Universidad Rafael Landívar

Facultad de Ingeniería

Ingeniería en Informática y Sistemas

Javier Andrés Morales

1210219

API: Endpoints por controlador

**Documentación de los Endpoints del Controlador de API Departamento**

Este controlador de API tiene como objetivo realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre la entidad "Departamento". A continuación, se presenta la documentación de cada uno de los endpoints definidos en este controlador:

**Create**

**POST /api/Departamento**

Este endpoint permite crear un nuevo departamento en el sistema.

Parámetros de entrada

El endpoint espera recibir un objeto JSON que represente un nuevo departamento. El formato del objeto es el siguiente:

* **IdDepartamento** (requerido): identificador único del departamento. Debe ser un número entero.
* **IdDireccion** (requerido): identificador de la dirección a la que pertenece el departamento. Debe ser un número entero.
* **NombreDepartamento** (requerido): nombre del departamento. Debe ser una cadena de caracteres.
* **DescripcionDepartamento** (opcional): descripción del departamento. Debe ser una cadena de caracteres.

Respuestas

* 200 OK: el departamento se creó correctamente.
* 400 Bad Request: se produjo un error al intentar crear el departamento.

**Reads**

**GET /api/Departamento**

Este endpoint permite obtener la lista de todos los departamentos del sistema.

Parámetros de entrada

El endpoint no espera recibir ningún parámetro.

Respuestas

* 200 OK: la lista de departamentos se devolvió correctamente en formato JSON.
* 404 Not Found: no se encontró ningún departamento en el sistema.

**GET /api/Departamento/{id}**

Este endpoint permite obtener los detalles de un departamento específico en el sistema.

Parámetros de entrada

El endpoint espera recibir un parámetro "id" que representa el identificador único del departamento a buscar.

Respuestas

* 200 OK: se devolvieron correctamente los detalles del departamento en formato JSON.
* 404 Not Found: no se encontró el departamento con el identificador especificado.

**Update**

**PUT /api/Departamento/{id}**

Este endpoint permite actualizar un departamento existente en el sistema.

Parámetros de entrada

El endpoint espera recibir un objeto JSON que represente el departamento actualizado. El formato del objeto es el mismo que el utilizado para crear un nuevo departamento.

Respuestas

* 200 OK: el departamento se actualizó correctamente.
* 400 Bad Request: se produjo un error al intentar actualizar el departamento.

**Delete**

**DELETE /api/Departamento/{id}**

Este endpoint permite eliminar un departamento existente en el sistema.

Parámetros de entrada

El endpoint espera recibir un parámetro "id" que representa el identificador único del departamento a eliminar.

Respuestas

* 200 OK: el departamento se eliminó correctamente.
* 404 Not Found: no se encontró el departamento con el identificador especificado.

**Documentación del Controlador DireccionController**

El Controlador **DireccionController** maneja las solicitudes HTTP relacionadas con la entidad "Dirección" en el sistema de manejo de empleados.

**Endpoints**

**Create**

Este endpoint crea una nueva dirección.

* **URL**: **/api/Direccion**
* **Método HTTP**: **POST**
* **Cuerpo de la solicitud**: objeto **SistemamanejoEmpleadosModel.Direccion** con los siguientes atributos:
  + **IdDireccion** (int): Identificador único de la dirección.
  + **NombreDireccion** (string): Nombre de la dirección.
  + **DescripcionDireccion** (string): Descripción de la dirección.
  + **Dpidirector** (int): Identificador único del director asociado a la dirección.
* **Respuesta exitosa**: código de estado HTTP **200 OK**.
* **Respuesta fallida**: código de estado HTTP **400 Bad Request**.

**Reads**

Estos endpoints leen una o varias direcciones.

