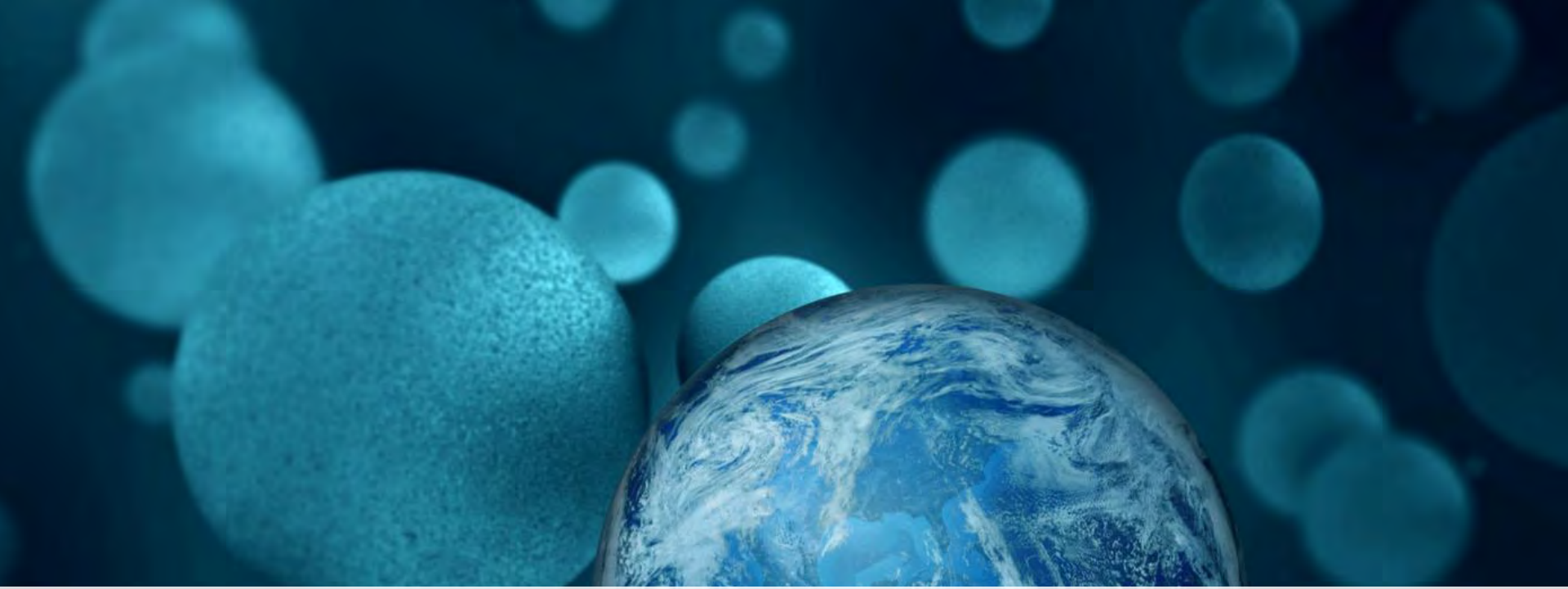
Leveraging Web Services – A Real World Example

ThermoFisher SCIENTIFIC



ThermoFisher SCIENTIFIC

Keen Haynes

Senior ColdFusion Engineer
17+ years with ColdFusion
keen.haynes@thermofisher.com

Doug Rehg

Senior ColdFusion Engineer
15+ years with ColdFusion
doug.rehg@thermofisher.com

What This Is – What It Is Not

This is not a “best practices” discussion but real world overview of how we did it at ThermoFisher and how we changed our approach with CF2016 and the API Manager

Agenda

- Soap Example
- RESTful Example
- Our Challenge
- Decisions We Faced
  - REST VS SOAP
  - Authentication
  - Tracking
- Leveraging CF 2016 and the API Manager
  - Security
  - Testing
  - Swagger Docs
  - Dashboard (Tracking)
- Application Demo
- Questions

Basic SOAP

```
<cfcomponent output="false" style="document">
   <cffunction name=" getAuthorDetails "  access="remote" output="no" returntype=" query"
   hint="returns author details">

         <cfargument name="authorid" required="false" default="0" type="numeric">

         <cfquery name="qAuthorDetails" datasource="#request.readDSN#">
              select * from app.authors
              <cfif arguments.authorid>
                   where authorid = #arguments.authorid#
              </cfif>
          </cfquery>
         <cfreturn qAuthorDetails>
     </cffunction>
</cfcomponent>
```

Calling SOAP

Pass Optional argument with value

```
<cfinvoke webservice="http://localhost/webservices/authorSoap.cfc?wsdl"
   method=" getAuthorDetails " returnvariable=“authDetails">
   <cfinvokeargument name=“authorid" value=“5" />
</cfinvoke>
```

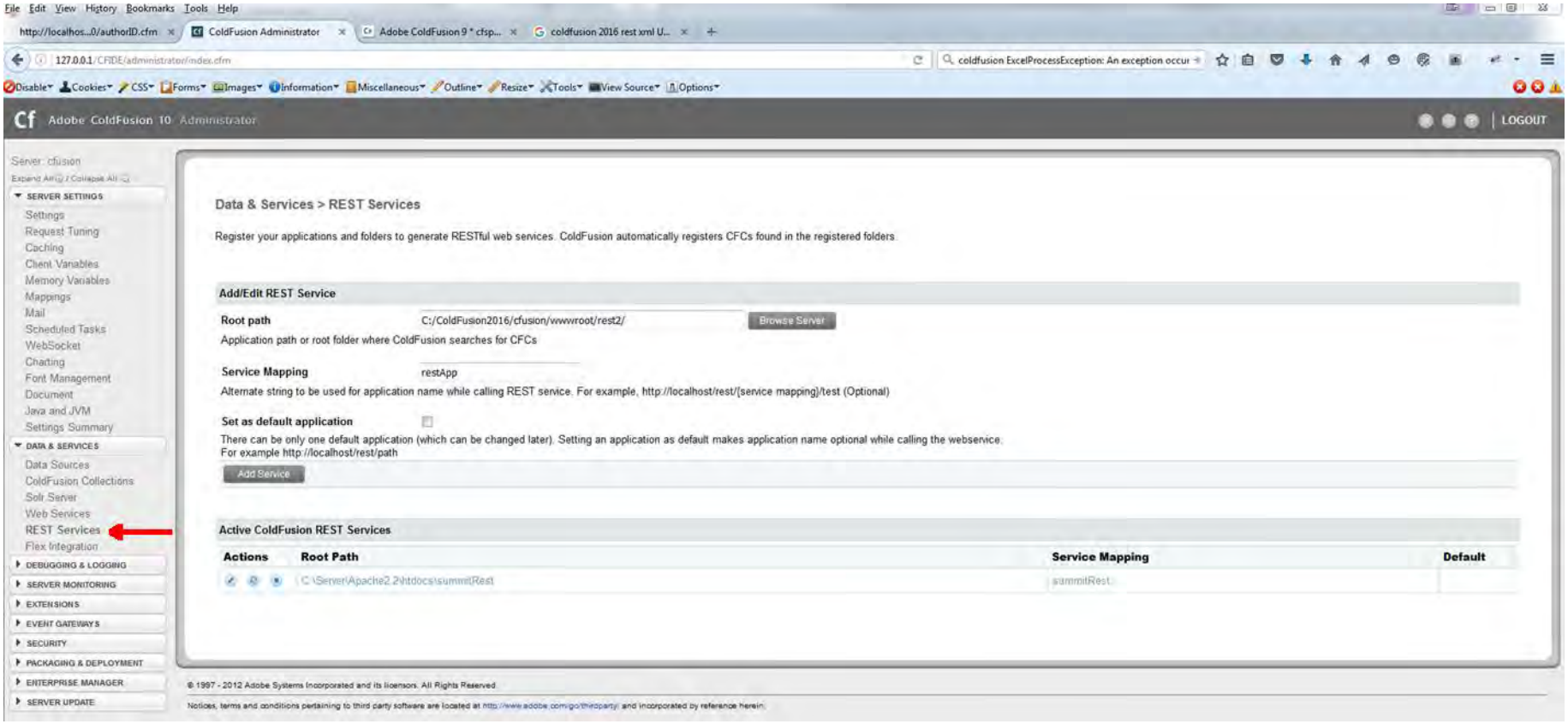
Pass Optional argument without value

```
<cfinvoke webservice="http://localhost/webservices/authorSoap.cfc?wsdl"
   method=" getAuthorDetails " returnvariable=“authDetails">
   <cfinvokeargument name=“authorid" value=““ omit=“true” />
</cfinvoke>
```

Basic REST

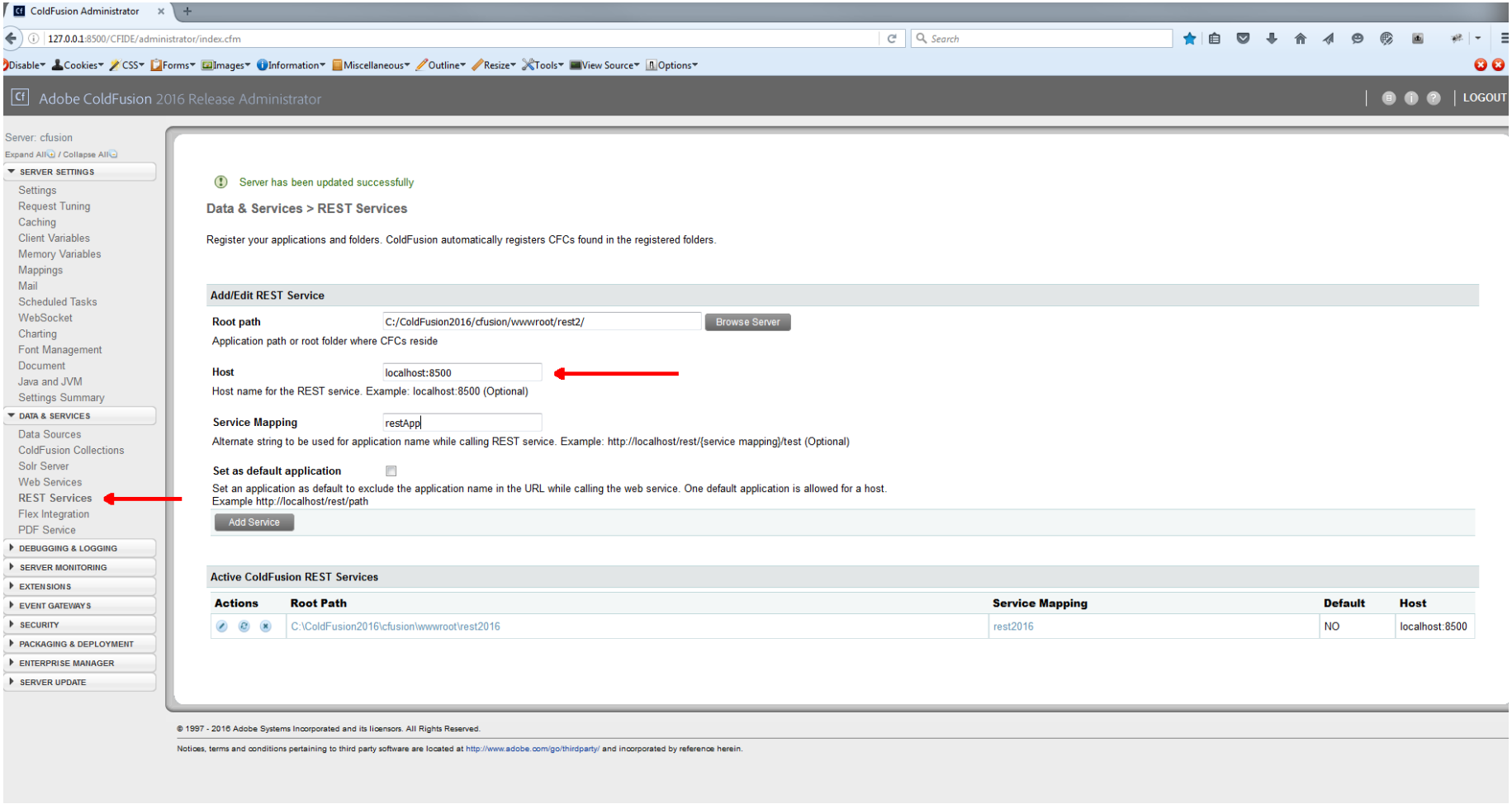
```
<cfcomponent rest="true" restpath="/authors" displayname="authors" >

    <cffunction name="getAuthorDetails" access="remote" httpmethod="GET"
    returntype="any" restpath="{authorid}" produces="application/xml">

        <cfargument name="authorid" required="true"  restargsource="Path"/>
        <cfquery name="qAuthorDetails" datasource="#request.readDSN#">
            select * from app.authors
            where authorid = #arguments.authorid#
         </cfquery>
        <cfreturn qAuthorDetails>
    </cffunction>
</cfcomponent>
```

Register Your REST Service - ColdFusion 10



ThermoFisher SCIENTIFIC

Register Your REST Service - CF2016



ThermoFisher SCIENTIFIC

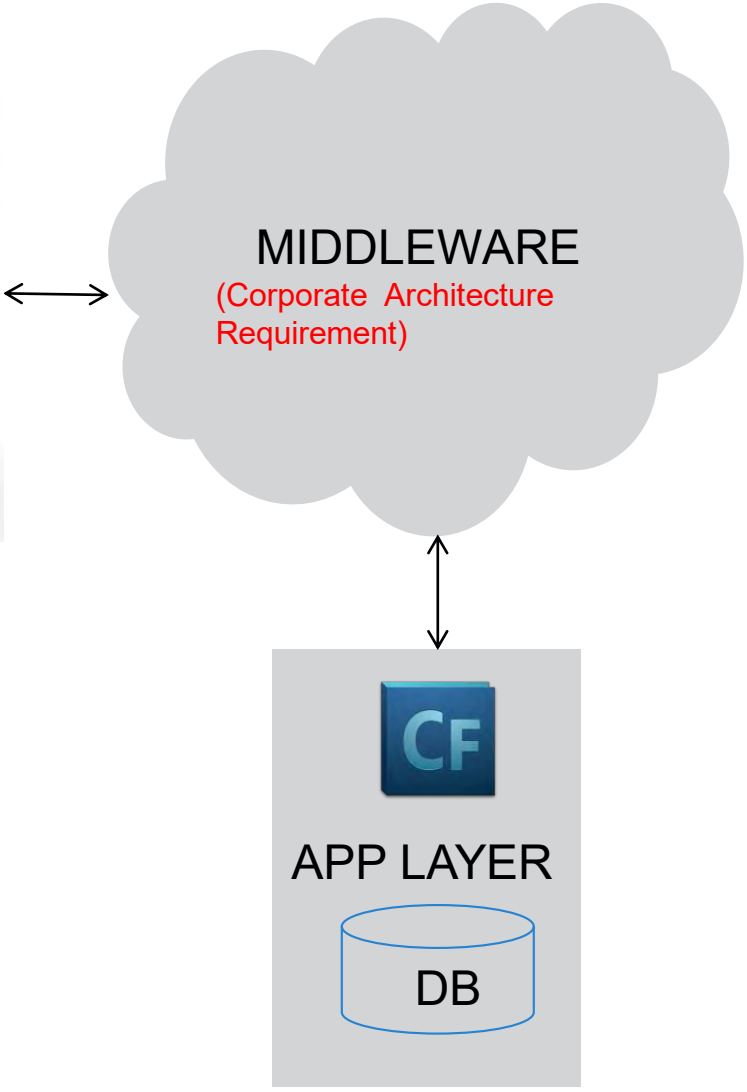
Calling Your REST Service

- <cfhttp url="http://localhost/rest/summitRest/authors/1" result="restResult" method="get">
- </cfhttp>

Our Challenge

- Provide our customers access to Thermo Fisher products real time on their own campus
- Make it super simple for customers to purchase and retrieve our products
- The products needed to be securely stored and refrigerated
- Inventory and replenishment needed to be automatic

Decisions We Faced



SOAP VS. RESTful

DOCUMENTATION

TESTING

SECURITY

Initial Solution – SOAP VS. RESTful

Soap

- RESTful not supported by CF8
- Middleware utilized SOAP

RESTful

- CF 10 introduced REST support
- Approval to bypass middleware

Initial Solution – SOAP VS. RESTful

SOAP

Documentation

- WSDL

Testing

- Custom CFML Test Harnesses
- SOAP UI

Security

- Security handled by Oracle OFM Middleware – Server Recognition

RESTful

Documentation

- No self documenting tools for REST
- Unfamiliar with SWAGGER
- Hand coded in CFML

