

TD n°4

	Tous les exercices utilisent Bootstrap pour la partie CSS. <code>ember install ember-bootstrap</code>
	Pour passer à la v4 de bootstrap (sass requis) : <code>npm install --save-dev ember-cli-sass</code> <code>ember generate ember-bootstrap --bootstrap-version=4</code>
	pour les fonts : <code>ember install ember-font-awesome</code>

- Projet **boards**
- Application gestion de projets SCRUM

Objectifs

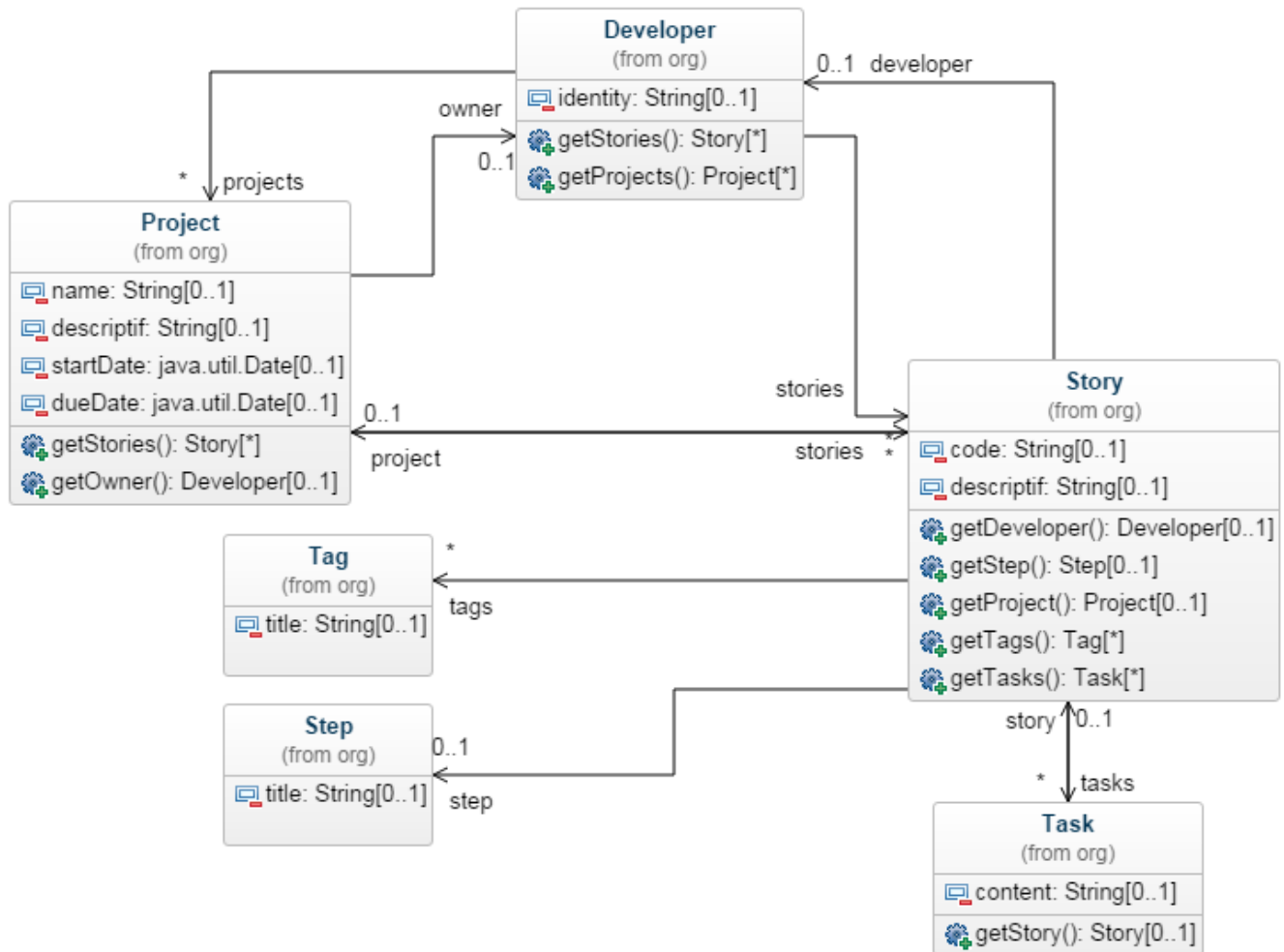
1. Se connecter à un service web externe
2. Créer des composants
3. Manipuler les models

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.



Création de la base de données

Démarrer mongoDb :

```
mongod
```

Démarrer le serveur **restheart** :

```
java -jar restheart.jar
```

Création du projet, intégration des composants

Créer le projet **boards**

Ajouter le plugin ember-cli-uuid pour la génération d'uuid côté client :

```
ember install ember-cli-uuid
```

Adapter & serializer

Générer l'adapter pour l'application :

```
ember g adapter application
```

Modifier l'adapter comme ci-dessous :

```
import DS from 'ember-data';
import { pluralize } from 'ember-inflector';

var Adapter=DS.RESTAdapter.extend({
  ajaxOptions: function(url, type, options) {
    var hash = this._super(url, type, options);
    if (type == 'POST' || type=='PUT') {
      hash.dataType = 'text';
    }
    return hash;
  },
  host: 'http://127.0.0.1:8080',
  namespace: 'projects',
  urlForDeleteRecord(id, modelName) {
    modelName=pluralize(modelName);
    return
this.get('host')+'/' +this.get('namespace')+'`/${modelName}/*?filter={_id:'${id}'}`;
  }
});

export default Adapter;
```

Générer le serializer pour l'application :

```
ember g serializer application
```

Modifier le serializer comme ci-dessous :

```
import DS from 'ember-data';

export default DS.JSONSerializer.extend({
  primaryKey: '_id',
  isNewSerializerAPI: true,
  extractId: function (modelClass, resourceHash) {
    if(resourceHash._id)
      return (resourceHash._id.$oid || resourceHash._id);
  },
  normalizeResponse(store, primaryModelClass, payload, id, requestType) {
    if(requestType=='createRecord')
      return this._super(store, primaryModelClass, {}, id, requestType);
    if(requestType=='updateRecord')
      return this._super(store, primaryModelClass, {}, id, requestType);
    if(requestType=='deleteRecord')
      return this._super(store, primaryModelClass, null, id, requestType);
  }
});
```

```
    if (payload._embedded)
      return this._super(store, primaryModelClass, payload._embedded, id,
requestType);
    return this._super(store, primaryModelClass, payload, id, requestType);
  },
  serializeId (snapshot, json) {
    let id = snapshot.id;
    json['_id'] = id;
  }
});
```

From:

<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**

Permanent link:

<http://slamwiki2.kobject.net/richclient/emberjs/td4?rev=1519667923>

Last update: **2019/08/31 14:38**

