Usage of REST API in webMethods Cloud

I want to create one API for searchAccount and getAccount integrations.

Hence both integrations deal with account, account will be the ressource.

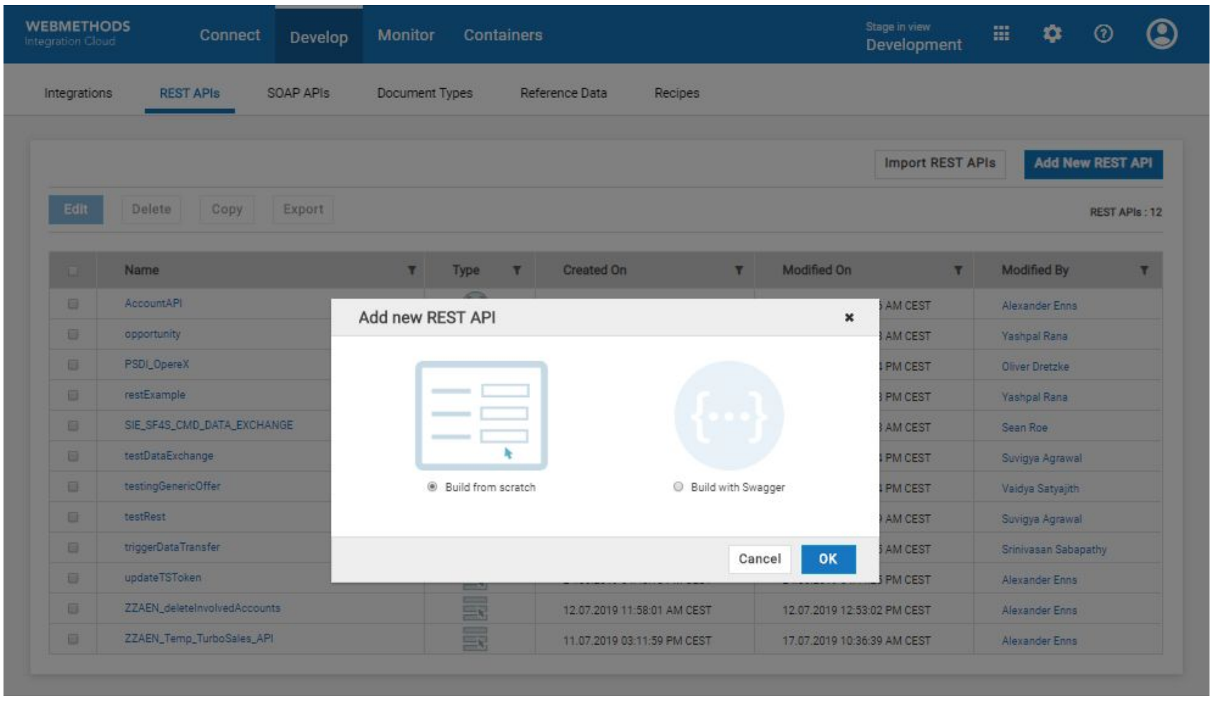
Structure of this API should be:

API

- account
    - POST should use this integration: SIE_SF4S_getAccDetails_API
    - GET should use this integration: SIE_SF4S_searchAccount_API

Approach 1 - build API from scratch

The Integrations are implemented, input and outputs are defined. Should be easy.



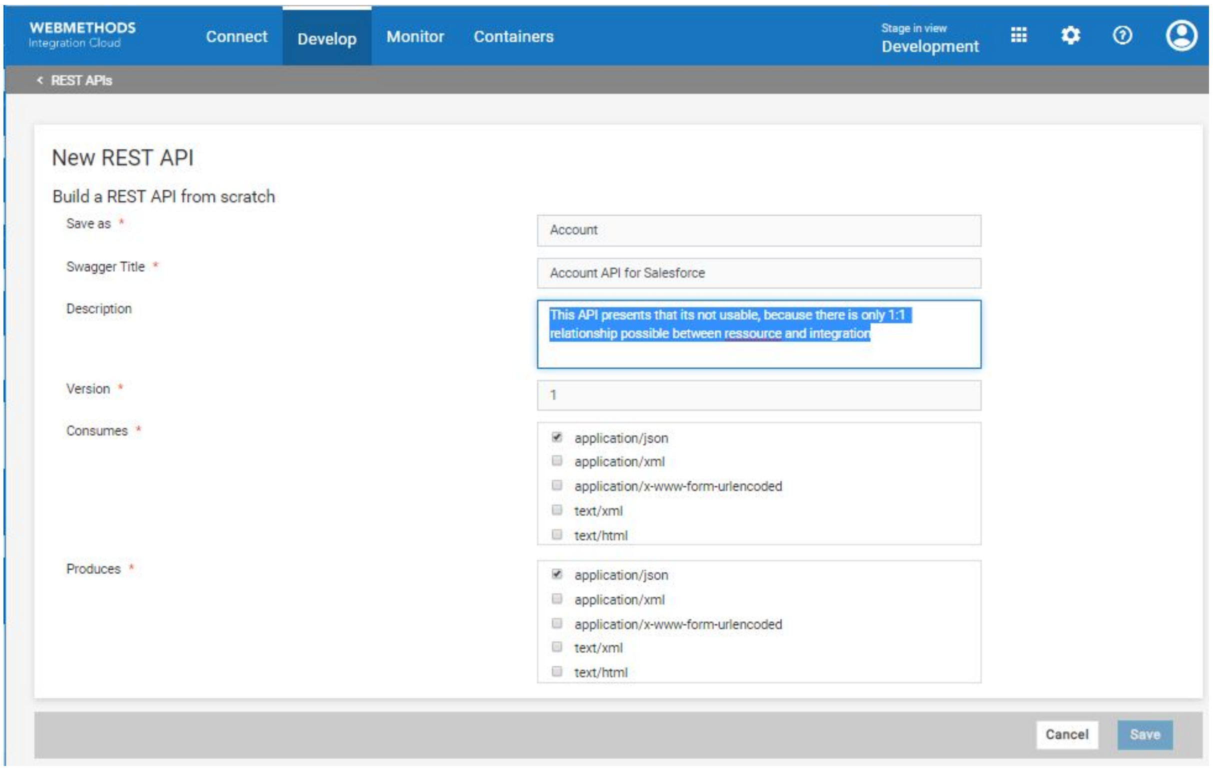
REST APIs : 12

Name	Type	Created On	Modified On	Modified By
AccountAPI			5 AM CEST	Alexander Enns
opportunity			3 AM CEST	Yashpal Rana
PSDI_OpereX			4 PM CEST	Oliver Dretzke
restExample			3 PM CEST	Yashpal Rana
SIE_SF4S_CMD_DATA_EXCHANGE			3 AM CEST	Sean Roe
testDataExchange			4 PM CEST	Suvigya Agrawal
testingGenericOffer			4 PM CEST	Vaidya Satyajith
testRest			9 AM CEST	Suvigya Agrawal
triggerDataTransfer			5 AM CEST	Srinivasan Sabapathy
updateTSToken			5 PM CEST	Alexander Enns
ZZAEN_deleteInvolvedAccounts		12.07.2019 11:58:01 AM CEST	12.07.2019 12:53:02 PM CEST	Alexander Enns
ZZAEN_Temp_TurboSales_API		11.07.2019 03:11:59 PM CEST	17.07.2019 10:36:39 AM CEST	Alexander Enns

Add new REST API

[✓ Build from scratch] Build with Swagger

Next step - define some API properties

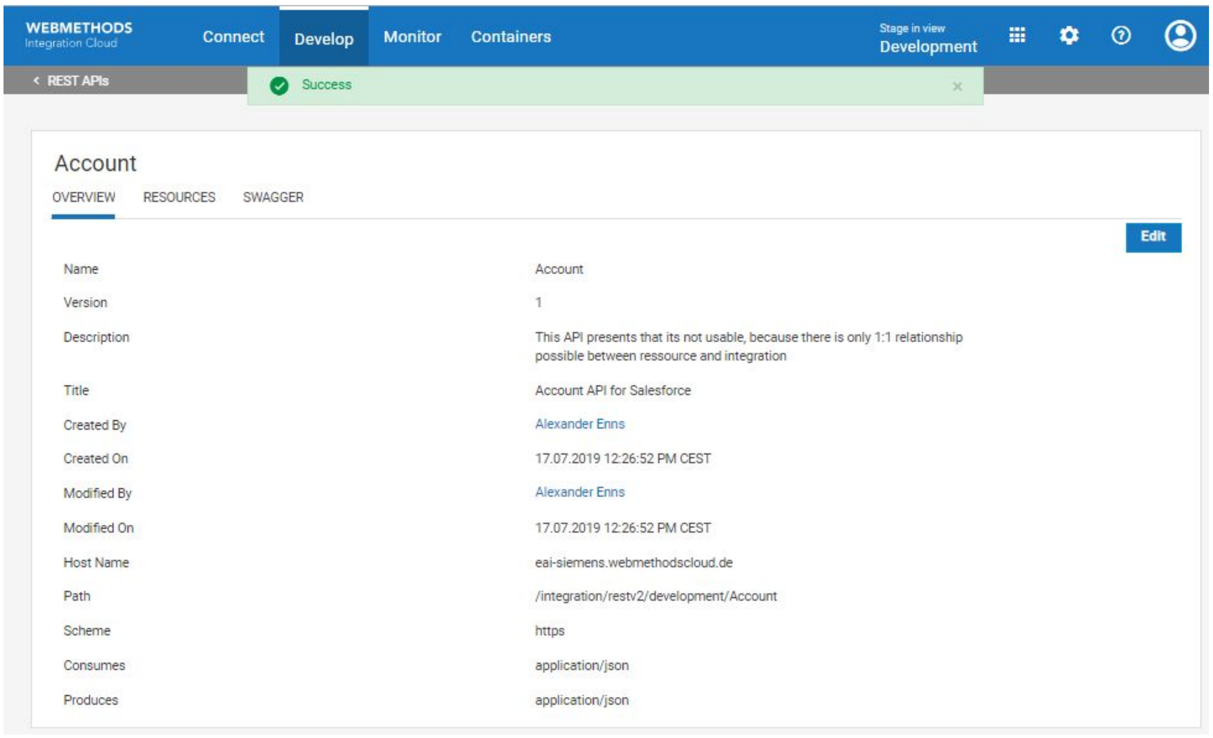


New REST API

Build a REST API from scratch

Save as *	Account
Swagger Title *	Account API for Salesforce
Description	This API presents that its not usable, because there is only 1:1 relationship possible between ressource and integration
Version *	1
Consumes *	✓ application/json ☐ application/xml ☐ application/x-www-form-urlencoded ☐ text/xml ☐ text/html
Produces *	✓ application/json ☐ application/xml ☐ application/x-www-form-urlencoded ☐ text/xml ☐ text/html

