

# ORA

## Outil de Reporting Avancé

Simonetti Julian

MacDonald Kendrick

# SOMMAIRE

I/ Informations.....	3
1/ Introduction.....	3
2/ Compléments d'information.....	3
II/ Documentation utilisateur.....	4
1/ Inscription.....	4
2/ Connexion.....	4
3/ Récupération de mot de passe.....	5
4/ Navigation générale.....	6
5/ Filtrage.....	6
6/ Drilldown.....	7
7/ Export d'image.....	7
8/ Administration.....	8
III/ Documentation technique.....	9
1/ Bibliothèques et frameworks.....	9
2/ Architecture et méthodes.....	10
A/ Squelette de l'application.....	10
B/ Détail des pages initiales.....	11
B.1/ Table des pages.....	11
B.2/ Codification des fichiers JavaScript.....	12
B.3/ Gestion des sessions.....	13
B.4/ Autres variables globales.....	13
B.5/ Détails sur la méthode de développement.....	14
3/ Architecture et méthodes : la base de données.....	16
A/ Informations générales.....	16
B/ Tables.....	16
C/ Modèle Conceptuel des Données (MCD).....	16

# I/ Informations

## 1/ Introduction

Dans le cadre de notre stage, nous avons intégré l'équipe PEPS d'Orange UPRSE en tant que développeurs.

L'application ORA (pour Outil de Reporting Avancé) est une solution web destinée à un usage en interne au pôle Planification. Elle a pour vocation de faciliter l'aperçu des données relatives aux activités du service, ainsi que leur analyse à des fins statistiques.

L'application est destinée à l'utilisation par le plus grand nombre des membres du service, ainsi son but premier est d'offrir simplicité et ergonomie tout en permettant une vue détaillée, complète et interactive des données. L'une des principales finalités étant de réduire la nécessité d'avoir recours à des outils parfois plus complexes et difficilement manipulables, par exemple des feuilles de calcul Excel.

Le principal type de représentation des données intégré à ORA est l'affichage sous forme de graphiques. ORA comprend une série de graphiques répondant aux besoins statistiques du service, et son architecture permet également son évolutivité: les bibliothèques utilisées (voir Bibliothèques et Frameworks) offrent des outils permettant de créer des graphes très divers et adaptatifs. On pourra également générer des tableaux et autres éventuels moyens de représenter des données.

## 2/ Compléments d'information

Le stage a duré 7 semaines, et nous étions sous la supervision directe de Lucas Courtel et sous les directives de Philippe Genre.

ORA est une application web principalement développée en PHP. Cela permet de rendre l'application dynamique en ayant recours à du traitement côté serveur. Par ailleurs, le caractère Web de l'application permet l'utilisation du JavaScript pour le traitement client et le traitement asynchrone via AJAX. On notera également l'utilisation de SQL pour la communication avec la base de données.

## II/ Documentation utilisateur

### 1/ Inscription

La première étape lorsque l'on souhaite utiliser ORA est de s'inscrire, pour cela, il suffit de remplir le formulaire.



Il est important de mettre une adresse mail valide en cas d'oubli de mot de passe.

Une fois le formulaire complété, appuyer sur Inscription permet de passer à la suite.

The screenshot shows the 'ORA' registration form. It has a black header with 'ORA' in orange. The title 'Inscription' is in bold. The form contains several input fields: 'Nom', 'Prénom', 'Adresse mail', 'Service', 'Identifiant', 'Mot de passe', and 'Confirmation du mot de passe'. Each field has a placeholder text 'Entrez votre [field name]'. At the bottom, there are two buttons: 'Inscription' (orange) and 'Retour' (blue).

### 2/ Connexion

Une fois l'inscription effectuée, il suffit de rentrer son identifiant et son mot de passe dans le menu de connexion pour accéder à l'application.

The screenshot shows the 'ORA' login form. It has a black header with 'ORA' in orange. The title 'Outil de Reporting Avancé Bienvenue' is in bold. The form contains two input fields: 'Identifiant' and 'Mot de passe'. Each field has a placeholder text 'Entrez votre [field name]'. At the bottom, there are two buttons: 'Connexion' (orange) and 'S'inscrire' (blue). Below the buttons, there is a link 'Mot de passe oublié ?' in blue.

