

Documentation technique

Pro à Pro - Planning des réceptions à venir - 2024



Table des matières

I. Introduction.....	3
A. Entreprise.....	3
B. Objectif.....	3
C. Vue d'ensemble.....	3
II. Architecture Global.....	4
A. Positionnement de la fonctionnalité.....	4
B. UML.....	5
III. Description de l'interface utilisateur.....	6
A. Filtre.....	6
B. Planning.....	6
IV. Backend (Côté Serveur).....	9
A. Utilisation de fonction.....	9
1. Vérification du token.....	9
2. Vérification de la tâche.....	9
B. Nouveaux Web Services.....	10
1. Fournisseurs.....	10
2. Réceptions fournisseurs.....	10
3. Informations réception.....	10
4. Détails commande.....	10
V. Frontend (Côté Client).....	12
A. Langages et technologies.....	12
B. AJAX.....	12
1. Réceptions fournisseurs.....	12
2. Informations réception.....	13
3. Détails commande.....	13
VI. Annexes.....	15
A. Fournisseurs.....	15
1. Contrôleur.....	15
2. Manager.....	15
B. Réceptions fournisseurs.....	16
1. Contrôleur.....	16
2. Manager.....	16
C. Informations réception.....	18
1. Contrôleur.....	18
2. Manager.....	18
D. Détails commande.....	20
1. Contrôleur.....	20
2. Manager.....	20

I. Introduction

A. Entreprise

Pro à Pro est une entreprise française d'environ 1 000 salariés. Elle est spécialisée dans la distribution de produits alimentaires frais et de qualité. Avec son réseau de logistique avancé, Pro à Pro livre ses clients en 24h.

B. Objectif

L'objectif de la fonctionnalité est de pouvoir visualiser les réceptions des commandes des fournisseurs. Cela permettra d'afficher sur un planning, par différent bloc distinguer par température, chaque commande fournisseur. On pourra ensuite avoir les détails de la commande en cliquant sur le bloc correspondant. Il est possible aussi de filtrer les commandes par date de réception, température et code fournisseur.

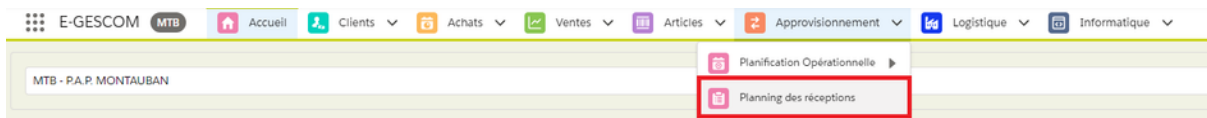
C. Vue d'ensemble

The screenshot displays the 'PRO à PRO' web application interface. The top navigation bar includes the company logo and a menu with items like 'E-GESCOM', 'MTB', 'Accueil', 'Clients', 'Achats', 'Ventes', 'Articles', 'Approvisionnement', 'Logistique', and 'Informatique'. The main header shows the 'Module Planning des réceptions' with a 'Retour à l'accueil' button. Below this, a breadcrumb trail reads 'Accueil > Approvisionnement > Planning des réceptions'. A 'Filtres' section contains input fields for 'Date de livraison' (format: jj/mm/aaaa), 'Service' (dropdown menu), and 'Code fournisseur'. To the right of these filters are buttons for 'Sec:', 'Frais:', and 'Gel:'. Below the filters is a table with a header row for 'Qual' and a data row for 'Inconnue'. The table's columns represent time slots from 00:00 to 23:00 in one-hour increments.

