

REST Integration (In Pega)

An Article By Amardeep

What is REST service in PEGA?



PEGA®



An Article By Amardeep Srivastava

Outcome: We will be leveraging the REST integration in Pega to Fetch the table records

.

Prerequisite :There should be one data type available with records.

What is REST ?What does it do ?

REST service in PEGA is a way to **expose Pega functionality to external systems** so that other applications can **send requests to Pega and get responses** using **HTTP methods**.

◆ What is a REST Service in PEGA?

A **REST Service** in Pega allows **external applications** (like Java, .NET, mobile apps, Postman, etc.) to **call Pega** using standard REST protocols such as:

- **GET**
- **POST**
- **PUT**
- **DELETE**

Pega processes the request, executes the configured logic (Activity / Data Transform), and returns a **JSON or XML response**.

◆ Why REST Service is used in PEGA?

- To integrate Pega with external systems

- To expose case creation, updates, or data retrieval
 - To support **microservices architecture**
 - Lightweight and **faster than SOAP**
-

◆ Key Components of REST Service in PEGA

1. Service Package

- Security & access control
- Authentication (Basic, OAuth 2.0)

2. Service REST Rule

- Defines endpoint URL
- HTTP method (GET/POST/etc.)
- Request & response mapping

3. Request Data Transform

- Maps incoming JSON/XML → Pega properties

4. Processing Logic

- Activity / Case creation / Data page

5. Response Data Transform

- Maps Pega data → JSON/XML response

Let's Start The Hands On :

The screenshot shows the Pega DEV STUDIO interface for the "Mahidra Vehicle Insurance" application. The left sidebar contains icons for Recents, Case types, Data types (selected), App, Records, and Favorites. The main header includes "DEV STUDIO", "Application: Mahidra Vehicle Insurance", "Configure", "Launch portal", "Create", a search bar, and a "DEVELOPMENT" dropdown. The top navigation bar shows "Data Type: PolicyDetails" and "Mahi-Data-Pol...". Below it, there are tabs for "Data model", "Usage", "Sources", "Records" (selected), "Views", "Test cases", and "Settings". The "Actions" and "Save" buttons are also visible.

Source

Local data storage Actions ▾

Export Import

Search... Q

Policy ID (requi...	EffectiveTo	EffectiveFrom	DecisionResult	MaxValue	MinValue	PolicyStatus	PolicyName
POL001	12/31/2025	1/1/2025	Reject	49	0	Active	Loan Eligibility Policy
POL002	12/31/2025	1/1/2025	Reject	99	50	Active	Loan Eligibility Policy
POL003	12/31/2025	1/1/2025	Reject	149	100	Active	Loan Eligibility Policy
POL004	12/31/2025	1/1/2025	Reject	199	150	Active	Loan Eligibility Policy
POL005	12/31/2025	1/1/2025	Reject	249	200	Active	Loan Eligibility Policy
POL006	12/31/2025	1/1/2025	Reject	299	250	Active	Loan Eligibility Policy
POL007	12/31/2025	1/1/2025	Reject	349	300	Active	Loan Eligibility Policy
POL008	12/31/2025	1/1/2025	Reject	399	350	Active	Loan Eligibility Policy
POL009	12/31/2025	1/1/2025	Reject	449	400	Active	Loan Eligibility Policy
POL010	12/31/2025	1/1/2025	Reject	499	450	Active	Loan Eligibility Policy
POL011	12/31/2025	1/1/2025	Refer	549	500	Active	Loan Eligibility Policy
POL012	12/31/2025	1/1/2025	Refer	599	550	Active	Loan Eligibility Policy

Agile Workbench Current work Scenario Testing Issues Tracer Clipboard Live UI Live Data Accessibility Performance PEGA

This is the Policy Details we will be fetching on the basis of Policy Id. So when end user is requesting for the Policy details upon providing the Policy id then Pega should provide the records associated with That id.

Implementation: Create the page property for Request and Response as shown and refer to the Data class.

Note: This Request and Response Page should be created under the context of Integration class.

