Projects

Remise Zippée du projet sur http://foad2.unicaen.fr/moodle/course/view.php?id=20911

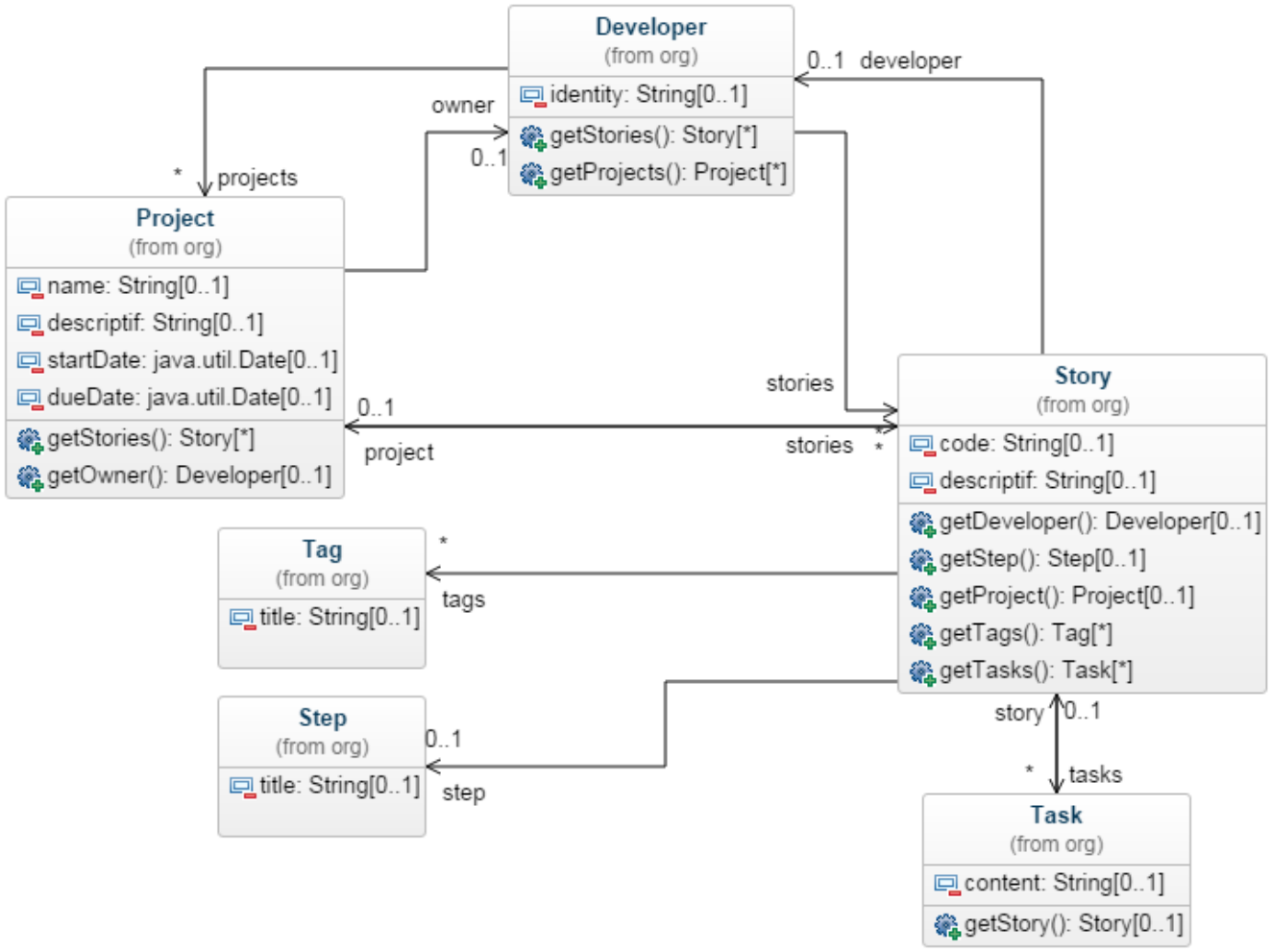
Attention au nommage **prenom.nom** !

-- Contexte

Vous travaillez sur un outil permettant de gérer des projets.

