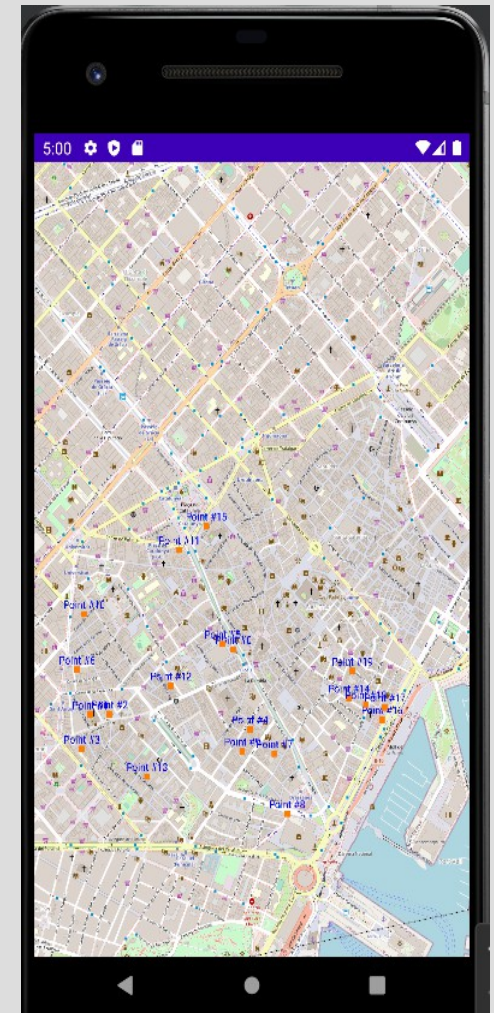
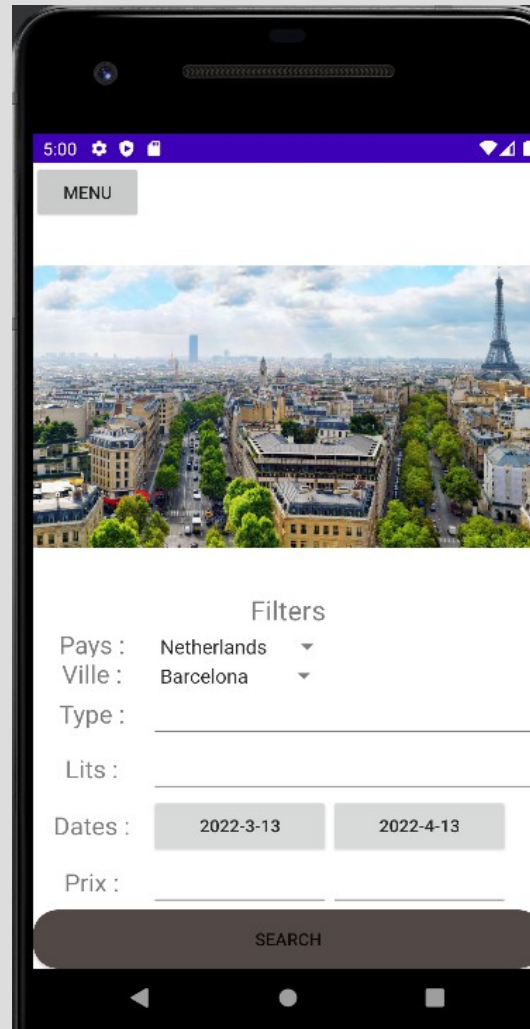
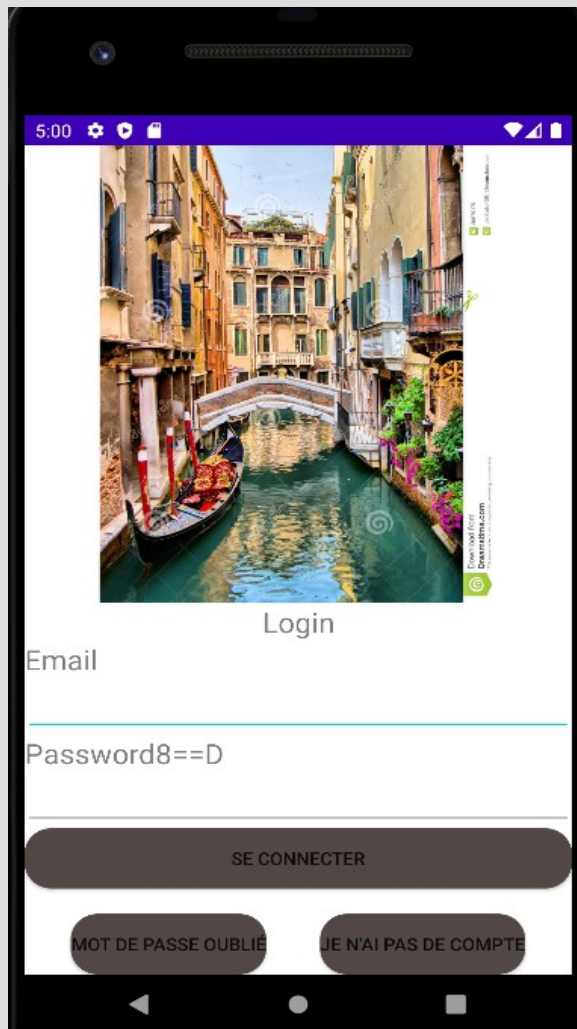