Testing

- Custom CFML Test Harnesses
- POSTMAN / BOOMERANG

Security

- Coded

Initial Solution – Books RESTful Code Snippet

```
<cfcomponent rest="true" restpath="/books" displayname="books"   extends="packetTracking">
   <cffunction name="getBookDetailsJSON" access="remote" httpmethod="GET" returntype="any"
     restpath="{bookid}"  produces="application/json">
     <cfargument name="bookid" required="true" restargsource="Path" type="string">
     <cfinclude template="inc_tracking.cfm">
     <cfquery name="qBookDetails" datasource="#request.readDSN#">
        select b.bookid, a.firstname, a.lastname, a.authorid, b.title, b.bookimage, b.bookdescription,
         b.isspotlight, b.genre
        from app.books b, app.authors a
        where a.authorid = b.authorid
        and b.bookid = #arguments.bookid#
     </cfquery>
     <cfset temp = restThrow(throw="false", errorCode="200", message="OK", detail="Action completed
     successfully.", data="#serializeJSON(qBookDetails)#", sourcename="#getfunctioncalledname()#")>
     <cfreturn qBookDetails>
</cffunction>
```

Initial Solution – inc_tracking.cfm Snippet

```
<cfset httpRequest = getHTTPRequestData()>
<cfset request.rest = {}>
<cfset request.rest["sessionId"] = createUUID()>
<cfset request.rest["tsStart"] = now()>
<cfset request.rest["restPath"] = CGI.PATH_INFO>

<cfif isDefined('httpRequest.Headers.Authorization')>
    <cfset request.rest["authString"] = ToString(ToBinary(
ListLast(GetHTTPRequestData().Headers.Authorization," ") ))>
    <cfset request.rest["memberEmail"] = ListFirst(request.rest.authString,":")>
    <cfset request.rest["memberPassword"] = ListLast(request.rest.authString,":")>
<cfelse>
    <cfset request.rest["authString"] = "NA">
    <cfset request.rest["memberEmail"] = "NA">
    <cfset request.rest["memberPassword"] = "NA">
</cfif>

<cfset theDir=GetDirectoryFromPath(GetCurrentTemplatePath())>
<cfset packetFileName = theDir & "packetTracker.xls">
```

Initial Solution – inc_tracking.cfm cont.

```
<cfif FileExists(packetFileName)>

    <cfspreadsheet action="read" src="#packetFileName#" sheetname="packets"
            name="packetTracker">
    <cfset spreadsheetAddRow(packetTracker,"#request.rest.sessionId#,#request.rest.restPath# ,
            #request.rest.authString#,,,,,#request.rest.tsStart#",2,1)>
    <cfspreadsheet action="write" filename="#packetFileName#" name="packetTracker"
            sheetname="packets" overwrite=true>

<cfelse>

    <cfset  theSheet = SpreadsheetNew("packetTracker")>
    <cfset spreadsheetAddRow(theSheet,"sessionId,restPath,authString,responseStatusCode,
            responseStatusMessage,responseStatusDetail,responseData,tsStart,tsEnd",1,1)>
    <cfset spreadsheetAddRow(theSheet,"#request.rest.sessionId#,#request.rest.restPath#,
            #request.rest.authString#,,,,,#request.rest.tsStart#",2,1)>
   <cfspreadsheet action="write" filename="#packetFileName#" name="theSheet"
            sheetname="packets" overwrite=true>

</cfif>
```

Initial Solution – tracking.cfc Snippet

```
<cffunction name="restThrow" returntype="STRING">

    <cfargument name="errorcode" type="any" default="" required="false">
    <cfargument name="message" type="string" default="" required="false">
    <cfargument name="detail" type="string" default="" required="false">
    <cfargument name="data" type="string" default="" required="false">
    <cfargument name="throw" type="boolean" default="true" required="false" hint="determines
            whether or not to throw the actual exception">
    <cfargument name="sendNotification" type="string" default="AllNon200s">
    <cfargument name="sourcename" type="string" default="none">

    <cfset returnval = serializeJSON(arguments)>

    <cfset theDir=GetDirectoryFromPath(GetCurrentTemplatePath())>
    <cfset packetFileName = theDir & "packetTracker.xls“>

     <cfspreadsheet action="read" src="#packetFileName#" sheetname="packets"
            name="theSheet">
     <cfset spreadsheetAddRow(theSheet, #request.rest.sessionId#,,,#arguments.errorcode#,
            #arguments.message#,#arguments.detail#,'#arguments.data#',,#NOW()#",2,1)>
     <cfspreadsheet action="write" filename="#packetFileName#" name="theSheet"
            sheetname="packets" overwrite=true>
```

Manual Tracking Spreadsheet

A1 sessionId

	A	B	C	D	E	F	G
1	sessionId	restPath	authString	responseStatusCode	responseStatusMessage	responseStatusDetail	responseData
2	A2428B8C-ACBA-BC9D-3771263E9DA8A106			200	OK	Action completed successfully.	{"COLUMNS":["AUTHORID","FIRSTNAME","LASTNAME","BIO","ISSPOTLIGHT"],"DATA":[
3	A2428B8C-ACBA-BC9D-3771263E9DA8A106	/summitRest/authors/authorsList	NA				
4	A23404BE-9FD0-C815-567F5074FBD68E80				Variable QAUTHORDETAILS is undefined.		
5	A23404BE-9FD0-C815-567F5074FBD68E80	/summitRest/authors/authorsList	NA				
6	3F38A5A7-9142-867C-874DEF0D4C9FC5FB			200	OK	Action completed successfully.	{"COLUMNS":["AUTHORID","FIRSTNAME","LASTNAME","BIO","ISSPOTLIGHT"],"DATA":[
7	3F38A5A7-9142-867C-874DEF0D4C9FC5FB	/summitRest/authors/8	NA				
8	3F3509B9-C2D9-9801-875905AD1E6EDD28			200	OK	Action completed successfully.	{"COLUMNS":["AUTHORID","FIRSTNAME","LASTNAME","BIO","ISSPOTLIGHT"],"DATA":[
9	3F3509B9-C2D9-9801-875905AD1E6EDD28	/summitRest/authors/1	NA				
10	3F329A96-C1EF-0A78-7922F3D8895E9535			404	Author not found	An author with ID 588 was not found	
11	3F329A96-C1EF-0A78-7922F3D8895E9535	/summitRest/authors/588	NA				
12	3BE8F2BC-A1A5-F87F-ECE55106B1A9E01A			200	OK	Action completed successfully.	{"EXECUTIONTIME":16,"SQL":"\n\t\t\t\tdelete from app.books\n\t\t\t\twhere bookid = 16 ","F
13	3BE8F2BC-A1A5-F87F-ECE55106B1A9E01A	/summitRest/books/16	super@demodata.com:demo				
14	3BDD3932-B434-8274-85994536B5FEE85D			200	OK	Action completed successfully.	{"EXECUTIONTIME":0,"SQL":"\n\t\t\t\tupdate app.books\n\t\t\t\tset authorID = 1,\n\t\t\t\ttitl
15	3BDD3932-B434-8274-85994536B5FEE85D	/summitRest/books/bookUpdate	super@demodata.com:demo				
16	3BD1D3F3-9670-7BFA-AAF28407B7BFE7B8			200	OK	Action completed successfully.	{"EXECUTIONTIME":0,"IDENTITYCOL":22,"SQL":"\n\t\t\t\tinsert into app.books (authorid, tit
17	3BD1D3F3-9670-7BFA-AAF28407B7BFE7B8	/summitRest/books/bookAdd	super@demodata.com:demo				
18	3BCF8DD7-072B-AD27-3F4EB8664580A175			200	OK	Action completed successfully.	{"COLUMNS":["BOOKID","FIRSTNAME","LASTNAME","AUTHORID","TITLE","THUMBNAIL
19	3BCF8DD7-072B-AD27-3F4EB8664580A175	/summitRest/books/bookList	NA				
20	3BC4B089-0245-2F01-27F32985BD1B466A			200	OK	Action completed successfully.	{"COLUMNS":["BOOKID","FIRSTNAME","LASTNAME","AUTHORID","TITLE","BOOKIMAG
21	3BC4B089-0245-2F01-27F32985BD1B466A	/summitRest/books/5/.xml	NA				
22	3BC45419-B558-8463-6E2E1837FF9A80B0			200	OK	Action completed successfully.	{"COLUMNS":["BOOKID","FIRSTNAME","LASTNAME","AUTHORID","TITLE","BOOKIMAG
23	3BC45419-B558-8463-6E2E1837FF9A80B0	/summitRest/books/5/.json	NA				
24	3B756C80-F9B4-C484-60C58D6A34BA8080			406	Bad Request	isspotlight is not provided	
25	3B756C80-F9B4-C484-60C58D6A34BA8080	/summitRest/authors/authorUpdate	super@demodata.com:demo				
26	3B3E0164-939A-ED66-EF836D2476514C40			200	OK	Action completed successfully.	{"EXECUTIONTIME":15,"SQL":"\r\n\t\t\t\tdelete from app.authors\r\n\t\t\t\twhere authorID =
27	3B3E0164-939A-ED66-EF836D2476514C40	/summitRest/authors/16	super@demodata.com:demo				
28	3B3AD491-B8FA-3961-26152CC8FC50B3B2			200	OK	Action completed successfully.	{"EXECUTIONTIME":31,"SQL":"\r\n\t\t\t\tupdate app.authors\r\n\t\t\t\tset firstname = 'John
29	3B3AD491-B8FA-3961-26152CC8FC50B3B2	/summitRest/authors/authorUpdate	super@demodata.com:demo				
30	3AF2041F-0386-3172-31302555315A4327			200	OK	Action completed successfully.	{"EXECUTIONTIME":0,"IDENTITYCOL":21,"SQL":"\r\n\t\t\t\tinsert into app.authors (firstnam
31	3AF2041F-0386-3172-31302555315A4327	/summitRest/authors/authorAdd	super@demodata.com:demo				
32	3AE6C1DE-A8D2-61C5-7F664E3B1FF9854F				Variable QADDAUTHOR is undefined.		
33	3AE6C1DE-A8D2-61C5-7F664E3B1FF9854F	/summitRest/authors/authorAdd	super@demodata.com:demo				
34	3AE1FB01-ACB4-97B2-22A70F456DA61348				Variable QADDAUTHOR is undefined.		
35	3AE1FB01-ACB4-97B2-22A70F456DA61348	/summitRest/authors/authorAdd	super@demodata.com:demo				
36	3AC61C6F-D56D-6B65-D614D473E5B8A7EC				Variable QAUTHORDETAILS is undefined.		
37	3AC61C6F-D56D-6B65-D614D473E5B8A7EC	/summitRest/authors/authorAdd	super@demodata.com:demo				
38	39548919-F6EE-DC8A-55C83DB00B7985E6			404	Author not found	An author with ID 588 was not found	
39	39548919-F6EE-DC8A-55C83DB00B7985E6	/summitRest/authors/588	NA				
40	39483033-D376-52D9-F7299B7A5A4535D4			200	OK	Action completed successfully.	{"COLUMNS":["AUTHORID","FIRSTNAME","LASTNAME","BIO","ISSPOTLIGHT"],"DATA":[
41	39483033-D376-52D9-F7299B7A5A4535D4	/summitRest/authors/5	NA				
42	39362B69-EF90-63EB-B5DF1AE4EFCD6104			200	OK	Action completed successfully.	{"COLUMNS":["AUTHORID"

Initial Solution – Books RESTful Code Snippet

```
<cffunction name="addBook" access="remote" httpmethod="POST" returntype="any"
restpath="bookAdd">
        <cfargument  name="structData"  type="struct"  required="true" >

        <cfinclude template="inc_tracking.cfm">
        <cfinclude template="inc_auth.cfm">   ←Security Call

        <cfquery name="qAddBook" datasource="#request.readDSN#" result="insertResults">
                insert into app.books (authorid, title, bookimage,thumbnailimage,
bookdescription, isspotlight, genre)
                values (#structData.authorid#, '#trim(structData.title)#',
'#trim(structData.bookimage)#', '#trim(structData.thumbnailimage)#',
'#trim(structData.bookdescription)#',                '#structData.isspotlight#',
'#trim(structData.genre)#')
        </cfquery>

        <cfset temp = restThrow(throw="false", errorCode="200", message="OK", detail="Action
completed successfully.", data="#serializeJSON(insertResults)#",
sourcename="#getfunctioncalledname()#")>

        <cfreturn insertResults>
</cffunction>
```

Initial Solution – Calling API With Security

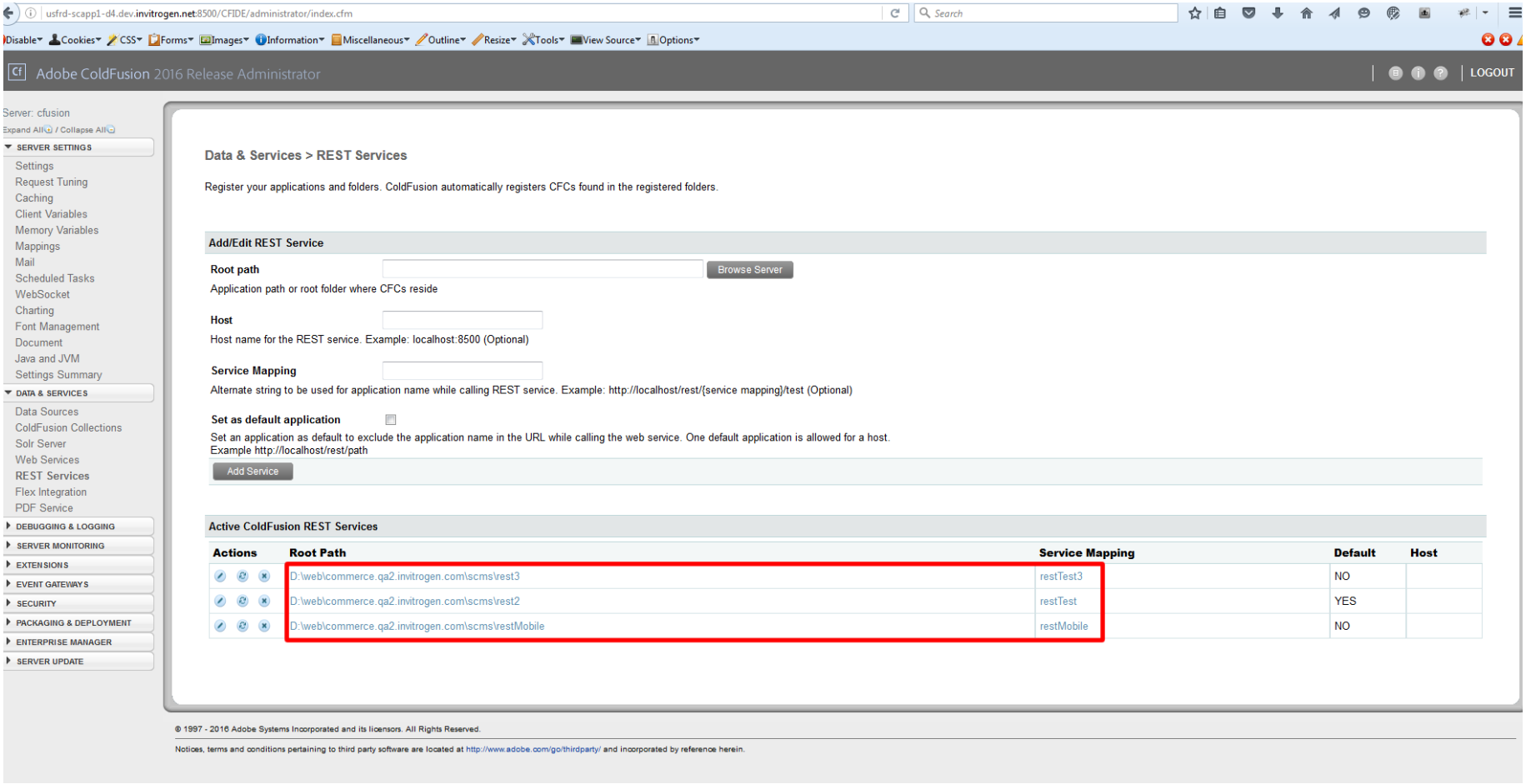
```
<cfset variables.startStr = "super@demodata.com:demo">
<cfset variables.base64Str = toBase64(variables.startStr)>

<cfset variables.jsonStr = {firstname="John", lastname="Stone", bio="John is from Carlsbad CA",
isspotlight ="Y"}>
<cfset variables.dataString = SerializeJSON(variables.jsonStr)>
<cfhttp url="http://localhost/rest/summitRest/authors/authorAdd" result="restResult" method="post">
  <cfhttpparam TYPE="Header"
               name="content-type"
               value="application/JSON">
                 <cfhttpparam type="Header"
               name="Authorization"
               value="#variables.base64Str#">
  <cfhttpparam TYPE="Body"
               name="structData"
               value="#variables.dataString#">

</cfhttp>
<cfdump var="#restResult#">
```