### 3/ Récupération de mot de passe

En cas d'oubli de mot de passe, vous pouvez utiliser l'option de changement automatique. Pour cela, précisez l'adresse mail du compte dont vous souhaitez réinitialiser le mot de passe. Si cette méthode venait à ne plus fonctionner, contactez un administrateur de l'application pour qu'il change manuellement votre mot de passe.

Un mail automatique sera envoyé, il suffira alors de cliquer sur le lien (ou si vous voulez l'ouvrir sur un navigateur spécifique, copier-coller l'URL donnée dans le mail) pour être amenés sur la page de réinitialisation de mot de passe.



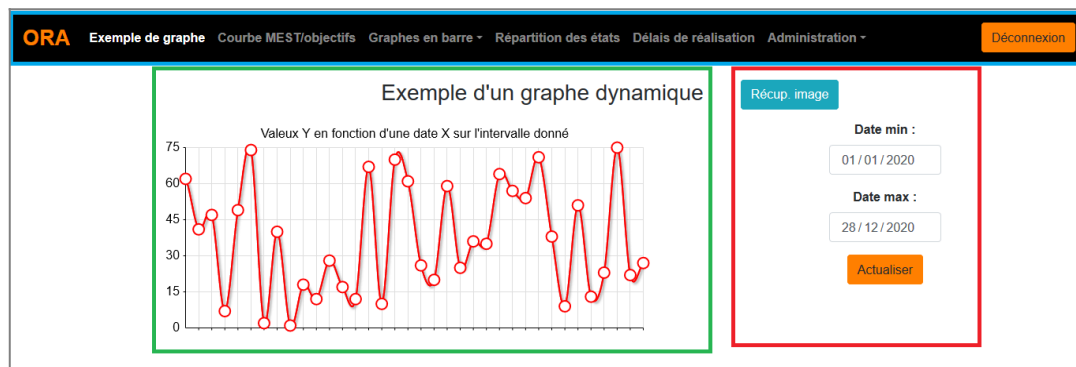
Une fois sur cette page, il vous suffira de rentrer votre nouveau de mot de passe. Si le nouveau mot de passe est identique à celui du compte dont vous voulez changer le mot de passe, aucun changement ne sera fait. Vous pourrez vous connecter normalement et la demande sera annulée.

The image shows a web form titled 'ORA' in a black header. Below the header, it says 'Changement du mot de passe pour l'utilisateur "kendrick"'. There are two input fields: 'Nouveau mot de passe :' and 'Confirmez le mot de passe :'. Below these fields are two buttons: 'Effectuer les changements' (orange) and 'Annuler' (blue).

**Note sur la confidentialité :** vos mots de passe ne sont pas enregistrés par la base de données sans être cryptés avant, et ne peuvent donc pas être consultés.

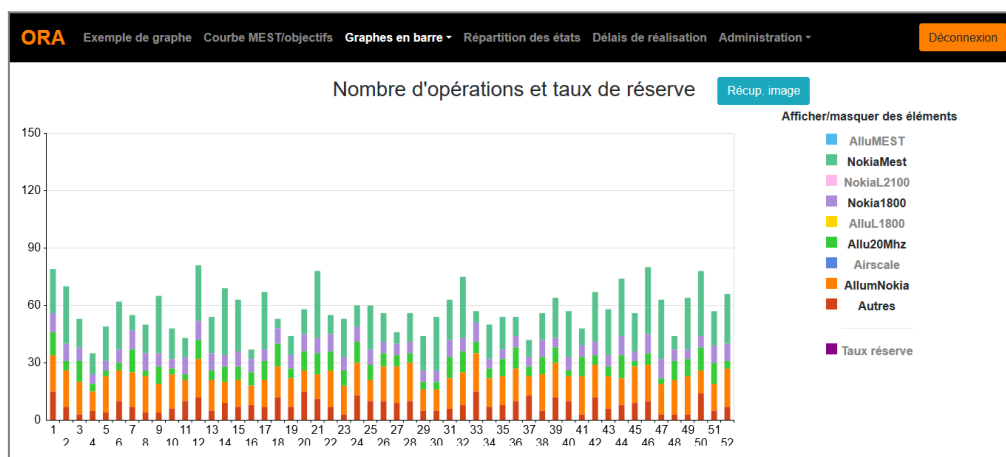
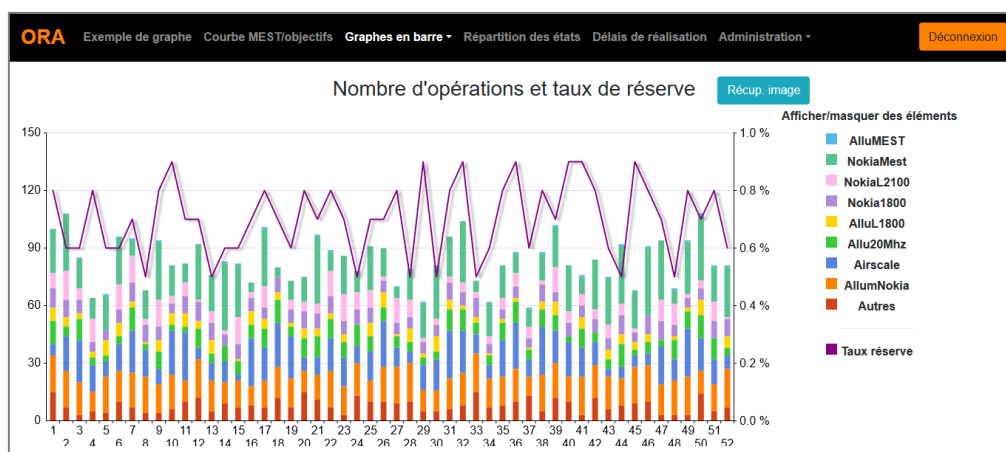
## 4/ Navigation générale

