📘 Guide: Using Business Partner gRPC Plugin with Postman

This guide is written for both **non-technical users** and **developers**. It explains how to create, read, update, delete, and list Business Partner records using Postman with the `ve.net.alge.grpc` plugin.

🧑‍💼 For Non-Technical Users: What Can You Do?

This system allows you to work with Business Partner records through gRPC requests. Here's how:

➕ Create a New Business Partner

Purpose: Add a new record to the system.

Fields & Description:

- `table_name`: Target table name. Example: `"C_BPartner"` (Business Partner)
- `attributes -> fields`: Contains each field's name, value type, and value.
    - `Name`: `string_value` – Business Partner's name (e.g., "Test Partner")
    - `Value`: `string_value` – Unique code or identifier (e.g., "TP-004")
    - `AD_Client_ID`: `number_value` – Client identifier
    - `AD_Org_ID`: `number_value` – Organization identifier
    - `Created` / `Updated`: `string_value` – Timestamps in ISO format
    - `C_BP_Group_ID`: `number_value` – Group ID like Vendor or Customer
    - `IsActive`: `bool_value` – Whether the record is active

- `Description`: `string_value` – Additional description

Example Request:

```
{
  "table_name": "C_BPartner",
  "attributes": {
    "fields": {
      "Name": { "string_value": "Test Partner" },
      "Value": { "string_value": "TP-004" },
      "AD_Client_ID": { "number_value": 11 },
      "AD_Org_ID": { "number_value": 0 },
      "Created": { "string_value": "2025-05-28T23:08:00Z" },
      "Updated": { "string_value": "2025-05-28T23:08:00Z" },
      "C_BP_Group_ID": { "number_value": 104 },
      "IsActive": { "bool_value": true },
      "Description": { "string_value": "New business partner for testing" }
    }
  }
}
```

🔍 View Business Partner Information

Purpose: View existing record data.

Using ID:

Fields & Description:

- `id`: Numeric ID of the business partner
- `table_name`: The table from which to retrieve the data (e.g., `"C_BPartner"`)

```
{
  "id": 1000002,
  "table_name": "C_BPartner"
}
```

Using Filter:

Fields & Description:

- `filters`: JSON string for conditions (e.g., search by Value)
- `table_name`: Table to apply the filter

```
{
  "filters": "[{\"name\": \"Value\", \"operator\": \"equal\", \"values\": \"TP-003\"}]",
  "table_name": "C_BPartner"
}
```

✏️ Update a Business Partner

Purpose: Change information about an existing partner.

Fields & Description:

- `table_name`: Table name (e.g., "C_BPartner")
- `id`: ID of the record to update
- `attributes -> fields`: Updated values for one or more fields
    - Example: Update `Name` or `Description`

Example Request:

```
{
  "table_name": "C_BPartner",
  "id": 1000003,
  "attributes": {
    "fields": {
      "Name": { "string_value": "Test Partner" },
      "Description": { "string_value": "New business partner for testing ..... " }
    }
  }
}
```

❌ Delete a Business Partner

Single Delete

Fields & Description:

- `table_name`: Table name (e.g., `"C_BPartner"`)
- `id`: ID of the record to delete

```
{
  "id": 1000002,
  "table_name": "C_BPartner"
}
```

Batch Delete

Purpose: Delete multiple Business Partner records at once.

Fields & Description:

- `table_name`: Table name
- `ids`: List of IDs to delete in one go

Example Request:

```
{
  "ids": [1000004, 1000005, 1000006],
  "table_name": "C_BPartner"
}
```

📋 List Business Partners

Purpose: Show a list of partners based on filter (e.g., all customers)

Fields & Description:

- `filters`: Filter condition as a string

- `sort_by`: Field to sort the result by
- `table_name`: The target table (e.g., `"C_BPartner"`)

Example Request:

```
{
  "filters": "[{\"name\": \"IsCustomer\", \"operator\": \"equal\", \"values\": \"Y\"}]",
  "sort_by": "C_BPartner_ID",
  "table_name": "C_BPartner"
}
```

👩‍💻 Developer Section: gRPC Setup in Postman

🛠️ Prerequisites

- ✅ Postman v10.14.0 or above
- ✅ gRPC server running (e.g., `localhost:50051`)
- ✅ All `.proto` files and their dependencies available

📥 Import `.proto` Files

1. Open Postman → **New** → **gRPC Request**
2. Enter `grpc://localhost:50051`
3. Click **Service Definition** → **Import .proto file**
4. Import from:

ve.net.alge.grpc\src\main\proto

📁 Add Import Paths for Dependencies

Note: These paths are only examples for reference. You must include all actual subfolders found inside the following path:

ve.net.alge.grpc\target\proto-dependencies\

Example paths to add in Postman:

ve.net.alge.grpc\src\main\proto
ve.net.alge.grpc\target\proto-dependencies\ce5781eccd28d0d1ae571cb0f9bf2e2f
ve.net.alge.grpc\target\proto-dependencies\fa24a8af1c95adcd549a9ba7ae615f63f
ve.net.alge.grpc\target\proto-dependencies\ff0fac2f3c693252fc9e8cce9a2bc5a

---