Voici les principales caractéristiques du système d'information :

- Chaque projet [**project**] possède un nom, un descriptif, une date de début et de fin, et un propriétaire (owner, qui est un développeur)
- L'équipe est constituée d'un ensemble de développeurs [**developer**].
- Chaque User story [**story**] a un code et un descriptif, et appartient à un projet.
- Il est possible de lui apposer des tags [**tags**], composés d'une couleur et d'un label.
- Elle peut être affectée à un développeur[**dev**] (qui a juste une identité).
- Elle peut contenir une liste de tâches [**tasks**], à réaliser ou réalisées.



-- Eléments à implémenter

- **Strict respect des normes, nommages et consignes** données ⇒ **1 point**
- Usage de **Browserify** pour créer un unique bundle ⇒ **1 point**

-- Squelette de l'application, routage (4 points)

//TODO 2.1.1

Le dossier root de votre application devra être de la forme : **projects-prenom.nom**

Créer la structure suivante :

Emplacement	**Fichier**	**Rôle**
/	index.html	Fichier principal
/app		Fichiers js
	app.js	Fichier principal de l'application (module)
	config.js	Fichier de routage et configuration

Emplacement	Fichier	Rôle
/app/controllers		contrôleurs
	myProjects.js	Définit le contrôleur **myProjectsController**
	project.js	Définit le contrôleur **projectController**
	story.js	Définit le contrôleur **storyController**
/app/services		services
/app/views		templates et vues HTML
/public/		Feuille(s) de styles + fichiers js externes

//TODO 2.1.2

Implémenter le routage suivant :

URL	Composante	Valeur
/home	Description	Affiche la liste des projets de l'utilisateur, et les projets auxquels il participe
	template	app/views/myProjects.html
	contrôleur	myProjectsController
	alias	myProjectsCtrl
/project/:_id	Description	Affiche le projet correspondant à **_id**
	template	app/views/project.html
	contrôleur	projectController
	alias	projectCtrl
/story/:_id	Description	Affiche la user story correspondant à **_id**
	template	app/views/story.html
	contrôleur	storyController
	alias	storyCtrl
/home	Route par défaut	

-- Service (3 points)

//TODO 2.2

Le service **daoService** simule la connexion à un web service (en réalité effectué à un web service connecté à MongoDB, ce qui explique la structure des données) ; il est défini de la façon suivante :