La quasi-totalité des pages de l'application ont une présentation similaire avec un menu de navigation (rectangle bleu sur l'image), une zone d'affichage (rectangle vert sur l'image) et un menu de contrôle (rectangle rouge sur l'image).



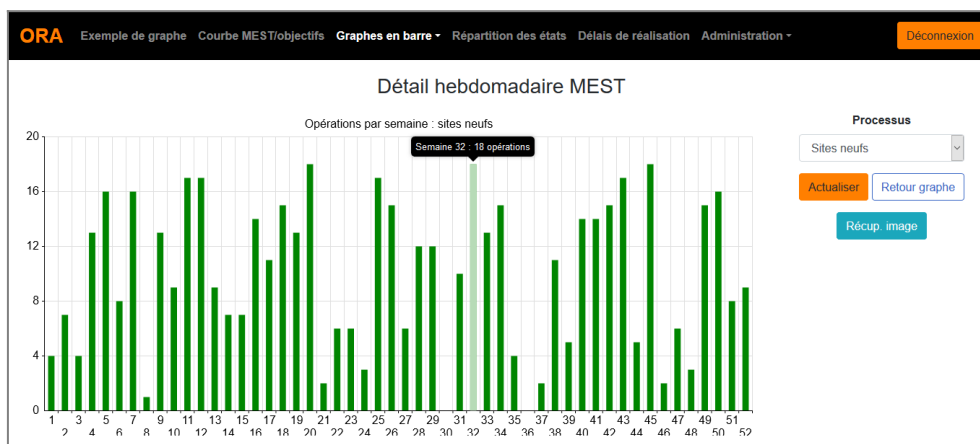
## 5/ Filtrage

Certains graphiques ont une légende disposant d'un système de filtrage, ce qui signifie qu'en cliquant dessus on peut choisir de rendre invisible certaines données.

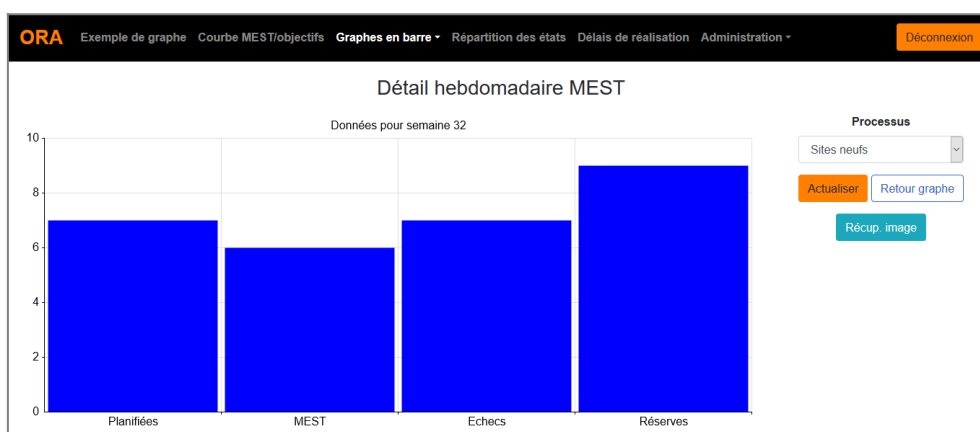


## 6/ Drilldown

Certains graphiques possèdent une fonctionnalité de drilldown, ce qui signifie qu'en cliquant sur une donnée du graphique, un nouveau graphique apparaît, affichant le détail de la donnée sélectionnée.