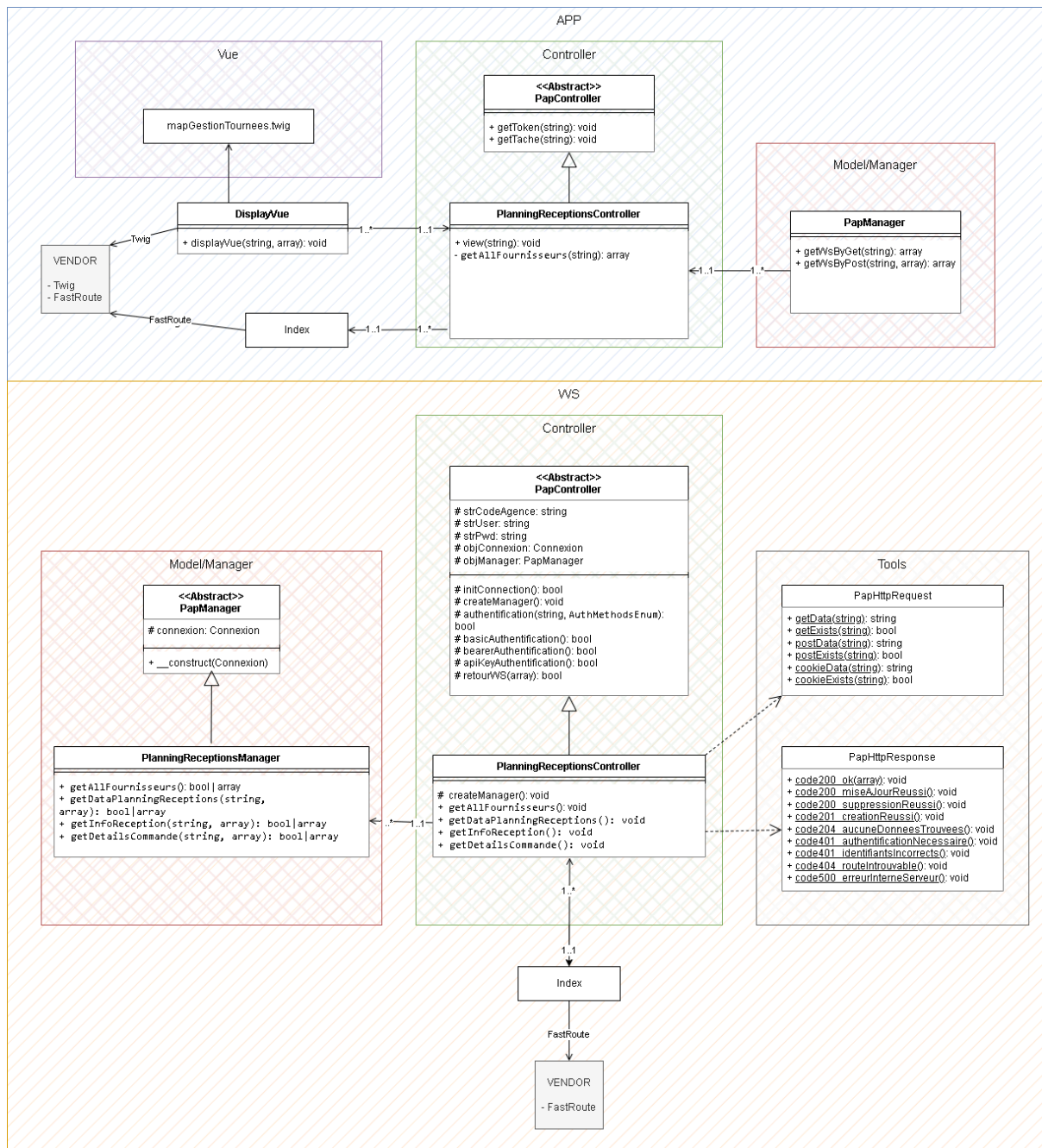
II. Architecture Global

A. Positionnement de la fonctionnalité

La fonctionnalité se trouve dans l'application web E-Gescom. Dans la catégorie "Approvisionnement > **Planning des réceptions**". On peut y accéder grâce à la barre de navigation ou par la page d'accueil. Pour pouvoir accéder à cette fonctionnalité il faut avoir la tâche "GESWPLREC".