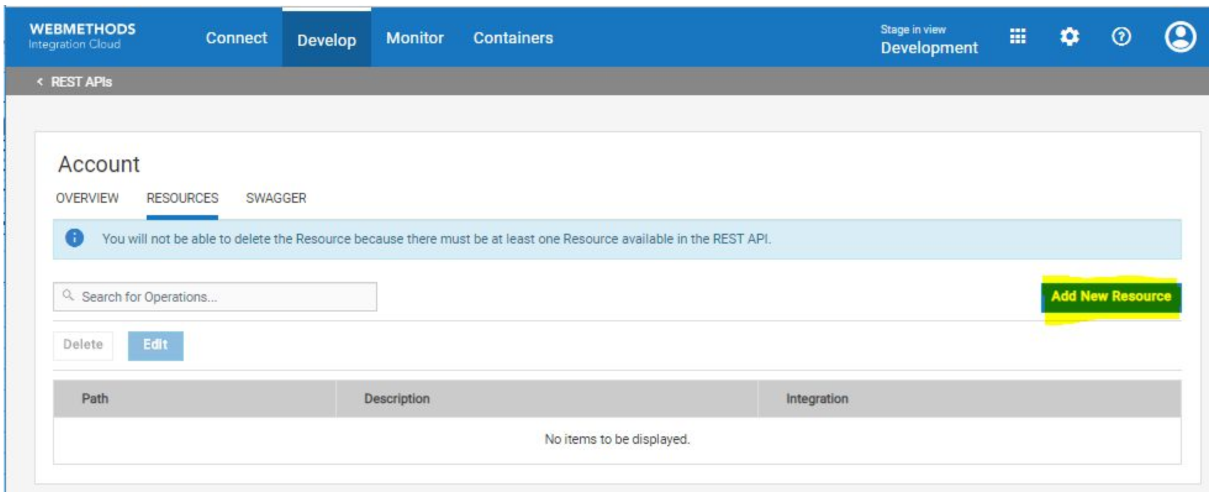
Works fine!



Account

Name	Account
Version	1
Description	This API presents that its not usable, because there is only 1:1 relationship possible between ressource and integration
Title	Account API for Salesforce
Created By	Alexander Enns
Created On	17.07.2019 12:26:52 PM CEST
Modified By	Alexander Enns
Modified On	17.07.2019 12:26:52 PM CEST
Host Name	eai-siemens.webmethodscloud.de
Path	/integration/restv2/development/Account
Scheme	https
Consumes	application/json
Produces	application/json

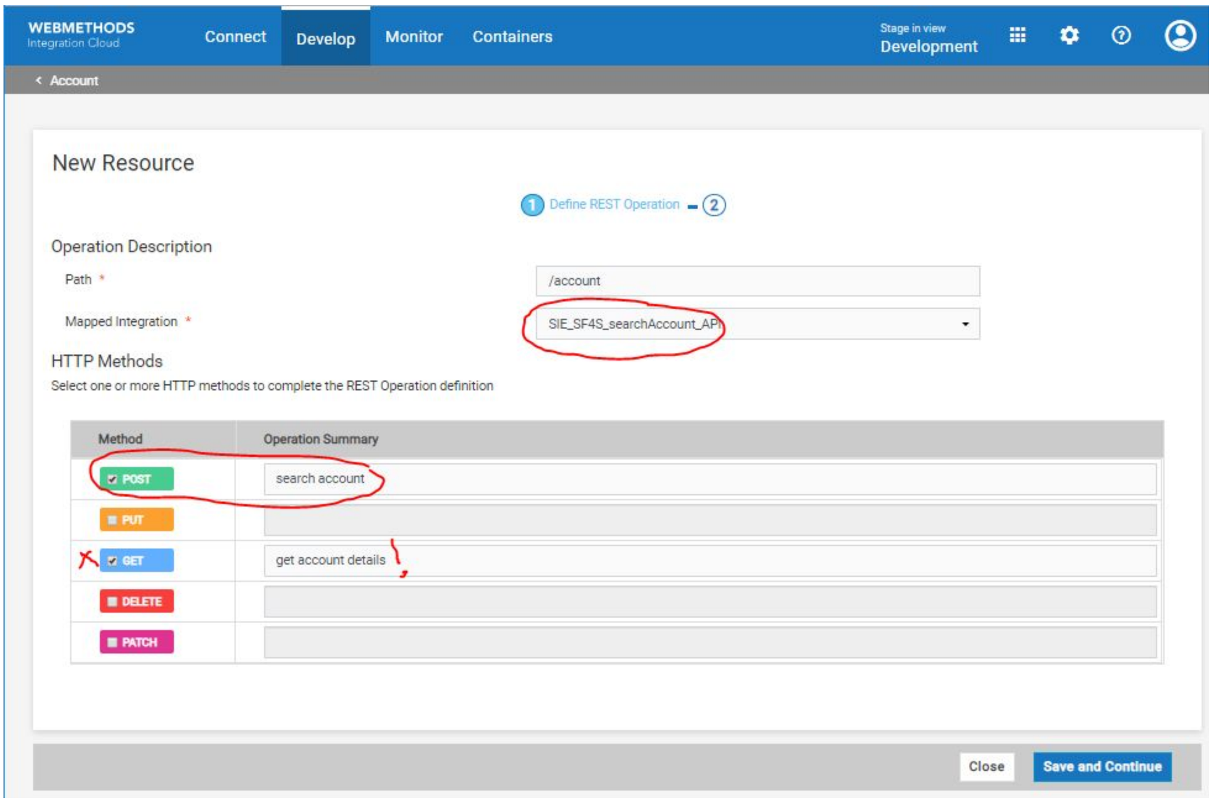
Switch to RESOURCES tab



You will not be able to delete the Resource because there must be at least one Resource available in the REST API.

No items to be displayed.

Click on **Add New Resource**



Operation Description

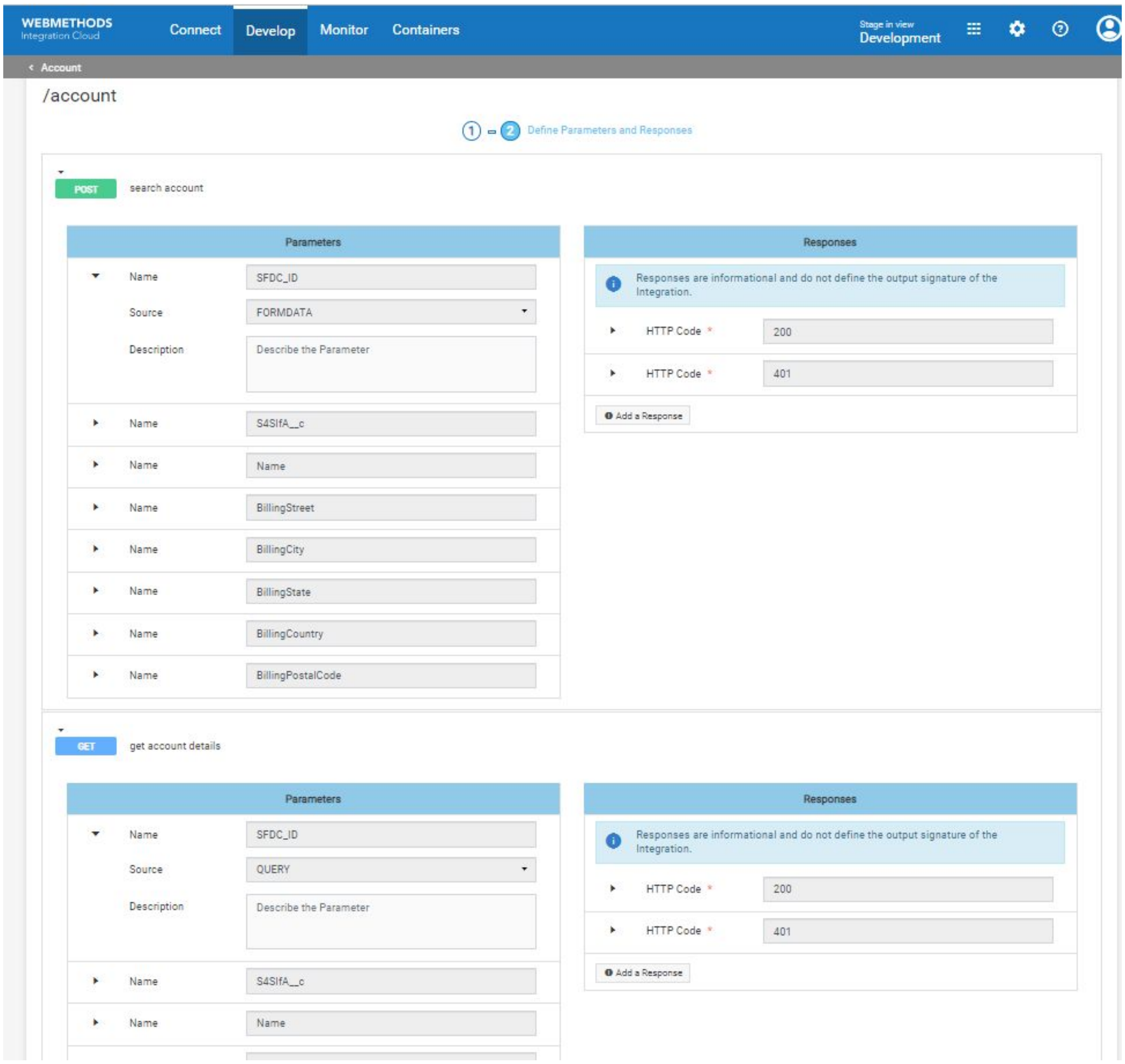
Path *	/account
Mapped Integration *	SIE_SF4S_searchAccount_API

HTTP Methods

Select one or more HTTP methods to complete the REST Operation definition

Method	Operation Summary
☑ POST	search account
PUT	
☑ GET	get account details
DELETE	
PATCH	

Here I notice a conflict! I actually have to map two methods to one integration. That's stupid.
Click Save and Continue


WEBMETHODS Integration Cloud
Connect
Develop
Monitor
Containers
Stage in view Development
Account
Define Parameters and Responses
Add a Response

/account

POST search account

Parameters	
Name	SFDC_ID
Source	FORMDATA
Description	Describe the Parameter
Name	S4SIfA__c
Name	Name
Name	BillingStreet
Name	BillingCity
Name	BillingState
Name	BillingCountry
Name	BillingPostalCode

Responses

Responses are informational and do not define the output signature of the Integration.

HTTP Code *	200
HTTP Code *	401

GET get account details

Parameters	
Name	SFDC_ID
Source	QUERY
Description	Describe the Parameter
Name	S4SIfA__c
Name	Name

Responses

Responses are informational and do not define the output signature of the Integration.