Portofolio

- H3BnB
 - Jeu python 2D
 - Fast4Food
 - IpGrabber
-
- Refonte dashboard web Solutions 909
 - Ecran reporting flux
 - Visualisations Kibana

H3BnB



H3BnB

models > database.php > ...

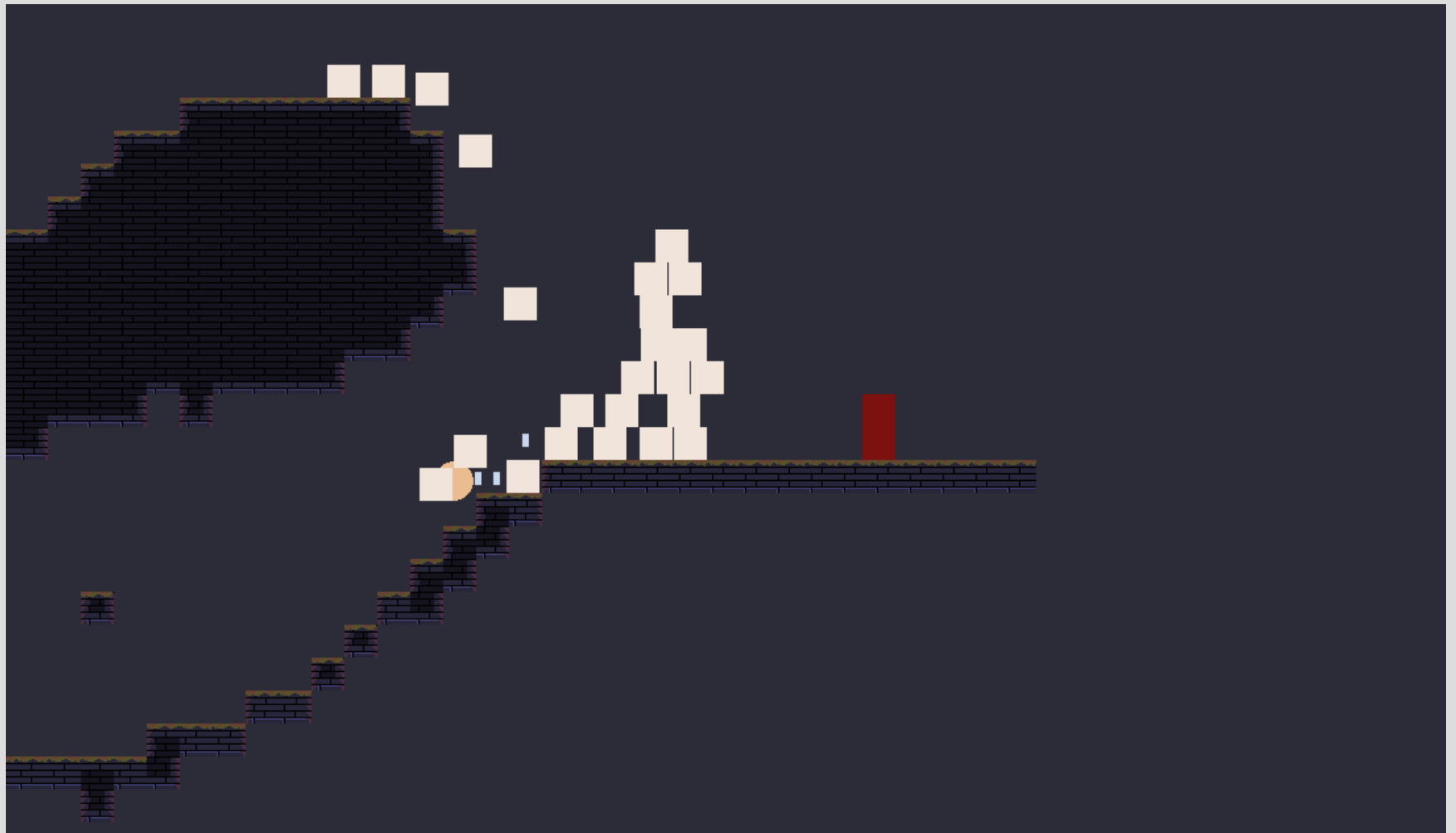
```
1  <?php
2  ob_start();
3  session_start();
4  class Database{
5
6      // specify your own database credentials
7      private $host = "localhost";
8      private $db_name = "h3bnbbdd";
9      private $username = "etienne";
10     private $password = "";
11     public $conn;
12
13     // get the database connection
14     public function getConnection(){
15
16         $this->conn = null;
17
18         try{
19             $this->conn = new PDO("mysql:host=" . $this->host . ";port=3306;dbname=" . $this->db_name.";charset=utf8", $this->username, $this->password);
20         }catch(PDOException $exception){
21             echo "Connection error: " . $exception->getMessage();
22         }
23
24         return $this->conn;
25     }
26 }
27 $database = new Database();
28 $db = $database->getConnection();
29 ?>
```