Initial Solution – inc_auth.cfm

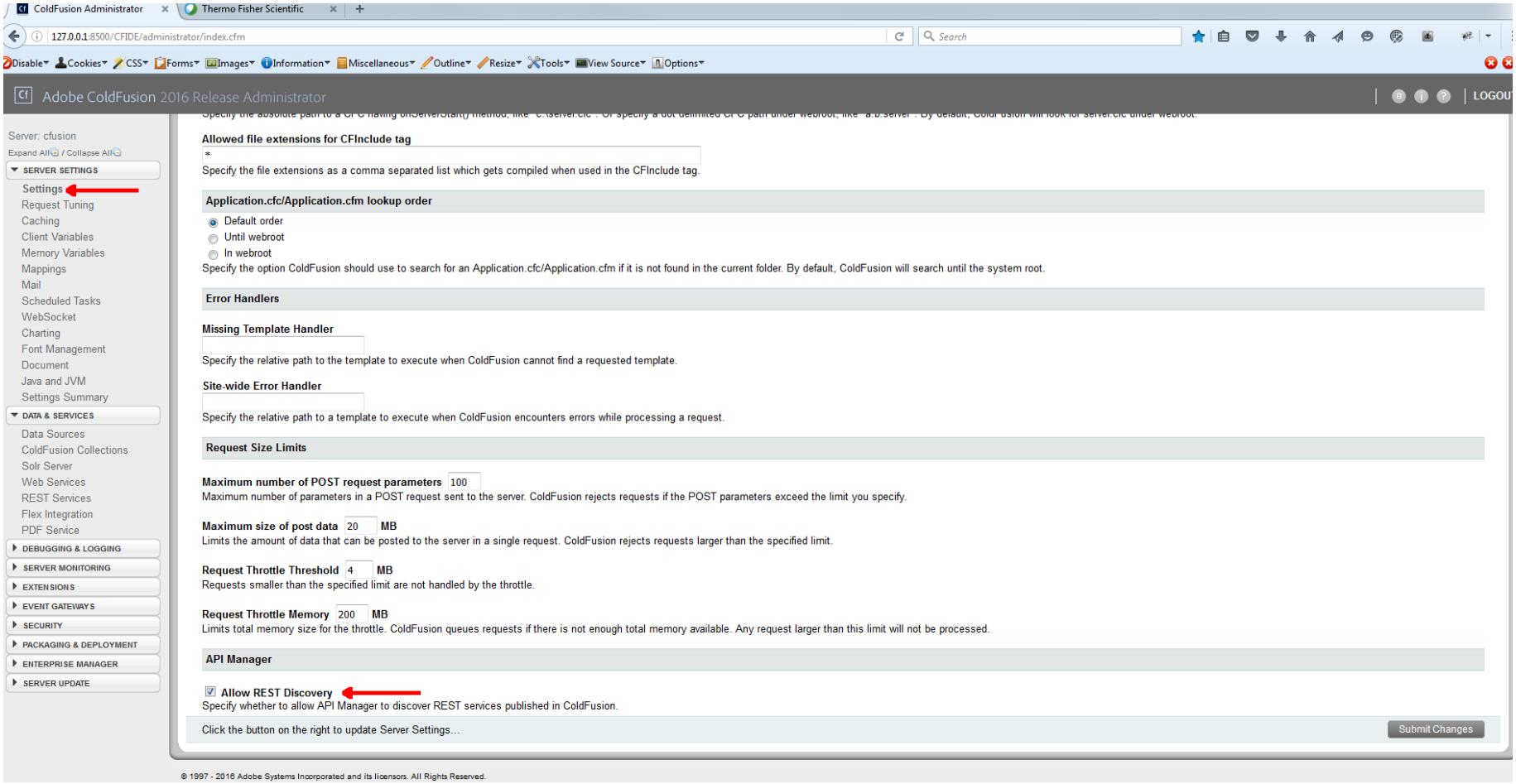
```
<cfset LOCAL.EncodedCredentials = GetHTTPRequestData().Headers.Authorization>
<cfset LOCAL.Credentials = ToString(ToBinary( LOCAL.EncodedCredentials ))>
<cfset LOCAL.memberEmail = ListFirst( LOCAL.Credentials, ":" )>
<cfset LOCAL.memberPassword = ListLast( LOCAL.Credentials, ":" )>

<cfquery name="getMember" datasource="#request.readDSN#">
	select *
	from app.members
	where lower(email)='#lcase(LOCAL.memberEmail)#'
	and memberpassword='#LOCAL.memberPassword#'
	and isactive = 'Y'
</cfquery>
<cfif getMember.recordcount EQ "0">
		<cfthrow errorcode="401" message="Unauthorized!" type="401" detail="Username or
Password incorrect">
<cfelse>
	<cfset arguments.memberid = getMember.memberid>
	<cfset arguments.memberfname = getMember.firstname>
	<cfset arguments.memberlname = getMember.lastname>
	<cfset arguments.memberemail = getMember.email>
	<cfset arguments.memberisadmin = getMember.isadmin>
</cfif>
```