projects, **stories**, **tags** et **devs** seront définis “en dur” à l'intérieur du service dans l'équivalent de variables privées :

```
    var _projects={"_embedded":[
    {"_id":{"$oid":"58d038b705d0b0b35f9764da"},"name":"Open-beer","descriptif":"A
free, public database and API for beer information.","startDate":"March 20, 2017
21:16:55"},
    {"_id":{"$oid":"58d038b705d0b0b35f9764d9"},"name":"Boards
analysis","descriptif":"AngularJS application + REST API","startDate":"March 20,
2017 21:16:55",
        "owner":{"identity":"Rod
Johnson","_id":{"$oid":"58d038b705d0b0b35f9764d5"}}},
    {"_id":{"$oid":"58d038b705d0b0b35f9764d8"},"name":"Boards
admin","descriptif":"Administration interface for Boards app with
javaFX","startDate":"March 20, 2017 21:16:55",
        "owner":{"identity":"Linus
Torvalds","_id":{"$oid":"58d038b705d0b0b35f9764d7"}}}
```

```
    ],"_id":"Project","_returned":3};
    var _stories={"_embedded":[
{"_id":{"$oid":"58d038b705d0b0b35f9764df"},"code":"Beer1","descriptif":"affichage
de la liste des bières /beers (L'affichage de la bière n'affiche pas le brasseur
associé)",
        "project":{"name":"Open-beer","descriptif":"A free, public database and API
for beer information.","startDate":"March 20, 2017
21:16:55","_id":{"$oid":"58d038b705d0b0b35f9764da"}},
        "developer":{"identity":"Linus
Torvalds","_id":{"$oid":"58d038b705d0b0b35f9764d7"}},
"tags":[{"title":"bug","color":"#EE0701","_id":{"$oid":"58d038b705d0b0b35f9764e0"}}
],
        "tasks":[]},
    {"_id":{"$oid":"58d038b705d0b0b35f9764de"},"code":"Dev1","descriptif":"En tant
que développeur, je peux consulter mes projets",
        "project":{"name":"Boards analysis","descriptif":"AngularJS application +
REST API","startDate":"March 20, 2017 21:16:55","owner":{"identity":"Rod
Johnson","_id":{"$oid":"58d038b705d0b0b35f9764d5"}},"_id":{"$oid":"58d038b705d0b0b3
5f9764d9"}},
        "developer":{"identity":"Linus
Torvalds","_id":{"$oid":"58d038b705d0b0b35f9764d7"}},
"tags":[{"title":"duplicate","color":"#CCCCCC","_id":{"$oid":"58d038b705d0b0b35f976
4e1"}}],
        "tasks":[{"content":"Descriptif cas
d'utilisation","closed":true},{"content":"Interfaces","closed":true}]},
    {"_id":{"$oid":"58d038b705d0b0b35f9764dd"},"code":"E140","descriptif":"En tant
que créateur, je souhaite gérer les utilisateurs [methods]",
        "project":{"name":"Boards admin","descriptif":"Administration interface for
Boards app with javaFX","startDate":"March 20, 2017 21:16:55",
        "owner":{"identity":"Linus
Torvalds","_id":{"$oid":"58d038b705d0b0b35f9764d7"}},"_id":{"$oid":"58d038b705d0b0b
35f9764d8"}},
        "tags":[],"tasks":[]},
    {"_id":{"$oid":"58d038b705d0b0b35f9764dc"},"code":"E120","descriptif":"En tant
que créateur, je veux créer / Modifier des quiz [methods]",
        "project":{"name":"Boards admin","descriptif":"Administration interface for
Boards app with javaFX","startDate":"March 20, 2017 21:16:55",
        "owner":{"identity":"Linus
Torvalds","_id":{"$oid":"58d038b705d0b0b35f9764d7"}},"_id":{"$oid":"58d038b705d0b0b
35f9764d8"}},
        "tags":[],
        "tasks":[{"content":"Implémentations méthodes
REST","closed":true},{"content":"Implémentations méthodes REST (suite et
fin)","closed":true}]},
    {"_id":{"$oid":"58d038b705d0b0b35f9764db"},"code":"B22","descriptif":"En tant
que créateur, je veux ajouter et gérer les réponses d'une question [methods]",
        "project":{"name":"Boards admin","descriptif":"Administration interface for
Boards app with javaFX","startDate":"March 20, 2017 21:16:55",
        "owner":{"identity":"Linus
Torvalds","_id":{"$oid":"58d038b705d0b0b35f9764d7"}},"_id":{"$oid":"58d038b705d0b0b
35f9764d8"}},
        "developer":{"identity":"Rod
Johnson","_id":{"$oid":"58d038b705d0b0b35f9764d5"}},
        "tags":[],
        "tasks":[]}
    ],"_id":"Story","_returned":5};
```

```
    var _devs={"_embedded":[
    {"_id":{"$oid":"58d038b705d0b0b35f9764d7"},"identity":"Linus Torvalds"},
    {"_id":{"$oid":"58d038b705d0b0b35f9764d6"},"identity":"James Gosling"},
    {"_id":{"$oid":"58d038b705d0b0b35f9764d5"},"identity":"Rod Johnson"}
    ],"_id":"Developer","_returned":3};
    var _tags={"_embedded":[
    {"_id":{"$oid":"58d038b705d0b0b35f9764e6"},"title":"wont
fix","color":"#FFFFFF"},
{"_id":{"$oid":"58d038b705d0b0b35f9764e5"},"title":"question","color":"#CC317C"},
{"_id":{"$oid":"58d038b705d0b0b35f9764e4"},"title":"invalid","color":"#E6E6E6"},
    {"_id":{"$oid":"58d038b705d0b0b35f9764e3"},"title":"help
wanted","color":"#128A0C"},
{"_id":{"$oid":"58d038b705d0b0b35f9764e2"},"title":"enhancement","color":"#84B6EB"}
,
{"_id":{"$oid":"58d038b705d0b0b35f9764e1"},"title":"duplicate","color":"#CCCCCC"},
    {"_id":{"$oid":"58d038b705d0b0b35f9764e0"},"title":"bug","color":"#EE0701"}
    ],"_id":"Tag","_returned":7};
```