The screenshot shows the Pega Dev Studio interface with the following details:

- Top Bar:** DEV STUDIO, Application: Mahidra Vehicle Insurance, Configure, Launch portal, Create, Search, DEVELOPMENT.
- Left Sidebar:** Recent cases, Case types, Data types, App, Records, Favorites.
- Middle Panel:**
 - Current Application:** Mahi-Int-ProductDetails
 - Current Class:** Data Model
 - Current Property:** RequestPage
 - Property Type:** Single Page
 - Page Definition:** Mahi-Data-PolicyDetails
 - Data Access:** Manual (selected radio button). A tooltip explains: "At run time, the user adds data to this property through the UI. Data transforms and other rules may be required to support this workflow."
- Bottom Bar:** Agile Workbench, Current work, Scenario Testing, Issues, Tracer, Clipboard, Live UI, Live Data, Accessibility, Performance, PEGA.

DEV STUDIO Application: Mahidra Vehicle Insurance Configure Launch portal Create

Search Q ✓ ? ⚡ DEVELOPMENT

3 applications Home Mahi-Data-Pol... ResponsePage

Edit Property: ResponsePage [Available]
CL: Mahi-Int-ProductDetails ID: ResponsePage RS: MahiInt:01-01-01

Delete Actions Save

Classes Branches

Mahi-Int-ProductDetails

Mahi-Int-ProductDetails Data Model

Property RequestPage ResponsePage

DecisionResult EffectiveFrom EffectiveTo MaxValue MinValue PolicyID PolicyName PolicyStatus

SysAdmin

Pinned classes Edit

Mahi-MahidraV-Int-GetCustomerDetails

Property type

Single Page

Page definition ★ Mahi-Data-PolicyDetails

Data access

Manual At run time, the user adds data to this property through the UI. Data transforms and other rules may be required to support this workflow.

Refer to a data page

Copy data from a data page

Agile Workbench Current work Scenario Testing Issues Tracer Clipboard Live UI Live Data Accessibility Performance PEGA

The screenshot shows the Pega DEV STUDIO interface for editing a ResponsePage property. The left sidebar contains navigation links for Recent, Case types, Data types, App, Records, and Favorites. The main workspace displays the 'Edit Property' screen for 'ResponsePage'. The 'General' tab is selected. The 'Page definition' field contains 'Mahi-Data-PolicyDetails'. Under 'Data access', the 'Manual' option is selected, with a note explaining that at runtime, the user adds data through the UI, requiring data transforms and other rules. The bottom navigation bar includes links for Agile Workbench, Current work, Scenario Testing, Issues, Tracer, Clipboard, Live UI, Live Data, Accessibility, Performance, and PEGA.

Now Lets create the Service Package rule rule by following this navigation :

The screenshot shows the Pega DEV STUDIO interface with the following navigation steps highlighted:

- Top navigation bar: Application: Mahidra Vehicle Insurance, Configure, Launch portal, Create.
- Search bar: Search icon, checkmark, question mark, refresh, DEVELOPMENT dropdown.
- Left sidebar:
 - Recent applications: 3 applications (dropdown).
 - Case types: Recents, Classes, Branches.
 - Data types: Case types, Data types (dropdown).
 - App: Records, Favorites.
- Main content area:
 - Application: Mahi-Data-Pol... ResponsePage
 - Edit Property: ResponsePage [Available]
 - CL: Mahi-Int-ProductDetails ID: ResponsePage RS: MahiInt:01-01-01
 - Action buttons: Delete, Actions, Save, Close.
- Central navigation menu:
 - General, Advanced, Specifications, History.
 - Create > Application Definition, Data Model, Decision, Integration-Connectors, Integration-Mapping.
 - Integration-Resources > Connect Request Processor, Email Account, Email Listener, File Listener, FTP Server, Integration System, JMS Listener, JMS MDB Listener, JMS Producer Model, JMS Transport, JNDI Server, MQ Listener, MQ Server.
 - Service Package (highlighted in blue) > Service Request Processor, SMS Account, Web Service Policy.
- Bottom status bar: localhost:8080/prweb/PRServlet/app/mahidra-vehicle-insurance/_EZO1Pk9BbGokdM-ArSgx99zofCityQm*/!STANDARD?pzPostData=1690485168#acer, Clipboard, Live UI, Live Data, Accessibility, Performance, PEGA.