Then Came CF 2016
And
The API Manager

Create API – Registered REST services

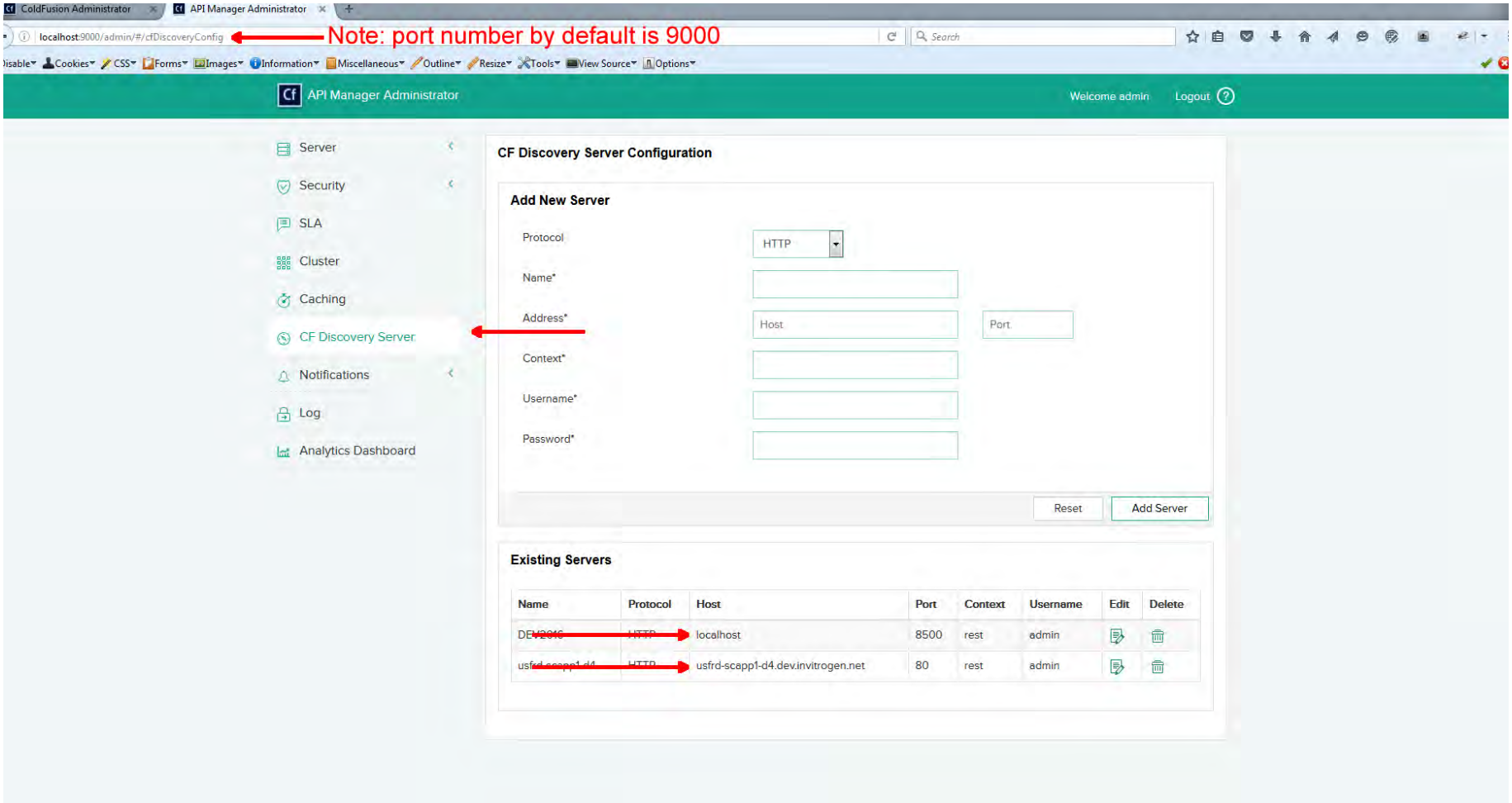


CF2016 Administrator – Allow REST Discovery

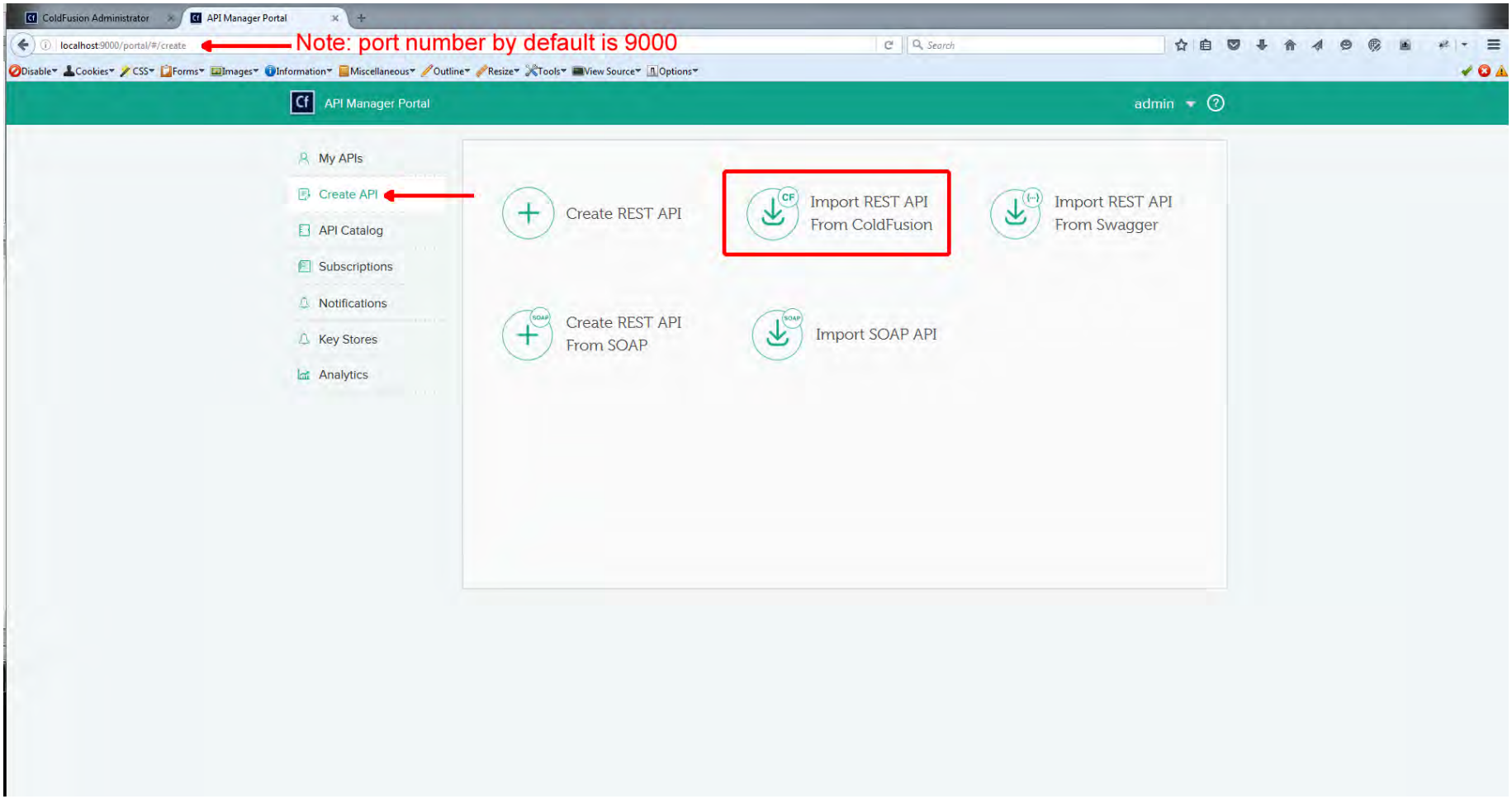


ThermoFisher SCIENTIFIC

API Manager Administrator – Server Discovery

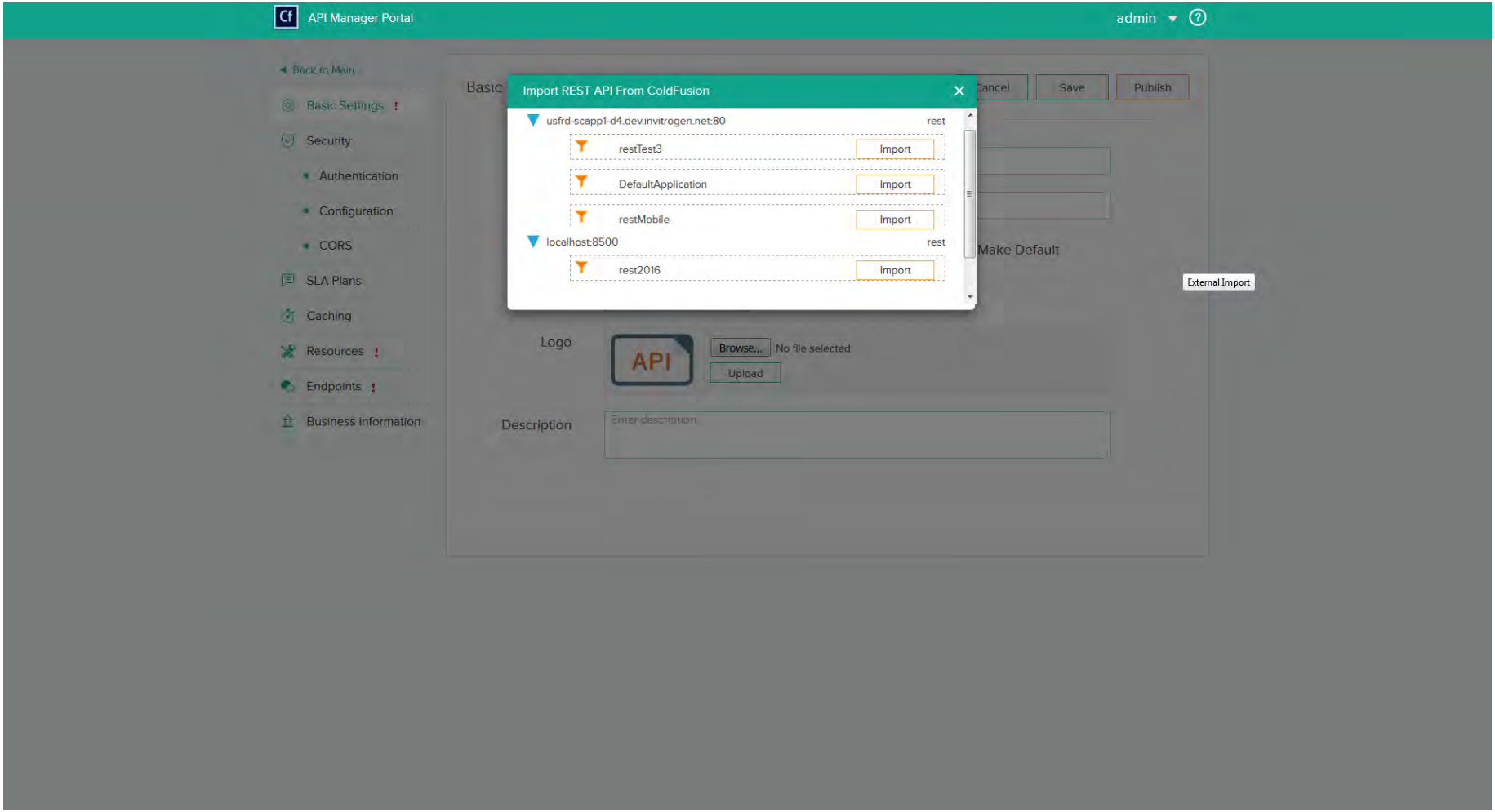


Create API - Import REST API From ColdFusion

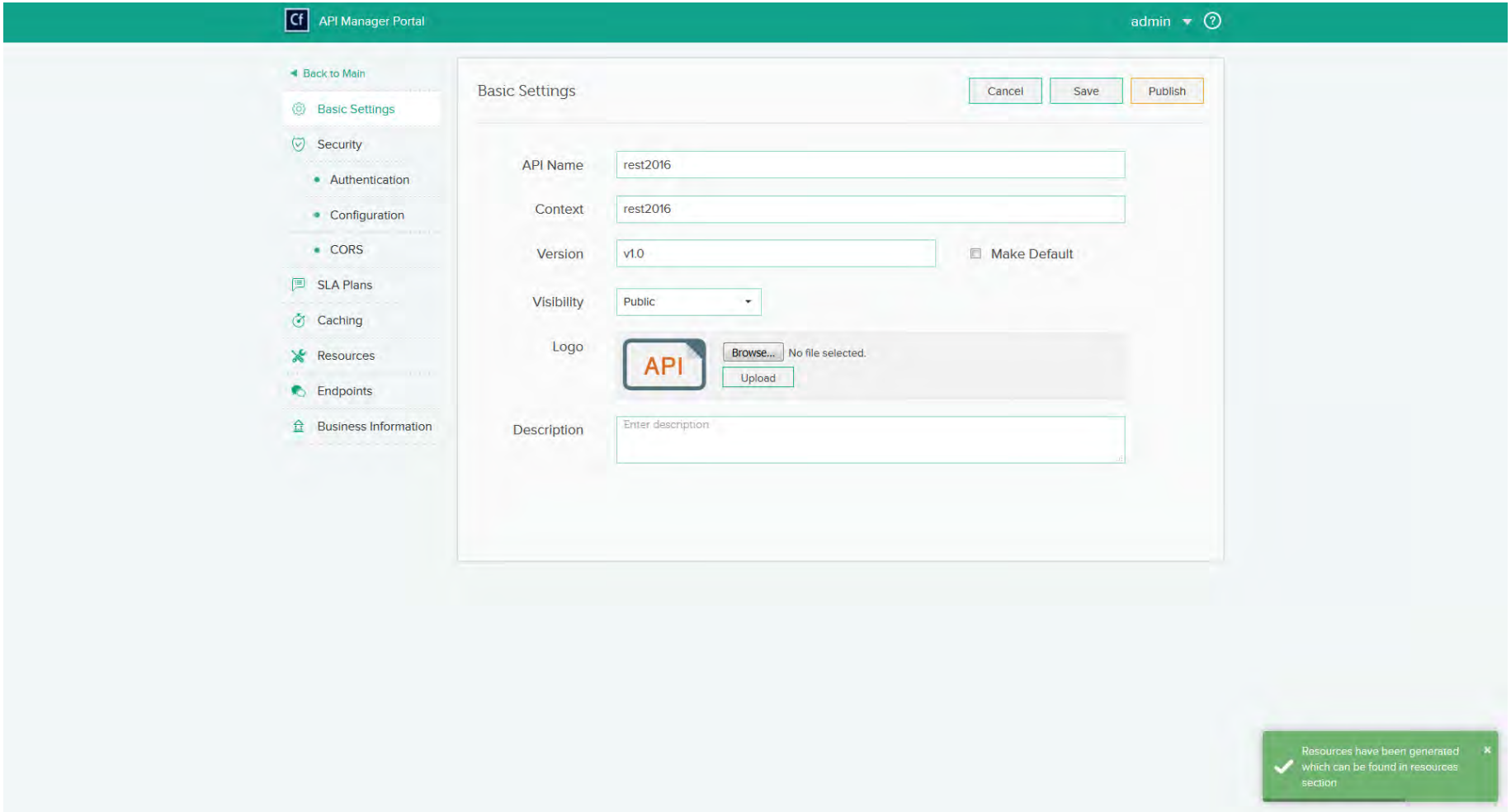


ThermoFisher SCIENTIFIC

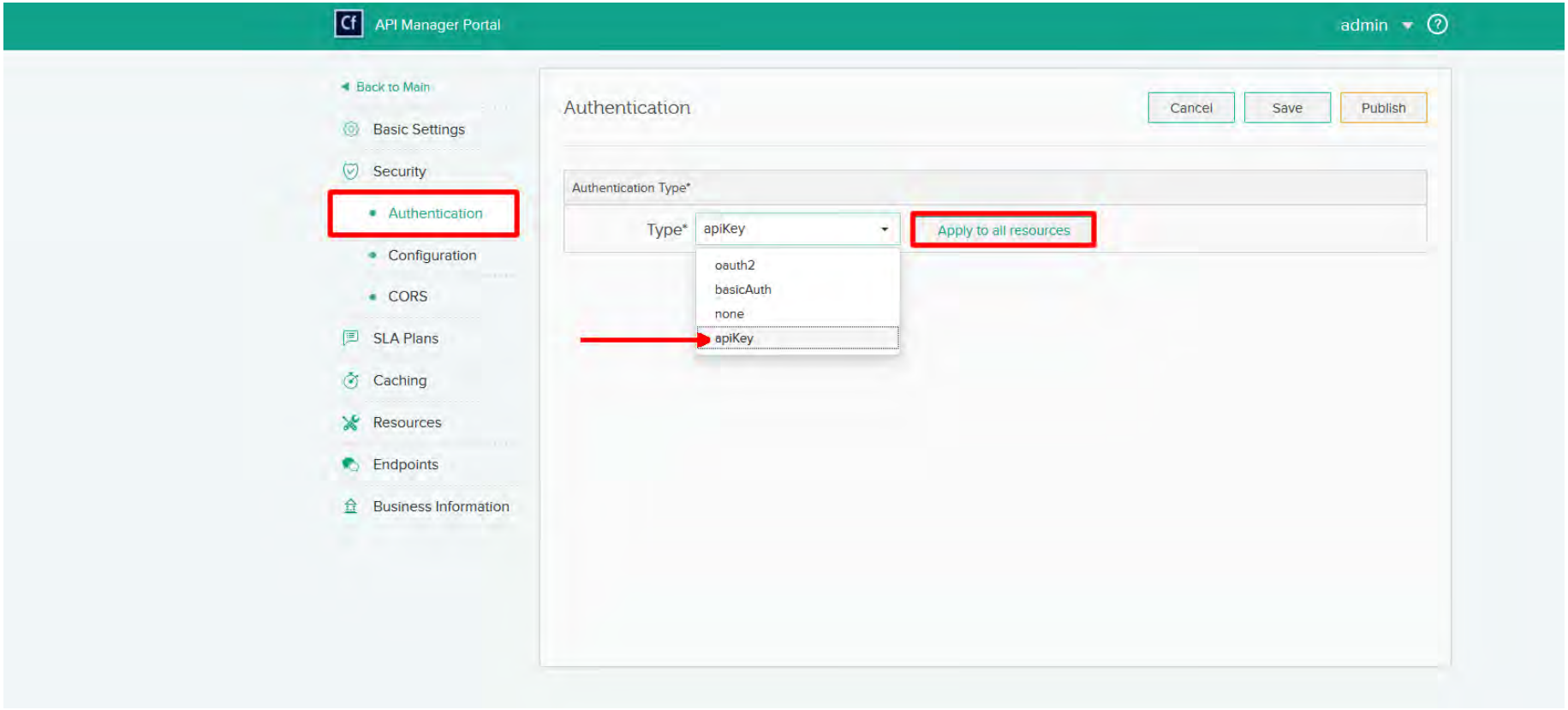
Create API - Select RESTful Service to Import