H3BnB

	Table	Action	Lignes	Type	Interclassement	Taille	Perte
<input type="checkbox"/>	biens	★ Parcourir Structure Rechercher Insérer Vider Supprimer	40	InnoDB	utf8mb4_general_ci	4,5 Mio	-
<input type="checkbox"/>	cities	★ Parcourir Structure Rechercher Insérer Vider Supprimer	3	InnoDB	utf8mb4_general_ci	32,0 kio	-
<input type="checkbox"/>	contrats	★ Parcourir Structure Rechercher Insérer Vider Supprimer	3	InnoDB	utf8mb4_general_ci	48,0 kio	-
<input type="checkbox"/>	countries	★ Parcourir Structure Rechercher Insérer Vider Supprimer	3	InnoDB	utf8mb4_general_ci	16,0 kio	-
<input type="checkbox"/>	user	★ Parcourir Structure Rechercher Insérer Vider Supprimer	3	InnoDB	utf8mb4_general_ci	16,0 kio	-
	5 tables	Somme	52	InnoDB	utf8mb4_general_ci	4,7 Mio	0 o

			1	beds	price	type	latitude	longitude	availabilities	description	idCity	idOwner	images
<input type="checkbox"/>	Éditer Copier Supprimer	1	1	150	Apartment	48.859705979478	2.3563040991133	ALL		Situated between Rambuteau and City hall, my flat ...	4	5	/9j/4AAQSkZJRgABAQEASABIAAD/2wBDAAgGBgc
<input type="checkbox"/>	Éditer Copier Supprimer	2	1	110	Apartment	48.856822795941	2.3585161884965	ALL		This modern, elegant and comfortable studio apartm...	4	5	/9j/4AAQSkZJRgABAQEASABIAAD/2wBDAAgGBgc
<input type="checkbox"/>	Éditer Copier Supprimer	3	1	75	Apartment	48.853547780061	2.3673635984799	ALL		L'appartement Bastille, situé à 5 minutes à pied d...	4	5	/9j/4AAQSkZJRgABAQEASABIAAD/2wBDAAgGBgc
										Appartement			

Jeu Python 2D



Jeu Python 2D

- .idea
- __pycache__
- Assets
- maps
- venv
- Block.py
- Enemy.py
- main.py
- map.txt
- player.py
- Projectiles.py
- settings.py
- Tileset.py

