

Employee

August 18, 2026

Employee

Kind: Class

Source: Pams/Logic/Pams.Business/Entity/Person/Employees.Model.cs (line 15)

Part of: Pams

`Employee` is a C# entity model that stores a person's employment identity, branch assignment, demographic details, and emergency contact information. It links an employee to a person record through `PersonID` and may link them to a company branch through `CompanyBranchID`.

Diagram

```
graph LR
    Employee[Employee]
    Employee --> Identity[Identity]
    Employee --> Assignment[Assignment]
    Employee --> Personal[Personal Details]
    Employee --> Emergency[Emergency Contact]

    Identity --> EmployeeID[EmployeeID: long]
    Identity --> PersonID[PersonID: long]

    Assignment --> CompanyBranchID[CompanyBranchID: long?]

    Personal --> NationalityCode[NationalityCode: string]
    Personal --> Gender[Gender: int?]
    Personal --> BirthDate[BirthDate: DateTime?]
    Personal --> MaritalStatus[MaritalStatus: int?]

    Emergency --> EmergencyContactName[EmergencyContactName: string]
    Emergency --> EmergencyContactAddress[EmergencyContactAddress: string]
    Emergency --> EmergencyContactNumber[EmergencyContactNumber: string]
```

Usage

```

type EmployeePayload = {
  employeeID: bigint;
  personID: bigint;
  companyBranchID: bigint | null;
  nationalityCode: string;
  gender: number | null;
  birthDate: string | null;
  maritalStatus: number | null;
  emergencyContactName: string;
  emergencyContactAddress: string;
  emergencyContactNumber: string;
};

async function createEmployee(employee: EmployeePayload) {
  const response = await fetch("/employees", {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify(employee),
  });

  if (!response.ok) {
    throw new Error("Employee creation failed.");
  }

  return response.json();
}

```

AI Coding Instructions

- Keep `EmployeeID` and `PersonID` mapped to their corresponding long-valued identifiers in the .NET API contract.
- Treat `CompanyBranchID`, `Gender`, `BirthDate`, and `MaritalStatus` as nullable fields.
- Send `BirthDate` as an ISO date-time string when transferring employee data through JSON.
- Preserve emergency contact fields as separate values; do not combine them into a single display field.

Relationships

- IMPORTS → `Person`
- IMPORTS → `Employee`
- IMPORTS → `PersonBindingModel`
- IMPORTS → `EntityBase`

- IMPORTS → Gender
- IMPORTS → MaritalStatus
- IMPORTS → PersonUser
- IMPORTS → PamsLinqDataContext

Used by

18 references from 18 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (18)

- BaseController — Pams/API/Pams.API/Controllers/BaseController.cs :515
- FilesController — Pams/API/Pams.API/Controllers/Common/FilesController.cs :27
- CompanyBranchController —
Pams/API/Pams.API/Controllers/Organization/CompanyBranchController.cs :21
- EmployeeController — Pams/API/Pams.API/Controllers/Person/EmployeeController.cs :29
- PersonAddressController —
Pams/API/Pams.API/Controllers/Person/PersonAddressController.cs :21
- PersonEmailController —
Pams/API/Pams.API/Controllers/Person/PersonEmailController.cs :22
- PersonController — Pams/API/Pams.API/Controllers/Person/PersonController.cs :36
- PersonPhoneController —
Pams/API/Pams.API/Controllers/Person/PersonPhoneController.cs :21

...and 10 more.