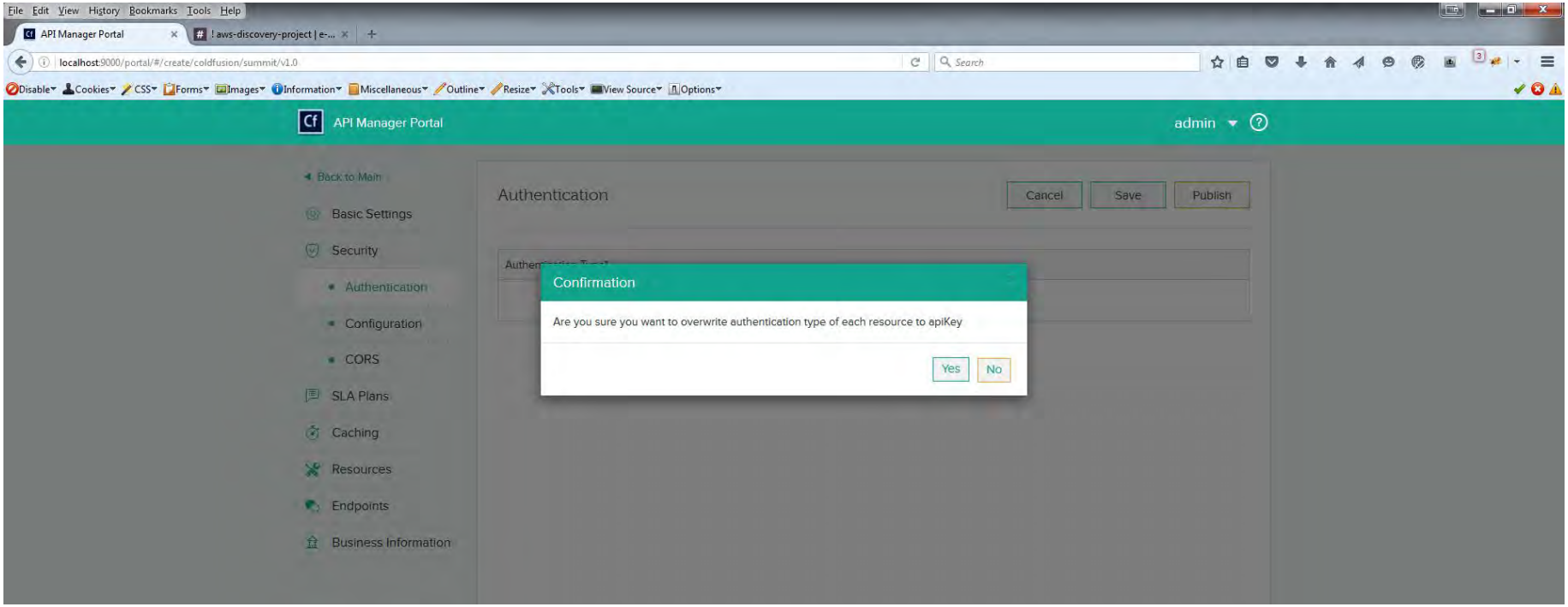
Create API - REST Import Complete



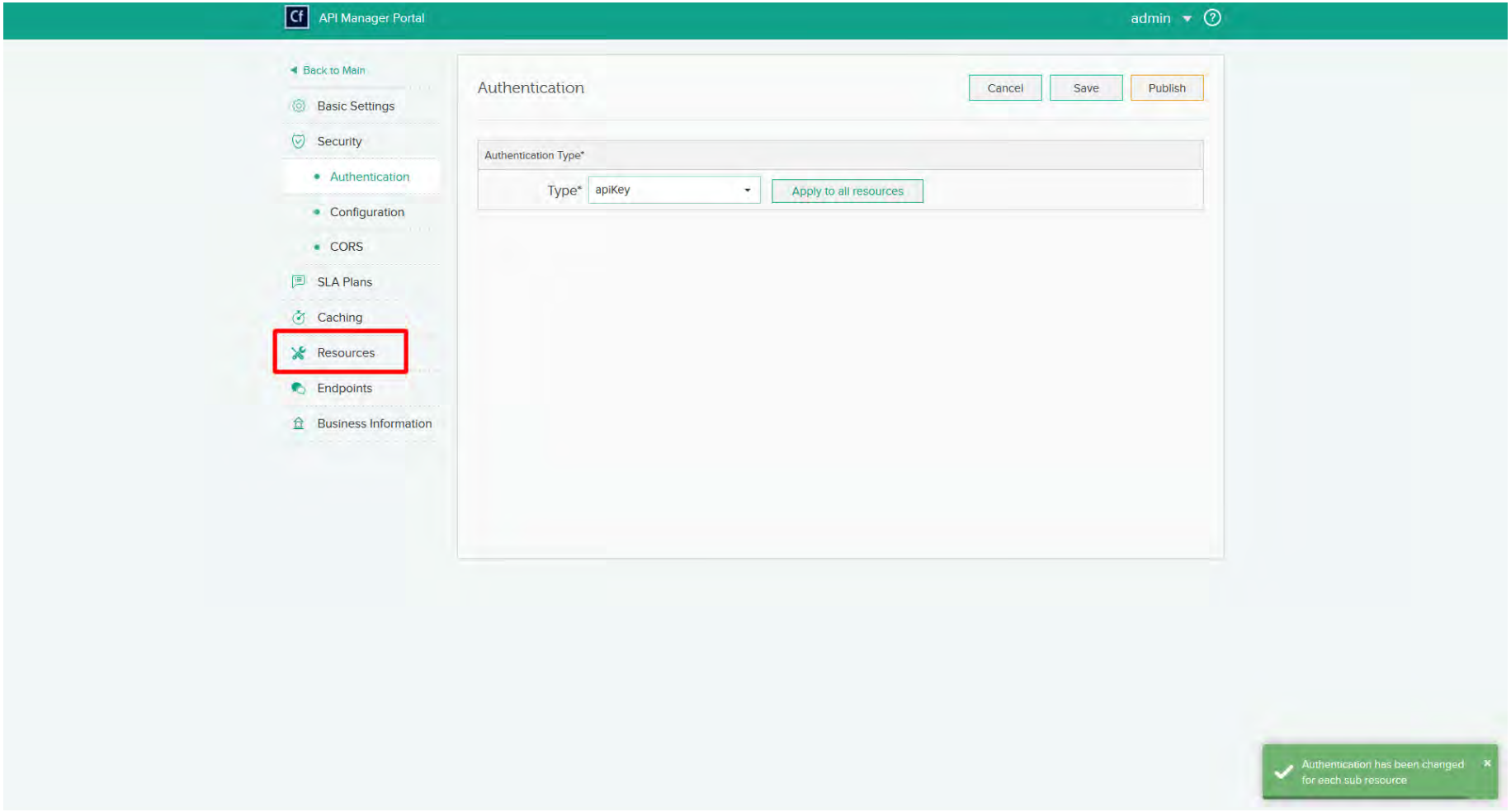
Create API - Apply Authentication



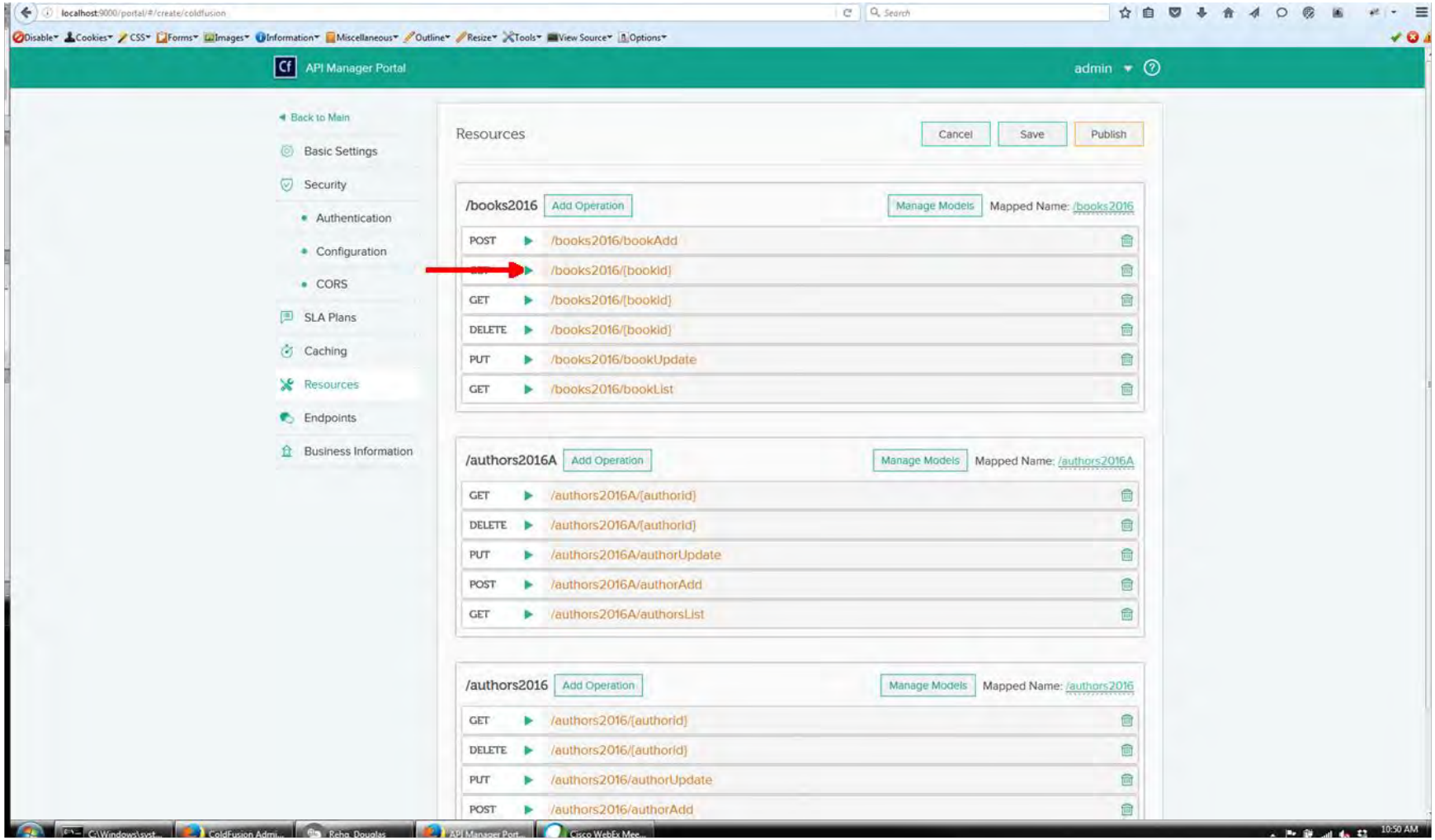
Create API - Confirm Authentication



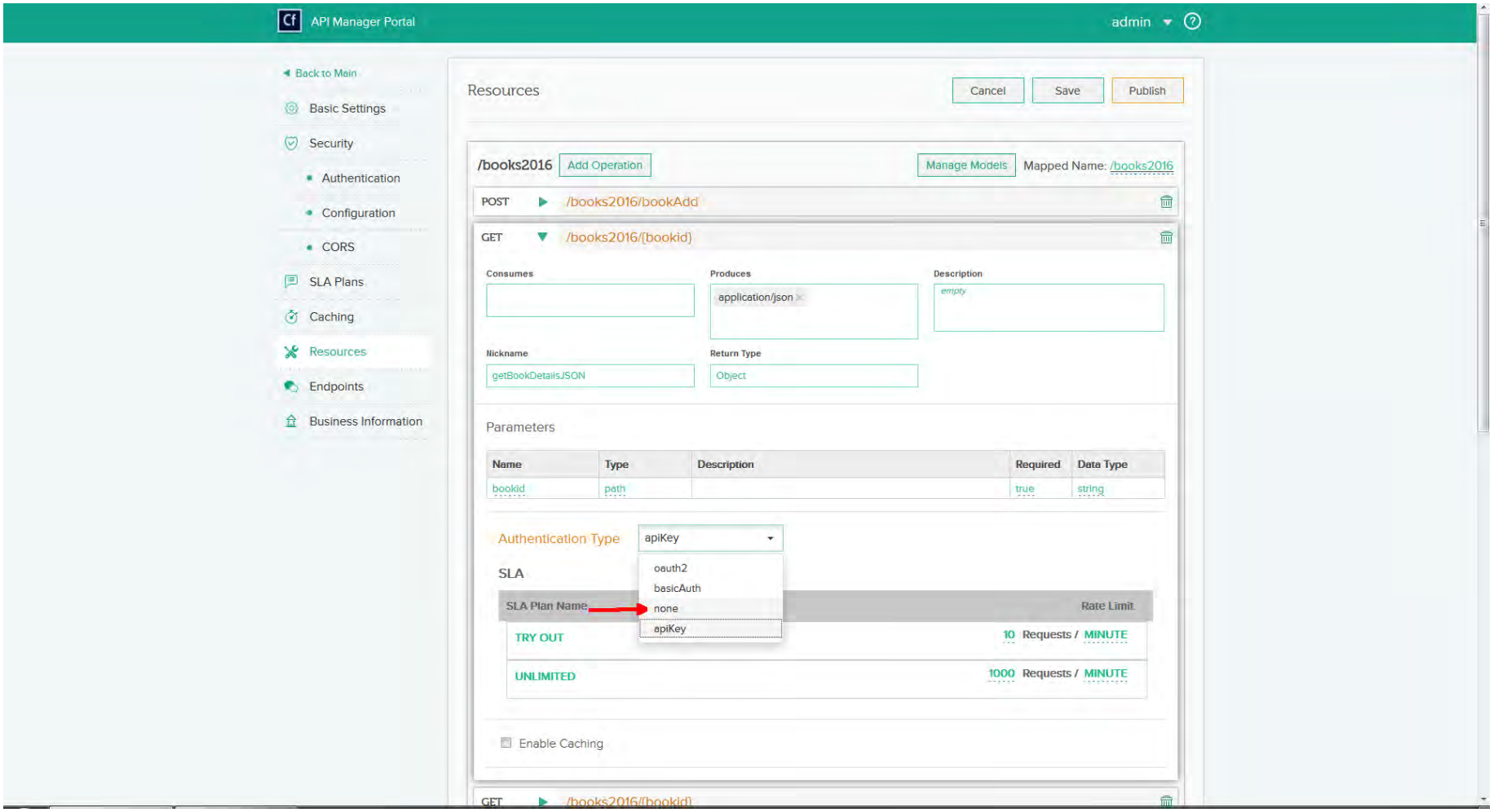
Create API – Select Resources



Create API – Select Get

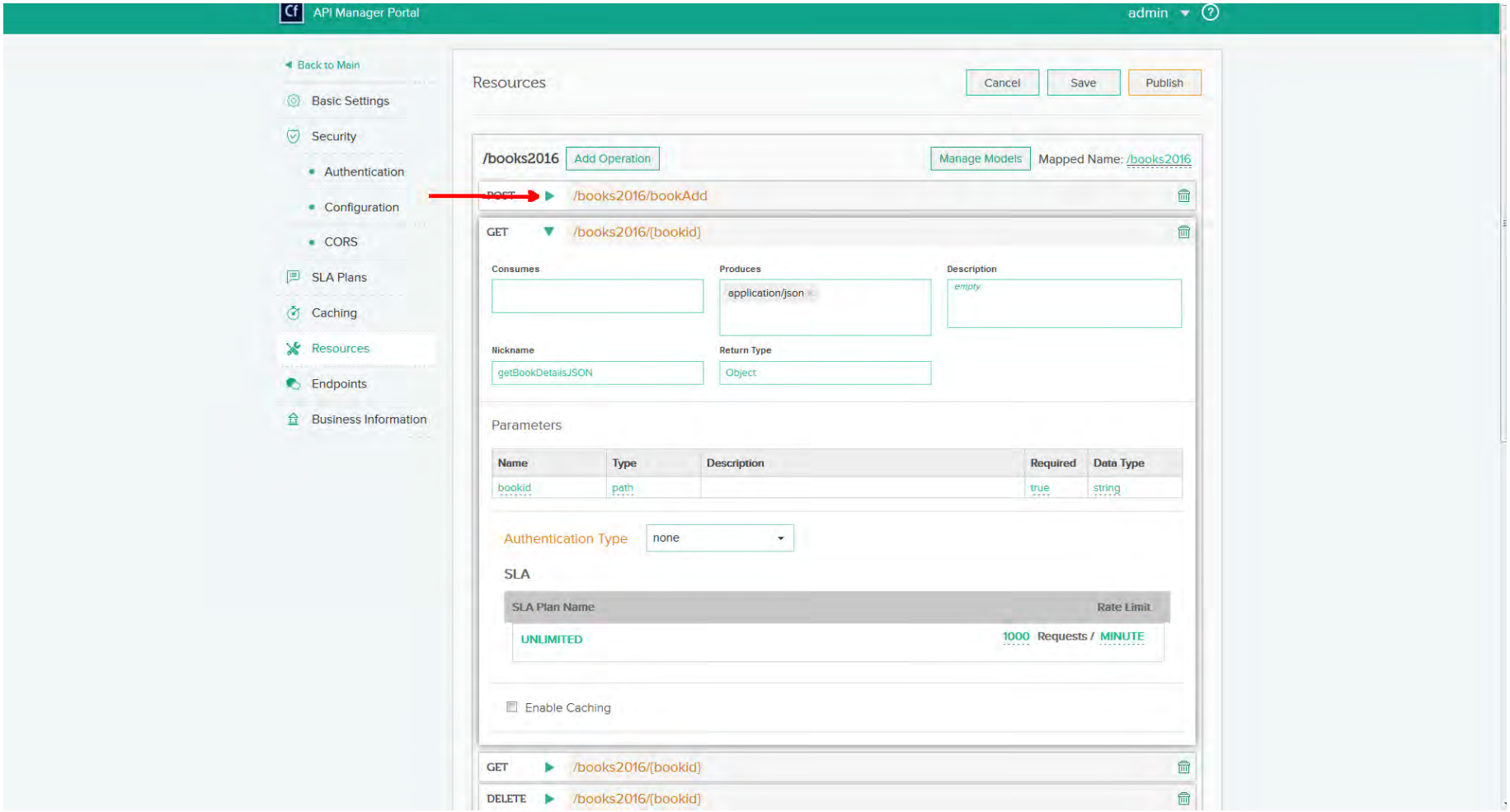


Set Authentication To “None”

