

SOLICITEL

Backend Api Routes

Software Engineering 2 Team 5 - 2024

Content table

1.	<i>API WIDE ERRORS</i>	3
2.	<i>USER AUTHORIZATION</i>	3
3.	<i>ROUTES</i>	3
3.1	Api	3
3.2	User	4
3.3	Proyecto	5
3.4	Solicitud	7

1. API WIDE ERRORS

- 400 Bad Request: {error: "Error on request" }
- 401 Unauthorized: {error: "Token not valid "} or {error: "No token"} or {error: "Token information not valid"}
- 403 Forbidden: { error: 'Auth not valid' }
- 404 Page not found: { error: "Page not found" }
- 500 Internal Server Error { error: "Something Broke", message: error message}

2. USER AUTHORIZATION

To perform any of the operations described below, the user must log in and obtain the JWT authorization token. This token must be used in the authorization field of the request header. This token will be obtained through the "/token" endpoint which is the only endpoint that does not require authorization.

3. ROUTES

3.1 Api

These endpoints are used to manage user login. To enter the endpoints of this section you must precede "api/".

Route	Send	Function	Response
POST /token	Username: String Password: String	Return Json with access and refresh token	200: Json list with tokens 401: error message, no active account found

3.2 User

Used for the creation of users and roles. To enter the endpoints of this section you must precede "user/".

Route	Send	Function	Response
GET /roles	None	Return Json with list of all content in user_role table	200: Json list of Roles 400: error message
POST /roles	Description: String	Add a Role to the database	200: {id, description} 400: error message, missing field
DELETE /roles/delete/<int:pk>	None	Delete Role with pk in the database	204: Delete complete 404: No role found with pk provided
POST /registro	Username: String Password: String	Add a User to the database	201: {id, username} 400: error message, User with username provided already exists
GET /rol	None	Return a Json with Role information of logged user	200: {id, description} 404: error message, no role found
GET /perfil	None	Return a Json with User profile information of logged user	200: {id, user, email, name, cedula} 500: error message, internal server error

3.3 Proyecto

Used for project and application management. To enter the endpoints of this section you must precede "api/".

Route	Send	Function	Response
GET /proyectos	None	Return Json with list of all projects created by the user logged	200: Json list of Proyectos or empty list if none
POST /proyectos	Nombre: String Project_budget: int Budget_items: List of Budget items	Add a Proyecto to the database	201: {id} 400: error message, missing field 500: internal server error
GET /proyectos/<int:pk>	None	Return Json with rows in proyectos table with the pk provided	200: {id, nombre, fecha_creacion, usuario_creacion, budget_items} 404: error message, no project found
PUT /proyectos/<int:pk>	Nombre: String Project_budget: int Usuario_creacion: int Budget_items: List of Budget items	Update a Proyecto with pk in the database	200: {id, nombre, fecha_creacion, Usuario_creacion, budget_items} 400: error message, missing field

			404: error message, no project found
DELETE /proyectos/<int:pk>	None	Delete Proyecto with pk in the database	204: Delete complete 404: error message, no project found
GET /proyectos/all	None	Return Json with list of all projects in database	200: Json list of Proyectos or empty if none 500: internal server error
GET /proyectos/<int:pk>/items	None	Return Json with list of all budget_items from Proyecto with pk in the database	200: Json list of Budget items or empty if none 404: error message, no project found 500: internal server error
POST /proyectos/<int:pk>/items	Budget_items: List of budget items {recurso, categoria, cantidad, valor, presupuesto, proyecto}	Add the list of budget_items to the project with pk provided in the database	200: {proyecto_id} 400: error message, missing field 404: error message, no project found 500: internal server error

3.4 Solicitud

Used to create and manage applications, along with their quotations and forms. To enter the endpoints of this section you must precede "api/".

Route	Send	Function	Response
GET /solicitudes	None	Return Json with list of all Solicitudes from the database created by logged user	200: Json list of Solicitudes or empty if none
GET /solicitudes/<int:pk>	None	Return Json with rows in solicitudes table with the pk provided	200: Json with Solicitud 404: error message, no solicitud found 500: internal server error
PUT /solicitudes/<int:pk>	Cotizacion_aceptada: Cotizacion	Update a Solicitud with pk in the database	200: {id} 400: error message, bad request 404: error message, no solicitud found 500: internal server error
GET /solicitudes/all	None	Return Json with list of all Solicitudes from the database	200: Json list of Solicitudes or empty if none 500: internal server error
POST /proyectos/<int:pk>/	Nombre: String Tema: String	Add a new Solicitud to the	201: {solicitud_id}

solicitud	Tipo: String Estado: Estado	project with pk provided in the database	400: error message, bad request 404: error message, no proyecto found 500: internal server error
GET solicitudes/<pk_s>/items	None	Return Json with list of all Items of Solicitud with pk_s from the database	200: Json list of Items or empty if none 500: internal server error
POST solicitudes/<pk_s>/items	item: int descripcion: String cantidad: int unidad: int	Add a new Item to the Solicitud with pk_s provided in the database	201: {id, item, solicitud, fecha_creacion, descripcion, cantidad, unidad} 400: error message, bad request 404: error message, no solicitud found 500: internal server error
GET /proyectos/<int:pk>/<int:pk_s>/estado	None	Return Json with Estado of Solicitud with the pk_s provided	200: Json with estado 404: error message, no solicitud found 500: internal server error
PUT proyecto/<int:pk_p>/solicitud/<int:pk_s>/	None	Update Estado from Solicitud with pk_s of Proyecto	200: Status success 400: error message, bad request

estado/<int:pk_e>		with pk_p in the database	404: error message, no Solicitud found 500: internal server error
GET solicitudes/<int:pk>/cotizaciones	None	Return Json with list of Cotizaciones from Solicitud with the pk provided	200: Json list of Cotizaciones or empty if none 404: error message, no Solicitud found 500: internal server error
POST solicitudes/<int:pk>/cotizaciones	Proveedor: String No_coti: int Monto: int url_coti: Url fecha_coti: Date	Add a new Cotizacion to the Solicitud with pk provided in the database	201: {solicitud_id, cotizacion_id} 400: error message, bad request 404: error message, no Solicitud found 500: internal server error
GET solicitud/<int:pk_s>/factura	None	Return Json with list of Factura from Solicitud with the pk provided	201: Json list of Facturas or empty if none 404: error message, no Solicitud found 500: internal server error
POST solicitud/<int:pk_s>/factura	Estado: String Monto: int Comentario: String	Add a new Factura to the Solicitud with pk_s provided in the database	201: {solicitud_id, factura_id} 400: error message, bad request

PUT solicitud/<int:pk_s>/ factura	Estado: String Monto: int Comentario: String	Update first Factura from Solicitud with pk_s in the database	201: {solicitud_id, factura_id} 400: error message, bad request 404: error message, no Solicitud found 500: internal server error
GET api/formulario/ <int:pk>	None	Return Json with Formulario of Solicitud with the pk provided	200: Json with Formulario 404: No formulario found 500: internal server error
POST api/formulario/ <int:pk>	Cedula_ruc: String Tipo_compra: String No_compra: String url_compra: Url tipo_acuerdo: String forma_pago: String tipo_pago: String tiempo: int ul_certi_banco: Url anticipo: Float nombre_banco: String tipo_cuenta: String numero_cuenta: String nombre_cuenta: String	Add a new Formulario to the Solicitud with pk provided in the database	201: {solicitud_id, formulario_id} 400: error message, bad request 404: error message, no Solicitud found 500: internal server error

	correo: String		
--	----------------	--	--