

Employee Excel Import Task - Complete Guide

1. Overview

- Upload Excel file containing employees via React frontend.
- API receives file, parses Excel to ExcelDataModel list.
- Map ExcelDataModel -> ImportEmployeeData using AutoMapper.
- Save an ImportMaster record + list of ImportEmployeeData.
- Admin triggers "Process" action that converts imported data into actual Employee records in DB.

2. Employee.cs

```
public class Employee : EntityBase {
    public Guid UserId { get; set; }
    public string EmployeeCode { get; set; }
    public string EmployeeName { get; set; }
    public DateTime? DateOfJoining { get; set; }
    public DateTime? DateOfBirth { get; set; }
    public Guid? DesignationId { get; set; }
    public double? PriorExperience { get; set; }
    public double? CompanyExperience { get; set; }
    public double? TotalExperience { get; set; }
    public Guid? DepartmentId { get; set; }
    public Guid? TechnologyId { get; set; }
    public Guid? ProjectId { get; set; }
    public DateTime? AllotedFrom { get; set; }
    public bool IsBillable { get; set; } = false;
    public Guid? UserRoleId { get; set; }
    public Guid? ReportingManager { get; set; }
    public Guid? ProjectLead { get; set; }
    public Guid? ProjectManager { get; set; }
    public Guid? DepartmentHead { get; set; }
    public Guid? PrimaryProjectId { get; set; }
    public Guid? LocationId { get; set; }
    public string Remarks { get; set; }
    public string EmailId { get; set; }
}
```

3. ExcelDataModel.cs

```
public class ExcelDataModel {
    public int SequenceNo { get; set; }
    public string EmployeeID { get; set; }
    public string Name { get; set; }
    public string DOJ { get; set; }
    public string Designation { get; set; }
    public string DOB { get; set; }
    public string PriorExperience { get; set; }
    public string CompanyExperience { get; set; }
    public string TotalExperience { get; set; }
    public string Department { get; set; }
    public string Technology { get; set; }
    public string Project { get; set; }
    public string AllotedFrom { get; set; }
    public string Billed { get; set; }
    public string UserRole { get; set; }
    public string ReportingManager { get; set; }
    public string ProjectLead { get; set; }
    public string ProjectManager { get; set; }
    public string DepartmentHead { get; set; }
    public string PrimaryProject { get; set; }
}
```

Employee Excel Import Task - Complete Guide

```
public string Location { get; set; }
public string Remarks { get; set; }
}
```

4. AutoMapper Profile

```
CreateMap<ExcelDataModel, ImportEmployeeData>()
    .ForMember(dest => dest.EmployeeId, opt => opt.MapFrom(src => src.EmployeeID))
    .ForMember(dest => dest.Name, opt => opt.MapFrom(src => src.Name))
    // Other properties mapped similarly...
```

5. ProcessEmployeeImport

```
public async Task ProcessEmployeeImport(Guid masterId) {
    var importData = await _context.ImportEmployeeData.Where(i => i.ImportMasterId == masterId).ToListAsync();
    var employees = new List<Employee>();
    foreach (var item in importData) {
        var employee = new Employee {
            UserId = Guid.NewGuid(),
            EmployeeCode = item.EmployeeId,
            EmployeeName = item.Name,
            DateOfJoining = ParseDate(item.DOJ),
            // more properties...
        };
        employees.Add(employee);
    }
    await _context.Employees.AddRangeAsync(employees);
    await _context.SaveChangesAsync();
}
```

6. ImportEmployeesController

```
[HttpPost("ImportEmployeeFile")]
public async Task<ActionResult> ImportEmployeeFile() {
    var file = Request.Form.Files.FirstOrDefault();
    if (file == null || file.Length == 0)
        return BadRequest("No file uploaded.");
    using (var stream = new MemoryStream()) {
        await file.CopyToAsync(stream);
        // Parse using EPPlus
    }
    await _importMasterRepository.SaveImport(master, employees);
    return Ok(new { Message = "File imported", ImportId = master.Id });
}
```

7. React Frontend Upload

```
const onUpload = async () => {
    const formData = new FormData();
    formData.append("file", file);
    await axios.post("/api/ImportEmployees/ImportEmployeeFile", formData, {
        headers: { "Content-Type": "multipart/form-data" }
    });
};
```

8. Checklist

- [x] Models: Employee, ExcelDataModel
- [x] AutoMapper Profile
- [x] Repository Save & Process Methods
- [x] API Controller Endpoints

Employee Excel Import Task - Complete Guide

- [x] React Frontend Upload Form
- [x] Excel Parser Utility (Optional)