

Universidad de San Carlos de Guatemala
Centro Universitario de Occidente División de Ciencias de la Ingeniería
Introducción a la Programación 2
Ing. Jose Granados Guevara
Sección: A

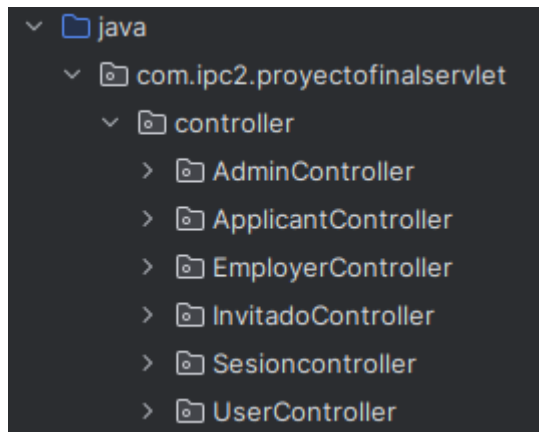


Manual Técnico

Carlos Raúl Alberto López Peláez 202031871

Quetzaltenango, 16 de enero del 2024

Controladores



La aplicación "proyectofinalservlet" tiene los siguientes controladores:

- AdminController se encargó de procesar las peticiones relacionadas con el administrador de la aplicación.
- ApplicantController se encargó de procesar las peticiones relacionadas con los Aplicantes.
- EmployerController se encargó de procesar las peticiones relacionadas con los Empleadores.
- InvitadoController se encargó de procesar las peticiones relacionadas con los Invitados.
- SesionController encargado de procesar las peticiones relacionadas con la Sesión.
- UserController encargado de procesar las peticiones relacionadas con los usuarios en general.

AdminController

El controlador AdminController se encarga de procesar las peticiones relacionadas con el administrador de la aplicación. Estas peticiones pueden incluir las siguientes:

Peticiones Get

- Endpoint: /cargar-categorias
Descripción: Retorna una lista de categorías desde la base de datos.
- Endpoint: /cargar-categoria
Descripción: Retorna los detalles de una categoría específica, identificada por su código.
- Endpoint: /listar-dashboard

Descripción: Retorna información del dashboard desde la base de datos

- Endpoint: /listar-comision
Descripción: Retorna información sobre comisiones desde la base de datos
- Endpoint: /listar-usuarios
Descripción: Retorna una lista de usuarios filtrados por el rol proporcionado como parámetro.
- Endpoint: /listar-usuario
Descripción: Retorna los detalles de un usuario específico, identificado por su código.
- Endpoint: /listar-usuario-especifico
Descripción: Retorna los detalles del usuario que realizó la solicitud.
- Endpoint: /listar-telefonos
Descripción: Retorna una lista de números de teléfono asociados a un usuario, identificado por su código.
- Endpoint: /listar-telefonos-usuario-especifico
Descripción: Retorna una lista de números de teléfono asociados al usuario que realizó la solicitud.
- Endpoint: /listar-curriculum
Descripción: Retorna el curriculum de un usuario en formato PDF, identificado por su código.
- Endpoint: /suspender-usuario
Descripción: Actualiza el estado de suspensión de un usuario, identificado por su nombre de usuario, según el estado proporcionado como parámetro.

Cada uno de estos servicios se encarga de manejar una operación específica dentro de la aplicación web, proporcionando la información solicitada desde la base de datos o realizando acciones en consecuencia.

Peticiones Post

Endpoint: /gestionar-categorias-crear

Descripción: Crea una nueva categoría en la base de datos. Lee los datos de la categoría desde el cuerpo de la solicitud en formato JSON.

Endpoint: /crear-registro-comision

Descripción: Registra una nueva comisión en la base de datos. Lee los datos de la comisión desde el cuerpo de la solicitud en formato JSON.

Endpoint: /crear-usuarios

Descripción: Crea un nuevo usuario en la base de datos. Lee los datos del usuario y sus teléfonos asociados desde el cuerpo de la solicitud en formato JSON.

Endpoint: /crear-telefonos

Descripción: Crea nuevos registros de números de teléfono en la base de datos. Lee los datos de los teléfonos desde el cuerpo de la solicitud en formato JSON.

Endpoint: /crear-telefonos-usuario

Descripción: Crea nuevos registros de números de teléfono asociados a un usuario específico en la base de datos. Lee los datos de los teléfonos desde el cuerpo de la solicitud en formato JSON.

En cada caso, se realizan validaciones antes de llevar a cabo la acción correspondiente, y se establecen códigos de respuesta adecuados para indicar el éxito o el fallo de la operación. Por ejemplo, se devuelve un código 201 (CREATED) en caso de éxito y códigos 400 (BAD_REQUEST) en caso de problemas como datos incorrectos o faltantes. Además, se verifica si el usuario que realiza la solicitud tiene el rol de "Administrador" antes de permitir el acceso a estas funcionalidades.

Peticiones Put

Endpoint: /gestionar-categorias-actualizar

Descripción: Actualiza los detalles de una categoría existente en la base de datos. Lee los nuevos datos de la categoría desde el cuerpo de la solicitud en formato JSON.

Endpoint: /actualizar-comision

Descripción: Actualiza la cantidad de una comisión existente en la base de datos. Lee los nuevos datos de la comisión desde el cuerpo de la solicitud en formato JSON.

Endpoint: /actualizar-usuario

Descripción: Actualiza los detalles de un usuario existente en la base de datos. Lee los nuevos datos del usuario desde el cuerpo de la solicitud en formato JSON.

Endpoint: /actualizar-telefonos

Descripción: Actualiza los números de teléfono asociados a un usuario específico en la base de datos. Lee los nuevos datos de los teléfonos desde el cuerpo de la solicitud en formato JSON.