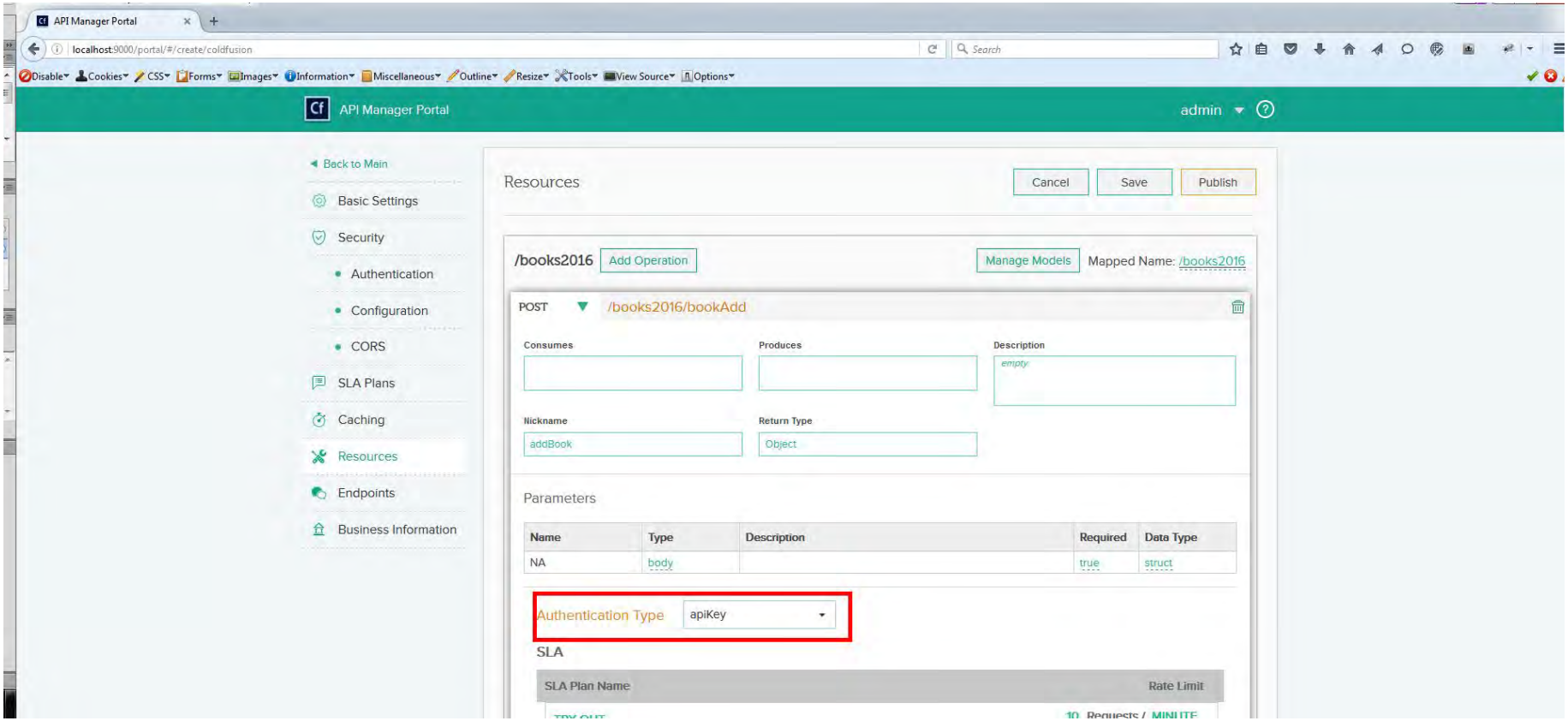


ThermoFisher SCIENTIFIC

Create API – Select Post

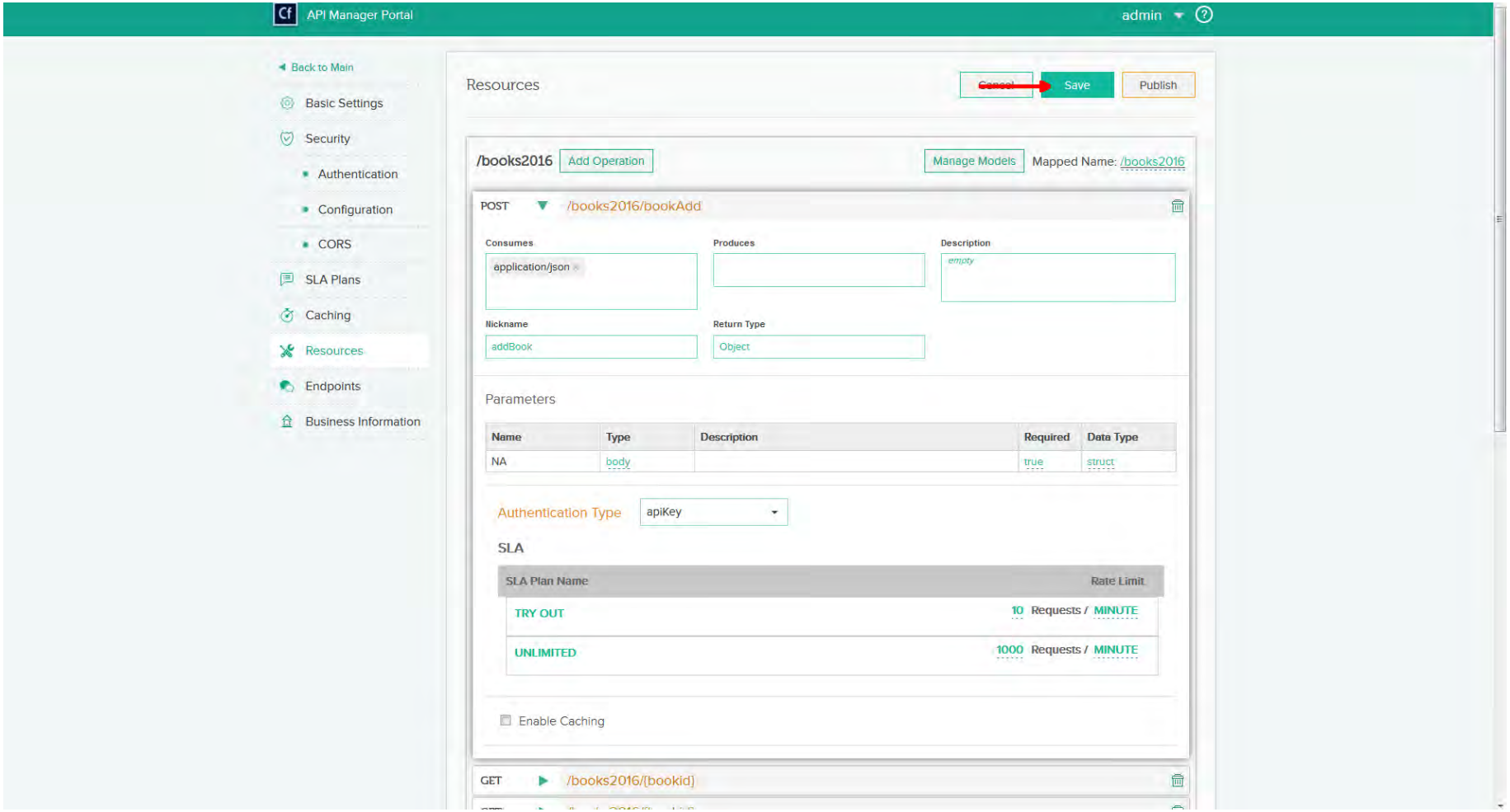


Create API – Confirm Authentication

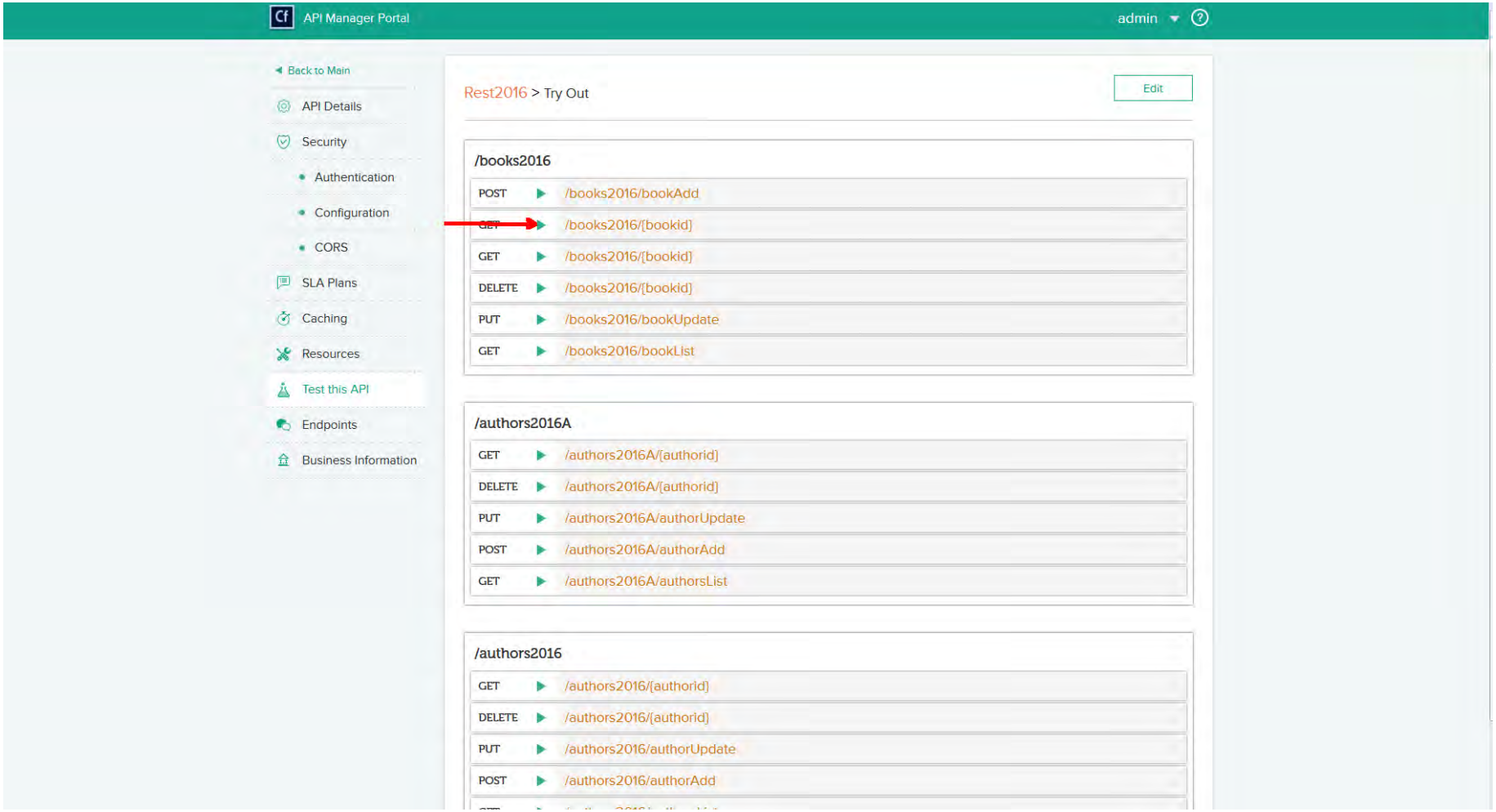


ThermoFisher SCIENTIFIC

Create API – Save Changes

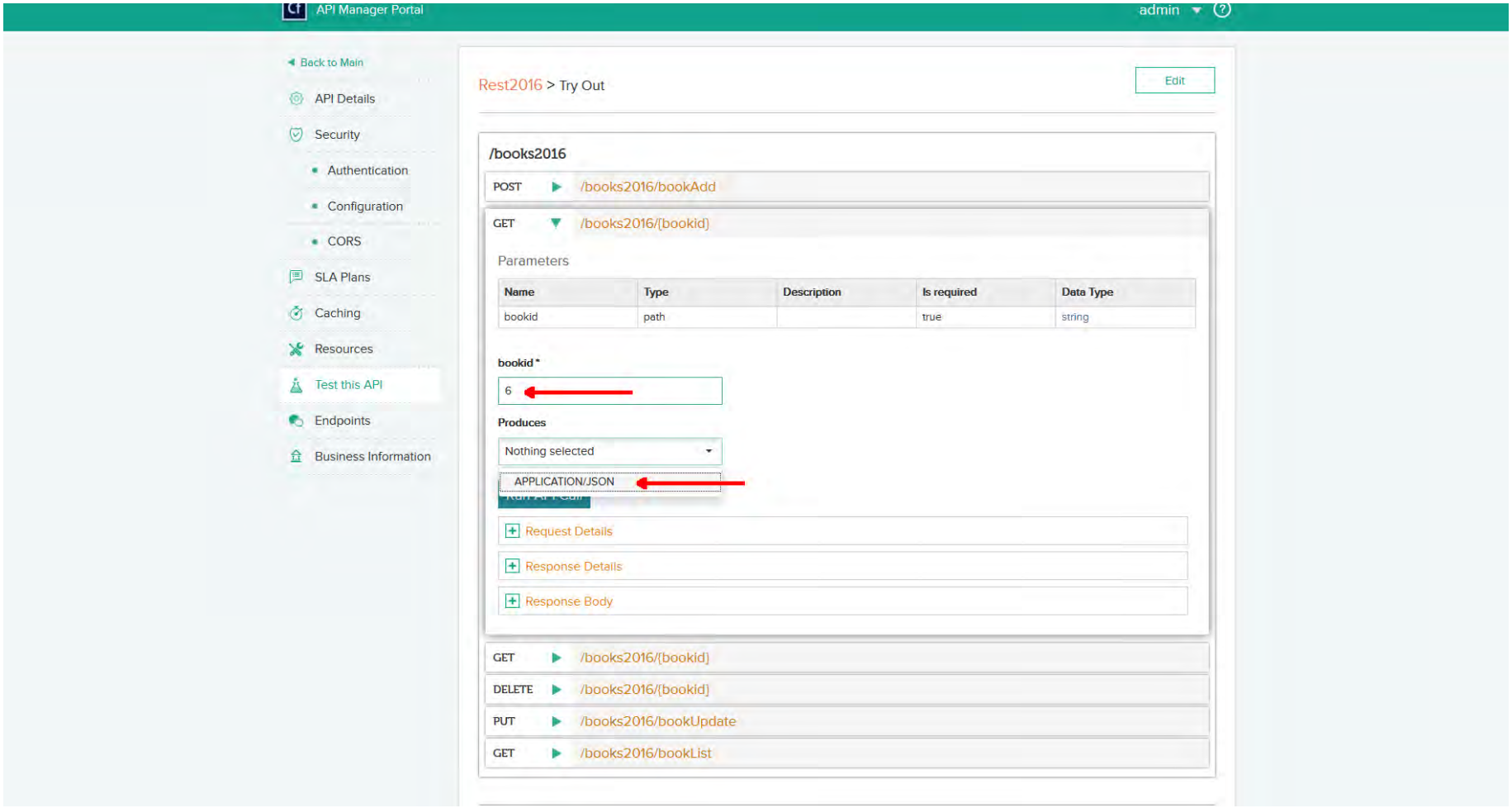


Test API - Get

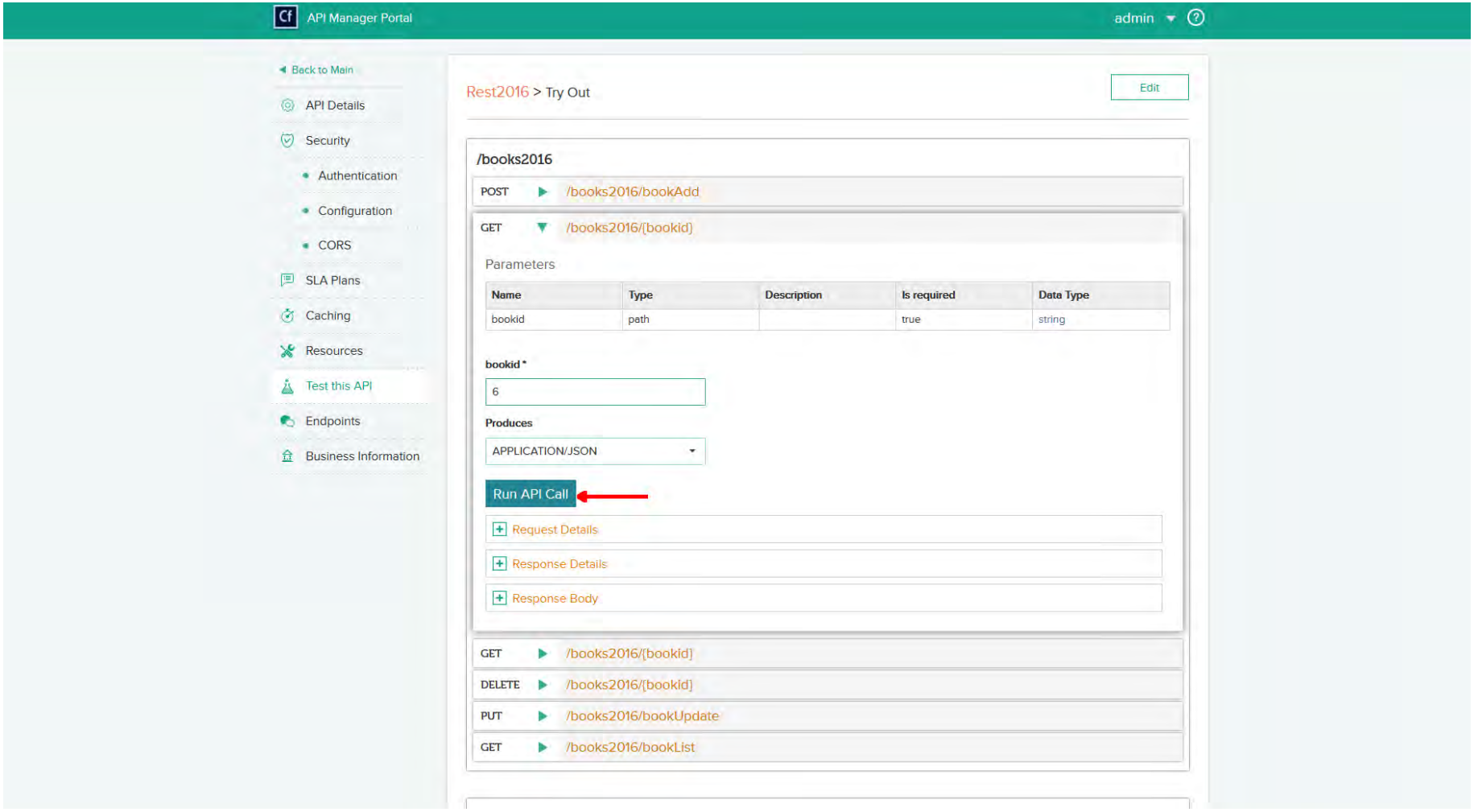


ThermoFisher SCIENTIFIC

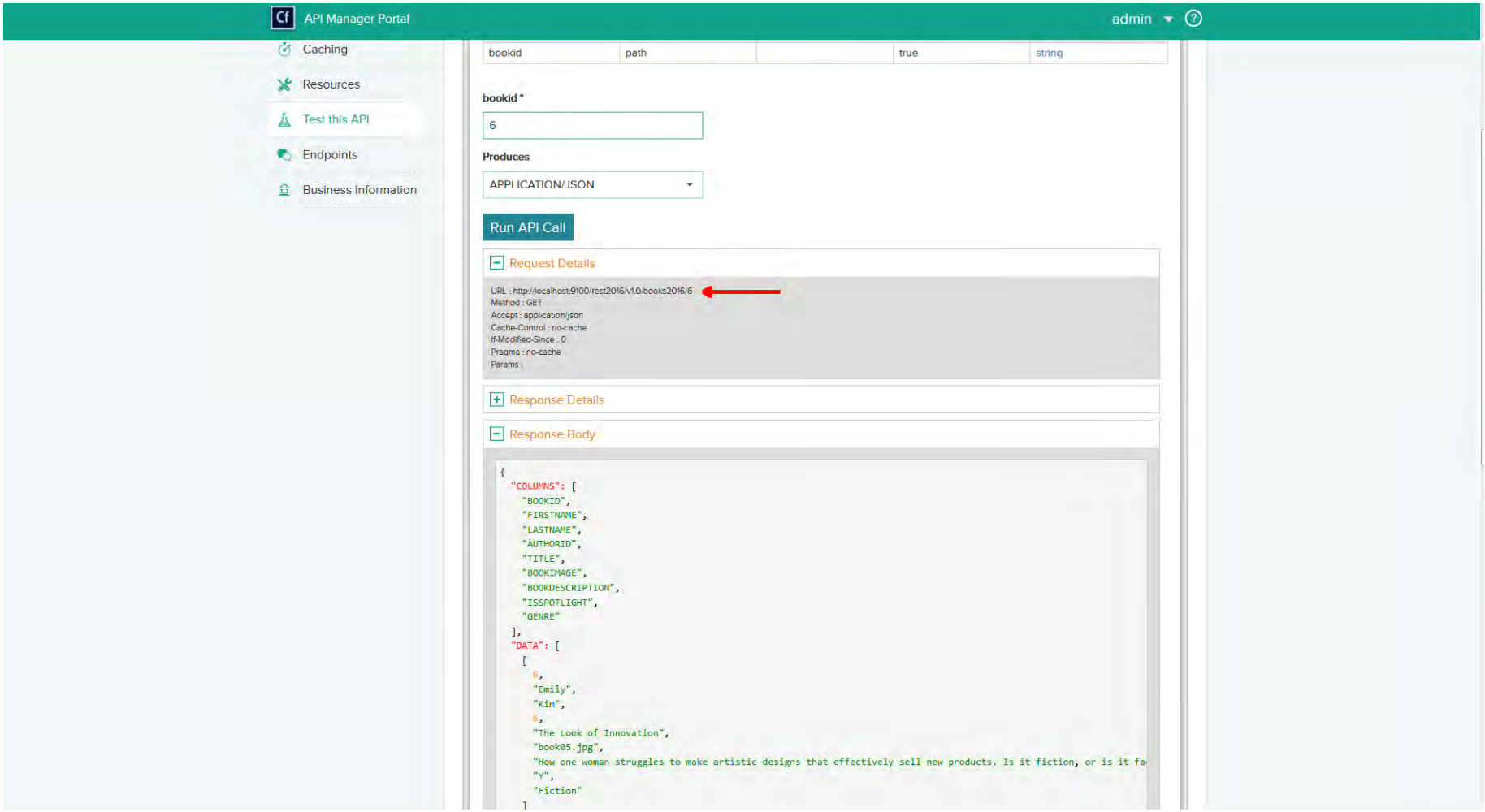
Test API - Get



Test API - Get



Test API - Get



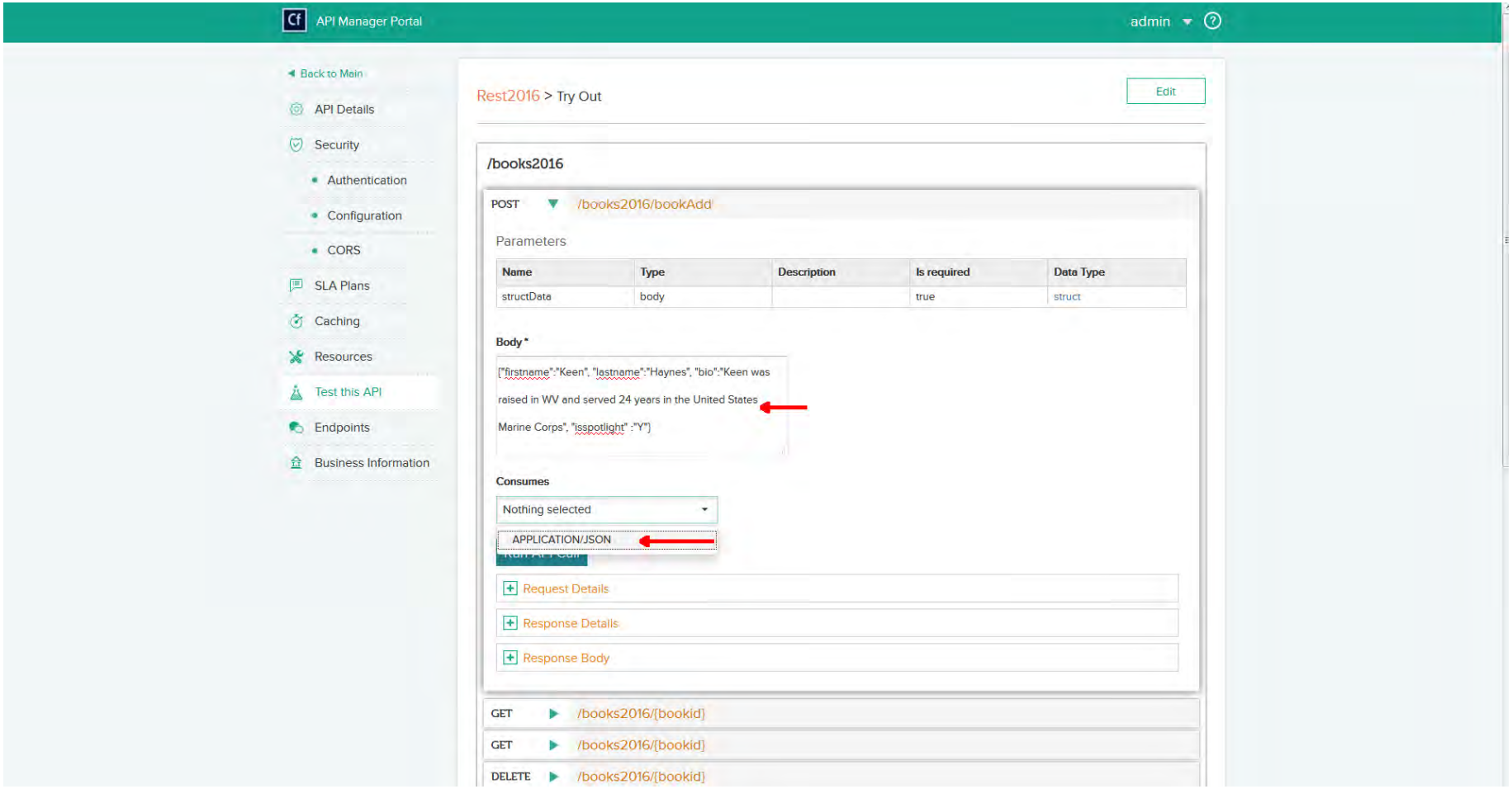
Compare Code - Get

```
<cffunction name="getBookDetails" access="remote" httpmethod="GET" returntype="any"
   restpath="{bookid}"  produces="application/json, application/xml">
   <cfargument name="bookid" required="true" restargsource="Path" type="string">

   <cfinclude template="inc_tracking.cfm">
   <cfquery name="qBookDetails" datasource="#request.readDSN#">
      select b.bookid, a.firstname, a.lastname, a.authorid, b.title, b.bookimage, b.bookdescription,
      b.isspotlight, b.genre
      from app.books b, app.authors a
      where a.authorid = b.authorid
      and b.bookid = #arguments.bookid#
   </cfquery>
   <cfset temp = restThrow(throw="false", errorCode="200", message="OK", detail="Action completed
   successfully.", data="#serializeJSON(insertResults)#", sourcename="#getfunctioncalledname()#")>

   <cfreturn qBookDetails>
</cffunction>
```

Compare Code – Call Get

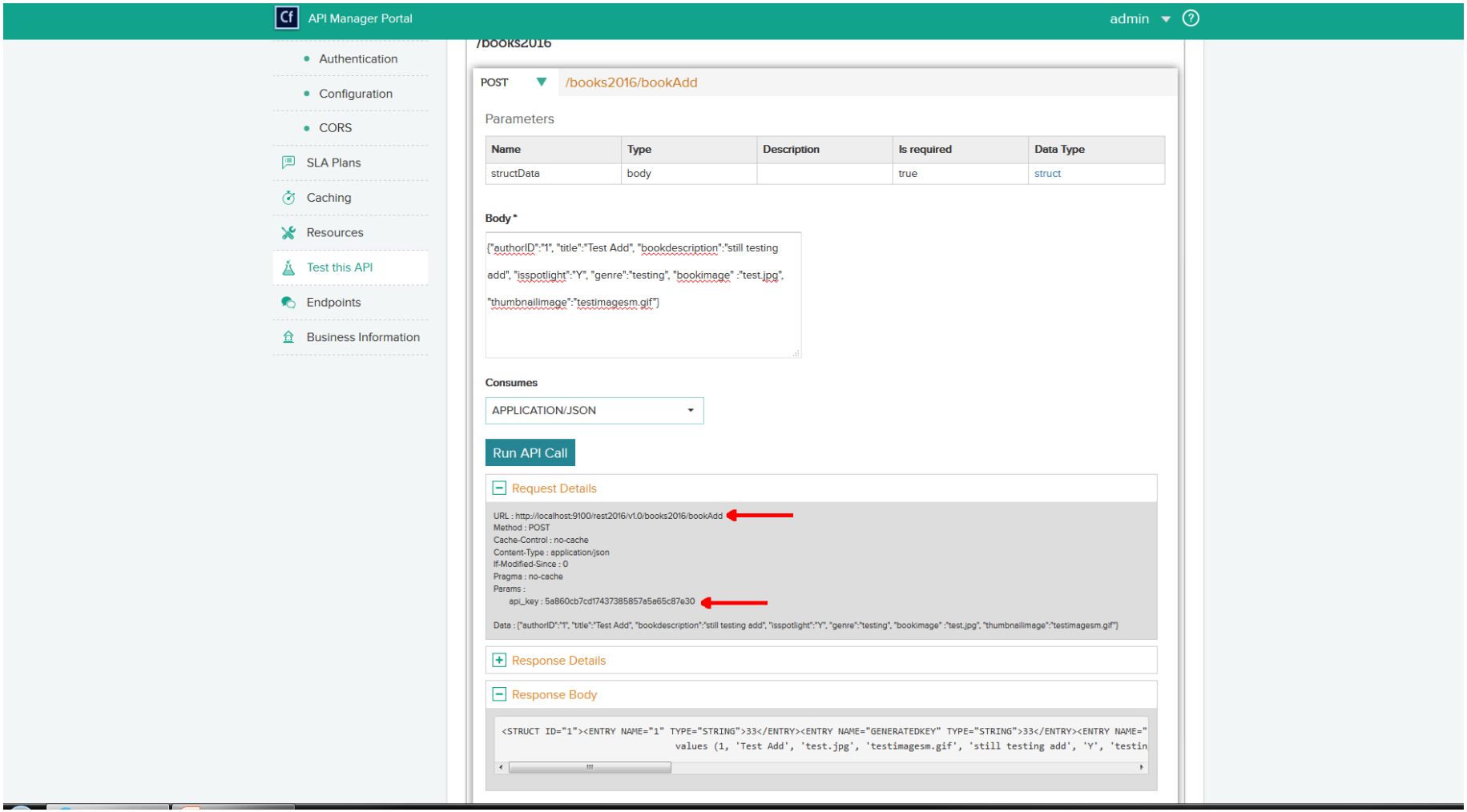
```
<cfhttp url="http://localhost:9100/rest2016/v1.0/books2016/24" result="restResult" method="get“>
<cfhttpparam type="header" name="Accept" value="application/json" />
</cfhttp>
<cfdump var="#restResult#">

<cfhttp url="http://localhost:8500/rest/rest2016/books2016/24/.xml" result="restResult2" method="get">
</cfhttp>
<cfdump var="#restResult2#">
```

Test API - Post



ThermoFisher SCIENTIFIC

Test API - Post



Compare Code

```
<cffunction name="addBook" access="remote" httpmethod="POST" returntype="any"  restpath="bookAdd">
   <cfargument  name="structData"  type="struct"  required="true" >

   <cfinclude template="inc_tracking.cfm">
   <cfinclude template="inc_auth.cfm">

   <cfquery name="qAddBook" datasource="#request.readDSN#" result="insertResults">
     insert into app.books (authorid, title, bookimage,thumbnailimage, bookdescription, isspotlight, genre)
     values (#structData.authorid#, '#trim(structData.title)#', '#trim(structData.bookimage)#',
     '#trim(structData.thumbnailimage)#', '#trim(structData.bookdescription)#',      '#structData.isspotlight#',
     '#trim(structData.genre)#')
   </cfquery>

   <cfset temp = restThrow(throw="false", errorCode="200", message="OK", detail="Action completed
   successfully.", data="#serializeJSON(insertResults)#", sourcename="#getfunctioncalledname()#")>

   <cfreturn insertResults>
</cffunction>
```

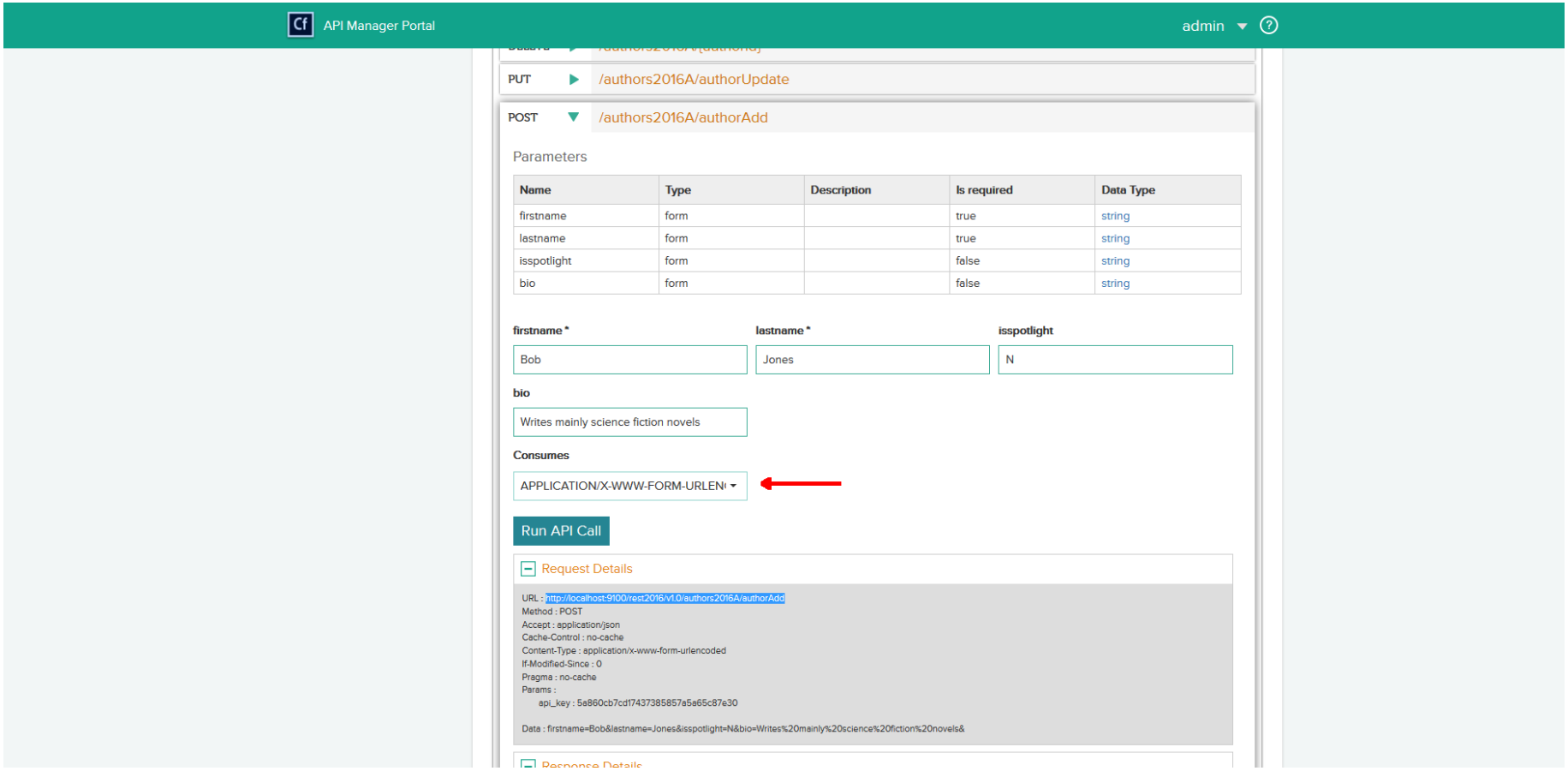
Compare Code

```
<cfset variables.jsonStr = {authorID=1, title="Test Add", bookdescription="sstill testing add", isspotlight="Y",
   genre="testing", bookimage ="test.jpg", thumbnailimage="testimagesm.gif"}>
<cfset variables.dataString = SerializeJSON(variables.jsonStr)>

<cfhttp url="http://localhost:9100/rest2016/v1.0/books2016/bookAdd" result="restResult" method="post">
  <cfhttpparam TYPE="Header"
               name="content-type"