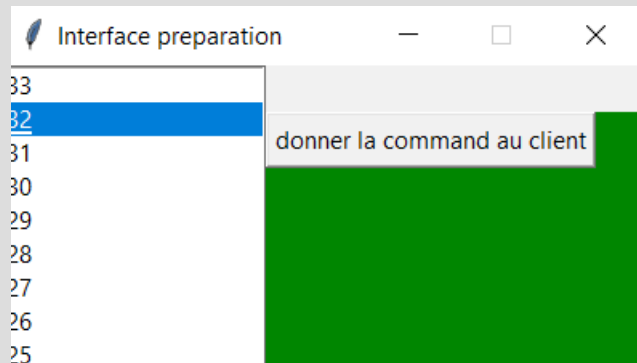
```
def loop(self):
    while self.running:
        self.events()
        self.update()
        self.camera()
        self.draw()
        self.clock.tick(framerate)
```

```
def events(self):
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            self.running = False
            pygame.quit()
        if event.type == pygame.KEYDOWN:
            self.pressed[event.key] = True
            if event.key == pygame.K_UP:
                self.player.jump()
        if event.type == pygame.KEYUP:
            self.pressed[event.key] = False
        if event.type == pygame.MOUSEBUTTONDOWN:
            self.player.shooting()
        if event.type == pygame.MOUSEMOTION:
            print(pygame.mouse.get_pos())
```

```
def update(self):
    if self.running:
        self.checkevents()
        self.gravity()
        for entity in self.gravity_affected:
            entity.brain()
            entity.move()

        for sprite in self.explosion_list:
            sprite.update()
```

Fast4Food



```
def __init__(self):

    self.root = tk.Tk()
    self.root.resizable(False, False)
    self.root.geometry('400x190')
    self.root.title('Interface preparation')

    self.listbox()
    self.but_listbox()

    self.info_frame()
    self.but_chg_stat()
    self.root.mainloop()
```

```
def chg_order(self):
    i = self.lb_listbox.curselection()
    print(i)
    # revoir
    self.displayed_order_id = self.lb_listbox.get(i)
    print(self.displayed_order_id)
    self.get_info()
    self.display()

def get_info(self):
    info = D.select_contient(self.displayed_order_id)
    self.to_print = []
    for item in info:
        item_id = item[0]
        qt = item[2]
        name = D.items_from_id(item_id)[0][0]

        if qt != 0:
            line = name + ' *' + str(qt)
            self.to_print.append(line)
```

IpGrabber

```
$os = getOS();  
$browser = getBrowser();  
$ip = getIp();  
  
$ipDetails = getIpDetails($ip);  
$country = $ipDetails['country'];  
$city = $ipDetails['city'];  
$postal = $ipDetails['postal'];  
  
$httpRef = $_SERVER['HTTP_REFERER'];
```

```
affichageConnectionData();  
$lastIdUser = insertIntoConnectionBase($c1, $os, $browser, $httpRef, $city, $postal, $country, $ip, $key, $iv);  
insertIntoKeyBase($c2, $lastIdUser, $iv, $key);
```

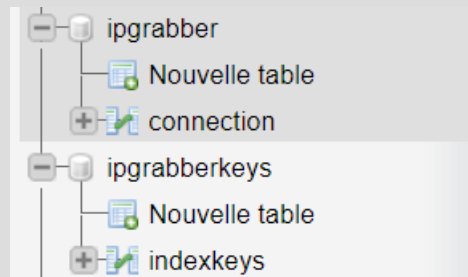
```
function insertIntoConnectionBase($conn, $os, $browser, $httpRef, $city, $postal, $country, $ip, $key, $iv){  
  
    $query = "INSERT INTO connection(os, browser, http_referer, ville, postal_code, pays, ip) VALUES(?,?,?,?,?,?,?)";  
    $stmt = $conn->prepare($query);  
    $queryArray = array(  
        encrypt($os, $key, $iv)[0],  
        encrypt($browser, $key, $iv)[0],  
        encrypt($httpRef, $key, $iv)[0],  
        encrypt($city, $key, $iv)[0],  
        encrypt($postal, $key, $iv)[0],  
        encrypt($country, $key, $iv)[0],  
        encrypt($ip, $key, $iv)[0]);  
    $stmt->execute($queryArray);  
    $lastIdUser = $conn->lastInsertId();  
    return $lastIdUser;  
}
```