Endpoint: /actualizar-contrasena

Descripción: Actualiza la contraseña de un usuario existente. Lee los nuevos datos de la contraseña desde el cuerpo de la solicitud en formato JSON y utiliza el servicio cambiarContrasena para llevar a cabo el cambio.

Peticiones Delete

Endpoint: /eliminar-usuario

Descripción: Elimina un usuario de la base de datos. Utiliza el parámetro username de la solicitud para identificar el usuario a eliminar.

Endpoint: /eliminar-categoria

Descripción: Elimina una categoría de la base de datos. Utiliza el parámetro código de la solicitud para identificar la categoría a eliminar.

Admin Reports

Generación de Informes PDF:

- El servlet responde a las solicitudes GET y genera informes PDF basados en plantillas JasperReport.
- Se utilizan diferentes plantillas JasperReport almacenadas en la ruta especificada en la variable resources.
- Se manejan casos especiales, como la generación de informes con parámetros de fechas y categorías, y se ajusta el nombre del informe y los parámetros en consecuencia.
- El contenido del informe se genera utilizando JasperReports y se envía como una respuesta HTTP con el tipo de contenido "application/pdf".

Admin Reports_Changer Servlet

Manejo de Solicitudes GET:

- El servlet maneja diferentes solicitudes GET según la URI proporcionada.
- Listar Top Empleadores:
Si la URI termina con "/listar-top-empleadores", se obtiene la lista de los mejores empleadores a través del método topEmpleadores y se envía como respuesta en formato JSON.

- Listar Cantidad Total:

Si la URI termina con "/listar-cantidad-total", se obtienen los parámetros de fecha, categoría y valor mediante el método obtenerParametros.

Se utiliza la información para decidir si se debe listar la cantidad total con o sin fecha. El resultado se envía como respuesta en formato JSON.

- Listar Registro de Comisión:
- Si la URI termina con "/listar-registro-comision", se obtiene la lista de registros de comisión a través del método listarRegistroComision y se envía como respuesta en formato JSON.

Ingresos por Fecha:

Si la URI termina con "/ingresos-fecha", se obtienen los parámetros de fecha mediante el método obtenerParametros.

Se utiliza la información para obtener la lista de ingresos totales por fecha a través del método `ingresosTotales`, y se envía como respuesta en formato JSON.

ApplicantController

El controlador `ApplicantController` se encarga de procesar las peticiones relacionadas con los solicitantes de empleo. Estas peticiones pueden incluir las siguientes:

Applicant PDF

Applicant Reports

Applicant Controller

Applicant Interview Controller

Applicant Nomination Controller

Applicant Reports Changer Controler

EmployerController

El controlador `EmployerController` se encarga de procesar las peticiones relacionadas con los empleadores. Estas peticiones pueden incluir las siguientes:

Employer Reports

Nomination Controller

Employer Reports Changer Controler

InvitadoController

El controlador `InvitadoController` se encarga de procesar las peticiones relacionadas con los visitantes de la aplicación. Estas peticiones pueden incluir las siguientes:

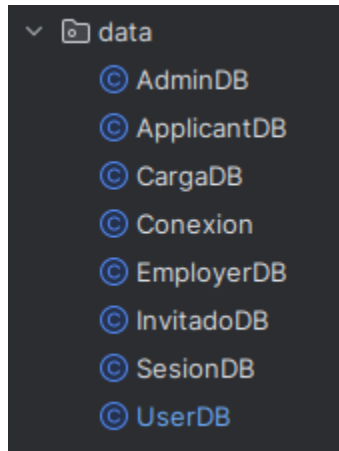
SesionController

El controlador `SesionController` se encarga de procesar las peticiones relacionadas con las sesiones de los usuarios. Estas peticiones pueden incluir las siguientes:

UserController

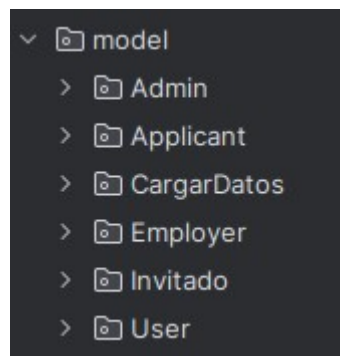
El controlador `UserController` se encarga de procesar las peticiones relacionadas con los usuarios en general. Estas peticiones pueden incluir las siguientes:

Peticiones a base de datos



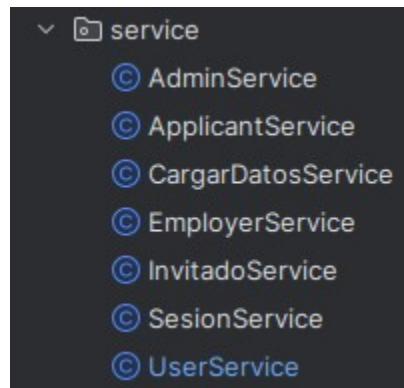
Para conectarse a la base de datos, los controladores utilizan la clase Connection de JDBC. Esta clase proporciona una interfaz para establecer una conexión con la base de datos y realizar operaciones sobre ella.

Modelos (procesado de datos para petición)



Los modelos solo sirven para procesar los datos en los servicios, servlet y la conexión a base de datos, son encargados de llevar la información y devolverla de ser necesario.

Servicios



Los servicios son consecuentes con los servlets es decir son intermediario entre la base de datos y el servlet , manejan los datos del servlet, su funcionalidad es cumplir con lo que el servlet pide

En general, los controladores procesan las peticiones de la siguiente manera:

El controlador recibe la petición del cliente.

El controlador utiliza el modelo para obtener los datos necesarios para procesar la petición.

El controlador procesa la petición.

El controlador devuelve la respuesta a la vista.

