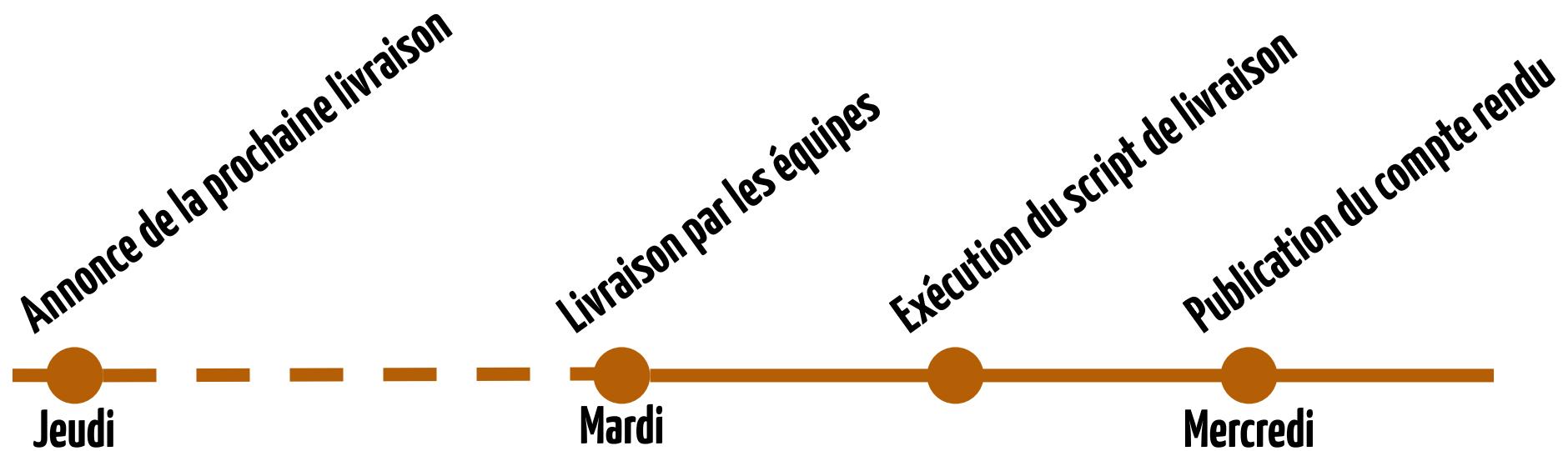


Championnat

Équipes, bateaux & arbitre



Déroulé d'une semaine type



Bonus de championnat



4

Premier rendu

Introduction au JSON

Mathias COUSTÉ
16.01.2020

1



Un format

Définition

“JavaScript Object Notation (JSON) est un format de données textuelles dérivé de la notation des objets du langage JavaScript.

Il permet de représenter de l'information structurée comme le permet XML par exemple.”

- Wikipedia

Un document JSON peut contenir:

- Un tableau : liste ordonnée de valeurs:
- Un objet: ensemble de paires “clé-valeur”

Une valeur JSON peut contenir:

- un tableau
- un objet
- un booléen (true/false)
- une chaîne de caractères
- un nombre
- *null*

Syntaxe du JSON

Tableau:

[<valeur 1>, <valeur 2>, <...>]

Objet:

{“<clef 1>”: <valeur 1>, “<clé 2>”: <valeur 2>}

Booléen:

true / false

Chaîne de caractères:

“<contenu de la chaîne de caractère>”

Nombre:

< - si négatif><partie entière>.<décimales>

Null:

null

2



Quelques exemples

Exemple 1

```
{           ----- Objet
  "goal": {           ----- Object
    "mode": "FRIGATE", ----- Chaîne de caractères
    "checkpoints": [ ----- Tableau
      {
        "done": false, ----- Objet
        "position": {           ----- Booléen
          "x": -150, ----- Nombre
          "y": 0,           ----- Nombre
          "orientation": 0.15 ----- Nombre
        },
        "shape": {           ----- Objet
          "type": "circle", ----- Chaîne de caractères
          "radius": 50       ----- Nombre
        }
      }
    ]
  }
}
```

Exemple 2

```
[  
  {  
    "sailorId": 0,           Tableau  
    "type": "MOVING",       Object  
    "xdistance": 5,          Nombre  
    "ydistance": 0           Chaîne de caractères  
  },  
  {  
    "sailorId": 0,           Objet  
    "type": "TURN",          Nombre  
    "rotation": 0            Chaîne de caractères  
  },  
  {  
    "sailorId": 2,           Objet  
    "type": "OAR"             Nombre  
  }]  
]
```

3



JSON et Java

JSON & Java

#1 - importer une bibliothèque

```
<!-- JSON serialization -->
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>2.10.1</version>
    <scope>provided</scope>
</dependency>
```

JSON & Java

#2 - créer votre modèle

Wind.java:

```
public class Wind {  
    double strength;  
  
    double orientation;  
  
}
```

JSON & Java

#3 - prenez un ObjectMapper, un JSON valide, et votre modèle

Mon JSON:

```
{  
    "strength": 125.6,  
    "orientation": 94.48  
}
```

Wind.java:

```
public class Wind {  
    double strength;  
    double orientation;  
}
```

MonParser.java:

```
ObjectMapper OBJECT_MAPPER = new ObjectMapper()
```

JSON & Java

#4 - donnez votre JSON à l'ObjectMapper

```
String monJSON = "{ \"strength\": 125.6, \"orientation\": 94.48 }";
```

```
Wind wind = OBJECT_MAPPER.readValue(monJSON, Wind.class)
```

JSON & Java



JSON & Java

/!\ - Renseignez vous sur:

1. ObjectMapper::readValue
2. ObjectMapper::writeValueAsString
3. ObjectMapper::registerModule
4. ObjectMapper::configure
5. @JsonIgnore
6. @JsonInclude

It's dangerous to go alone!
Take this.

<https://www.baeldung.com/jackson-object-mapper-tutorial>



Q&A