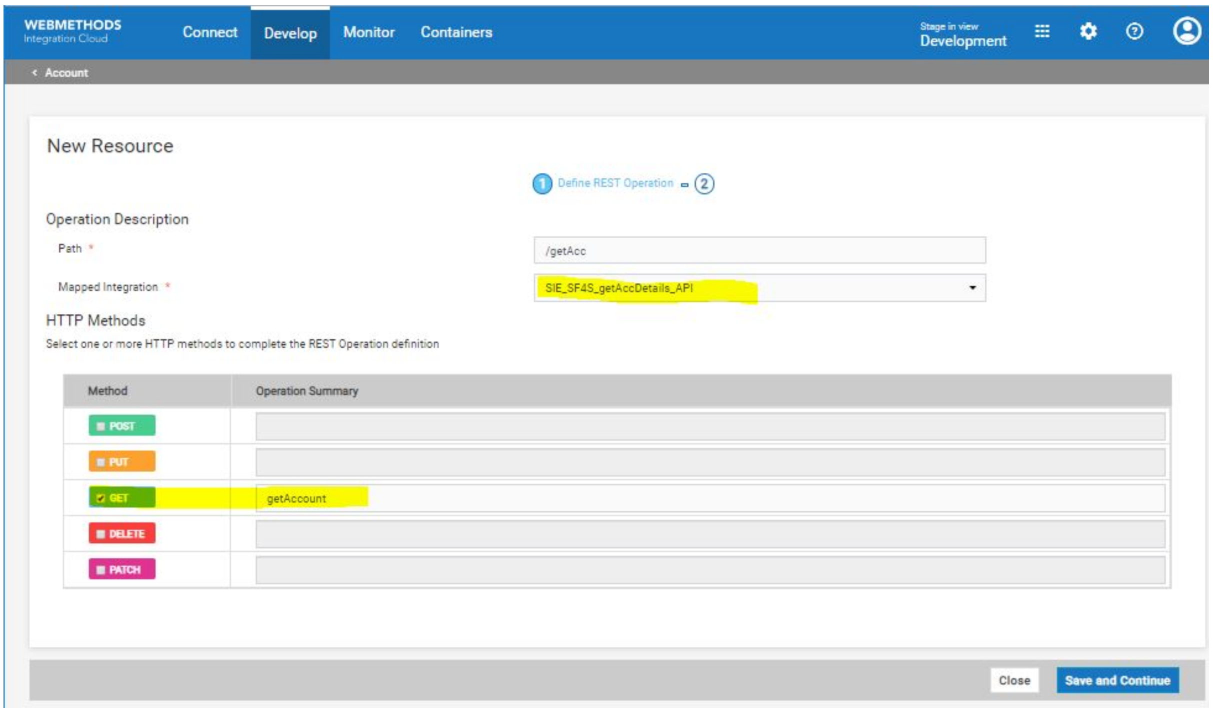
HTTP Code *	200
HTTP Code *	401

As you can see both methods have same signature for input. Of course! There is the same integration behind them. WMIC is smart in case of “Source” of the parameter. In case of GET it is a QUERY. But I don’t want to have same Input for getAccount as I have for searchAccount.

This approach is crashed by limitation of this tool.

Workaround:

Add another resource.



New Resource

1 Define REST Operation – 2

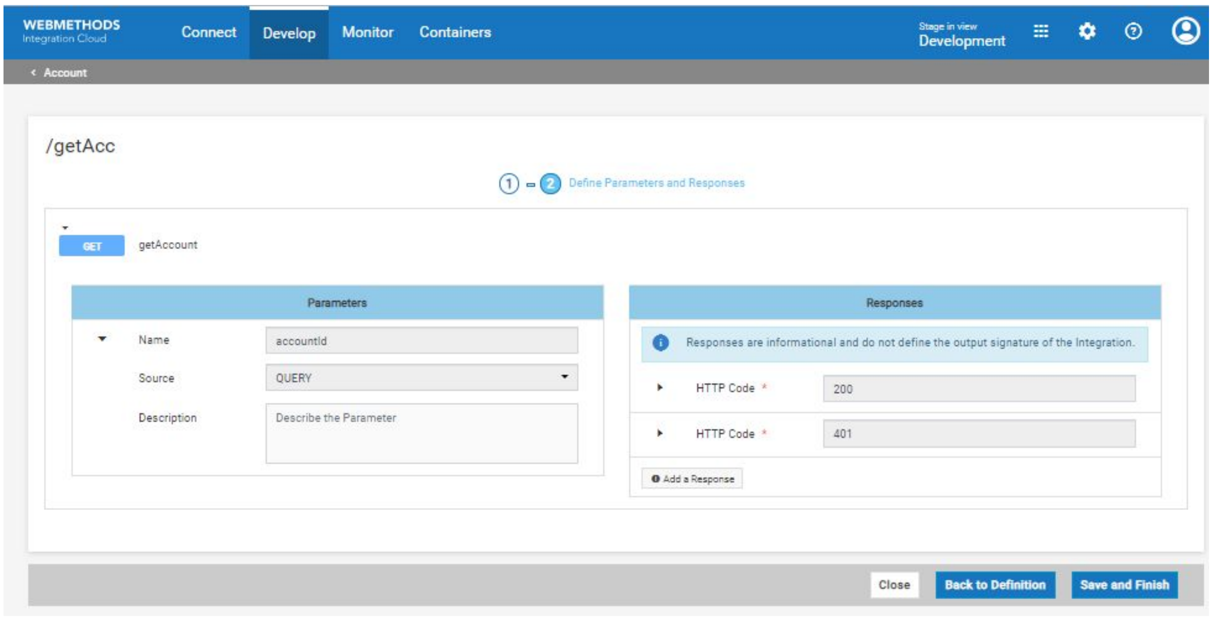
Operation Description

Path *	/getAcc
Mapped Integration *	SIE_SF4S_getAccDetails_API

HTTP Methods

Select one or more HTTP methods to complete the REST Operation definition

Method	Operation Summary
POST	
PUT	
✓ GET	getAccount
DELETE	
PATCH	



/getAcc

1 – 2 Define Parameters and Responses

GET getAccount

Parameters

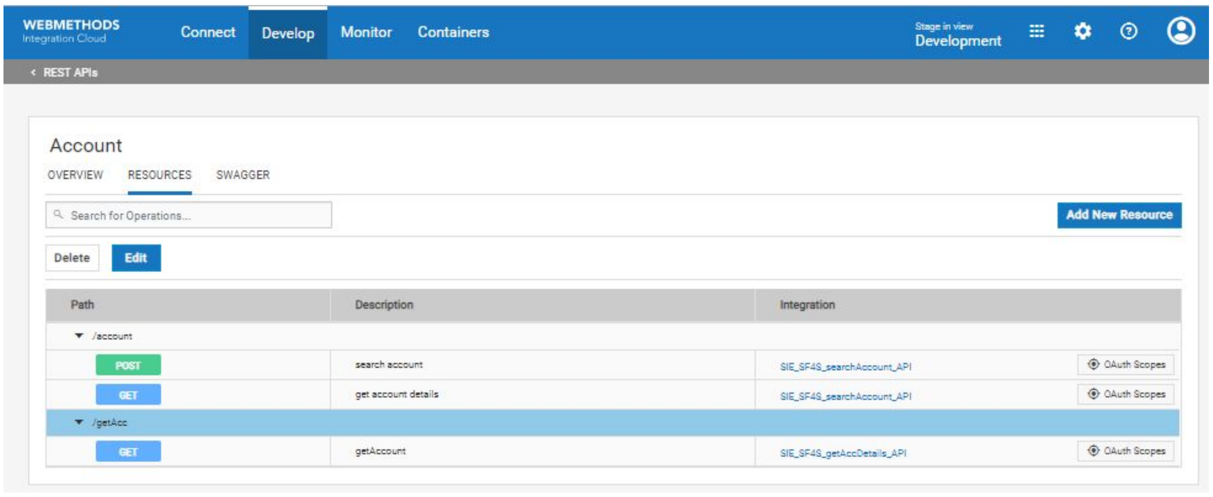
Name	accountId
Source	QUERY
Description	Describe the Parameter

Responses

Responses are informational and do not define the output signature of the Integration.

HTTP Code *	200
HTTP Code *	401

getAccount now have only ressource Id as input - great!
But now we have this structure:



Account

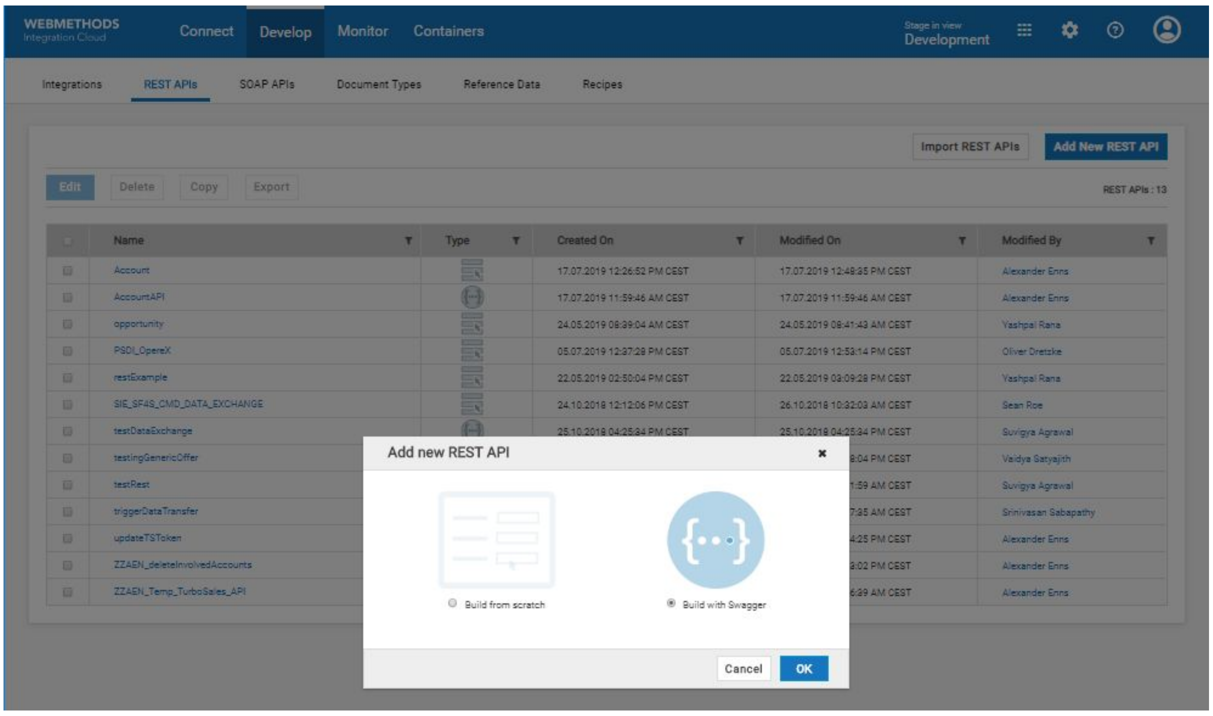
Path	Description	Integration
/account		
POST	search account	SIE_SF4S_searchAccount_API
GET	get account details	SIE_SF4S_searchAccount_API
/getAcc		
GET	getAccount	SIE_SF4S_getAccDetails_API

That’s an well known antipattern in REST!