DEV STUDIO Application: Mahidra Vehicle Insurance Configure Launch portal Create Search DEVELOPMENT

3 applications Classes Branches **Mahi-Int-ProductDetails** **Mahi-Int-ProductDetails** Data Model Property SysAdmin

Pinned classes **Edit** **Mahi-MahidraV-Int-GetCustomerDetails**

Edit Service Package : PolicydetailsRest
ID: PolicydetailsRest RS: MahidraV [Edit]

This record has 1 info warning (including 1 unjustified) [Review/Edit](#)

Context Pooling Open API History

Context

Processing mode **Stateless**

Service access group **MahidraV:Authors**

Requires authentication Suppress Show-HTML

Methods

No Service Methods found for this Service Package. Try clicking "Refresh."

Refresh

localhost:8080/prweb/PRServlet/app/mahidra-vehicle-insurance/_EZ01Pk9BBbGokdM-ArSgx99zofGityQm*/!STANDARD?pzPostData=1690485168#acer Clipboard Live UI Live Data Accessibility Performance PEGA

Lets create the REST Service through wizard:

Create Service REST ? Cancel **Create and open**

Service REST Record Configuration

Service name ★:

Service package name: Service version:

URI Template ★: Build URI Identifier:

Context

Mahidra Vehicle Insurance Mahindra Auto Insurance Cosmos Rules Pega Platform

Add to ruleset ★:

Current work item

Work item to associate: C

DEV STUDIO Application: Mahidra Vehicle Insurance Configure Launch portal Create

Search Q ? DEVELOPMENT

3 applications Classes Branches

Mahi-Int-ProductDetails

Mahi-Int-ProductDetails Data Model SysAdmin

Pinned classes Mahi-MahidraV-Int-GetCustomerDetails

Edit

Home New Service Package POLICYDETAILS... Mahi-Data-Pol...

Edit Service REST: GETProductDetails [Available]
ID: PolicydetailsRest • v1 • policyid_varc0459556ff4da06d49f16e2e0eddf458 RS: MahiInt:01-01-01

Actions Save X

Service Methods Specifications OpenAPI History

Resource properties

Resource * Mahi-Int-ProductDetails

Service endpoint URL
http://localhost:8080/prweb/api/PolicydetailsRest/v1/PolicyID/{PolicyID}

URI template
PolicyID/{PolicyID}

URI path parameters

Name	Description	Map to	Map to key
PolicyID		Clipboard	Param.PolicyID

Processing options

End Requestor When Done Enable service SLA with fallback activity

Method is read-only

Execution mode
Execute synchronously

Lightweight clipboard mode [?](#)

Provide values Use application settings

Enable

Request processor

The screenshot shows the Service Studio interface for the 'Mahidra Vehicle Insurance' application. On the left sidebar, under 'Pinned classes', the 'Mahi-Int-ProductDetails' class is highlighted with a red box. In the main workspace, the 'Edit Service REST' screen is displayed for the 'GETProductDetails' service. A second red box highlights the 'Mahi-Int-ProductDetails' entry in the 'Resource properties' section, specifically the 'Resource' field. The service endpoint URL is set to 'http://localhost:8080/prweb/api/PolicydetailsRest/v1/PolicyID/{PolicyID}'. The 'URI template' is 'PolicyID/{PolicyID}'. Under 'Processing options', the 'Method is read-only' checkbox is selected. The 'Execution mode' dropdown is set to 'Execute synchronously'. In the 'Lightweight clipboard mode' section, the 'Provide values' radio button is selected. The 'Request processor' field is empty.

As of now we are only fetching the records so we will creating everything (Activity etc)under this .

The screenshot shows the Pega Platform interface for editing a service. The top navigation bar includes 'Home', 'New', 'Service Package', 'POLICYDETAILS...', and 'Mahi-Data-Pol...'. The main title is 'Edit Service REST: GETProductDetails [Available]'. Below it, the service ID is listed as 'ID: PolicydetailsRest • v1 • policyid_varc0459556ff4da06d49f16e2e0eddf458' and the RS is 'MahiInt:01-01-01'. On the right, there are 'Actions' and 'Save' buttons. The 'Methods' tab is selected, showing five methods: GET, POST, PUT, PATCH, and DELETE. At the bottom, there are links for 'Issues', 'Tracer', 'Clipboard', 'Live UI', 'Live Data', 'Accessibility', 'Performance', and the 'PEGA' logo.

