Smart Gateway

User manual and Web Service Documentation

- Introduction

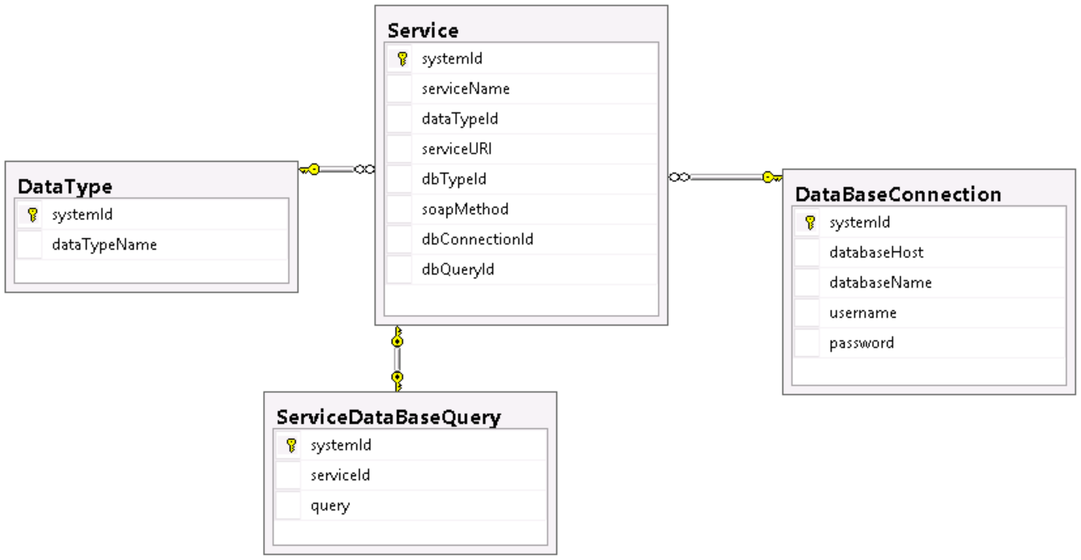
Smart Gateway has been designed to utilize SmartHub with wider flexibility of integrability with variant platforms of service providers. Smart Gateway connects with variant platforms through providing connectivity with different data formats: Soap, XML, and Database.

Smart Gateway is a data-based configuration system, service developers don't have to write codes to integrate with other services/payment providers, it's just a matter of data configurations. It's developed based on Node JS platform.

- Supported Data Format:
    - Soap
    - XML
    - Database: till now supports two database platforms: SQL Server, MySQL. Only Oracle is missed due to the absence of a testing platform.

- **Smart-Gateway Database Structure**

Smart-Gateway designed based on the concept of services, each service has its attributes (data format type, API link, database configs, etc). The below diagram shows the database schema.



- **Service Table:**

Define the attributes of the service:

- systemId: it's the service ID that Smart Gateway will use to get the service info.
- serviceName: Name of the service.
- dataTypeId: To define the data type that service will need to connect with API. It's a foreign key of **DataType** table.
- serviceURI: The API link of the service.
- dbTypeId: Define the database type of service provider, in case service provider don't have API for their service.
- soapMethod: If data type was soap you have to provide the method of soap that service will be processed on.
- dbConnectionId: Define the connectivity variables of database that Smart Gateway will connect with. It's foreign key of **DataBaseConnection** table.
- dbQueryId: Indicates to the query that will be processed on service provider's database to obtain/provide certain info. It's foreign key of ServiceDataBaseQuery.

- DataType Table:

Indicates to the data format of service. They are fixed variables, changing data of table doesn't affect the Smart Gateway system. it contains three types only:

1- Soap
2- XML
3- Database

- DataBaseConnection Table:

This table dedicated to storing the connection attributes of service providers' databases.

- systemId: It's the id that used as foreign key on Service table.
- databaseHost: The host IP of database.
- databaseName: Name of the database.
- username: username of the credential to access the database.
- password: password of credential.

- ServiceDataBaseQuery Table:

This table contains the queries of database services, it has:

- systemId: the ID that used as foreign key on Service table.
- query: the query that will be processed on host database.

Service Creation on Smart Gateway

Based on service' data type admin can create the service by providing suitable attributes. For instance, the creation of SOAP service required following data must be provided on the table of Service:

- service name : "test soap service"
- dataTypeId : 1
- serviceURI : https://www.crcind.com/csp/samples/SOAP.Demo.CLS?WSDL
- soapMethod : SumNumbers

Database service, on the other hand, requires to define database connectivity variables on DataBaseConnection table, service's query that will be processed on ServiceDataBaseQuery table then fill the service info on Service table with providing ids of other tables, for instance:

- service name : "Get Order Info"
- dataTypeId : 2
- dbConnectionId : 1
- dbQueryId : 1

The database query must be provided alongside the field name attached with @ mark so Smart Gateway can recognize the value that should be attached to the query before the processing. Example:

"SELECT * FROM orders where id = ***@orderId***"

While orderId should be sent on the request data in JSON format.