“CRUD function names should never be used in URIs.”

Therefore we need to create a SAG Incident.

Approach 2 - build API from swagger

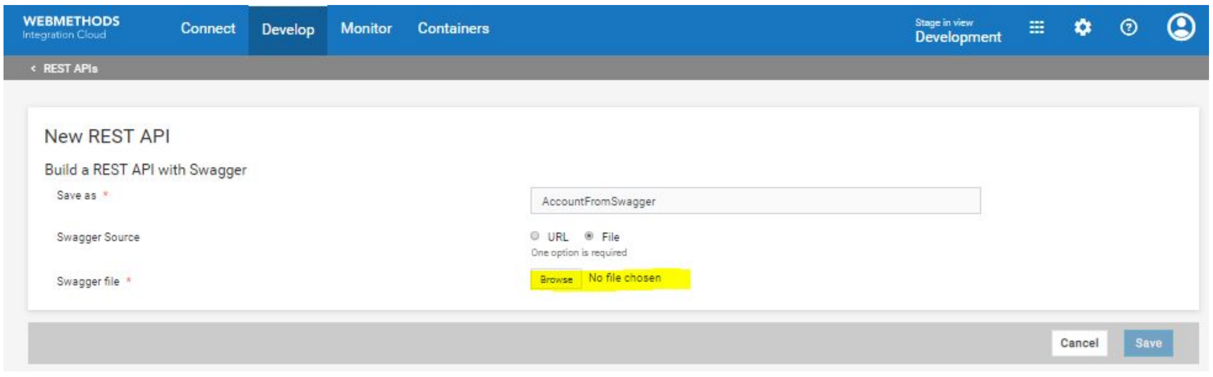


Name	Type	Created On	Modified On	Modified By
Account		17.07.2019 12:26:52 PM CEST	17.07.2019 12:48:35 PM CEST	Alexander Enns
AccountAPI		17.07.2019 11:59:46 AM CEST	17.07.2019 11:59:46 AM CEST	Alexander Enns
opportunity		24.05.2019 08:39:04 AM CEST	24.05.2019 08:41:43 AM CEST	Yashpal Rana
PSDI_OpereX		05.07.2019 12:37:28 PM CEST	05.07.2019 12:53:14 PM CEST	Oliver Dretzke
restExample		22.05.2019 02:50:04 PM CEST	22.05.2019 03:09:28 PM CEST	Yashpal Rana
SIE_SF4S_CMD_DATA_EXCHANGE		24.10.2018 12:12:06 PM CEST	26.10.2018 10:32:03 AM CEST	Sean Roe
testDataExchange		25.10.2018 04:25:34 PM CEST	25.10.2018 04:25:34 PM CEST	Suvigya Agrawal
testingGenericOffer			8:04 PM CEST	Vaidya Satyajith
testRest			1:59 AM CEST	Suvigya Agrawal
triggerDataTransfer			7:35 AM CEST	Srinivasan Sabapathy
updateTSToken			4:25 PM CEST	Alexander Enns
ZZAEN_deleteInvolvedAccounts			3:02 PM CEST	Alexander Enns
ZZAEN_Temp_TurboSales_API			6:39 AM CEST	Alexander Enns

Add new REST API

Build from scratch [✓ Build with Swagger]

Cancel OK



New REST API

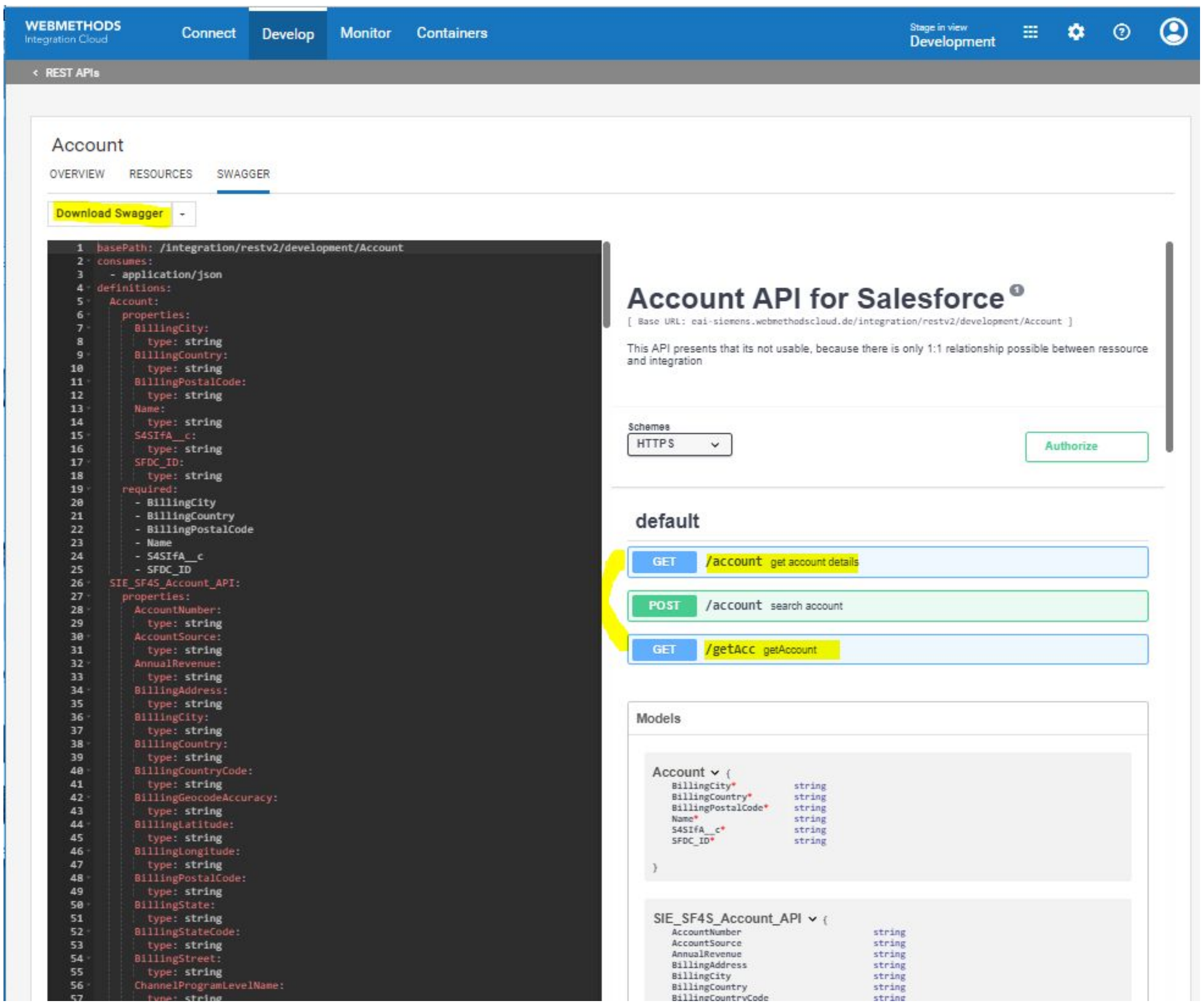
Build a REST API with Swagger

Save as *	AccountFromSwagger
Swagger Source	URL [✓ File] One option is required
Swagger file *	Browse No file chosen

I need a swagger file (YAML or JSON).

Since I don't want to write a swagger from scratch, I use the swagger from my previous API.

Idea is: merge both paths (account and getAcc) in swagger.



```
basePath: /integration/restv2/development/Account
consumes:
  - application/json
definitions:
  Account:
    properties:
      BillingCity:
        type: string
      BillingCountry:
        type: string
      BillingPostalCode:
        type: string
      Name:
        type: string
      S4SIfA__c:
        type: string
      SFDC_ID:
        type: string
    required:
      - BillingCity
      - BillingCountry
      - BillingPostalCode
      - Name
      - S4SIfA__c
      - SFDC_ID
  SIE_SF4S_Account_API:
    properties:
      AccountNumber:
        type: string
      AccountSource:
        type: string
      AnnualRevenue:
        type: string
      BillingAddress:
        type: string
      BillingCity:
        type: string
      BillingCountry:
        type: string
      BillingCountryCode:
        type: string
      BillingGeocodeAccuracy:
        type: string
      BillingLatitude:
        type: string
      BillingLongitude:
        type: string
      BillingPostalCode:
        type: string
      BillingState:
        type: string
      BillingStateCode:
        type: string
      BillingStreet:
        type: string
      ChannelProgramLevelName:
        type: string
```

Account API for Salesforce

[Base URL: cai-siemens.webmethodscloud.de/integration/restv2/development/Account]

This API presents that its not usable, because there is only 1:1 relationship possible between ressource and integration

```
GET   /account  get account details
POST  /account  search account
GET   /getAcc   getAccount
```

```
Account {
    BillingCity*          string
    BillingCountry*       string
    BillingPostalCode*    string
    Name*                 string
    S4SIfA__c*            string
    SFDC_ID*              string
}

SIE_SF4S_Account_API {
    AccountNumber                   string
    AccountSource                   string
    AnnualRevenue                   string
    BillingAddress                  string
    BillingCity                     string
    BillingCountry                  string
    BillingCountryCode              string
```

Download as JSON.

Manipulation of WMIC json:

1. merge paths (acc and searchAcc) - be aware of correct syntax.
2. Tip: change operationId definition. You should be able to see which Integration is used.