               value="application/JSON">
  <cfhttpparam TYPE="header"
               name="api_key" value="5a860cb7cd17437385857a5a65c87e30">
  <cfhttpparam TYPE="Body"
               name="structData"
               value="#variables.dataString#">
  </cfhttp>

<cfdump var="#restResult#">
```

Compare Code

```
<cffunction name="addAuthor" access="remote" httpmethod="POST" returntype="any"
  restpath="authorAdd">
    <cfargument name="firstname" required="true" restargsource="form" type="string">
    <cfargument name="lastname" required="true" restargsource="form" type="string">
    <cfargument name="isspotlight" required="false" default="N" restargsource="form" type="string">
    <cfargument name="bio" required="false" default="" restargsource="form" type="string">

    <cfquery name="qAddAuthor" datasource="#request.readDSN#" result="insertResults">
        insert into app.authors (firstname, lastname,bio,isspotlight)
        values ('#trim(arguments.firstname)#', '#trim(arguments.lastname)#',
'#trim(arguments.bio)#', '#arguments.isspotlight#')
      </cfquery>

       <cfreturn insertResults>
    </cffunction>
```

Test API



Compare Code

```
<cfhttp url="http://localhost:9100/rest2016/v1.0/authors2016A/authorAdd" result="restResult"
   method="post">

  <cfhttpparam TYPE="Header" name="content-type" value="application/x-www-form-urlencoded">
  <cfhttpparam TYPE="header" name="api_key" value="5a860cb7cd17437385857a5a65c87e30">
  <cfhttpparam TYPE="FormField" name="firstname"  value="Keen">
  <cfhttpparam TYPE="FormField" name="lastname"  value="Haynes">
  <cfhttpparam TYPE="FormField" name="isspotlight" value="Y">
  <cfhttpparam TYPE="FormField" name="bio"  value="A retired United States Marine, Keen grew up in
   West Virginia.  When not coding he can be found riding his Harley">


</cfhttp>


<cfdump var="#restResult#">
```

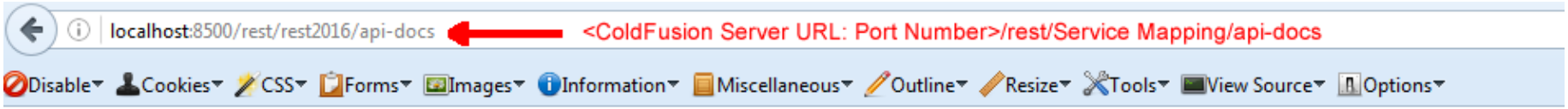
Generating Swagger Docs – Application.cfc

RDS Query Viewer | *Application.cfc | authors2016.cfc | books2016.cfc | authors2016A.cfc

```
1  <cfcomponent output="false">
2      <cfparam name="request.readdsn" default="cfbookclub">
3      <cfset this.name = "bookclubDemo2016">
4      <cfset this.applicationTimeout = createTimeSpan(2,0,0,0)>
5      <cfset this.clientManagement = "true">
6      <cfset this.sessionManagement = "true">
7      <cfset this.loginStorage = "session">
8      <cfset this.sessionTimeout = createTimeSpan(0,0,20,0)>
9      <cfset this.setClientCookies = "true">
10     <cfset this.setDomainCookies = "true">
11     <cfset this.scriptProtect = "false">
12     <cfset this.restsettings.generateRestDoc="true">
13     <cfset this.restsettings.restDocInfo.title="CFSummit">
14     <cfset this.restsettings.restDocInfo.apiVersion="1.0">
15     <cfset this.restsettings.restDocInfo.description="These are the REST services used in CF Summit demo">
16     <!--- <cfset this.restsettings.restDocInfo.termOfServiceUrl="http://thermofisher.com">--->
17     <cfset this.restsettings.restDocInfo.contact="keen.haynes@thermofisher.com">
18     <!---<cfset this.restsettings.restDocInfo.license="adobe 1.0">--->
19     <!---<cfset this.restsettings.restDocInfo.licenseUrl="http://thermofisher.com">--->
20
21     <cffunction name="onApplicationStart" returnType="void" output="false">
```