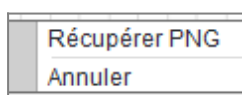
Il est possible de revenir en arrière avec le bouton "Retour graphe"



## 7/ Export d'image

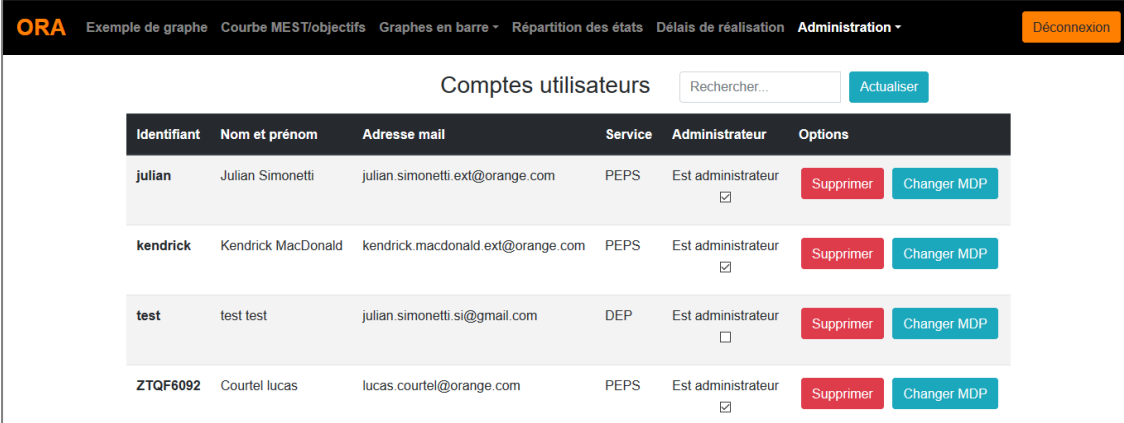
Pour récupérer une image, il existe le bouton prévu à cet effet permettant d'obtenir une image du graphique en cours de visionnage.

Aussi, il y a sur certains graphiques un menu contextuel permettant d'accéder à cette fonctionnalité en faisant clic droit.



## 8/ Administration

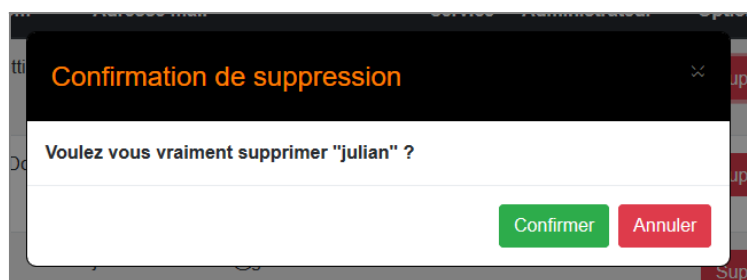
Un utilisateur ayant les droits d'administrateur aura accès au menu d'administration, qui offre plusieurs fonctions et une liste des utilisateurs inscrits.



Comptes utilisateurs					
Identifiant	Nom et prénom	Adresse mail	Service	Administrateur	Options
julian	Julian Simonetti	julian.simonetti.ext@orange.com	PEPS	Est administrateur <input checked="" type="checkbox"/>	<button>Supprimer</button> <button>Changer MDP</button>
kendrick	Kendrick MacDonald	kendrick.macdonald.ext@orange.com	PEPS	Est administrateur <input checked="" type="checkbox"/>	<button>Supprimer</button> <button>Changer MDP</button>
test	test test	julian.simonetti.si@gmail.com	DEP	Est administrateur <input type="checkbox"/>	<button>Supprimer</button> <button>Changer MDP</button>
ZTQF6092	Courtél Lucas	lucas.courtél@orange.com	PEPS	Est administrateur <input checked="" type="checkbox"/>	<button>Supprimer</button> <button>Changer MDP</button>

Cocher ou décocher la case "Est administrateur" permet de donner accès à la page d'administration à un utilisateur.