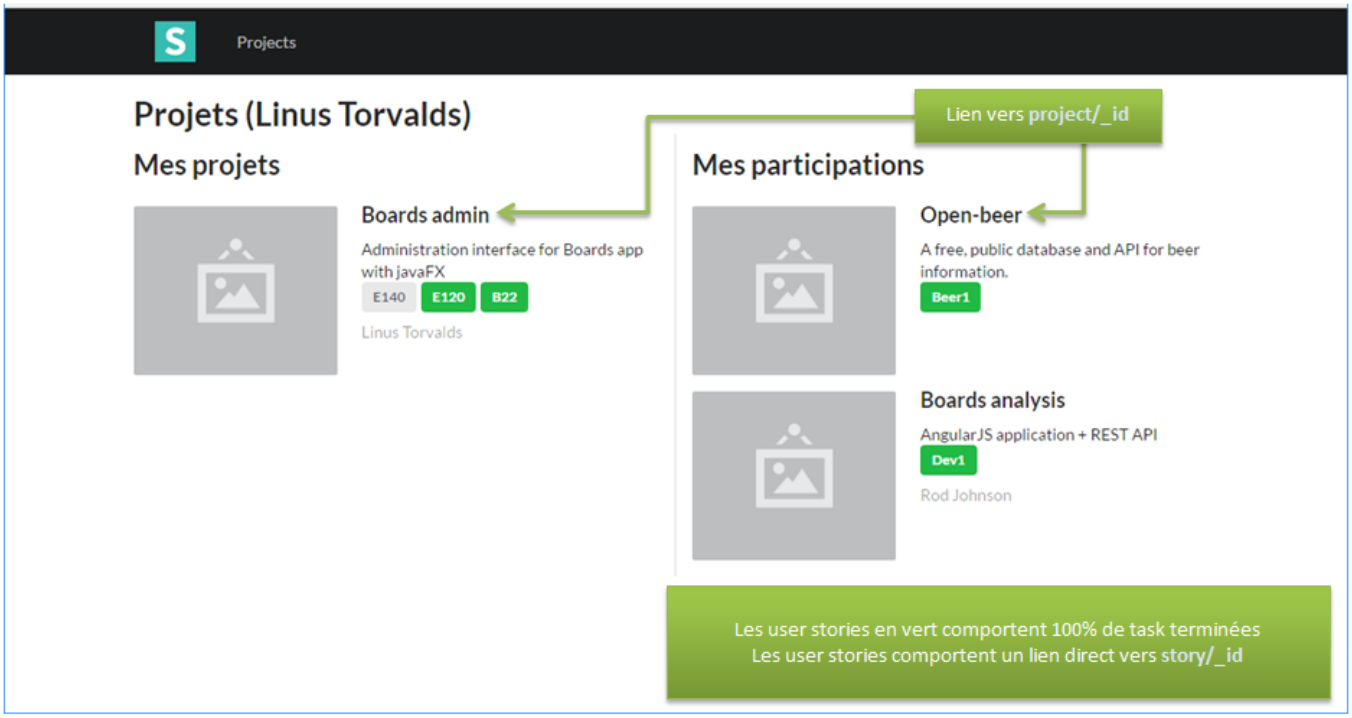
Service	**daoService.js** (daoService)
Variables privées	_projects Tableau des Projets
	_stories Tableau des user stories disponibles
	_devs Tableau des développeurs
	_tags Tableau des tags disponibles
Méthodes publiques	getProjects() retourne le tableau des projets
	getStories() retourne les **user stories**
	getTags() retourne les **tags**
	getDevs() Retourne les **devs**
	getMyProjects(_idDev) Retourne les projets dont un développeur est propriétaire
	getMyParticipations(_idDev) Retourne les projets auxquels un développeur participe (au travers des stories)
	getProjectStories(_idProject) Retourne les user stories d'un projet

Créez **daoService**, implémentez les méthodes publiques.
Il faudra ensuite injecter **daoService** aux 2 contrôleurs **myProjectsController** et **projectController**

--Url /home (3 points)

//TODO 2.3

Affiche la liste des projets de l'utilisateur (owner du project) + les projets auquel l'utilisateur participe .



