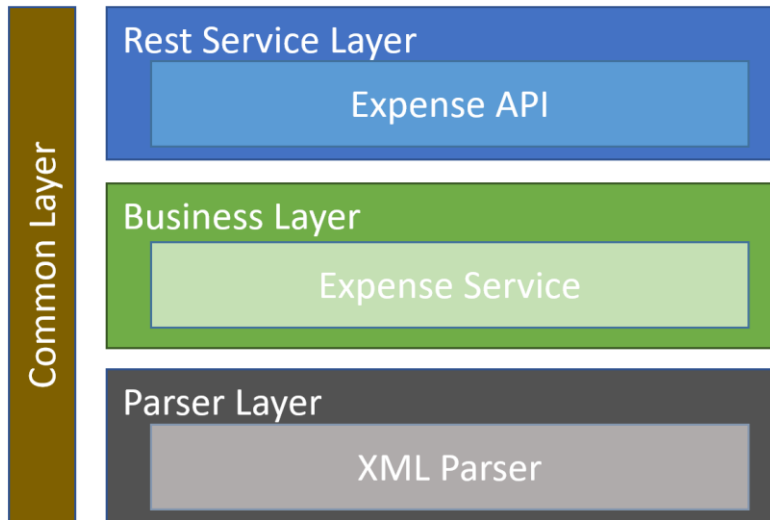


# Serko Expense API

## 1. High Level Architecture



### Rest Service Layer (SerkoExpense.Api):

- Contains the Web API endpoint to import data.
- CORS enabled using the web.config settings.
  - Current settings allow any origin, any domain and any method.
  - Used web.config settings instead of configuring in WebApiConfig because it will allow to change CORS settings without rebuild and redeploy.
- Handled exceptions using custom exception attribute(class=ExceptionFilter) because then all the exceptions will be handled at single point.
  - This class is available under Filters folder in WebApi project.
  - The filter registered in WebApiConfig.
  - This filter logs the exception using log4net and throw custom exception message to user. Log4net configuration are under log4net section in web.config

### Business Layer ( SerkoExpense.Business):

- Contains the Expense service which calculate the GST and total excluding GST.
- The GST percentage is configured under app settings (key= GST) in Web.config of Web API project.

### Parser Layer (SerkoExpense.XmlParser):

- Contains the XML parser which will validate the input string and extract the xml content.

### Common Layer (SerkoExpense.Common):

- Contains the common classes that is shared among all other layers.

## 2. API endpoint, request and response

**Endpoint Url:** <http://localhost:51214/Expense/ImportData>

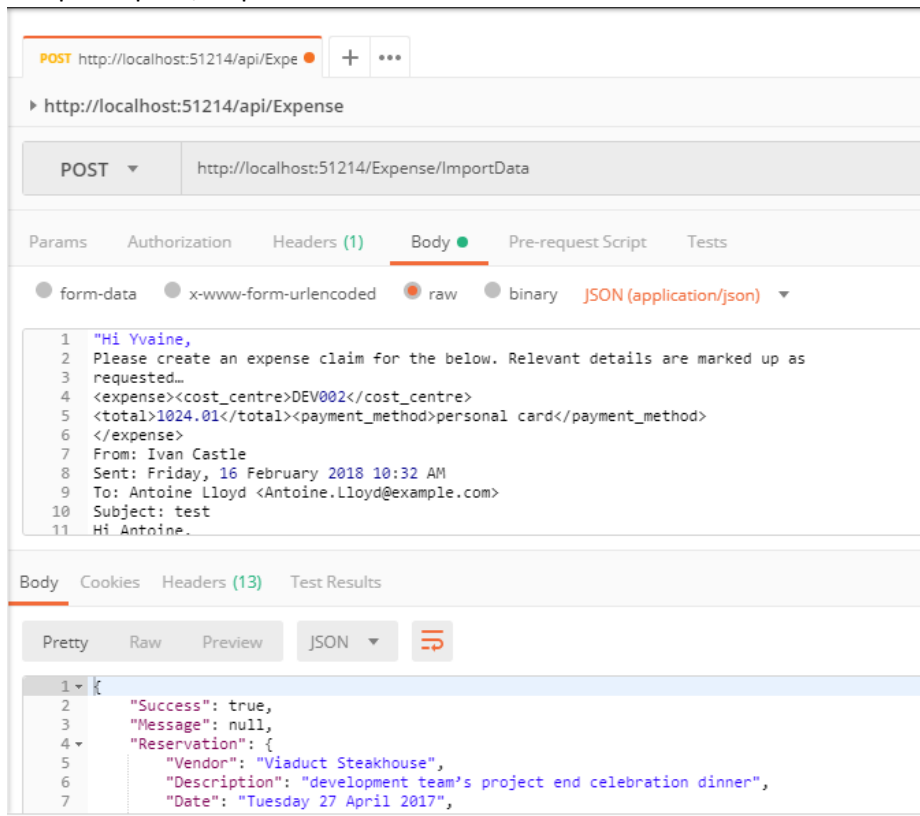
**Request:** string

- This is the text received via email
- Send string in JSON format

**Response:** ImportResponse

- This contains three properties
  - Success- Indicates whether import success or not
  - Message- Contains the error message if import unsuccessful
  - Reservation-Contains the extracted and calculated data

Sample request/response from Postman:



### 3. Testing

Unit test project (SerkoExpense.Api.Test) is done to test the Expense API. Test methods available to test following scenarios.

No	Scenario	Test Method	Status
1	Opening tags that have no corresponding closing tag. In this case the whole message should be rejected.	ImportData_ShouldRejectIfCorrespondingClosingTagIsMissing	Pass
2	Missing <total>. In this case the whole message should be rejected.	ImportData_ShouldRejectIfTotalTagIsMissing	Pass
3	Missing <cost_centre>. In this case the 'cost centre' field in the output should be defaulted to 'UNKNOWN'.	ImportData_ShouldReciveUnknowForEmptyCostCenter	Pass
4	extracted and calculated data should be available when xml is valid	ImportData_ShouldReturnExtractedAndCalculatedDataWhenXMLIsValied	Pass

### 4. Assumptions

1. Value of <total> can be zero
2. Value of <total> cannot be negative
3. <total> cannot have empty string as value (<total> </total> is not allowed)
4. <total> cannot be empty tag (<total> </total> is not allowed)
5. Self-closing tags will not consider as valid, therefore will be ignored.