Next : Lets Create an activity

Edit Activity: FetchPolicyDetails [Available]
CL: Mahi-Int-ProductDetails ID: FetchPolicyDetails RS: MahidraVint:01-01-01

This record has 3 severe or moderate warnings (including 3 unjustified) and 1 info warning [Review/Edit](#)

Steps Parameters Pages & Classes Security Test cases Specifications History

1. Label Method Step page Description
Loop When Page-New PolicyPage Create a page Jump

Method Parameters

Name	Value
Data Transform	
PageList	
NewClass	

2. Loop When Property-Set Set property values Jump

Method Parameters

*PropertyName	*PropertyValue
Param.PolicyID	Primary.RequestPage.PolicyID

+

3. Loop When Obj-Open PolicyPage Open record Jump

Method Parameters

Name	Value
OpenClass	Mahi-Data-PolicyDetails
Lock	<input checked="" type="checkbox"/>
ReleaseOnCommit	<input checked="" type="checkbox"/>
LockInfoPage	

*PropertyName	*PropertyValue
PolicyPage.PolicyID	Param.PolicyID

+

The screenshot shows the 'Edit Activity' interface for 'FetchPolicyDetails'. It displays three steps: 1. Create a page (Method: Page-New), 2. Set property values (Method: Property-Set), and 3. Open record (Method: Obj-Open). Each step has its own set of method parameters. The 'Method Parameters' section for step 1 includes fields for Data Transform, PageList, and NewClass. The 'Method Parameters' section for step 2 includes a table mapping Param.PolicyID to Primary.RequestPage.PolicyID. The 'Method Parameters' section for step 3 includes fields for OpenClass (set to Mahi-Data-PolicyDetails), Lock (checked), ReleaseOnCommit (checked), and LockInfoPage (empty). A warning message at the top indicates 3 severe or moderate warnings and 1 info warning.

4. [] Loop When ▾ Page-Copy PolicyPage Copy page Jump []

Method Parameters

CopyFrom	PolicyPage
*CopyInto	Primary.ResponsePage
Data Transform	[]
PageList	[]

+ Add a step Collapse all steps

Scenario Testing Issues Tracer Clipboard Live UI Live Data Accessibility Performance PEGA

Edit Activity: FetchPolicyDetails [Available]
CL: Mahi-Int-ProductDetails ID: FetchPolicyDetails RS: MahidraVInt:01-01-01
Delete Actions Save X

This record has 3 severe or moderate warnings (including 3 unjustified) and 1 info warning Review/Edit

Steps Parameters Pages & Classes Security Test cases Specifications History

Page name	Class
PolicyPage	Mahi-Data-PolicyDetails

+

Testing Part :

One should have postman tool to Test other than Pega platform environment.

Copy the Url from the Service package / Service : "<http://localhost:8080/prweb/api/PolicydetailsRest/v1/PolicyID/{Policy}>"

The screenshot shows the Pega Dev Studio interface with the following details:

- Application:** Mahindra Vehicle Insurance
- Service:** PolicydetailsRest
- Resource:** Mahi-Int-ProductDetails
- Service endpoint URL:** `http://localhost:8080/prweb/api/PolicydetailsRest/v1/PolicyID/{Policy}`
- Processing options:**
 - Execution mode: Execute synchronously
 - Request processor: Off
 - Service monitoring: Off
- Map to key:** Primary.RequestPage.PolicyID

A red box highlights the "Service endpoint URL" field.

Open the PostMan Tool: Pass valid key : <http://localhost:8080/prweb/api/PolicydetailsRest/v1/PolicyID/{key }>

<http://localhost:8080/prweb/api/PolicydetailsRest/v1/PolicyID/{ POL010}>

The screenshot shows the Postman application interface. On the left, the sidebar displays 'Amardeep Srivastava's Workspace' with sections for Collections, Environments, History, Flows, and API Network. Under 'Collections', 'My Collection' is expanded, showing requests for 'Get data', 'Post data', 'New Request', and 'REST API basics: CRUD, test & variable'. The 'Get data' request is selected. The main workspace shows a 'Get data' request with the URL 'http://localhost:8080/prweb/api/Mahindr...'. The 'Params' tab is active, showing query parameters: 'Key' (Value: 'http://localhost:8080/prweb/api/Mahindr...'), 'Key' (Value: 'POST http://localhost:8080/prweb/PRRestServ...'), and 'Value' (Description: 'Bulk Edit'). To the right, a 'Visualize response as table' panel is open, containing a code editor with a CSS snippet:

```
.customer-table th, .customer-table td {  
    border: 1px solid #ddd;  
    padding: 12px;  
    text-align: left;  
}  
.customer-table th {  
    background-color: #4CAF50;  
    color: white;  
    font-weight: bold;  
}
```