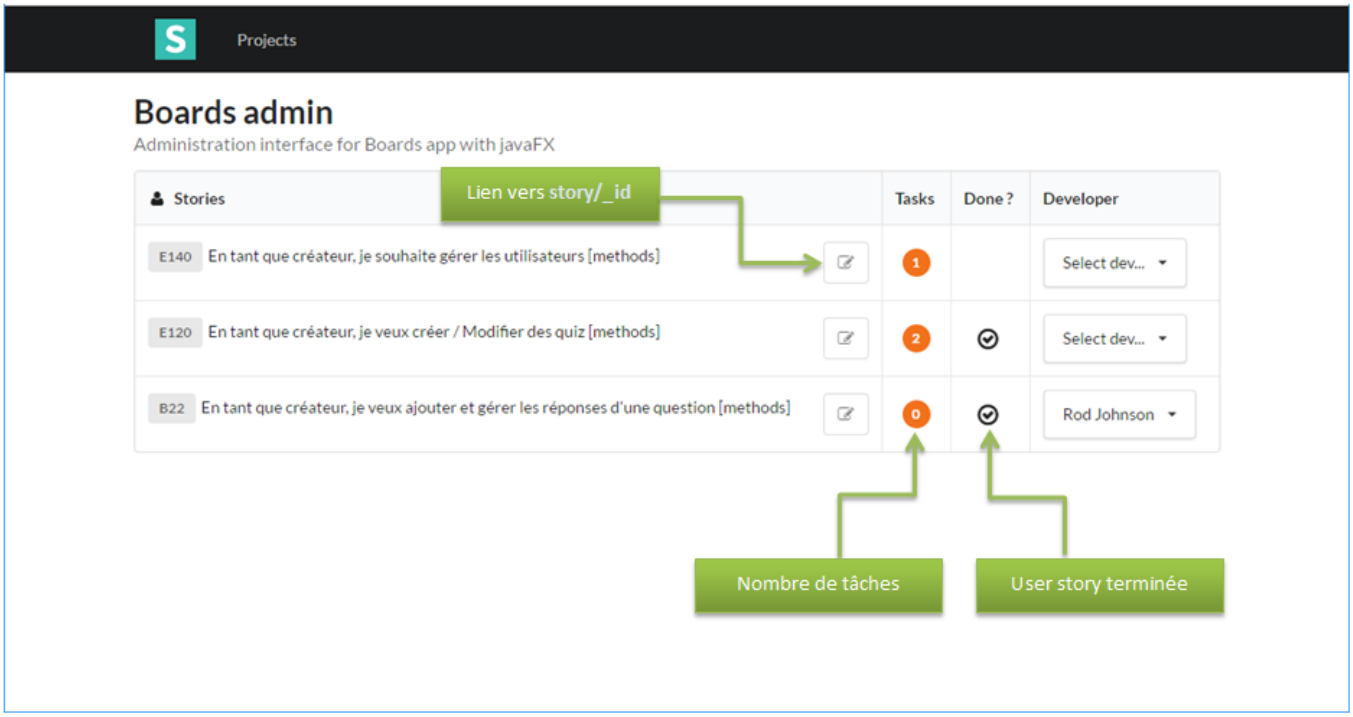
Créer le template **app/views/myProjects.html**, associé au contrôleur **/app/controllers/myProjectsController.js**

Controller	**myProjectsController.js** (myProjectsController)
Variables publiques	projectsOwner Tableau des projets possédés
	projectsWorker Tableau des projets auxquels le développeur participe
Méthode privée	initUser() Initialise l'utilisateur connecté à partir du premier développeur trouvé dans les devs (ne sert pas à grand chose, mais à faire quand même)

--Url /project/_id (3 points)

//TODO 2.4

Affiche le projet dont l'identifiant ($oid) est passé en paramètre.