IpGrabber

```
//$connexion = new mysqli($dbHost, $dbUser, $dbPass,$dbName) or die("Connect failed: %s\n". $connexion -> error);
$connexion1 = new PDO("mysql:host=" . $dbHost . ";port=3306;dbname=" . $dbName.";charset=utf8", $dbUser, $dbPass );

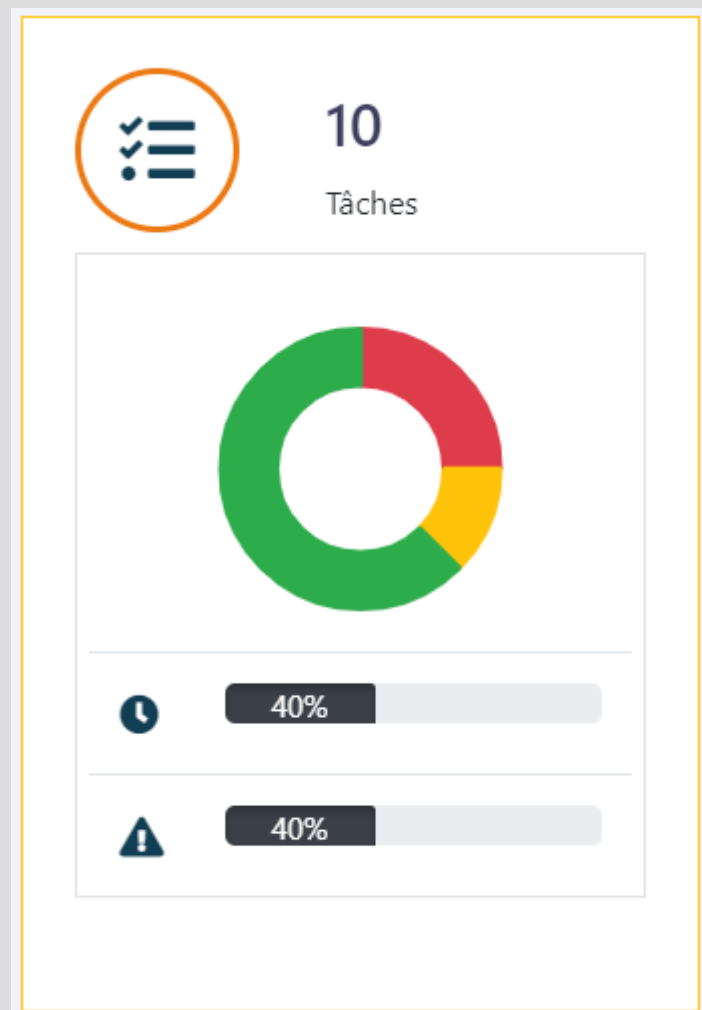
//$connexion = new mysqli($dbHost, $dbUser, $dbPass,$dbName) or die("Connect failed: %s\n". $connexion -> error);
$connexion2 = new PDO("mysql:host=" . $dbHost . ";port=3306;dbname=" . $db2Name.";charset=utf8", $db2User, $dbPass );

return [$connexion1, $connexion2];
```