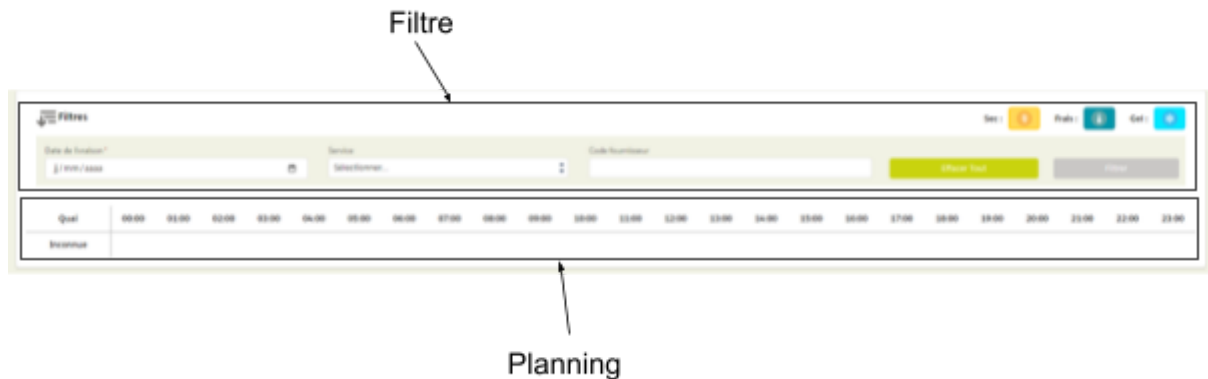


B. UML



III. Description de l'interface utilisateur

L'interface utilisateur sur E-GESCOM pour la fonctionnalité de planning des réceptions à venir se présente en deux parties. La première est le filtre. La seconde partie est le planning.



A. Filtre

Le filtre contient 3 champs :

- **Date de livraison** : Permet de filtrer sur la date de livraison d'une commande. (*Obligatoire*)
- **Températures** : Permet de filtrer sur la température des commandes (Sec, Frais ou Gel).
- **Code fournisseur** : Permet de filtrer un fournisseur précis..

Le filtre contient aussi 2 boutons :

- **Effacer tout** : Permet d'effacer tous les éléments du filtre et le résultat afficher sur la map.
- **Filtrer** : Permet de filtrer avec les champs remplis dans le filtre.

B. Planning


Le planning permet d'afficher la résultante du filtre après avoir appuyé sur le bouton Filtrer. Elle affichera toutes les commandes par fournisseurs qui ont une date de livraison égale à celle entrée dans le filtre.

Chaque bloc de commande par fournisseur à une couleur attitrée ce qui correspond à sa température mais aussi un quai qui peut être connu ou inconnu.



Si on clique sur un bloc correspondant à une réception d'un fournisseur, il y a un modal qui s'ouvre affichant les informations de la livraison et les détails des différentes commandes passer au fournisseur dont la livraison est pour la date entrée dans le filtre.

Infos réception

Code fournisseur : 400034
 Nom fournisseur : AMAND SAS
 Date de livraison : 18/12/2023
 Heure de livraison : 06:00
 Service : 
 Quai de réception : Inconnu

Commandes :

Détails Cde 1048974 Cde 1048977

Nombre de commande : 2
 Nombre total de palette : 1
 Poids brut total : 346.24 kg
 Volume total : 0.856 m³

En premier lieu nous avons les informations de la réception avec le code fournisseur, son nom, la date de livraison, l'heure de livraison, la température et le quai de réception. Ensuite là où il y a les différents onglets on peut retrouver l'onglet général "Détails" où l'on retrouve les informations générales de toutes les commandes et ensuite sur les autres onglets on a les informations détaillées de chaque commande comme ci-dessous.

