

# *Candy Store Saga*

Créez votre site marchand avec



# INTRODUCTION

Le but de ce TP consiste à créer à l'aide d'AngularJS Candy Store Saga, un site de vente en ligne de confiseries.

Lien utile : <https://docs.angularjs.org/tutorial>

# INTRODUCTION

Le squelette de l'application est fourni, vous complétez le code html du fichier `main.html` et le code Javascript des fichiers *main.js*, *app.js* et *dataservice.js*. Vous n'aurez pas besoin de modifier les autres fichiers.

Le squelette est structuré de la façon suivante :

**app/images** dossier contenant les images

**app/scripts** dossier contenant les fichiers Javascript

**app/styles** dossier contenant les feuilles de style CSS

**app/views** dossiers contenant les différentes vues HTML

# MISE EN ROUTE

De la même manière que le TP NodeJS, ajoutez l'url du dossier node dans la variable d'environnement PATH.

Si l'url du TP NodeJS est toujours présente, supprimez la avant d'insérer la nouvelle.

# MISE EN ROUTE

Placez-vous dans le dossier CandyStore et exécutez la commande *grunt serve*

Le site que vous allez compléter s'ouvre alors dans votre navigateur (cela peut prendre quelques secondes)

# TRAVAIL A RÉALISER

## Partie 1 - Le contrôleur

1. Créez un contrôleur **MainCtrl** dans le fichier *main.js*.
2. Attachez la vue *main.html* avec le contrôleur à la route “/” dans module **candyStoreApp** avec l’alias *main*.
3. Ajoutez une directive pour que tous les produits s’affichent dans le *product container*

# TRAVAIL A RÉALISER

## Partie 1 - Le contrôleur

4. Dans le fichier `dataservice.js`, récupérez les données du fichier <https://raw.githubusercontent.com/mspegagne/AngularSFFS/master/data.json> grâce au service `$http`.
5. Affichez le nom du produit dans la balise `<h3>` et son prix dans la balise `<em>`  
*hint : Observez le contenu du fichier json*
6. Le prix de certaines confiseries ne sont pas affichés correctement, utilisez un filtre pour afficher le prix en euros avec les centimes.

# TRAVAIL A RÉALISER

## Partie 1 - Des détails sur votre confiserie !

Un produit possède, en autres, les propriétés suivantes :

- **canPurchase** qui est un booléen indiquant si le produit peut être acheté
  - **soldOut** qui est un booléen indiquant si le produit est en rupture de stock
1. Utiliser une directive qui permette d'afficher le bouton "Acheter" seulement si la propriété **canPurchase** d'un produit est vraie.
  2. Lorsque qu'un produit est en rupture de stock, essayez de le dissimuler de la page HTML.



# TRAVAIL A RÉALISER

## Partie 1 - Des détails sur votre confiserie !

3. Il y a 3 types de confiseries : Glaces, Gâteaux et Tartes. Modifiez la valeur de la propriété **soldOut** pour chacun des produits afin que les 3 catégories s'affichent.
4. Amusez-vous à modifier les valeurs de ces propriétés pour activer/déactiver la possibilité d'achat des produits.

# TRAVAIL A RÉALISER

## Partie 2 - Une galerie photo appétissante

1. Affichez la première image correspondante à chaque catégorie de confiserie.
2. Faites en sorte que la balise `<li>` se répète pour chaque image de la catégorie pour ainsi afficher toutes les images d'une catégorie de confiserie.

# TRAVAIL A RÉALISER

## Partie 2 - La gestion des onglets

1. Ajoutez au contrôleur la propriété **tab** et initialisez-la avec la valeur souhaitée.

*tip: Cette valeur doit être comprise entre 1 et 3 car il y a 3 onglets (Description, Référence et Évaluations)*

2. Créez une méthode **setTab** qui permette de modifier la valeur de l'onglet sélectionné. Ajouter également une méthode **isSet** qui renvoie vrai si la valeur passée en paramètre est égale à celle de l'onglet actuellement sélectionné.

# TRAVAIL A RÉALISER

## Partie 2 - La gestion des onglets

3. Dans le fichier html, ajoutez des directives pour :
  - changer la valeur de l'onglet sélectionné
  - afficher l'information liée à l'onglet sélectionné et seulement celui-ci
  - ajouter la classe **active** à chaque onglet.

*hint: utiliser la directive ng-class*

4. Affichez les informations contenues dans les propriétés **description** et **ref** d'un produit au bon endroit dans votre page HTML.

# TRAVAIL A RÉALISER

## Partie 3 - L'onglet Évaluations

1. Dans la partie concernant l'onglet "Evaluations" du fichier html, faites apparaître toutes les évaluations des internautes. Tous les éléments d'une évaluation (**stars**, **body** et **author**) doivent apparaître.
2. Ajoutez les directives **ng-model** correspondantes pour toutes les entrées du formulaire d'ajout d'une évaluation.

# TRAVAIL A RÉALISER

## Partie 3 - L'onglet Évaluations

3. Rendez obligatoires la sélection d'un nombre d'étoiles et la saisie de l'adresse mail.

*hint: utiliser l'attribut html **required***

4. Complétez votre contrôleur avec la propriété **review**, initialement vide, représentant l'évaluation laissée par un internaute.

# TRAVAIL A RÉALISER

## Partie 3 - L'onglet Évaluations

5. Ajoutez la fonction **addReview**. Cette fonction prend en paramètre le produit pour lequel on veut ajouter une évaluation. Elle ajoute la date courante dans l'attribut **createdOn** d'une **review** puis ajoute l'évaluation à la liste des évaluations du produit donné.

*hint: Utiliser l'objet javascript Date et la méthode push(), réinitialiser l'objet **review** à vide après l'ajout d'une évaluation*

6. Ajoutez la directive appropriée dans le formulaire pour ajouter une évaluation lors de la validation du formulaire

*tip: utiliser la propriété \$valid qui renvoie vrai si le contenu du formulaire est valide*

***Vous pouvez partir en week-end :)***