

City

August 18, 2026

City

Kind: Class

Source: Pams/Logic/Pams.Business/Entity/Common/City.Model.cs (line 14)

Part of: [Pams](#)

`City` is a C# entity model that stores city identity, naming, country, governate, and company association data. It is used where the system needs to represent a city record and connect it to related geographic or company-scoped data.

Diagram

```
graph LR
    City[City]
    City --> CityID[CityID: int]
    City --> GovernateID[GovernateID: int?]
    City --> CountryCode[CountryCode: string]
    City --> CompanyId[CompanyId: int?]
    City --> Name[Name: string]
```

Usage

```
type City = {
    CityID: number;
    GovernateID: number | null;
    CountryCode: string;
    CompanyId: number | null;
    Name: string;
};

async function getCity(cityId: number): Promise<City> {
    const response = await fetch(`/api/cities/${cityId}`);

    if (!response.ok) {
        throw new Error("Unable to load city.");
    }
}
```

```

    }

    return response.json();
}

const city = await getCity(selectedCityId);

console.log(`${city.Name}, ${city.CountryCode}`);

```

AI Coding Instructions

- Keep property names aligned with the C# model: `CityID` , `GovernateID` , `CountryCode` , `CompanyId` , and `Name` .
- Treat `GovernateID` and `CompanyId` as nullable values when mapping the entity to API contracts or TypeScript types.
- Preserve `CountryCode` as a string; do not convert it to a numeric identifier.
- Use `CityID` as the city record identifier when creating routes, queries, or update requests.
- Confirm company-scoping behavior before filtering by `CompanyId` , since the field can be null.

Relationships

- IMPORTS → `City`
- IMPORTS → `EntityBase`
- IMPORTS → `PamsLinqDataContext`

Used by

13 references from 13 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (13)

- `CityController` — `Pams/API/Pams.API/Controllers/Common/CityController.cs` :20
- `CompanyController` — `Pams/API/Pams.API/Controllers/Organization/CompanyController.cs` :26
- `EmployeeController` — `Pams/API/Pams.API/Controllers/Person/EmployeeController.cs` :29
- `PersonAddressController` — `Pams/API/Pams.API/Controllers/Person/PersonAddressController.cs` :21

- Account — Pams/Data/Pams.Data/PamsLinq5.designer.cs :166863
- AccountViewModel —
Pams/Logic/Pams.Business/BindingModels/Person/AccountsBindingModel.cs :6
- PersonUserBindingModel —
Pams/Logic/Pams.Business/BindingModels/Security/PersonUserBindingModel.cs :7
- ReturnData — Pams/Logic/Pams.Business/Component/NewSales/SalesOfferLogic.cs :2637

...and 5 more.