Commandes :

Details **Cde 1048974** Cde 1048977

Numéro de commande : 1048974

Créateur de la commande : PEREZ Bertrand

Nombre de palette : 1

Poids brut : 173.12 kg

Volume : 0.367 m³

Détails de la commande :

Afficher éléments

Rechercher:

Activité ▲	Code article ◆	Unité de vente ◆	Désignation ◆	Marque ◆	Quantité ◆	Prix brut ◆
FRA	157729	SCH - Sachet	Boudin blanc BN 125g x8	Bernaudeau	160	5.2925 €

Affichage de 1 à 1 sur 1 éléments

Précédent Suiv

IV. Backend (Côté Serveur)

A. Utilisation de fonction

1. Vérification du token

- **Nom** : *checkToken*
- **Description** : Vérifier si l'utilisateur est bien connecté avant de lui afficher la page.
- **Classe** : *PapController*
- **Paramètre** : *Aucun*
- **Retour** : *Aucun*
- **Exemple** : `$this->checkToken();`
- **Code** :

```
public function checkToken(): void
{
    if (empty($_SESSION["token"])) {
        header("Location: /app/");
    }
}
```

2. Vérification de la tâche

- **Nom** : *checkTache*
- **Description** : Vérifier si l'utilisateur a les droits pour pouvoir accéder à la page.
- **Classe** : *PapController*
- **Paramètre** :
 - Nom de la tâche → **Obligatoire**
- **Retour** : *Aucun*
- **Exemple** : `$this->checkTache("GESWPLREC");`
- **Code** :

```
public function checkTache(string $tache): void
{
    $tache = strtoupper($tache);
    $agence = $_GET["agence"];
    if (array_key_exists($tache, $_SESSION["Tache_Menu_Dashboard"]) &&
        ($_SESSION["Tache_Menu_Dashboard"][$tache]["ACCES"] != "0")) {
        header("Location: /app/dashboard?agence=" . $agence);
        exit();
    }
}
```

B. Nouveaux Web Services

1. Fournisseurs

- **Description** : Permet de récupérer tous les fournisseurs.
- **Route** :
 - **API** : POST
 - **Paramètres** : aucun
 - **URL** :
/ws/approvisionnement/planning-receptions/{agence}/fournisseurs
- **Code** :
 - **Contrôleur** : [PlanningReceptionsController](#)
 - **Manager** : [PlanningReceptionsManager](#)

2. Réceptions fournisseurs

- **Description** : Permet de récupérer toutes les réceptions de chaque fournisseur.
- **Route** :
 - **API** : POST
 - **Paramètre** :
 - Date de livraison → **Obligatoire**
 - Température
 - Code fournisseur
 - **URL** : /ws/approvisionnement/planning-receptions/{agence}/data
- **Code** :
 - **Contrôleur** : [PlanningReceptionsController](#)
 - **Manager** : [PlanningReceptionsManager](#)

3. Informations réception

- **Description** : Permet de récupérer les informations d'une réception fournisseur.
- **Route** :
 - **API** : POST
 - **Paramètres** :
 - Date de livraison → **Obligatoire**
 - Code fournisseur → **Obligatoire**
 - Température → **Obligatoire**
 - Heure de réception → **Obligatoire**
 - Quai de réception → **Obligatoire**
 - **URL** :
/ws/approvisionnement/planning-receptions/{agence}/reception
- **Code** :
 - **Contrôleur** : [PlanningReceptionsController](#)
 - **Manager** : [PlanningReceptionsManager](#)

4. Détails commande

- **Description** : Permet de récupérer le détail d'une commande d'une réception fournisseur.

- **Route :**
 - **API :** *POST*
 - **Paramètres :**
 - Code fournisseur → **Obligatoire**
 - Numéro commande → **Obligatoire**
 - **URL :**
/ws/approvisionnement/planning-receptions/{agence}/commande
- **Code :**
 - **Contrôleur :** [*PlanningReceptionsController*](#)
 - **Manager :** [*PlanningReceptionsManager*](#)

V. Frontend (Côté Client)

A. Langages et technologies

- **TWIG** : Moteur de rendu permettant de construire une page avec des templates.
- **Javascript** : Langage de programmation côté client.
- **HTML** : Langage de balisage permettant de construire une page web.
- **CSS** : Langage permettant d'ajouter du style à la page.