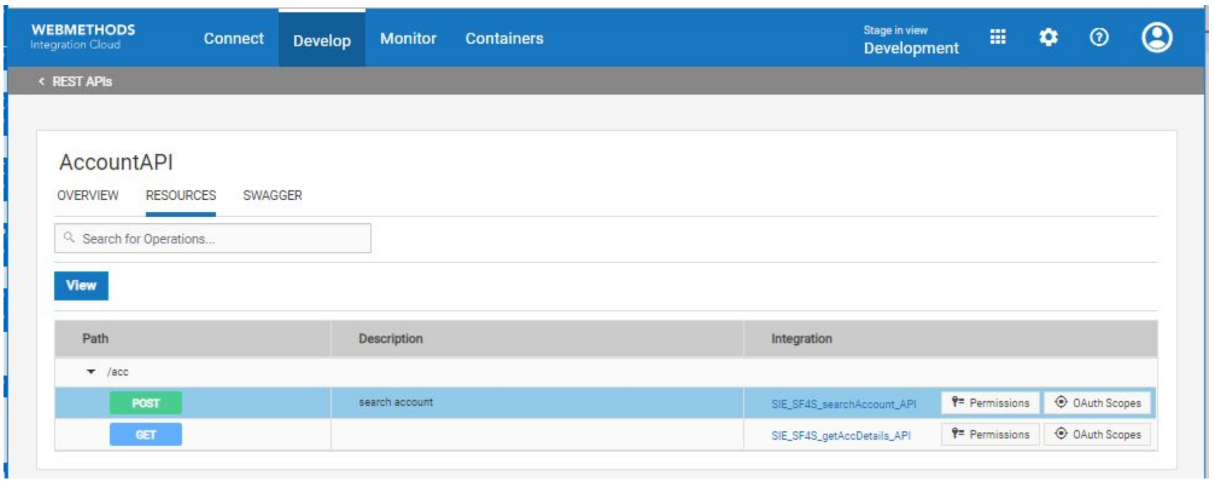
```
"paths":{
     "/account":{
        "get":{
           "description":"qa2 Test with: 0010Q00000QrpPkQAJ",
           "operationId":"SIE_SF4S_getAccDetails_API",
```

3. (realized later) move security _basicAuth by putting this after “path” and remove it from REST methods

```
"security":[
   {
      "_basicAuth":[

      ]
   }
]
```

Works fine!


WEBMETHODS Integration Cloud | Connect | Develop | Monitor | Containers | Stage in view Development

‹ REST APIs

AccountAPI

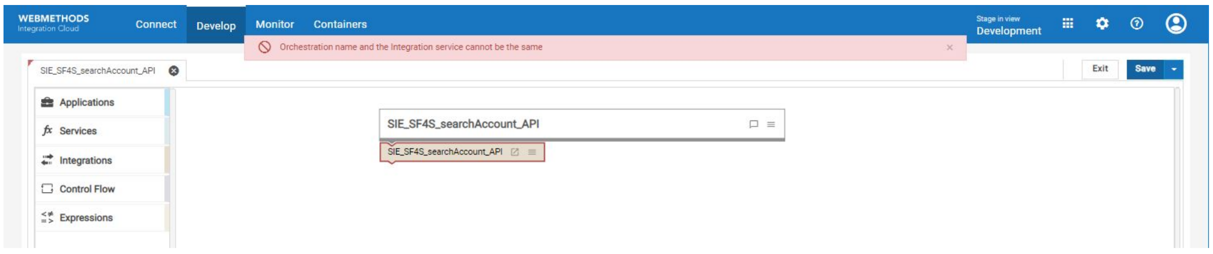
OVERVIEW | RESOURCES | SWAGGER

Search for Operations...

View

Path	Description	Integration
/acc		
POST	search account	SIE_SF4S_searchAccount_API Permissions OAuth Scopes
GET		SIE_SF4S_getAccDetails_API Permissions OAuth Scopes

But! My operationId must be different - argghh!


WEBMETHODS Integration Cloud | Connect | Develop | Monitor | Containers | Stage in view Development

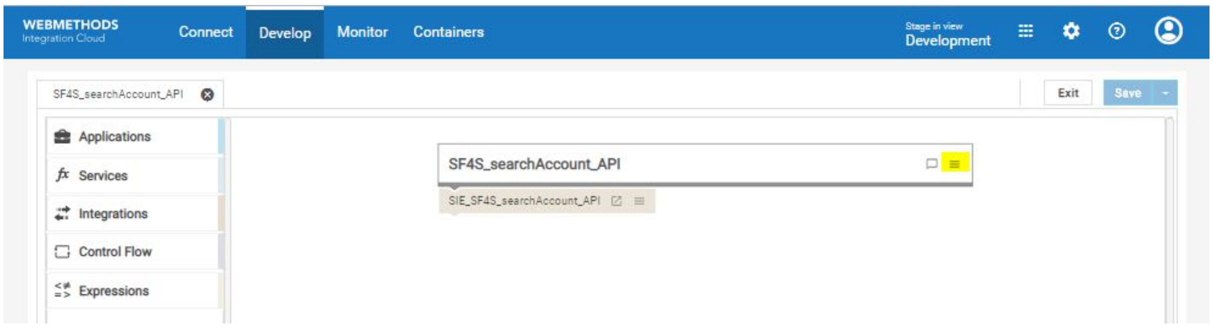
Orchestration name and the Integration service cannot be the same

SIE_SF4S_searchAccount_API | Exit | Save

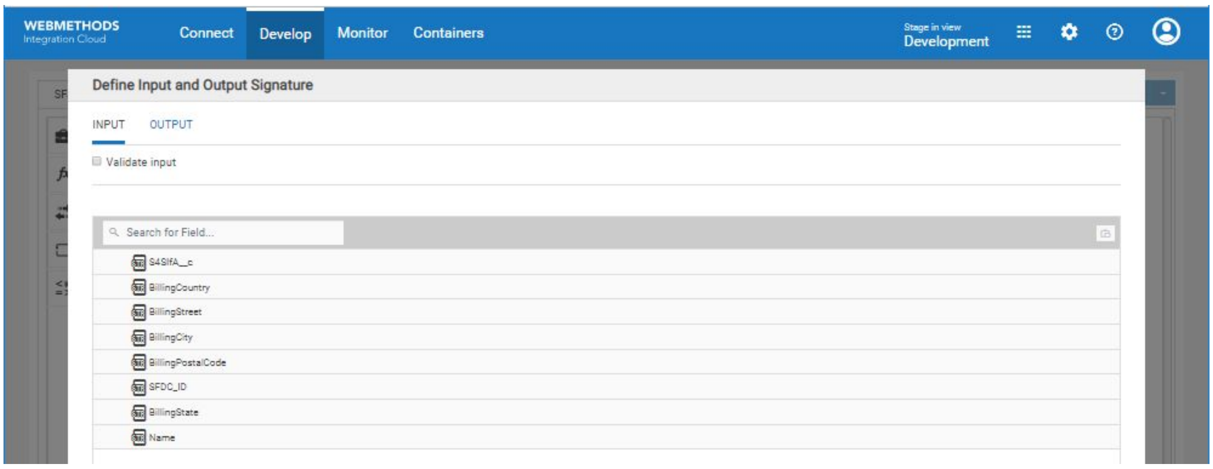
Applications
Services
Integrations
Control Flow
Expressions

SIE_SF4S_searchAccount_API
SIE_SF4S_searchAccount_API

I renamed the operationIds in my swagger. Now I can implement my operations:



Now I have to check, which input and output was generated.



Define Input and Output Signature

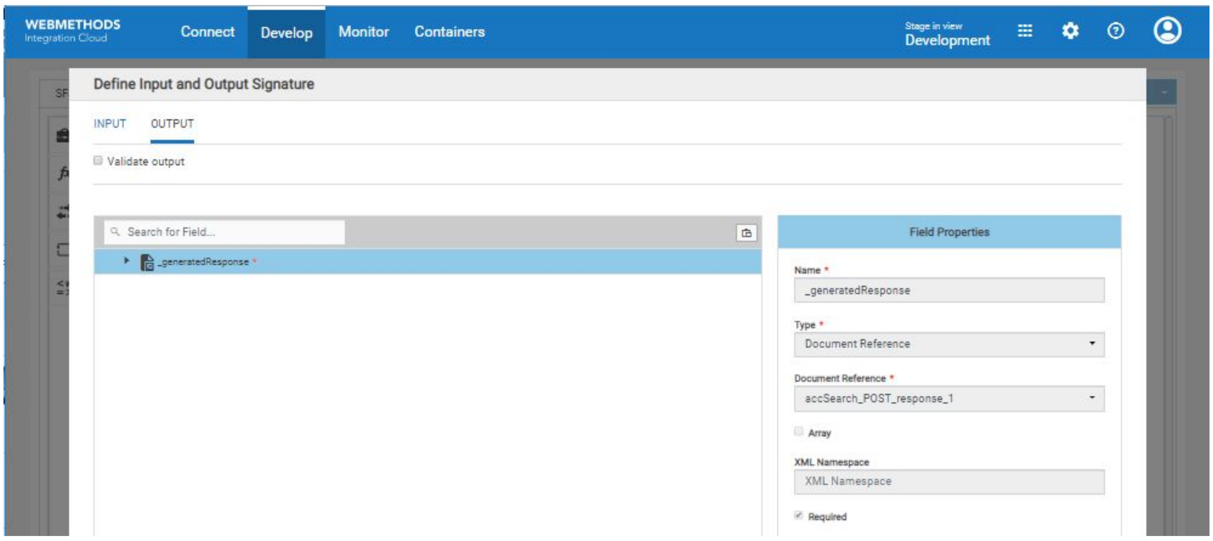
INPUT OUTPUT

Validate input

Search for Field...

S4SIfA__c
BillingCountry
BillingStreet
BillingCity
BillingPostalCode
SFDC_ID
BillingState
Name

Input is OK.



Define Input and Output Signature

INPUT OUTPUT

Validate output

Search for Field...

_generatedResponse *

Field Properties

Name *
_generatedResponse

Type *
Document Reference

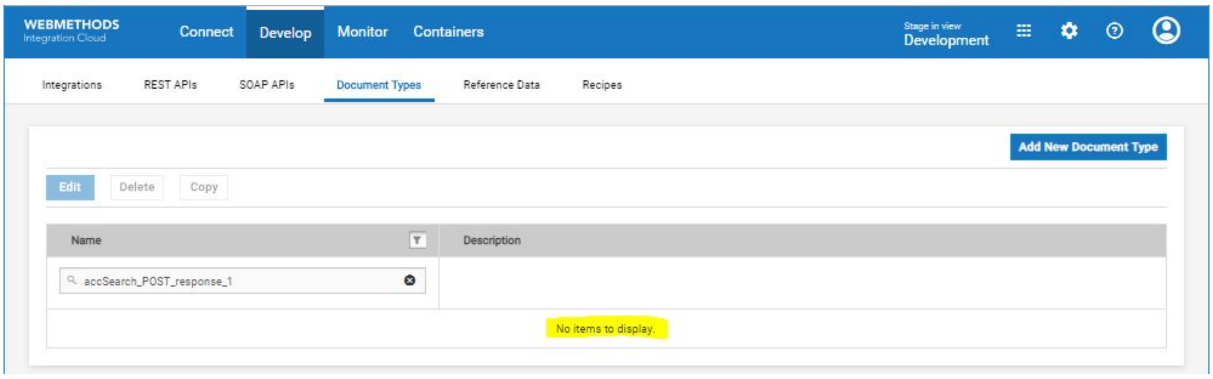
Document Reference *
accSearch_POST_response_1