The screenshot shows the Postman application interface. On the left, the sidebar includes 'Collections' (selected), 'Environments', 'History', 'Flows', and a '+' button. The main workspace shows 'Amardeep Srivastava's Workspace' with 'My Collection'. A 'Get data' request is selected, showing the URL: `http://localhost:8080/prjweb/api/PolicydetailsRest/v1/PolicyID/POL010`. The 'Params' tab is active. To the right, a 'Visualize response as table' panel is open, displaying the following table:

Key	Value	Description
Key	Value	Description

Below the table, a post-response script is shown:

```
        }
.customer-table th, .customer-table td {
    border: 1px solid #ddd;
    padding: 12px;
    text-align: left;
}
.customer-table th {
    background-color: #4CAF50;
    color: white;
    font-weight: bold;
}
```

Hit send .

Bingo!!We Got the result.

Home Workspaces API Network

Search Postman Ctrl K

Invite Upgrade

Amardeep Srivastava's Workspace

collections New Import

My Collection

- GET Get data
- POST Post data
- GET New Request

REST API basics: CRUD, test & variable

- GET Get data
- POST Post data
- PUT Update data
- DEL Delete data

HTTP My Collection / Get data

Save Share

GET http://localhost:8080/prweb/api/PolicydetailsRe... Send

Docs Params Auth Headers (7) Body Scripts Settings Cookies

Body 200 OK 142 ms 848 B

{ } JSON Preview Visualize

pxUpdateDateTime	2025-12-18T19:09:21.647Z
pzInsKey	Mahi-Data-PolicyDetails POL010
pxUpdateOpName	mahiauto
MaxValue	499
pxUpdateOperator	mahiauto@ins.com
pxObjClass	Mahi-Data-PolicyDetails
pxCreateOperator	mahiauto@ins.com
DecisionResult	Reject
pxCreateDateTime	2025-12-18T19:09:21.647Z
PolicyName	Loan Eligibility Policy
PolicyStatus	Active
MinValue	450
pxCreateSystemID	pega
EffectiveFrom	2025-01-01
pxCommitDateTime	1970-01-01T00:00:00.000Z
pxCreateOpName	mahiauto
EffectiveTo	2025-12-31
PolicyID	POL010

Visualize response as table

GET Get data Visualize response as table

I'll create a table visualization for the response data from your current request.

Allow agent to update post response of request?

Post-response script

```
.customer-table th, .customer-table td { border: 1px solid #ddd; padding: 12px; text-align: left; }
.customer-table th { background-color: #4CAF50; color: white; font-weight: bold; }
.customer-table tr:nth-child(even) { background-color: #f9f9f9; }
.customer-table tr:hover { background-color: #f1f1f1; }
```

Approve Reject

Waiting for approval...

Mahindra app.txt - Note... Describe v... for Skills.

Cloud View Find and replace Console Terminal

Json:

```
{  
    "pxUpdateSystemID": "pega",  
    "pxSaveDateTime": "1970-01-01T00:00:00.000Z",  
    "pxUpdateDateTime": "2025-12-18T19:09:21.647Z",  
    "pzInsKey": "Mahi-Data-PolicyDetails POL010",  
    "pxUpdateOpName": "mahiauto",  
    "MaxValue": 499,  
    "pxUpdateOperator": "mahiauto@ins.com",  
    "pxObjClass": "Mahi-Data-PolicyDetails",  
    "pxCreateOperator": "mahiauto@ins.com",  
    "DecisionResult": "Reject",  
    "pxCreateDateTime": "2025-12-18T19:09:21.647Z",  
    "PolicyName": "Loan Eligibility Policy",  
    "PolicyStatus": "Active",  
    "MinValue": 450,  
    "pxCreateSystemID": "pega",  
    "EffectiveFrom": "2025-01-01",
```

```
"pxCommitDateTime": "1970-01-01T00:00:00.000Z",
"pxCreateOpName": "mahiauto",
"EffectiveTo": "2025-12-31",
"PolicyID": "POL010"
}
```

.

Hope you all like this article it takes lot of effort to create these type of documents please share and follow me and give like and tell your thoughts in comment please.

PEGA REST SERVICE



BUSTED!!

REST

An Article By Amardeep Srivastava