* **URL**: **/api/Direccion**
* **Método HTTP**: **GET**
* **Respuesta exitosa**: objeto JSON que contiene un arreglo de objetos **SistemamanejoEmpleadosModel.Direccion** con los siguientes atributos:
  + **IdDireccion** (int): Identificador único de la dirección.
  + **NombreDireccion** (string): Nombre de la dirección.
  + **DescripcionDireccion** (string): Descripción de la dirección.
  + **Dpidirector** (int): Identificador único del director asociado a la dirección.
* **Respuesta fallida**: código de estado HTTP **404 Not Found**.
* **URL**: **/api/Direccion/{id}**
* **Método HTTP**: **GET**
* **Parámetros de la solicitud**: **id** (int) es el identificador único de la dirección.
* **Respuesta exitosa**: objeto JSON que contiene un objeto **SistemamanejoEmpleadosModel.Direccion** con los siguientes atributos:
  + **IdDireccion** (int): Identificador único de la dirección.
  + **NombreDireccion** (string): Nombre de la dirección.
  + **DescripcionDireccion** (string): Descripción de la dirección.
  + **Dpidirector** (int): Identificador único del director asociado a la dirección.
* **Respuesta fallida**: código de estado HTTP **404 Not Found**.

**Update**

Este endpoint actualiza una dirección existente.

* **URL**: **/api/Direccion/{id}**
* **Método HTTP**: **PUT**
* **Parámetros de la solicitud**: **id** (int) es el identificador único de la dirección.
* **Cuerpo de la solicitud**: objeto **SistemamanejoEmpleadosModel.Direccion** con los siguientes atributos:
  + **IdDireccion** (int): Identificador único de la dirección.
  + **NombreDireccion** (string): Nombre de la dirección.
  + **DescripcionDireccion** (string): Descripción de la dirección.
  + **Dpidirector** (int): Identificador único del director asociado a la dirección.
* **Respuesta exitosa**: código de estado HTTP **200 OK**.
* **Respuesta fallida**: código de estado HTTP **400 Bad Request**.

**Delete**

Este endpoint elimina una dirección existente.

* **URL**: **/api/Direccion/{id}**
* **Método HTTP**: **DELETE**
* **Parámetros de la solicitud**: **id** (int) es el identificador único de la dirección.
* **Respuesta exitosa**: código de estado HTTP `200 OK

**Documentación del Controlador DirectorController**

* **Crear un director:**
  + **Endpoint: POST /api/Director**
  + **Body: objeto SistemamanejoEmpleadosModel.Director con propiedades Dpi y Bonos**
  + **Descripción: Crea un nuevo director con los datos proporcionados en el cuerpo de la solicitud.**
* **Obtener todos los directores:**
  + **Endpoint: GET /api/Director**
  + **Descripción: Obtiene una lista de todos los directores de la empresa con sus datos, incluyendo el nombre de la persona asociada al DPI del director.**
* **Obtener un director específico:**
  + **Endpoint: GET /api/Director/{id}**
  + **Parámetros de URL: id (entero largo) - el DPI del director que se desea obtener**
  + **Descripción: Obtiene los datos de un director específico con el DPI proporcionado.**
* **Actualizar un director:**
  + **Endpoint: PUT /api/Director/{id}**
  + **Parámetros de URL: id (entero largo) - el DPI del director que se desea actualizar**
  + **Body: objeto SistemamanejoEmpleadosModel.Director con las propiedades Dpi y Bonos actualizadas**
  + **Descripción: Actualiza los datos de un director específico con el DPI proporcionado.**
* **Eliminar un director:**
  + **Endpoint: DELETE /api/Director/{Dpi}**
  + **Parámetros de URL: Dpi (entero largo) - el DPI del director que se desea eliminar**
  + **Descripción: Elimina un director específico con el DPI proporcionado.**

**Documentación del Controlador EmpleadoController**

Crear un nuevo empleado:

* Método: POST
* Endpoint: /api/empleado
* Parámetros: Objeto JSON con la información del nuevo empleado
* Respuesta: Ok

Obtener todos los empleados:

* Método: GET
* Endpoint: /api/empleado
* Respuesta: Objeto JSON con una lista de todos los empleados

Obtener un empleado específico:

* Método: GET
* Endpoint: /api/empleado/{id}
* Parámetros: ID del empleado a obtener
* Respuesta: Objeto JSON con la información del empleado específico

Editar un empleado:

* Método: PUT
* Endpoint: /api/empleado/{id}
* Parámetros: Objeto JSON con la información editada del empleado y el ID del empleado a editar
* Respuesta: Ok si se editó correctamente, BadRequest en caso contrario

Eliminar un empleado:

* Método: DELETE
* Endpoint: /api/empleado/{id}
* Parámetros: ID del empleado a eliminar
* Respuesta: Ok si se eliminó correctamente, BadRequest en caso contrario

**Documentación del Controlador PersonaController**

POST: api/Persona Descripción: Crea una nueva persona. Parámetros: Un objeto de tipo SistemamanejoEmpleadosModel.Persona. Método HTTP: POST

GET: api/Persona Descripción: Retorna una lista de todas las personas. Parámetros: Ninguno. Método HTTP: GET

GET: api/Persona/{id} Descripción: Retorna una persona específica. Parámetros: El id de la persona que se desea retornar. Método HTTP: GET

PUT: api/Persona/{id} Descripción: Actualiza una persona existente. Parámetros: El id de la persona que se desea actualizar y un objeto de tipo SistemamanejoEmpleadosModel.Persona. Método HTTP: PUT

DELETE: api/Persona/{Dpi} Descripción: Elimina una persona existente. Parámetros: El Dpi de la persona que se desea eliminar. Método HTTP: DELETE

**Documentación del Controlador PuestoController**

POST api/Puesto

Crea un nuevo puesto en la base de datos. Se espera un objeto JSON con la siguiente estructura:

* idPuesto (int): El identificador único del puesto.
* nombrePuesto (string): El nombre del puesto.
* descripcionPuesto (string): Una descripción del puesto.

Obtener todos los puestos

GET api/Puesto

Retorna todos los puestos almacenados en la base de datos.

Obtener un puesto por id

GET api/Puesto/{id}

Retorna el puesto con el id especificado. El id se especifica en la URL de la solicitud.

Actualizar un puesto

PUT api/Puesto/{id}

Actualiza los datos del puesto con el id especificado. Se espera un objeto JSON con la siguiente estructura:

* idPuesto (int): El identificador único del puesto.
* nombrePuesto (string): El nombre actualizado del puesto.
* descripcionPuesto (string): La descripción actualizada del puesto.

Eliminar un puesto

DELETE api/Puesto/{id}

Elimina el puesto con el id especificado. El id se especifica en la URL de la solicitud.

**Documentación del Controlador TareaController**

* POST /api/Tarea
  + Crea una nueva tarea en la base de datos.
  + Parámetros requeridos: un objeto JSON que incluya los siguientes campos:
    - Idtarea (long)
    - DpiempleadoAsignado (long)
    - NombreTarea (string)
    - RequerimientosTarea (string)
    - FechaCreacion (DateTime)
    - FechaLimite (DateTime)
    - Estado (string)
    - Comentarios (string)
* GET /api/Tarea
  + Obtiene una lista de todas las tareas en la base de datos.
  + Parámetros requeridos: ninguno.
* GET /api/Tarea/{id}
  + Obtiene una tarea específica de la base de datos según su Idtarea.
  + Parámetros requeridos: un número entero que representa el Idtarea de la tarea que se desea obtener.
* PUT /api/Tarea/{id}
  + Actualiza una tarea existente en la base de datos.
  + Parámetros requeridos: un objeto JSON que incluya los siguientes campos:
    - Idtarea (long)
    - DpiempleadoAsignado (long)
    - NombreTarea (string)
    - RequerimientosTarea (string)
    - FechaCreacion (DateTime)
    - FechaLimite (DateTime)
    - Estado (string)
    - Comentarios (string)
* DELETE /api/Tarea/{id}
  + Elimina una tarea específica de la base de datos según su Idtarea.
  + Parámetros requeridos: un número entero que representa el Idtarea de la tarea que se desea eliminar.