Array

XML Namespace
XML Namespace

✓ Required

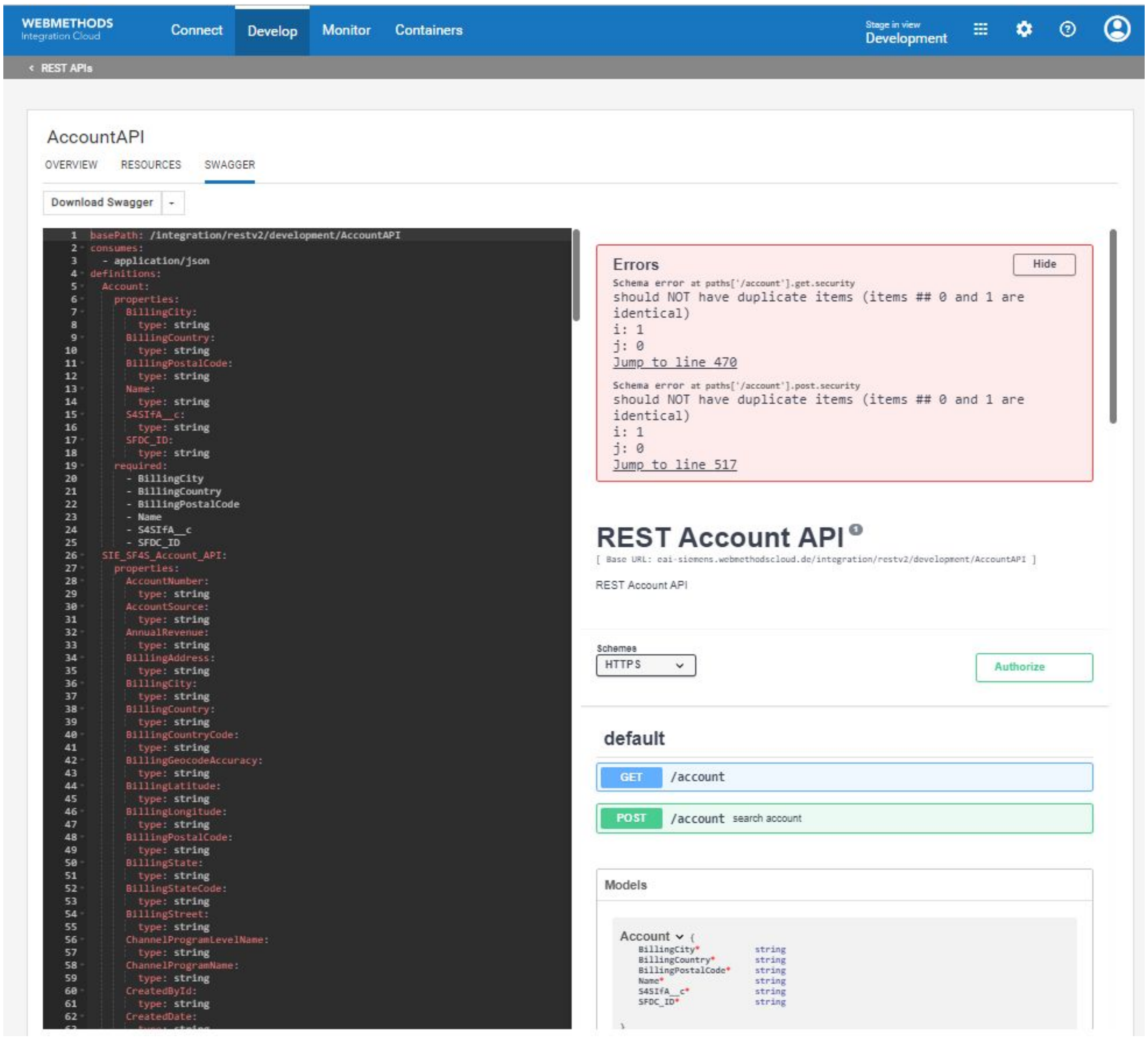
Output is a strange new doc type. I cannot expand this to see which fields are there. I need to search this doc type in WMIC doc type registry.



accSearch_POST_response_1

No items to display.

It is missing!
Maybe WMIC translates it automatically. Let's test our API.



```
basePath: /integration/restv2/development/AccountAPI
consumes:
  - application/json
definitions:
  Account:
    properties:
      BillingCity:
        type: string
      BillingCountry:
        type: string
      BillingPostalCode:
        type: string
      Name:
        type: string
      S4SIfA__c:
        type: string
      SFDC_ID:
        type: string
    required:
      - BillingCity
      - BillingCountry
      - BillingPostalCode
      - Name
      - S4SIfA__c
      - SFDC_ID
  SIE_SF4S_Account_API:
    properties:
      AccountNumber:
        type: string
      AccountSource:
        type: string
      AnnualRevenue:
        type: string
      BillingAddress:
        type: string
      BillingCity:
        type: string
      BillingCountry:
        type: string
      BillingCountryCode:
        type: string
      BillingGeocodeAccuracy:
        type: string
      BillingLatitude:
        type: string
      BillingLongitude:
        type: string
      BillingPostalCode:
        type: string
      BillingState:
        type: string
      BillingStateCode:
        type: string
      BillingStreet:
        type: string
      ChannelProgramLevelName:
        type: string
      ChannelProgramName:
        type: string
      CreatedById:
        type: string
      CreatedDate:
```

```
Schema error at paths['/account'].get.security
should NOT have duplicate items (items ## 0 and 1 are
identical)
i: 1
j: 0
Jump to line 470

Schema error at paths['/account'].post.security
should NOT have duplicate items (items ## 0 and 1 are
identical)
i: 1
j: 0
Jump to line 517
```

REST Account API

[Base URL: cai-siemens.webmethodscloud.de/integration/restv2/development/AccountAPI]

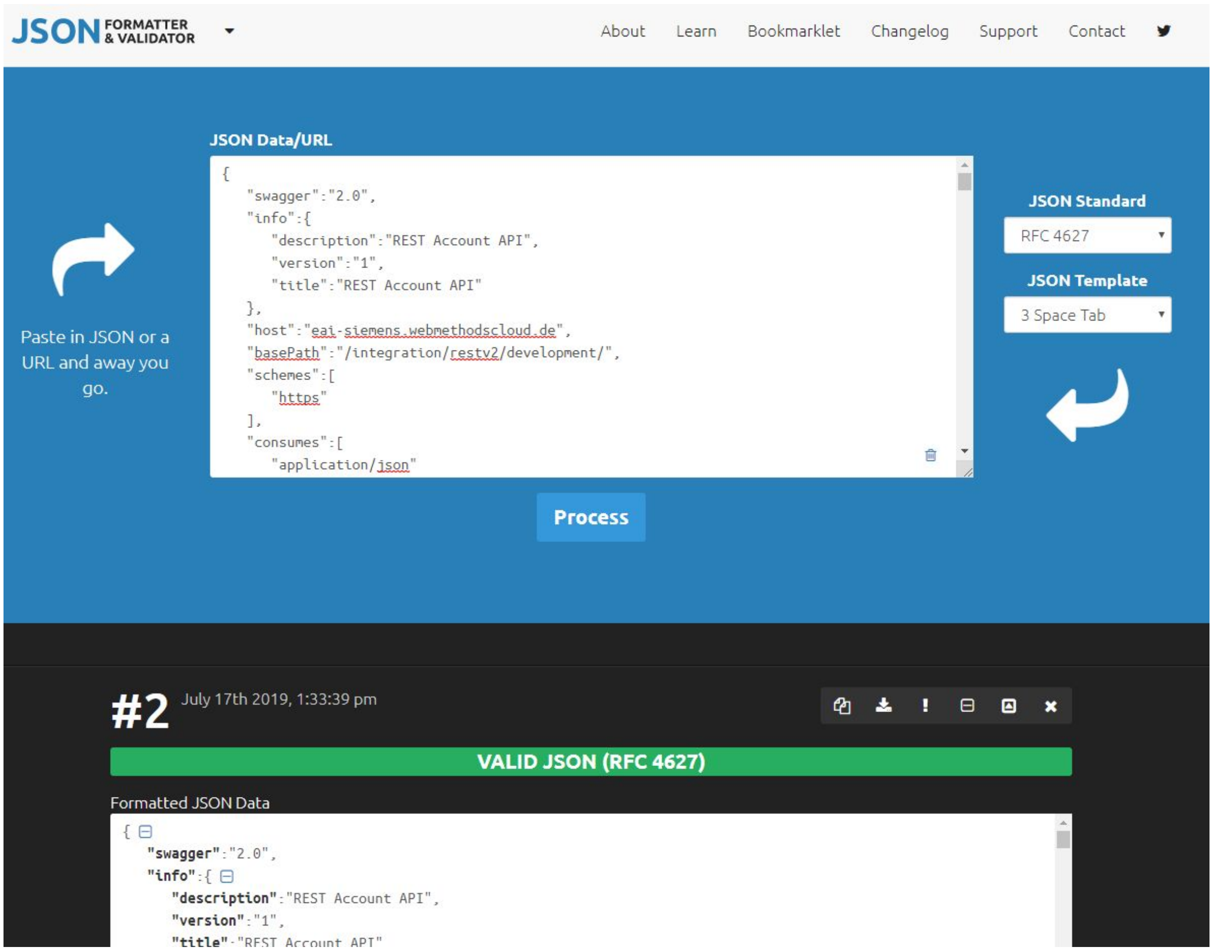
REST Account API

GET /account

POST /account search account

```
Account ∨ {
  BillingCity*          string
  BillingCountry*       string
  BillingPostalCode*    string
  Name*                 string
  S4SIfA__c*            string
  SFDC_ID*              string
}
```

Next surprise. WMIC see some schema errors in my swagger.
Let's check if my swagger is valid:



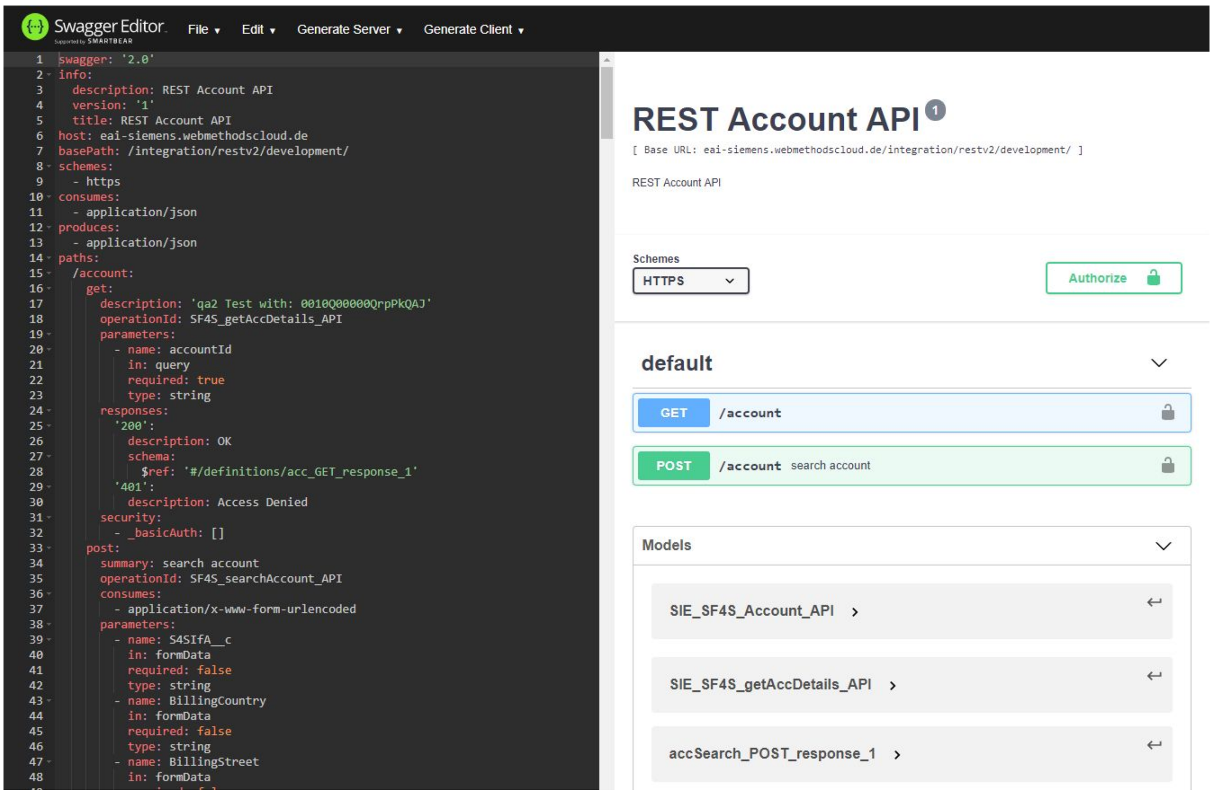
JSON Data/URL

```
{
   "swagger":"2.0",
   "info":{
      "description":"REST Account API",
      "version":"1",
      "title":"REST Account API"
   },
   "host":"eai-siemens.webmethodscloud.de",
   "basePath":"/integration/restv2/development/",
   "schemes":[
      "https"
   ],
   "consumes":[
      "application/json"
```

Formatted JSON Data

```
{
   "swagger":"2.0",
   "info":{
      "description":"REST Account API",
      "version":"1",
      "title":"REST Account API"
```

Same on https://editor.swagger.io/



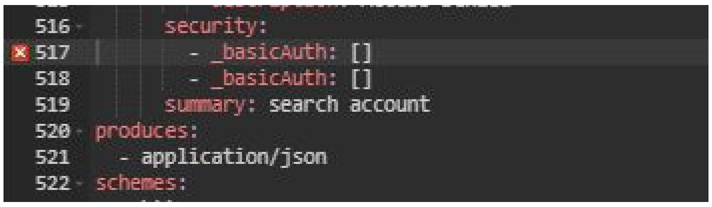
```
swagger: '2.0'
info:
  description: REST Account API
  version: '1'
  title: REST Account API
host: eai-siemens.webmethodscloud.de
basePath: /integration/restv2/development/
schemes:
  - https
consumes:
  - application/json
produces:
  - application/json
paths:
  /account:
    get:
      description: 'qa2 Test with: 0010Q00000QrpPkQAJ'
      operationId: SF4S_getAccDetails_API
      parameters:
        - name: accountId
          in: query
          required: true
          type: string
      responses:
        '200':
          description: OK
          schema:
            $ref: '#/definitions/acc_GET_response_1'
        '401':
          description: Access Denied
      security:
        - _basicAuth: []
    post:
      summary: search account
      operationId: SF4S_searchAccount_API
      consumes:
        - application/x-www-form-urlencoded
      parameters:
        - name: S4SIfA__c
          in: formData
          required: false
          type: string
        - name: BillingCountry
          in: formData
          required: false
          type: string
        - name: BillingStreet
          in: formData
```

REST Account API

[Base URL: eai-siemens.webmethodscloud.de/integration/restv2/development/]

REST Account API

WMIC shows this errors:
Errors
Hide
Schema error at paths['/account'].get.security
should NOT have duplicate items (items ## 0 and 1 are identical)
i: 1
j: 0
Jump to line 470
Schema error at paths['/account'].post.security
should NOT have duplicate items (items ## 0 and 1 are identical)
i: 1
j: 0
Jump to line 517

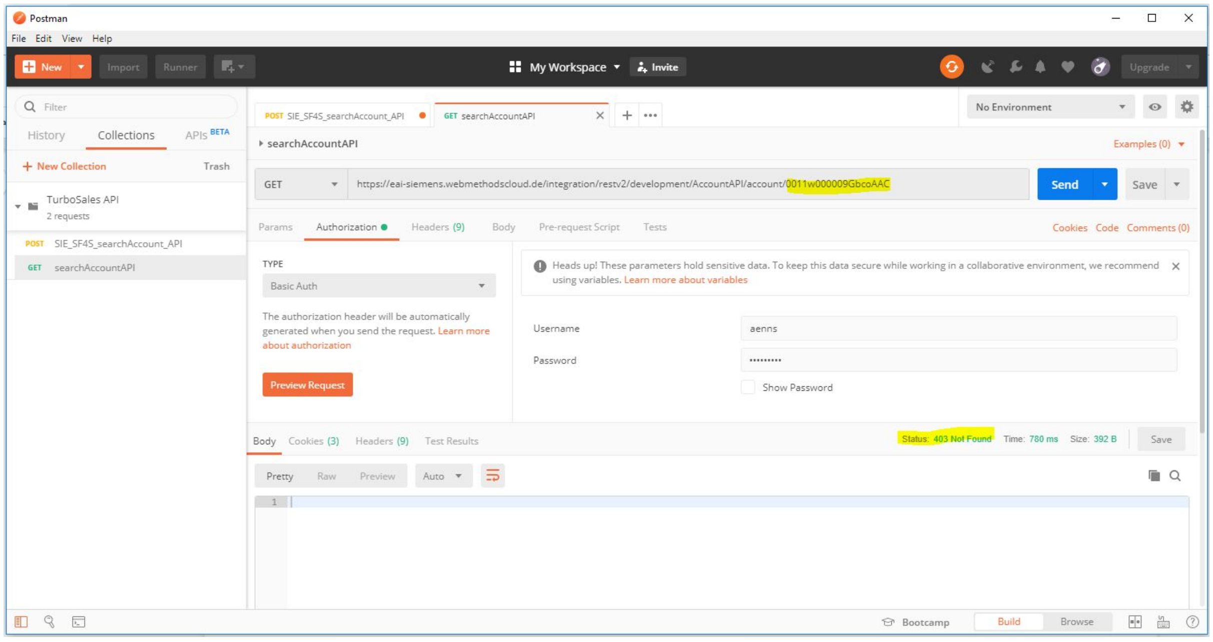


```
516        security:
517          - _basicAuth: []
518          - _basicAuth: []
519        summary: search account
520  produces:
521    - application/json
522  schemes:
```

There are two _basicAuth entries. Maybe I added them in my JSON.
No, I should remove it in get and post section and add single one on path level. See list

After regenerate it seems OK. Test again:

Test from Postman responses 403 Not found.
Used URL:
https://eai-siemens.webmethodscloud.de/integration/restv2/development/AccountAPI/account/0011w000009GbcoAAC



GET https://eai-siemens.webmethodscloud.de/integration/restv2/development/AccountAPI/account/0011w000009GbcoAAC

TYPE
Basic Auth

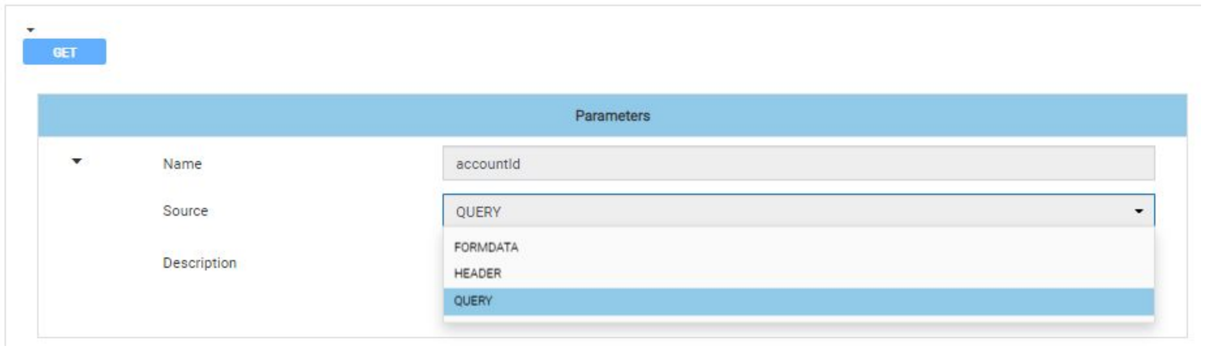
The authorization header will be automatically generated when you send the request. Learn more about authorization

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables

Username aenns
Password ••••••••••
Show Password

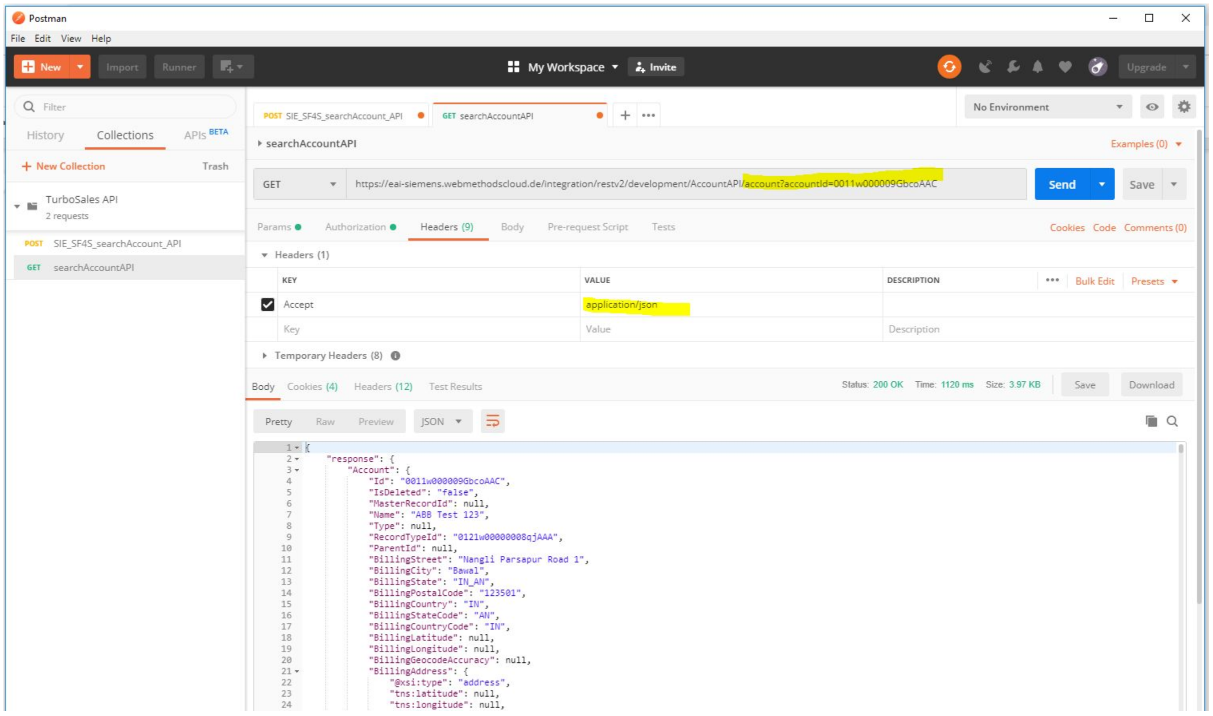
Status: 403 Not Found Time: 780 ms Size: 392 B

Of course. accountId is defined as query and not as path segment parameter.