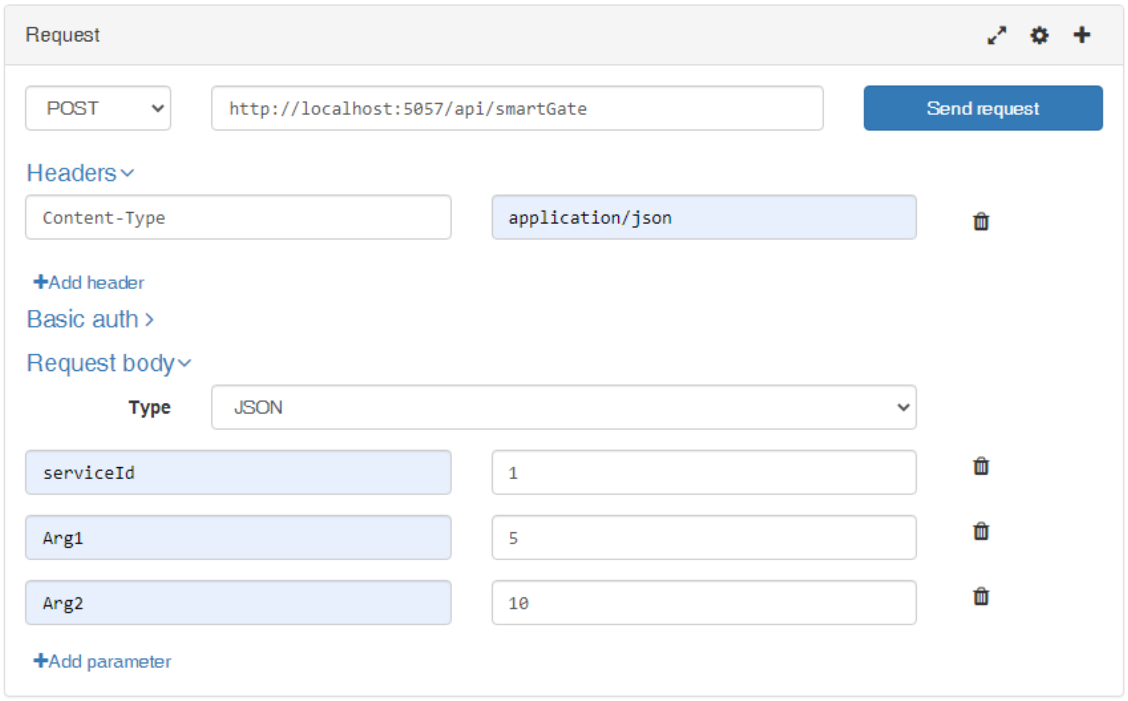
How Smart Gateway Works

Smart Gateway receive the request on JSON format, request data must have the *serviceId* alongside the other data.

Based on the given type Smart Gateway convert the request data format (JSON) to destination data format, and call the service provider's platforms based on the given properties.

Below is an example of SOAP service that's do sum two numbers and bring back the result:

Request:



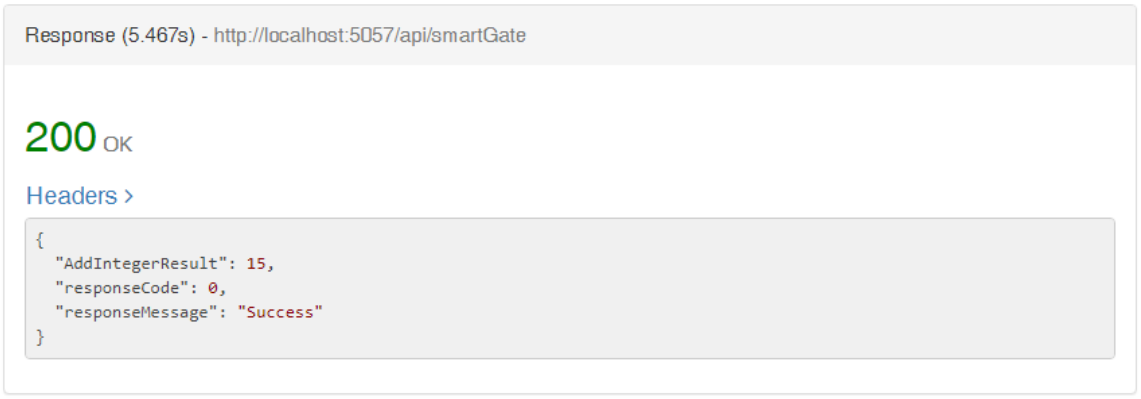
```
http://localhost:5057/api/smartGate
```

Content-Type	application/json

Type: JSON

serviceId	1
Arg1	5
Arg2	10

Response:



Response (5.467s) - http://localhost:5057/api/smartGate

200 OK

```
{
  "AddIntegerResult": 15,
  "responseCode": 0,
  "responseMessage": "Success"
}
```