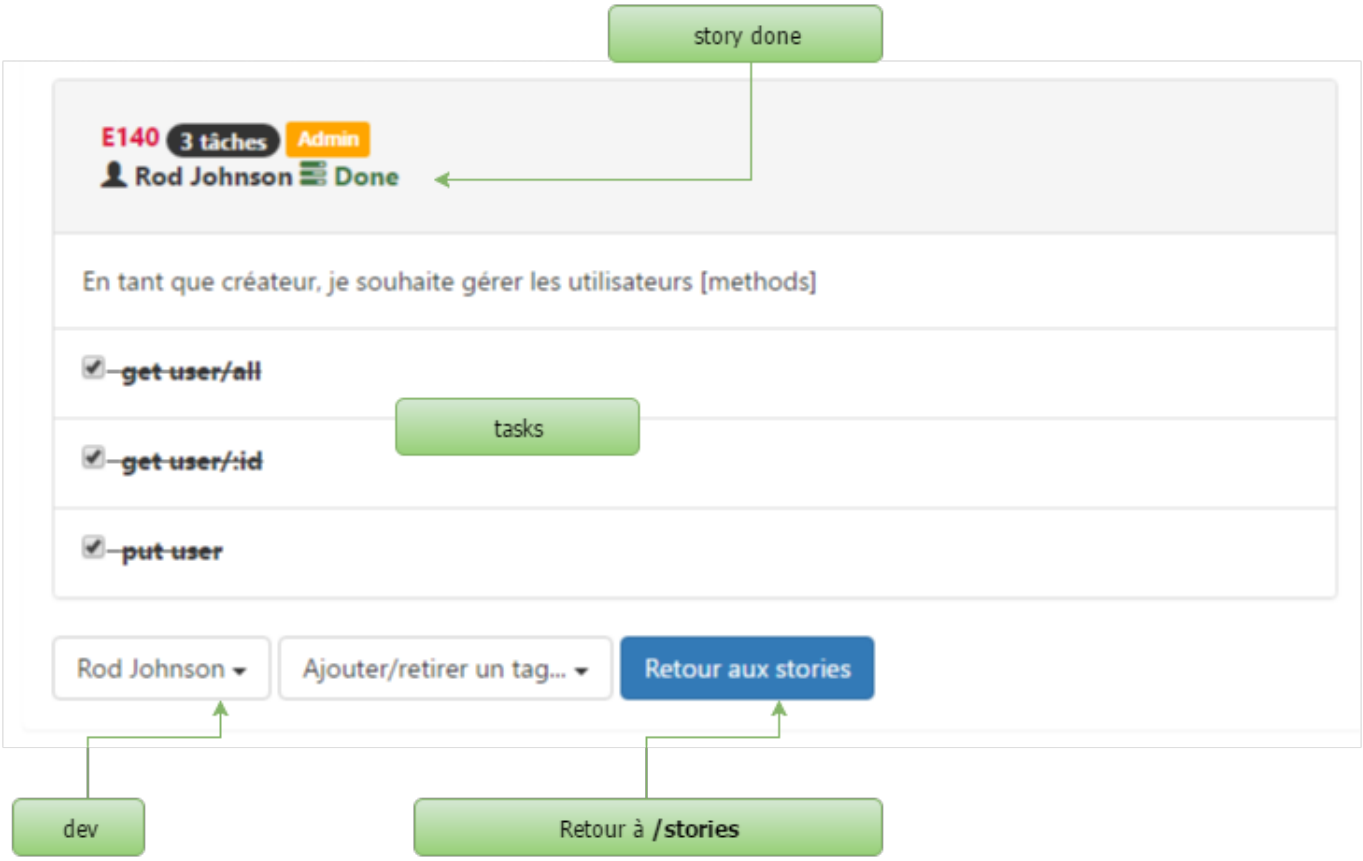
Créer le template **app/views/project.html**, associé au contrôleur **/app/controllers/projectController.js**

Controller	**projectController.js** (projectController)
Variables publiques	project Projet actif
	devs Tableau de tous les développeurs
Méthodes publiques	setDev(dev,story) Affecte le développeur dev à la user story

-- Url /story/:_id (5 points)

//TODO 2.5

Affiche la **story** correspondant au **code** passé dans l'url.