Name	accountId
Source	QUERY
Description	

Now I try to test it in WMIC
With URL params it works. But that’s not the best idea to use REST. The common usage of query params is if you want to search something. Query params returns empty lists or objects. Path param return 404 Not found if resource is not in the provider system. But OK. Our Request URL should also be:
https://eai-siemens.webmethodscloud.de/integration/restv2/development/AccountAPI/account?accountId=0011w000009GbcoAAC
Now it works in Postman:



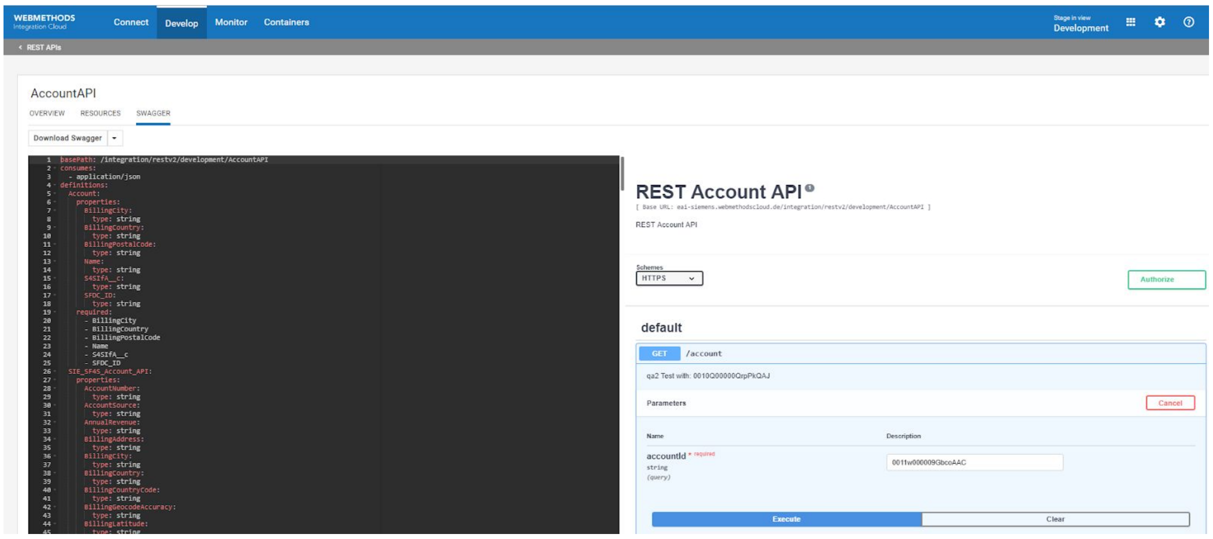
GET https://eai-siemens.webmethodscloud.de/integration/restv2/development/AccountAPI/account?accountId=0011w000009GbcoAAC

KEY	VALUE	DESCRIPTION
✓ Accept	application/json	
Key	Value	Description

Status: 200 OK Time: 1120 ms Size: 3.97 KB

```
{
    "response": {
        "Account": {
            "Id": "0011w000009GbcoAAC",
            "IsDeleted": "false",
            "MasterRecordId": null,
            "Name": "ABB Test 123",
            "Type": null,
            "RecordTypeId": "0121w00000008qjAAA",
            "ParentId": null,
            "BillingStreet": "Nangli Parsapur Road 1",
            "BillingCity": "Bawal",
            "BillingState": "IN_AN",
            "BillingPostalCode": "123501",
            "BillingCountry": "IN",
            "BillingStateCode": "AN",
            "BillingCountryCode": "IN",
            "BillingLatitude": null,
            "BillingLongitude": null,
            "BillingGeocodeAccuracy": null,
            "BillingAddress": {
                "@xsi:type": "address",
                "tns:latitude": null,
                "tns:longitude": null,
```

And in WMIC UI:



```
basePath: /integration/restv2/development/AccountAPI
consumes:
  - application/json
definitions:
  Account:
    properties:
      BillingCity:
        type: string
      BillingCountry:
        type: string
      BillingPostalCode:
        type: string
      Name:
        type: string
      S4SIfA__c:
        type: string
      SFDC_ID:
        type: string
    required:
      - BillingCity
      - BillingCountry
      - BillingPostalCode
      - Name
      - S4SIfA__c
      - SFDC_ID
  SIE_SF4S_Account_API:
    properties:
      AccountNumber:
        type: string
      AccountSource:
        type: string
      AnnualRevenue:
        type: string
      BillingAddress:
        type: string
      BillingCity:
        type: string
      BillingCountry:
        type: string
      BillingCountryCode:
        type: string
      BillingGeocodeAccuracy:
        type: string
      BillingLatitude:
```

REST Account API
[Base URL: eai-siemens.webmethodscloud.de/integration/restv2/development/AccountAPI]
REST Account API

Schemes: HTTPS

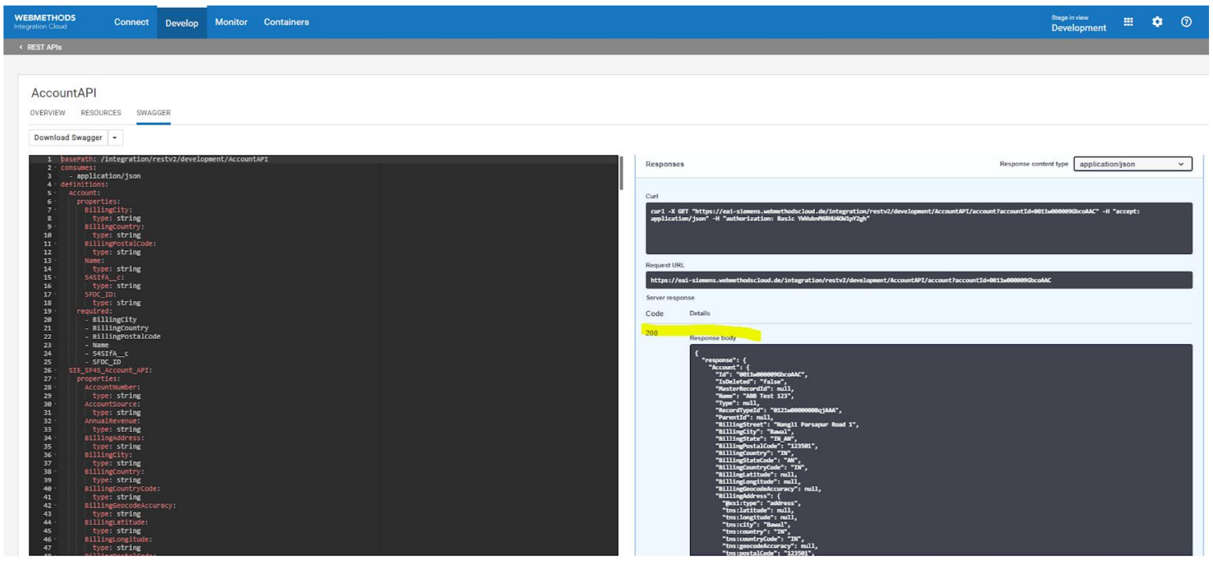
default

GET /account

qa2 Test with: 0010Q00000QrpPkQAJ

Parameters

Name	Description
accountId * required string (query)	0011w000009GbcoAAC



```
basePath: /integration/restv2/development/AccountAPI
consumes:
  - application/json
definitions:
  Account:
    properties:
      BillingCity:
        type: string
      BillingCountry:
        type: string
      BillingPostalCode:
        type: string
      Name:
        type: string
      S4SIfA__c:
        type: string
      SFDC_ID:
        type: string
    required:
      - BillingCity
      - BillingCountry
      - BillingPostalCode
      - Name
      - S4SIfA__c
      - SFDC_ID
  SIE_SF4S_Account_API:
    properties:
      AccountNumber:
        type: string
      AccountSource:
        type: string
      AnnualRevenue:
        type: string
      BillingAddress:
        type: string
      BillingCity:
        type: string
      BillingCountry:
        type: string
      BillingCountryCode:
        type: string
      BillingGeocodeAccuracy:
        type: string
      BillingLatitude:
        type: string
      BillingLongitude:
        type: string
```

Responses

Response content type: application/json

Curl

```
curl -X GET "https://eai-siemens.webmethodscloud.de/integration/restv2/development/AccountAPI/account?accountId=0011w000009GbcoAAC" -H "accept: application/json" -H "authorization: Basic YWVubnM6RHU4OW1pY2gh"
```

Request URL

```
https://eai-siemens.webmethodscloud.de/integration/restv2/development/AccountAPI/account?accountId=0011w000009GbcoAAC
```

Server response

Code Details

200

Response body

```
{
  "response": {
    "Account": {
      "Id": "0011w000009GbcoAAC",
      "IsDeleted": "false",
      "MasterRecordId": null,
      "Name": "ABB Test 123",
      "Type": null,
      "RecordTypeId": "0121w00000008qjAAA",
      "ParentId": null,
      "BillingStreet": "Nangli Parsapur Road 1",
      "BillingCity": "Bawal",
      "BillingState": "IN_AN",
      "BillingPostalCode": "123501",
      "BillingCountry": "IN",
      "BillingStateCode": "AN",
      "BillingCountryCode": "IN",
      "BillingLatitude": null,
      "BillingLongitude": null,
      "BillingGeocodeAccuracy": null,
      "BillingAddress": {
        "@xsi:type": "address",
        "tns:latitude": null,
        "tns:longitude": null,
        "tns:city": "Bawal",
        "tns:country": "IN",
        "tns:countryCode": "IN",
        "tns:geocodeAccuracy": null,
        "tns:postalCode": "123501",
```

Conclusion:

- Use swagger first approach
- use query params in GET services