B. AJAX

1. Réceptions fournisseurs

- **Nom** : `getDataPlanningReceptions`
- **Paramètres** :
 - **date** : *String*
 - **service** : *String*
 - **codeFournisseur** : *String*
- **Méthode** : *POST*
- **Type** : *JSON*
- **Code** :

```
function getDataClientCommandes(date, service = "", codeFournisseur =
"") {
    let data;
    $.ajax({
        async: false,
        type: "POST",
        url: "/ws/approvisionnement/planning-receptions/{ { agence
    }}/data",
        data: {
            date: date,
            service: service,
            codeFournisseur: codeFournisseur
        },
        dataType: "json",
        success: function (result) {
            data = result;
        },
        error: function (req, err) {
            console.log("Error : " + err);
        }
    });
    return data;
}
```

2. Informations réception

- **Nom** : *getDataInfosReception*
- **Paramètres** :
 - **Date de livraison** : *String*
 - **Service de préparation** : *String*
 - **Code fournisseur** : *String*
 - **Heure de livraison** : *String*
 - **Quai de livraison** : *String*
- **Méthode** : *POST*
- **Type** : *JSON*
- **Code** :

```
function getDataInfosReception(date, service, codeFournisseur, heure,
quai) {
    let data;
    $.ajax({
        async: false,
        type: "POST",
        url: "/ws/approvisionnement/planning-receptions/{ { agence
    }}/reception",
        data: {
            date: date,
            service: service,
            codeFournisseur: codeFournisseur,
            heure: heure,
            quai: quai
        },
        dataType: "json",
        success: function (result) {
            data = result;
        },
        error: function (req, err) {
            console.log("Error : " + err);
        }
    });
    return data;
}
```

3. Détails commande

- **Nom** : *getDataDetailsCommande*
- **Paramètres** :
 - **Code fournisseur** : *String*
 - **Numéro de commande** : *String*
- **Méthode** : *POST*
- **Type** : *JSON*

- **Code :**

```
function getDataDetailsCommande(codeFournisseur, numCommande) {  
    let data;  
    $.ajax({  
        async: false,  
        type: "GET",  
        url: "/ws/approvisionnement/planning-receptions/{ { agence  
    }}/commande",  
        data: {  
            codeFournisseur: codeFournisseur,  
            numCommande: numCommande  
        },  
        dataType: "json",  
        success: function (result) {  
            data = result;  
        },  
        error: function (req, err) {  
            console.log("Error : " + err);  
        }  
    });  
    return data;  
}
```

VI. Annexes

A. Fournisseurs

1. Contrôleur

```
public function getAllFournisseurs(String $agence): void
{
    $boolConnexionReussi = $this->authentification($agence,
AuthMethodsEnum::BEARER);
    if (!$boolConnexionReussi) {
        return;
    }

    $this->createManager();
    $tabDonnees = $this->_objManager->getDataAllFournisseurs();

    if ($tabDonnees === false) {
        PapHttpResponse::code500_erreurInterneServeur();
    } elseif (!empty($tabDonnees)) {
        PapHttpResponse::code200_ok($tabDonnees);
    } else {
        PapHttpResponse::code204_aucuneDonneesTrouvees();
    }
}
```

2. Manager

```
public function getDataAllFournisseurs() {

    $request = "select distinct
                zffou as code_fournisseur,
                zfnom as nom_fournisseur
            from adfoba
            order by zffou";

    try {
        return $this->_connexion->requeteDb($request, [], true);
    } catch (Error $e) {
        return false;
    }
}
```

