Paramètres URL

- **Sensible à la casse** (clé et valeur)
- L'**absence** de **valeur** a le même effet que l'absence du **paramètre** lui-même

Obligatoires

L'absence d'un seul ou plusieurs paramètres obligatoires dans l'URL entraine l'arrêt prématuré de l'initialisation de la page et affiche un message d'erreur.

data
{String}
Je génère l'url du web service en faisant « *<protocole>://<ip> :<port>/<id>/<data ser terminant par '_gestion.ini'> ?<tous les autres paramètres de l'URL>* ».
Ne pas inclure le « / » initial. Le point de départ est le dossier racine d'Oris

id
{String} @Unique
(Colonne de la base contenant l') ID de la tâche.
Doit être unique

Doit être l'id ("&id=id") **« automatique » donné par la base pour permettre la création de nouvelles tâches** car, pour des raisons de sécurité, il n'est pas possible de définir un ID à la création et, du coup, si une colonne *(au lieu de l'id généré automatiquement par la base) est utilisée, sa valeur sera vide, et la tâche ne sera pas créée visuellement. (Un ID devant être unique, s'il existe des dupliqués, seul la dernière Tâche sera instanciée).*

start
{Date}
(Colonne de la base contenant la) Date de début de la tâche.
Format ISO 8601 (Zulu Time) → "2019-06-14T09:01:49.722Z" ou "2019-06-14T09:01:49Z" (sans les millisecondes)
~~*ou* **Format « Short Date » Européen** → DD/MM/YYY~~

end
{Date}
(Colonne de la base contenant la) Date de fin de la tâche.
Format ISO 8601 (Zulu Time) → "2019-06-14T09:01:49.722Z" ou "2019-06-14T09:01:49Z"
Doit être **strictement supérieure** à la **date de début** de la tâche.
Si **absente**, la Tâche sera une sera visuellement représentée en tant que **milestone**.

Optionnels

Les paramètres optionnels peuvent être omis de l'URL ou bien avoir une valeur vide dans la base.

Natifs

title
{String}

Titre du graphique
La valeur est donnée dans l'URL. Il ne s'agit pas d'une colonne de la base.

subtitle:
{String}
Sous-titre du graphique
La valeur est donnée dans l'URL. Il ne s'agit pas d'une colonne de la base.

name
{String}
Définit le nom de la tâche. Celui-ci est visible sans avoir à survoler.
L'**absence de valeur valide** ne pose pas de problème.

category
{String}
Définit le libellé de la ligne sur laquelle se trouve cette tâche. Permet de regrouper plusieurs tâches sur la même ligne.
L'**absence de valeur valide** (*une chaine de caractère vide est invalide*) entrainera la création d'une **ligne par défaut** nommée automatiquement par GanttCharts.

dependency
{String}
ID de la tâche dont celle courante dépend.
Une flèche partira de celle dont l'ID est précisé ici, vers celle courante.
Un **ID invalide** ne pose **pas un problème** (autre que le fait qu'il n'y aura pas de flèche, mais **cette erreur ne sera pas signalée**)

complete
{Number}
Pourcentage d'avancement d'une tâche.
La valeur doit être **entre 0 et 1 (inclus)**, idéalement. Il est cependant possible de saisir une valeur entre 0 et 100 mais attention car 1 sera toujours interprété comme « 100 % ».

color
{code RGB}
Couleur de la tâche.
Au choix, un **code rgb** ou **rgba**, **court** ou **long**, **avec** ou **sans « # » initial**.
L'absence de valeur valide laissera HighCharts décider arbitrairement d'une couleur (cycle d'une douzaine de couleur et affectant automatiquement **la couleur aux lignes** et, par extension, aux tâches sur celle-ci, à moins d'être précisé par l'utilisateur).

height et width
{Number}

&height déconseillé car crée des bugs visuels

Hauteur et largeur du graphique (inclut le titre et les lignes de l'axe X, "11px" de hauteur, donc le graphique réel ne fait qu'une partie de cette valeur).

Préciser ces valeurs **désactive la responsivité.**
Mélanger &minwidth et &width prendra est déconseillé car illogique. De plus, si &width < &minwidth, une bordure apparaitra à la largeur précisée par &width et le graphique « passera derrière ».

Champs customisés

Il est possible de rajouter des champs customisés au formulaire de création/édition de Tâche. Pour cela, il faut préciser les ID des colonnes de la base contenant la valeur, séparés par des **points virgules** (« ; », par exemple : « &inputs-id=**<id>**;**<id>**;… ») et, avec le même format, les labels des champs à créer.

L'association label/valeur sera visible dans la bulle d'info, lors de survol de Tâche, sous la forme « **<label>** : **<valeur de la BD>** ».

&inputs-id et &inputs-label doivent être **présents en même temps pour activer la fonctionnalité**

inputs-id

{Array d'ID, String}

Si une valeur est **vide**, ou **n'existe pas** dans la base (ne correspond à aucune colonne), ce champ du formulaire ne sera pas créé

inputs-label

{Array de String}

Précise les labels des champs du formulaire sont précisés.

Même **format** que pour &inputs-id.

Les valeurs **vides** sont **autorisées**.

Bonus

En tant que paramètres bonus, ils n'existent pas par défaut / ne sont pas traités par HighCharts mais sont accessibles par l'attribut « userOptions » d'un Point/tâche. Libre à nous de les implémenter.

icon-left et icon-right
{String}
Soit le String contient un point (« . »). L'URI, est alors passée telle quelle, à un tag <img> de taille 24x24px, situé en dehors de la tâche (*peut chevaucher un autre élément du graphique*) :

- si elle commence par un slash « / », le chemin démarre de la racine du serveur, « www » (un fichier « www/img/icon.jpg », le champ de la base sera « /img/icon.jpg »
- sinon, le chemin est relatif au fichier « index.html ». Il est possible de remonter l'arborescence via « ../icon.jpg »

Soit le String ne contient pas de point (« . ») et est alors interprétée comme le nom d'une icône FontAwesome (*version 4.7.0*), un SVG, prédéfini, sans arrière-plan, qui apparaitra sur la tâche.

Il est possible de préciser la taille et l'animation de l'icône lors que l'on utilise FontAwesome *(les animations sont toutes dans le sens des aiguilles d'une montre par contre, on peut rajouter une classe pour préciser l'inverse)*, et la couleur via les classes Bootstrap 4.3.

xprecision
{Number}
Précision, **en heures**, du Drag/Drop. Le déplacement de la tâche se fera par intervalles de <xprecision> heures (snap).

grid
{boolean}
"true" pour afficher des lignes verticales à chaque tick X, format alors une grille permettant de plus facilement lire le Gantt.

current
{boolean}
"true" pour afficher une ligne rouge verticale indiquant la date d'aujourd'hui.

xinterval
{Number}
Intervalle, **en jours**, souhaité entre deux ticks X.

xlabel
{string}
Changer le formatage de l'axe X supérieur (par défaut, le numéro de semaine de l'année).
La valeur doit respecter le format suivant, sans le « % ». Ne supporte qu'une valeur pour l'instant, donc pas vraiment utilisable (à la limite, « B », pour afficher le nom du mois. Couplé à « &xinterval=30 », cela est légèrement utile…).