Il est aussi possible de supprimer un compte si besoin, ou d'en changer le mot de passe manuellement.





## III/ Documentation technique

### 1/ Bibliothèques et frameworks

#### A/ Bootstrap 4.0 (JavaScript)

Le framework Bootstrap permet la mise en page des éléments HTML grâce à un système de containers et de grille, offre des styles prédéfinis et apporte également différentes fonctionnalités pré-implémentées (par exemple certains types de formulaires ou de boîtes de dialogue).

Documentation : <https://getbootstrap.com/docs/4.0/>

#### B/ jQuery (JavaScript)

La bibliothèque jQuery permet de faciliter certaines opérations en JavaScript et aide également à l'utilisation des méthodes AJAX du langage.

Documentation : <https://api.jquery.com/>

#### C/ RGraph (JavaScript)

La bibliothèque RGraph permet de générer les graphiques sur les pages. C'est un outil gratuit se basant sur l'utilisation des canvas HTML. Le nombre d'options et de paramètres est grand, tout comme les types de graphes disponibles. Il s'agit de la bibliothèque la plus importante de ORA.

Documentation : <https://www.rgraph.net/canvas/>

#### D/ html2canvas (JavaScript)

Il s'agit d'une petite bibliothèque permettant de générer des images à partir du contenu d'une page. L'une des fonctionnalités principales d'ORA étant de pouvoir exporter les graphes sous forme d'image, RGraph supporte la plupart du temps l'exportation d'un graphe individuel, cependant lorsqu'on souhaitera récupérer plusieurs graphes et/ou des données ailleurs sur la page en même temps, cette bibliothèque offre cette possibilité.

Documentation : <https://html2canvas.hertzen.com/documentation>

#### E/ PHPMailer (PHP)

Cette bibliothèque permet de générer des mails et de pouvoir les envoyer depuis un serveur SMTP, le tout à partir du PHP. Elle est utilisée pour le système de réinitialisation des mots de passe utilisateurs.

Documentation : <https://github.com/PHPMailer/PHPMailer>

## 2/ Architecture et méthodes

### A/ Squelette de l'application

Assets/JS : fichiers JavaScript développés et utilisés dans l'application

Assets/Styles : fichiers CSS

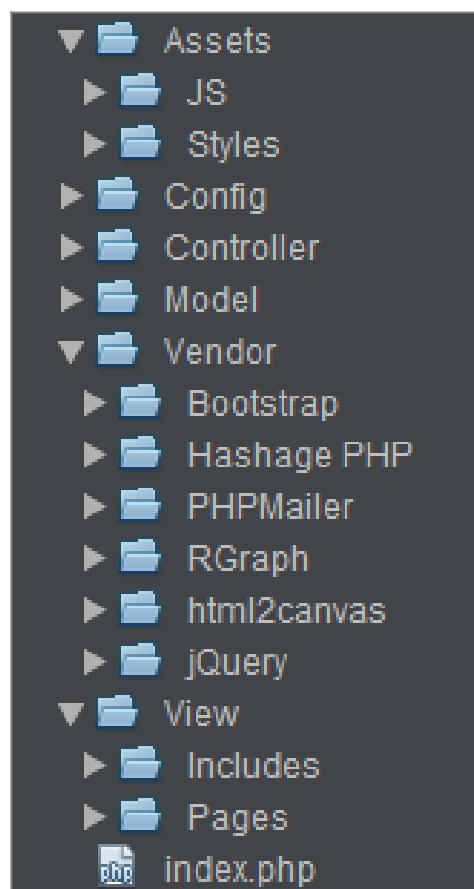
Config : fichiers PHP définissant certains objets et constantes utilisés dans l'application

Controller : fichiers PHP contrôleurs, permettant d'appeler les bons modèles et également les bons fichiers JavaScript à inclure dans la page HTML

Model : fichiers PHP modèles, permettant de faire la liaison avec la base de données. On y retrouvera également les PHP appelés par AJAX sur certaines pages

Vendor : fichiers des bibliothèques utilisées dans l'application (voir Bibliothèques et Frameworks)

View/Includes : parties du HTML qui sont communes à certains groupes de pages (ex: entêtes et pieds de page)



View/Pages : HTML et PHP qui composent la vue de chaque page (voir détail des pages initiales)

index.php : point d'entrée unique de l'application. Chaque appel par le client se fera via cette page, avec le détail de la destination dans des variables telles que le POST, le GET ou les variables de session

