Request Params : RP

Response : res,

Response.status : RS creation :201, OK=200

(…) : (request,response,next)

Controllers :

1. Agencies.js
   * 1. getAllAgencies(req, res,next) =>( res -> all agencies )
     2. createAgency( … )=>( res -> 200)
     3. getAgencyById( … ) => (res -> agency )
     4. addEmployeeToAgency ( … )=> ( (RP = idEmployee, idAgency) -> 200 )
     5. addSubsidiaryToAgency(…)=>( (RP = idSubsidiary, idAgency) -> 200 )
2. Auth.js
   * 1. Login( … )=>((RP= login, ps)->RS=200)
     2. Signup(…) => ((RP=employe)->200)
3. Employees.js
   * 1. createEmployee(…)=>(res-> 200 )
     2. getAllEmployees(…)=>(res -> employees)
     3. getEmployeeId(…)=>(res -> employee)
4. Publications.js
   * 1. getAllPublications(…)=>(res -> publications)
     2. createPublications(…)=>(res -> 200)
     3. getPublicationById(…)=>(res -> publication)

Middleware :

1. Auth.js : (contient une seul methode pour controller l’authentification)
2. PublicationMiddelware.js (control toutes les gestions des publications)
   * 1. canApprovePublication (pour contrôler qui peut approuver une publication)

Models : (les entites)

1. Agency
2. Employee
3. Publication

Routes

1. Agencies
   * 1. Get( ‘/’, authMiddelware, AgencyController.getAllAgencies)
     2. Post(‘/’, authMiddelware,AgencyController.createAgency)
     3. Get(‘/:id’, authMiddelware,AgencyController.getAgencyById)
     4. post("/:idAgency/addSubsidiary\_id::idSubsidiary", authMiddelware, agencyController.addSubsidiaryToAgency);
     5. post("/:idAgency/addEmployee\_id::idEmployee", authMiddelware ,agencyController.addEmployeeToAgency);
2. Auth
   * 1. Post(‘/’, authMiddelware,authController.signup)
     2. Post(‘/’, authMiddelware,authController.login)
3. Employees
   * 1. Get(‘/’, authMiddelware,EmployeController.getAllEmployees)
     2. Post(‘/’, authMiddelware,EmployeController.createEmploye)
     3. Get(‘/ :id’, authMiddelware,EmployeController.getEmployById)
4. Publication
   * 1. Get(‘/’, authMiddelware, PubCtl.getAllPublications)
     2. Get(‘/:id’,authMiddelware, PubCtl.getPublicationById)

NB!!

Pour toutes les requêtes qui passent par authMiddelware il doivent ajouter la secureToken, reçue après le signin , à leurs headers