B. Réceptions fournisseurs

1. Contrôleur

```
public function getDataPlanningReceptions(String $agence): void
{
    $boolConnexionReussi = $this->authentification($agence,
AuthMethodsEnum::BEARER);
    if (!$boolConnexionReussi) {
        return;
    }

    $date = PapHttpRequest::getData("date");
    $service = PapHttpRequest::getData("service");
    $codeFournisseur = PapHttpRequest::getData("codeFournisseur");

    $params = [
        "agence" => $agence,
        "date" => $date
    ];

    if (!empty($service)) {
        $params["service"] = $service;
    }

    if (!empty($codeFournisseur)) {
        $params["codeFournisseur"] = $codeFournisseur;
    }

    $this->createManager();
    $tabDonnees =
$this->_objManager->getDataPlanningReceptions($params);

    if ($tabDonnees === false) {
        PapHttpResponse::code500_erreurInterneServeur();
    } elseif (!empty($tabDonnees)) {
        PapHttpResponse::code200_ok($tabDonnees);
    } else {
        PapHttpResponse::code204_aucuneDonneesTrouvees();
    }
}
```

2. Manager

```
public function getDataPlanningReceptions(Array $params) {
```



```

$request = "select zcefo as code_fournisseur,
            zfnom as nom_fournisseur,
            zcesv as service,
            zcefi as date_livraison,
            zcehl as heure_livraison,
            zcect as code_transporteur,
            zcent as nom_transporteur,
            zcequ as quai_reception,
            count( distinct zcenc) as Nbre_commande,
            case sum(zcepc)
                when 0 then f_mteqpa( zceag, sum(ARLONG *
ARLARG * ARHAUT * ZCDQC)/ 1000000000, sum(ARPB * ZCDQC)/ 1000)
                else sum(zcepc) end as estimation_palettes,
            sum(ARPB * ZCDQC)/ 1000 as poids_brut,
            sum(ARLONG * ARLARG * ARHAUT * ZCDQC)/ 1000000000 as
volume

            from adcfen
            join adcfde on zceag = zcdag and zcenc = zcdnu and ZCDAN
= 'N'

            join gcarte on zcdar = arartt
            join adfoba on zcefo = zffou
            where zceag = ?
                and zceru = 'N'
                and zcean = 'N'
                and zcefi = ?
            " . (array_key_exists("service", $params) ? " and
zcesv = ?" : "") . (array_key_exists("codeFournisseur", $params) ? " and
zcefo = ?" : "") . "
            group by zcefo,
                zfnom,
                zcesv,
                zcefi,
                zcehl,
                zcect,
                zcent,
                zcequ,
                zceag
            order by zcequ;";

    try {
        return $this->_connexion->requeteDb($request, $params,
true);
    } catch (Error $e) {
        return false;
    }
}

```

C. Informations réception

1. Contrôleur

```
public function getDataInfoReception(String $agence): void
{
    $boolConnexionReussi = $this->authentification($agence,
AuthMethodsEnum::BEARER);
    if (!$boolConnexionReussi) {
        return;
    }

    $date = PapHttpRequest::getData("date");
    $service = PapHttpRequest::getData("service");
    $codeFournisseur = PapHttpRequest::getData("codeFournisseur");
    $heure = PapHttpRequest::getData("heure");
    $quai = PapHttpRequest::getData("quai");

    $params = [
        "agence" => $agence,
        "date" => $date,
        "service" => $service,
        "codeFournisseur" => $codeFournisseur,
        "heure" => intval($heure),
        "quai" => $quai
    ];

    $this->createManager();
    $tabDonnees = $this->_objManager->getInfoReception($params);

    if ($tabDonnees === false) {
        PapHttpResponse::code500_erreurInterneServeur();
    } elseif (!empty($tabDonnees)) {
        PapHttpResponse::code200_ok($tabDonnees);
    } else {
        PapHttpResponse::code204_aucuneDonneesTrouvees();
    }
}
```