## **B/ Détail des pages initiales**

### **B.1/ Table des pages**

Lors du développement des pages de départ ont été faites et nommées génériquement. Voici leur légende :

page1.php : Exemple de graphe dynamique utilisant AJAX et des données factices

page2.php : Courbe du nombre de MEST par rapport aux objectifs, à l'échelle d'une année, les données sont divisées par semaine

page3.php : Graphes en donut montrant la répartition des états de tickets SWAN sur une période définie par l'utilisateur (par défaut : la semaine complète précédente). Deux autres graphes en donut montrent la répartition des états par cause et par typologie, en laissant le choix à l'utilisateur de consulter la répartition des échecs, reports ou réserves. Des tableaux, dynamiques également, donneront le détail des valeurs et des couleurs correspondantes

page4.php : Détail hebdomadaire des MEST sur l'année, les données sont divisées par semaine. L'utilisateur a le choix du processus concerné (sites neufs, réaménagements, new réaménagements, accélération sites neufs). Il s'agit d'un graphe en "Drilldown", un clic sur une donnée affiche un nouveau graphe. Le niveau suivant donne le détail des états des opérations pour cette semaine et ce processus

page5.php : Graphe combiné barres-courbe donnant le nombre d'opérations par typologie et par semaine sur une année. Le graphique affiche également une courbe du taux de réserve pour chaque semaine. La légende est interactive et permet d'afficher ou masquer des éléments du graphe

page6.php : Graphe en radar donnant les délais de réalisation entre chaque jalon du processus, de la MADLL à la MEST. Deux jeux de données sont affichés : les données sur la période de temps demandée (par défaut l'année complète précédente) et un jeu de données de référence (par exemple la moyenne des années précédentes). Il est possible, au lieu de filtrer sur une période de temps, de choisir une opération en particulier via son Code 14.

page7.php : Graphe donnant le détail du nombre d'opérations réalisées pour chaque semaine de l'année complète en comparaison avec le nombre d'opérations planifiées. L'utilisateur peut avoir une vue sur l'année complète ou sur un trimestre en particulier. Il s'agit d'un graphe en Drilldown. Un clic sur une donnée affiche un graphe donnant le détail de ce rapport planifié/réalisé spécifiquement pour chaque aménageur. Un autre clic sur un détail par aménageur affiche un troisième graphe avec une répartition des états d'opération pour cette semaine et cet aménageur

identification.php : page de connexion à l'application. Un formulaire de réinitialisation du mot de passe est intégré. On peut accéder à la page d'inscription depuis cette page

inscription.php : page de création d'un compte ORA

confirmInscription.php : brève page qui confirme que le compte a bien été créé et permet de retourner à la page d'identification

AdminReset.php : page accessible uniquement depuis un mail de réinitialisation de mot de passe, permettant de choisir un nouveau mot de passe et de soumettre sa demande

administrationAcc.php : page d'administration des comptes, visible et accessible uniquement en étant connecté avec un compte disposant des droits administrateur

## **B.2/ Codification des fichiers JavaScript**

Les fichiers JavaScript ne sont pas numérotés mais codifiés. Voici leur correspondance avec les pages :

adaptatif.js : page 1

administration.js : page d'administration

amen\_\*.js : page 7 : détail par aménageur

bars\_\*.js : page 4 : détail hebdomadaire des MEST

combo\_\*.js : page 5 : graphe combiné opérations par typologie et taux de réserve

courbe\_mest.js : page 2

donut(s)\_\*.js : page 3 : répartition des états

radar\_\*.js : page 6 : délais de réalisation

verifPasse\*.js : code gérant les formulaires avec saisie de mot de passe

reset\_mdp.js : gère la réinitialisation de mot de passe depuis la page de demande de reset

### **B.3/ Gestion des sessions**

#### B.3.1/ Variables de session

L'accès à ORA nécessite d'ouvrir une session utilisateur. Voici les informations associées à l'utilisateur qui sont retenues (index de la variable SESSION => signification) :

utilUsername : identifiant de l'utilisateur

utilNom : nom de famille de l'utilisateur

utilPrenom : prénom de l'utilisateur

utilMail : adresse mail de l'utilisateur

utilService : service de l'utilisateur

utilNiveauPriv : niveau de privilèges de l'utilisateur

#### B.3.2/ Niveaux de privilège

En date du 19/02/2021, deux niveaux de privilèges sont implémentés :

- 1 : utilisateur normal
- 256 : administrateur

Des valeurs numériques, qui par ailleurs sont volontairement éloignées, sont utilisées pour laisser la place à une éventuelle évolution du système de privilèges.