</

Dashboard Solutions 909



Dashboard Solutions 909

```
<!-- Tâches -->
<div class="col-lg-6 col-md-6 organisme">
  <div class="card <?php echo($_SESSION['connected_user_droits'] != 1 ? '' : 'border-warning')?>">
    <div class="card-body">
      <div class="stat-widget-one pb-2">
        <a href="<?= SERVER_ROOT.'/modules/conformite/traitement-taches.php'?>">
          <div class="stat-icon dib"><i class="fa fa-tasks" title="Tâches" style="color:#0e384d; border-color:#eb730f;"></i></div>
        </a>
        <div class="stat-content dib">
          <div class="stat-digit h5">
            <?php
              $request = $bdd->prepare("SELECT COUNT(*) as NBTASK FROM traitement_taches, lien_org_tra where ta_action <> 'D' and ta_tra_id = lot_id_tra and lot_id_tra = ".$_SESSION['connected_user_lot_id']."");
              $request->execute(array($_SESSION['connected_user_organisme'], 'DC'));
              $data=$request->fetch();
              $NBTASK = $data['NBTASK'];
              echo $NBTASK;
              $request->closeCursor();
              $lateTasks = rand(0, $NBTASK);
              if ($NBTASK == 0){
                $percLateTasks = 0;
                $taskGreen = 0;
                $taskOrange = 0;
                $taskDark = 0;
              } else {
                $request = $bdd->prepare("SELECT count(*) as NBWA FROM traitement_taches, lien_org_tra where ta_action <> 'D' and ta_tra_id = lot_id_tra and lot_id_tra = ".$_SESSION['connected_user_lot_id']."");
                $request->execute(array($_SESSION['connected_user_organisme'], 'WA', 'IP'));
                $data=$request->fetch();
                $taskOrange = $data['NBWA'];
                $request->closeCursor();

                $request = $bdd->prepare("SELECT count(*) as NBNC FROM traitement_taches, lien_org_tra where ta_action <> 'D' and ta_tra_id = lot_id_tra and lot_id_tra = ".$_SESSION['connected_user_lot_id']."");
                $request->execute(array($_SESSION['connected_user_organisme'], 'NC'));
                $data=$request->fetch();
                $taskRed = $data['NBNC'];
                $request->closeCursor();
```

Dashboard Solutions 909

```
<div class="card pt-0 " style="color:#0e384d;">
  <div class="card-body <?php echo($NBTASK != 0 ? 'pb-0': '')?>">

    <?php if($NBTASK != 0){?>

      <div class="row justify-content-center pb-0">
        <div class="col-8 pb-3">
          <canvas id="cmChartOrgaTaches"></canvas>
        </div>
        <table class="table col-12 pb-0 mb-0">
          <tr>
            <td class="col-2"><i class="fa fa-clock" title="Tâches en retard" style="color:#0e384d; border-color:#eb730f;"></i></td>
            <td class="col-10">
              <div class="progress">
                <div class="progress-bar bg-dark" role="progressbar" title="<?= $lateTasks?> tâches en retard" style="width:<?= $percLateTasks?>% "></div>
              </div>
            </td>
          </tr>

          <tr>
            <td class="col-2"><i class="fa fa-exclamation-triangle" title="Tâches urgentes" style="color:#0e384d; border-color:#eb730f;"></i></td>
            <td class="col-10">
              <div class="progress">
                <div class="progress-bar bg-dark" role="progressbar" title="<?= $urgTasks ?> âches urgentes" style="width:<?= $percUrgTasks?>% "></div>
              </div>
            </td>
          </tr>
        </table>
      </div>

      <script>
        const configCmOrgaTac = {
          type: 'doughnut',
          data: {
            labels: [
              'A commencer',
              'A terminer',
              'Terminées',
            ]
          }
        }
      </script>
    }
  </div>
</div>
```

Reporting Routes

```
<!-- DATATABLE -->
<div class="row py-4 mt-4" style="line-height: 1.8; margin-right: 0px;">
  <div class="col-12 ml-2 mr-2">
    <div class="bdf-datatable">
      <table aria-describedby="tabledescription"
        class="table table-striped table-condensed bdf-table"
        #mf="mfDataTable"
        [mfData]="rapportAgoraTrefleList"
        [mfRowsOnPage]="pageSize"
      >
        <thead>
          <tr style="word-wrap: break-word; align-items: center;">
            <th *ngFor="let column of rapportAgoraTrefleListColumns"
              style="text-align: center;"
              class="p-0" scope="col"
              id="{{column.key}}"
            >
              <ng-container *ngIf="column.key!='alerteCheckBox'; else checkboxBlock">
                <bdf-ng-datatable-sorter [by]="column.sortedKey" class="justify-content-center p-0" style="word-wrap: break-w
                  <br/><p class="columnTitle pl-0 pr-0 mr-0 ml-0" >{{column.label}}</p>
                </bdf-ng-datatable-sorter>
              </ng-container>
              <ng-template #checkboxBlock id="alerteCheckBox">
                <div id="alerteCheckBox" class="justify-content-center px-0 ml-1 mr-1">
```