2. Manager

```
public function getInfoReception(Array $params) {
    $request = "select zcefo as code_fournisseur,
                zcesv as service,
                zcenc as numero_commande,
                zcerc as reference_commande,
                zcefi as date_livraison,
                zcehl as heure_livraison,
                zcect as code_transporteur,
```

```

        zcent as nom_transporteur,
        zceic as createur_commande,
        upnm as nom_createur_commande,
        zcequ as quai_reception,
        case zcepc
            when 0 then f_mteqpa( zceag, sum(ARLONG * ARLARG
* ARHAUT * ZCDQC)/ 1000000000, sum(ARPB * ZCDQC)/ 1000)
            else zcepc end as nbre_palette_sur_commande,
        sum(ARPB * ZCDQC)/ 1000 as poids_brut,
        sum(ARLONG * ARLARG * ARHAUT * ZCDQC)/ 1000000000 as
volume
        from dcf_fr.adcfen
        join adcfde on zceag = zcdag and zcenc = zcdnu and ZCDAN
= 'N'
        join gcarte on zcdar = arartt
        join adutil on zceic = uprf
        where zceag = ?      -- agence en cours
            and zceru = 'N'  -- cde non reçue
            and zcean = 'N' -- cde non annulée
            and zcefi = ?    -- à choisir dans les filtres
            and zcesv = ?    -- choix possible dans les filtres
mais pas obligatoire
            and zcefo = ?    -- fournisseur choisi lors du clic
            and zcehl = ?    -- heure de livraison choisi lors du
clic
            and zcequ = ?    -- quai choisi lors du clic
        group by zcefo,
            zcesv,
            zcenc,
            zceru,
            zcefi,
            zcehl,
            zcect,
            zcent,
            zceic,
            zcequ,
            zcepc,
            upnm,
            zceag ;";

        try {
            return $this->_connexion->requeteDb($request, $params,
true);
        } catch (Error $e) {
            return false;
        }
    }
}

```

D. Détails commande

1. Contrôleur

```
public function getDetailsCommande(String $agence): void
{
    $boolConnexionReussi = $this->authentification($agence,
AuthMethodsEnum::BEARER);
    if (!$boolConnexionReussi) {
        return;
    }

    $codeFournisseur = PapHttpRequest::getData("codeFournisseur");
    $numCommande = PapHttpRequest::getData("numCommande");

    $params = [
        "agence" => $agence,
        "codeFournisseur" => $codeFournisseur,
        "numCommande" => $numCommande
    ];

    $this->createManager();
    $tabDonnees = $this->_objManager->getDetailsCommande($agence,
$params);

    if ($tabDonnees === false) {
        PapHttpResponse::code500_erreurInterneServeur();
    } elseif (!empty($tabDonnees)) {
        PapHttpResponse::code200_ok($tabDonnees);
    } else {
        PapHttpResponse::code204_aucuneDonneesTrouvees();
    }
}
```

2. Manager

```
public function getDataPlateformesAndTournees(array $params): bool|array
{
    $strRequest = "select distinct
        trim(code_plth) as code_plateforme,
        trim(lib_plth) as nom_plateforme,
        (trim(code_plth) concat case when trim(lib_plth) = ''
then '' else (' - ' concat trim(lib_plth)) end) as libelle_plateforme,
        trim(tltol) as code_tournee,
        trim(tltxt) as nom_tournee,
        (trim(tltol) concat case when trim(tltxt) = '' then ''
```

```

else (' - ' concat trim(tltxt)) end) as libelle_tournee
    from gcto
    right join dic_plth on code_plth = tlplt and supr_plth = 'N'
    where tlage = ?
    and tlann = 'N'
    order by trim(code_plth), trim(tlitol);";

try {
    return $this->_connexion->requeteDb($strRequest, $params, true);
} catch (Error $e) {
    return false;
}
}

```

Insérer une commande dans une tournée

Contrôleur

```

public function insertCmdToTournee(string $agence) {

    $connexion = $this->authentication($agence, AuthMethodsEnum::BEARER);

    if (!$connexion) {
        return;
    }

    $agence = urldecode(PapHttpRequest::getData("agence", ""));
    $commande = intval(urldecode(PapHttpRequest::getData("commande", 0)));
    $date = urldecode(PapHttpRequest::getData("date"));
    $plateforme = urldecode(PapHttpRequest::getData("plateforme", ""));
    $tournee = urldecode(PapHttpRequest::getD

```