Un administrateur a la possibilité d'accéder aux pages d'administration des comptes, et peuvent ainsi supprimer des comptes ou changer leur mot de passe manuellement (fonctionnalité destinée à disparaître à mesure que la réinitialisation du mot de passe par l'utilisateur sera perfectionnée et flexible).

### **B.4/ Autres variables globales**

On notera parfois l'utilisation d'index 'error\*' dans le POST ou le REQUEST pour permettre de transmettre certains marqueurs d'erreur à travers les pages. Cela facilite la gestion de l'affichage de certains messages d'erreur.

## B.5/ Détails sur la méthode de développement

### B.5.1/ Génération des graphes

```
graphes[0] = new RGraph.Bar({
  id: 'processus',
  data: data,
  options: {
    title: 'Opérations par semaine : ' + listeProc[processus],
    xaxisLabels: labels,
    customlabels: labels,
    tooltips: 'Semaine ${property:customlabels[${dataset}]} : ${value_formatted} opérations',
    tooltipsHotspotXonly: true,
    tooltipsEvent: 'mousemove',
    textAccessible: false,
    contextmenu: [
      ['Récupérer PNG', RGraph.showPNG],
      null,
      ['Annuler', function () {}]
    ],
    colors: ['green']
  }
});
```

La génération d'un graphe se présente toujours comme ceci dans le Javascript. La liste des options est disponible sur la documentation de RGraph : <https://www.rgraph.net/canvas/>.

### B.5.2/ Pré-requis à l'exportation

Pour que les graphes puissent être exportés sans transparence :

- S'assurer que l'option `textAccessible: false` est renseignée à la génération
- Avant de dessiner le graphe (par exemple avec la méthode `.draw()`), effectuer la ligne suivante :

```
RGraph.clear(canvas, 'white');
```

Où `canvas` est l'objet de DOM du canvas où le graphe sera dessiné.

### B.5.3/ Résultats des appels AJAX

Les fichiers PHP appelés par AJAX prennent toujours cette forme :

```
<?php

header('Content-Type: text/html; charset=utf-8');
session_start();
include '../Config/includeodin.php';
include '../Config/CONST.php';
include '../Config/objetPDO.php';

$borne1 = $_POST['borne1'];
$borne2 = $_POST['borne2'];
//var_dump($_POST);
//echo $borne1.' '.$borne2;
$c = getObjetPDO();

$req = 'SELECT dataDate, dataValeur '
      . 'FROM outilreport_data1 '
      . 'WHERE dataDate BETWEEN :borne1 AND :borne2';

$stmt = $c->prepare($req);
$stmt->bindParam(':borne1', $borne1);
$stmt->bindParam(':borne2', $borne2);
//var_dump($stmt);
$stmt->execute();

$labelsOutput = [];
$dataOutput = [];

$resultat = $stmt->fetchAll(PDO::FETCH_ASSOC);

foreach ($resultat as $ligne) {
    array_push($labelsOutput, $ligne['dataDate']);
    array_push($dataOutput, (int)$ligne['dataValeur']);
}

$output = [$labelsOutput, $dataOutput];
echo json_encode($output);
```

Pour RGraph il est toujours plus pratique de retourner les valeurs sous la forme d'un grand array unique.

### 3/ Architecture et méthodes : la base de données

#### A/ Informations générales

La base de données repose sur le SGBD MariaDB, évolution de MySQL.

#### B/ Tables

Les tables sont toutes situées dans la base ICP.

Les tables au préfixe outilreport\_\* sont les tables utilisées pour le développement. Le numéro apposé à data\* représente la page concernée dans l'appli. Ces tables sont vouées à disparaître à mesure que de vraies données sont utilisées.

Les tables au préfixe ORA\_\* sont les tables valides et définitives, contenant de vraies données.

#### C/ Modèle Conceptuel des Données (MCD)

En date du 19/02/2021, la base de données liée à ORA n'est pas terminée. En revanche, une ébauche du MCD a été faite. Le modèle est amené à évoluer.

Sont implémentées les tables :

- Opération
- Bande
- BandesPourOpération
- TicketSWAN