Reporting Routes

```
extractData(data){
  let routes = []
  for (let i = 0; i < data.length; i++){
    let entry = data[i]
    let routePath = entry["route"]
    let timestamp = entry["timestamp"]
    let time = timestamp.split('T')[1].slice(0,4)
    let date = {date:timestamp.split('T')[0], time:time.slice(0,2)+'h'+time.slice(2,4)}
    let route = {
      route:routePath,
      client: entry["client"],
      typeReleves: entry["typeReleves"],
      fluxMft: entry["fluxMft"],
      idParticipant: entry["idParticipant"],
      canal: entry["canalLibelle"],
      heureLimite: entry["heureLimite"],
      nbFichiers:entry["nbFichiers"],
      dates:[date]
    }

    if (routes.length!=0){
      let addroute = true
      routes.forEach(element => {
        if (element.route == routePath){
          addroute = false
        }
      })
    }
    if (addroute){
      routes.push(route)
    }
  }
}
```

Reporting Routes

```
public void addFileIfAbsent(String route, String fileName){  
  
    routes = reportingAgoraTrefleConfig.getAgoraTrefleRoutesList().stream().map(agoraTrefleRoute -> agoraTrefleRoute)  
  
    if (routes.contains(route) || route.contains("216I344S16")) {  
        routesListInfo = reportingAgoraTrefleConfig.getAgoraTrefleRoutesList();  
        routesListInfo.forEach(agoraTrefleRoute -> {  
            if (route.equals(agoraTrefleRoute.getRoute())){  
  
                String id = route + StringTools.COLON + DateTools.getTodayStrDate() + StringTools.COLON + fileName  
  
                ReportingRouteModel reportingRouteModel = new ReportingRouteModel();  
  
                reportingRouteModel.setId(id.trim());  
                reportingRouteModel.setRoute(agoraTrefleRoute.getRoute());  
                reportingRouteModel.setFluxMft(agoraTrefleRoute.getFluxMft());  
                reportingRouteModel.setTimestamp(DateTools.getStrHistoriqueDateTime());  
                reportingRouteModel.setClient(agoraTrefleRoute.getClient());  
                reportingRouteModel.setFilename(fileName);  
                reportingRouteModel.setMeta_id(id);  
                reportingRouteModel.setTypeRelevés(agoraTrefleRoute.getTypeRelevés());  
                reportingRouteModel.setIdParticipant(agoraTrefleRoute.getIdParticipant());  
                reportingRouteModel.setCanalLibelle(agoraTrefleRoute.getCanalLibelle());  
            }  
        });  
    }  
}
```

Reporting Kibana

```
buildVizDoublonsReport(data){
  let ligneEnTete = ["msg_id", "meta_step", "Count", "meta_appname", "meta_dir"]
  let toExport = []
  toExport.push(ligneEnTete)
  data.forEach(element => {
    let toPush = []
    let msgId = "" + element["msgId"].toString()
    let metaStep = element["metaStep"]
    let count = 1
    let metaAppname = element["metaAppname"]
    let metaDir = element["metaDir"]
    let newroute = true
    toExport.forEach(entry => {
      if (entry[0] == msgId){
        newroute = false
        entry[2] += 1
      }
    })
    if (newroute){
      toPush.push(msgId, metaStep, count, metaAppname, metaDir)
      toExport.push(toPush)
    }
  })
  let name = "viz Doublons msg_id / astra_ops"
  this.buildCsv(name, toExport)
```

```
@Query("{\"bool\":{\"must\": [{\"terms\": {\"meta_step\": [\"EPE_IN\", \"MPH_IN\"]}}, {\"match\": {\"meta_dir\": \"ACQUIS\"}}]}\")
Stream<OperationModel> findAllByDateForDoublons(String gte, String lte);
```

```
@Query("{\"bool\":{\"must\": [{\"terms\": {\"tec_reference_fichier_bdf\": [\"30001ZZZTESSI\", \"30001SPCR - IMAG\"]}}, {\"match\": {\"meta_dir\": \"ACQUIS\"}}]}\")
Stream<OperationModel> findAllByDateForTessi(String gte, String lte);
```