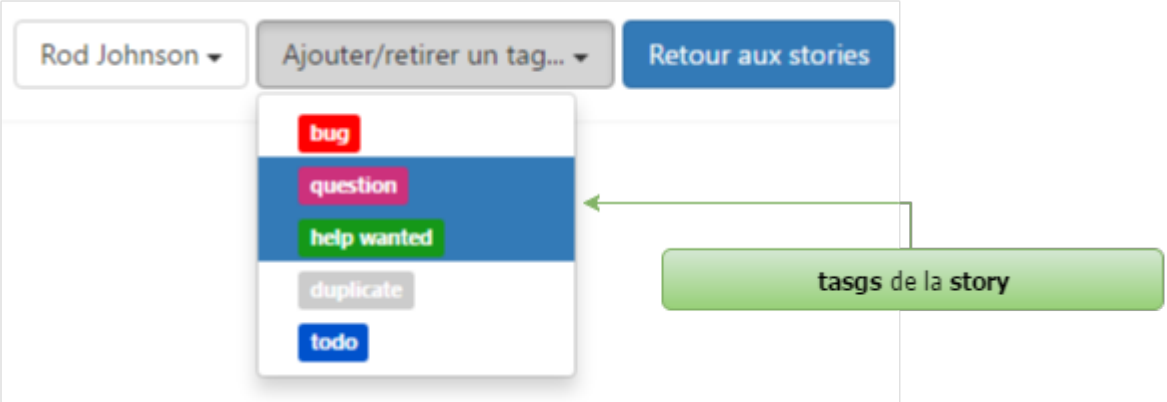
Comportement de l'interface :

- Une task **done** est barrée
- Les cases à cocher permettent de faire passer une tâche de non réalisée (done=false) à réalisée (done=true) et inversement

Menu Dev :

- Le dev actif est sélectionné (classe bs **active**)
- Le clic sur un autre dev change l'affectation
- L'élément **Retirer l'affectation** passe le dev de la story à **null**

Menu Tags :



- Les tags présents dans la story sont sélectionnés (classe bs **active**)
- Le clic sur un tag l'ajoute ou le retire à la liste des tags de la story

Créer le template **app/views/story.html**, associé au contrôleur **/app/controllers/story.js**

Controller	**story.js** (storyController)
Variables privées	story user story à afficher
	devs Tableau de tous les développeurs
	tags Tableau de tous les tags
Méthodes publiques	toggleDone(task) Bascule de vrai à faux et inversement le membre **done** de la tâche **task** passée en paramètre
	setDev(dev) Assigne le développeur **dev** à la **story**
	indexOfTag(tag) Retourne l'index du **tag** passé en paramètre dans la liste des **tags** de la **story** (-1 s'il n'est pas trouvé)
	toggleTag(tag) Ajoute ou retire le **tag** passé en paramètre de la liste des **tags** de la story

-- A poursuivre...

//TODO 2.6

Idées de fonctionnalités supplémentaires à implémenter :

Dans la page story/:_id :

- Permettre l'ajout/modification/suppression de tasks
- Permettre l'ajout/modification/suppression de nouveaux tags

-- Ressources

HTML

```
<!DOCTYPE html>
<html data-ng-cloak>
```

```
<head>
<meta charset="UTF-8">
<title></title>
    <base href="/yourSite/"/>
    <link rel="stylesheet" type="text/css"
href="https://cdnjs.cloudflare.com/ajax/libs/semantic-ui/2.2.7/semantic.min.css">
</head>
<body>

    <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
    <script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.6.3/angular.min.js"></script>
    <script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.6.3/angular-route.min.js"></s
cript>
    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/semantic-ui/2.2.7/semantic.min.js"></sc
ript>
</body>
</html>
```

.htaccess pour Apache

```
<IfModule mod_rewrite.c>
    Options +FollowSymlinks
    RewriteEngine On
    RewriteBase /yourSite/
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteCond %{REQUEST_URI} !.*\.(css|js|html|png|jpg|jpeg|gif|txt|ttf|woff)
    RewriteRule (.*) index.html [L]
</IfModule>
```

Semantic-UI : composants utilisés

- view::item
- collections::grid
- collections::table