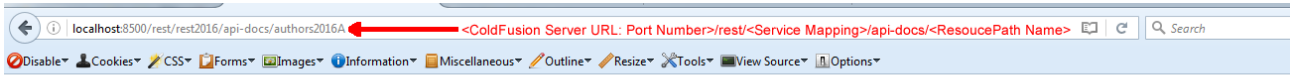
[API Manager Help / Generating Swagger documents]

Generating Swagger Docs – Resource Listing



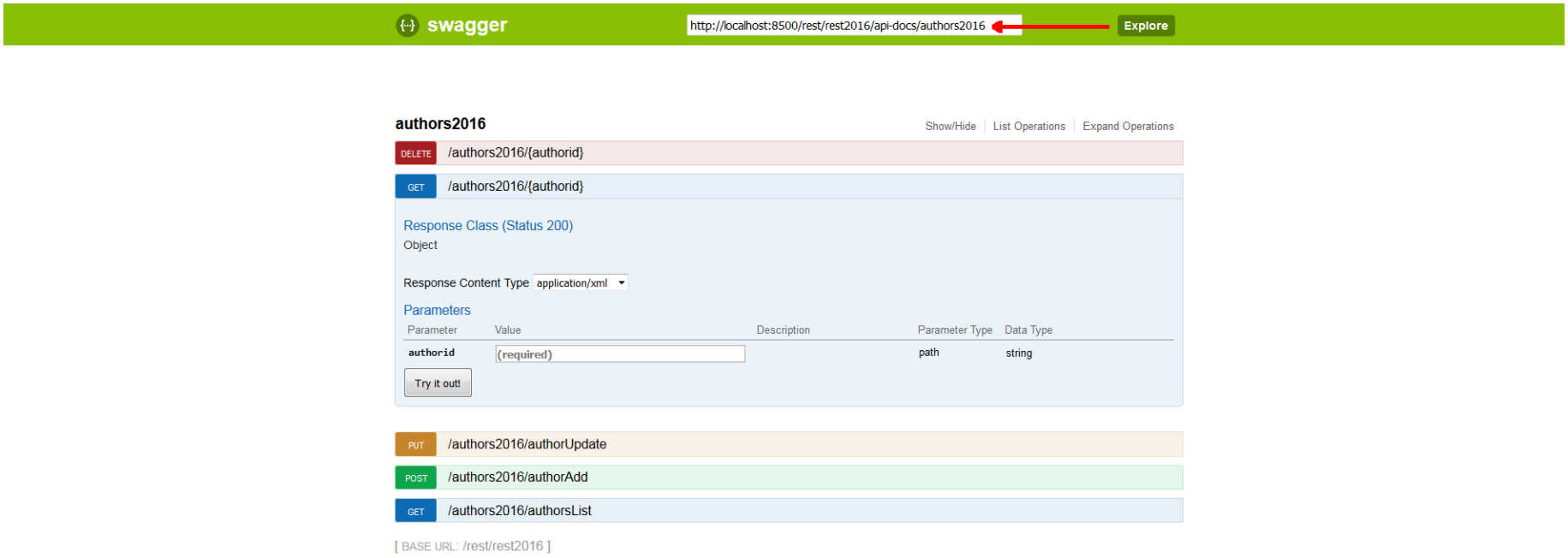

```
{
    "apiVersion":"1.0",
    "swaggerVersion":"1.2",
    "apis":[
        {
            "path":"/authors2016"
        },
        {
            "path":"/authors2016A"
        },
        {
            "path":"/books2016"
        }
    ],
    "info":{
        "title":"CFSummit",
        "description":"These are the REST services used in CF Summit demo",
        "contact":"keen.haynes@thermofisher.com"
    }
}
```

Generating Swagger Docs – API Declaration

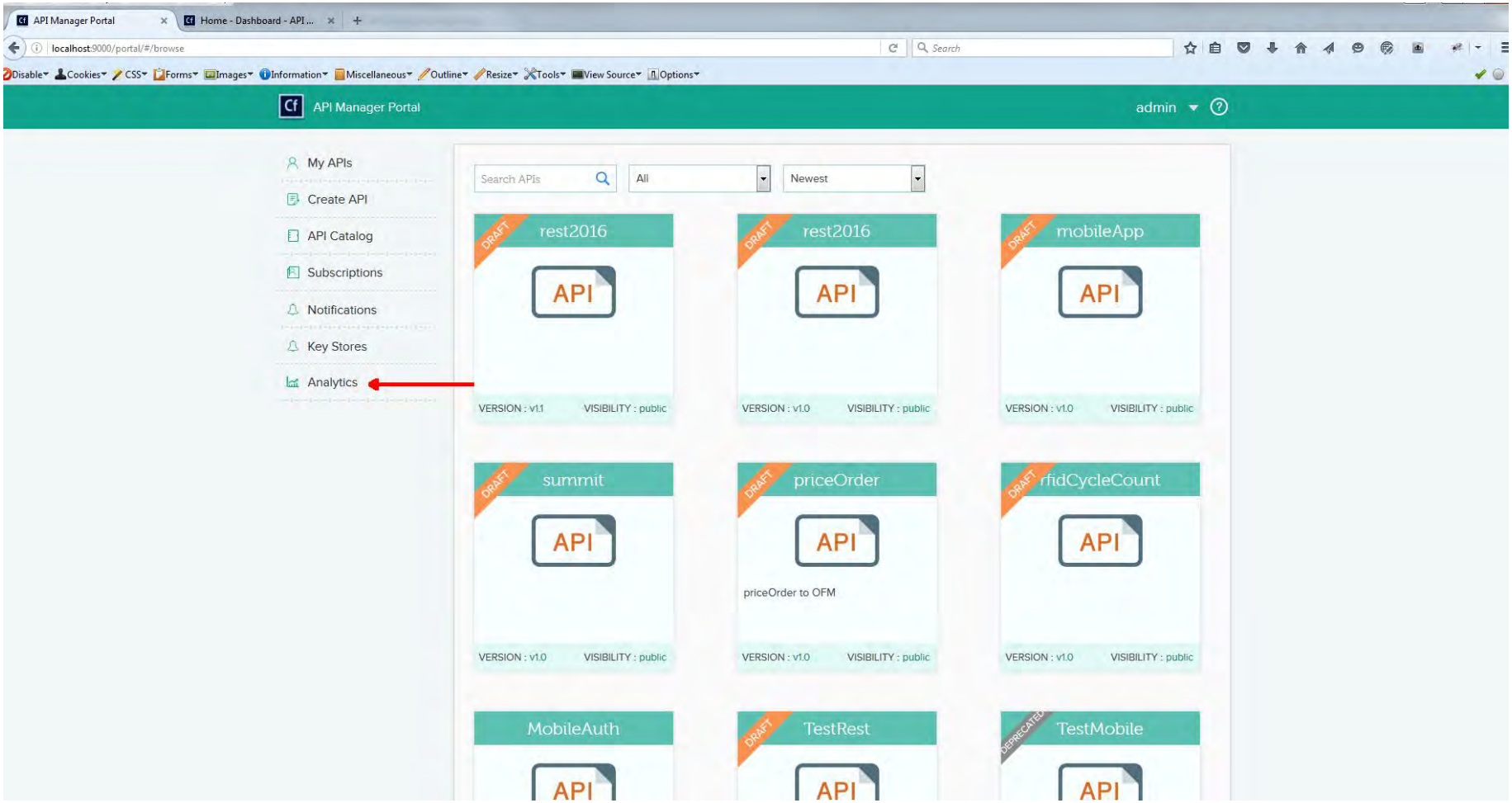



```
{
    "swaggerVersion":"1.2",
    "apiVersion":"1.0",
    "basePath":"http://localhost:8500/rest/rest2016",
    "resourcePath":"/authors2016A",
    "apis":[
        {
            "path":"/authors2016A/{authorid}",
            "description":"",
            "operations":[
                {
                    "nickname":"getAuthorDetails",
                    "method":"GET",
                    "summary":"",
                    "type":"Object",
                    "produces":[
                        "application/xml",
                        "application/json"
                    ],
                    "parameters":[
                        {
                            "name":"authorid",
                            "paramType":"path",
                            "allowMultiple":false,
                            "required":true,
                            "type":"string"
                        }
                    ]
                },
                {
                    "nickname":"deleteAuthor",
                    "method":"DELETE",
                    "summary":"",
                    "type":"Object",
                    "parameters":[
                        {
                            "name":"authorid",
                            "paramType":"path",
                            "allowMultiple":false,
                            "required":true,
                            "type":"string"
                        }
                    ]
                }
            ]
        },
        {
            "path":"/authors2016A/authorUpdate",
            "description":"",
            "operations":[
                {
                    "nickname":"updateAuthor",
                    "method":"PUT",
                    "summary":"",
                    "type":"Object",
                    "parameters":[
                        {
                            "name":"authorID",
                            "paramType":"form",
                            "allowMultiple":false,
                            "required":true,
                            "type":"string"
                        },
                        {
                            "name":"firstname",
                            "paramType":"form",
                            "allowMultiple":false,
                            "required":true,
```

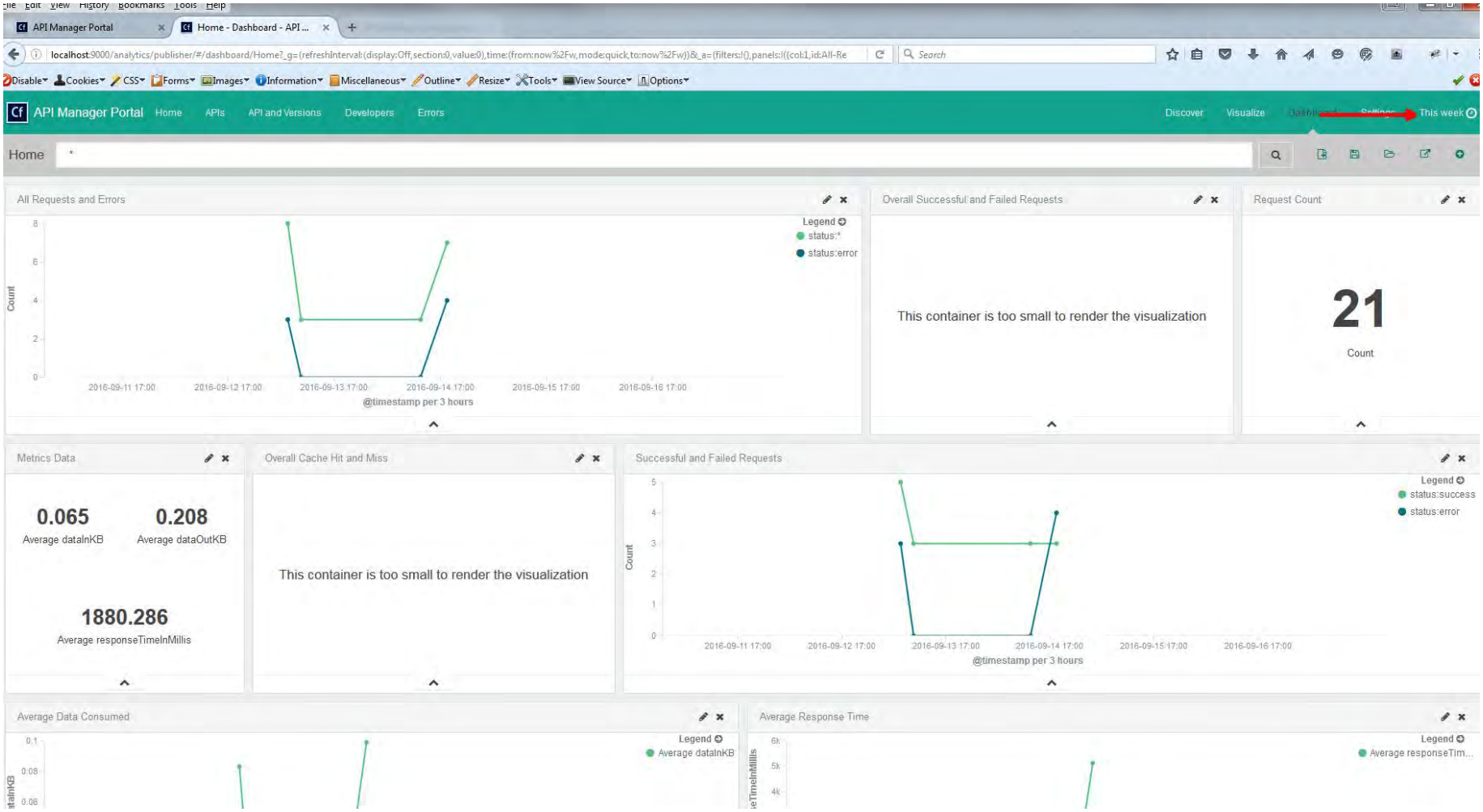
Swagger UI – Calling API Declaration



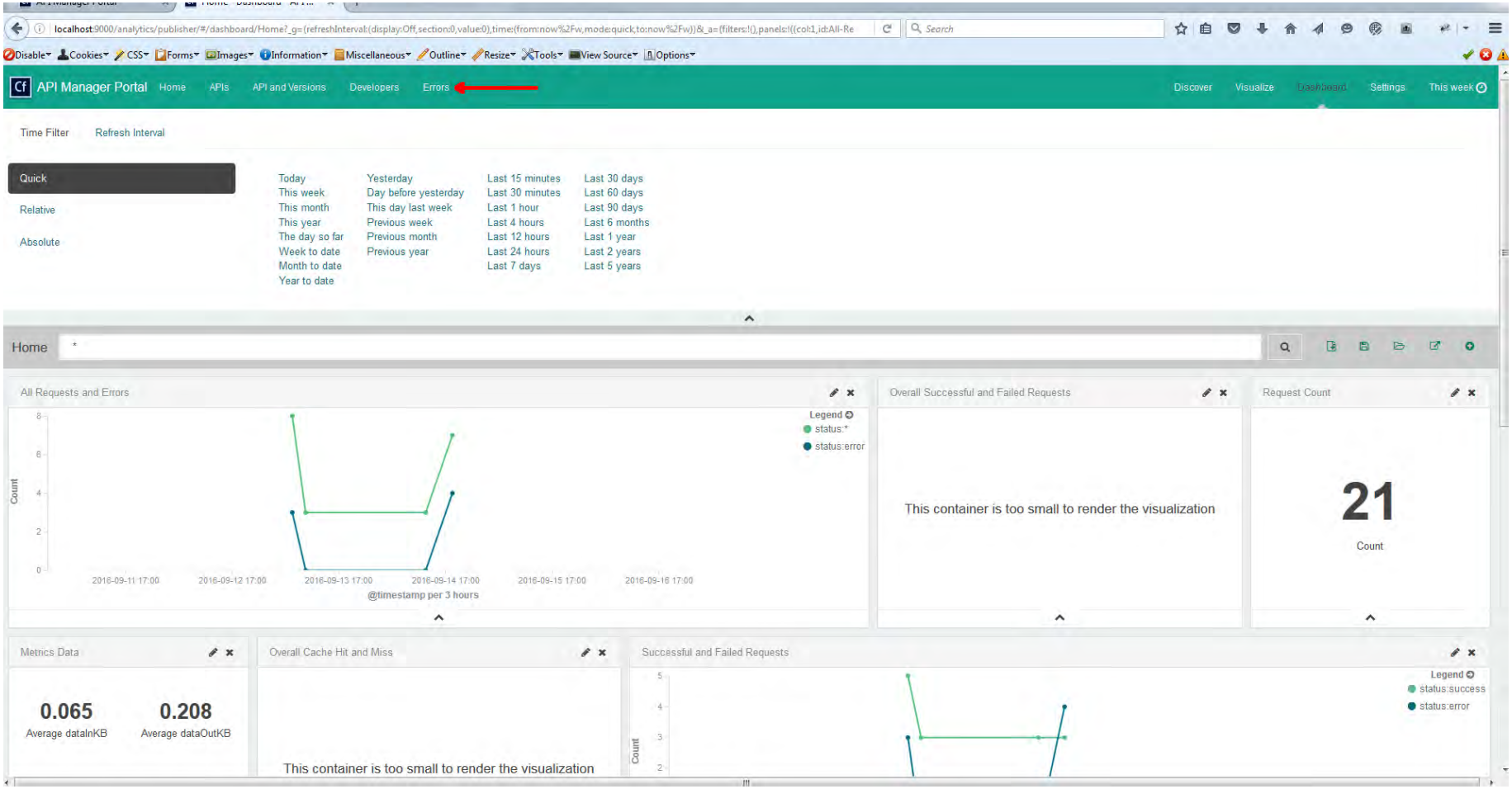
Analytics Dashboard



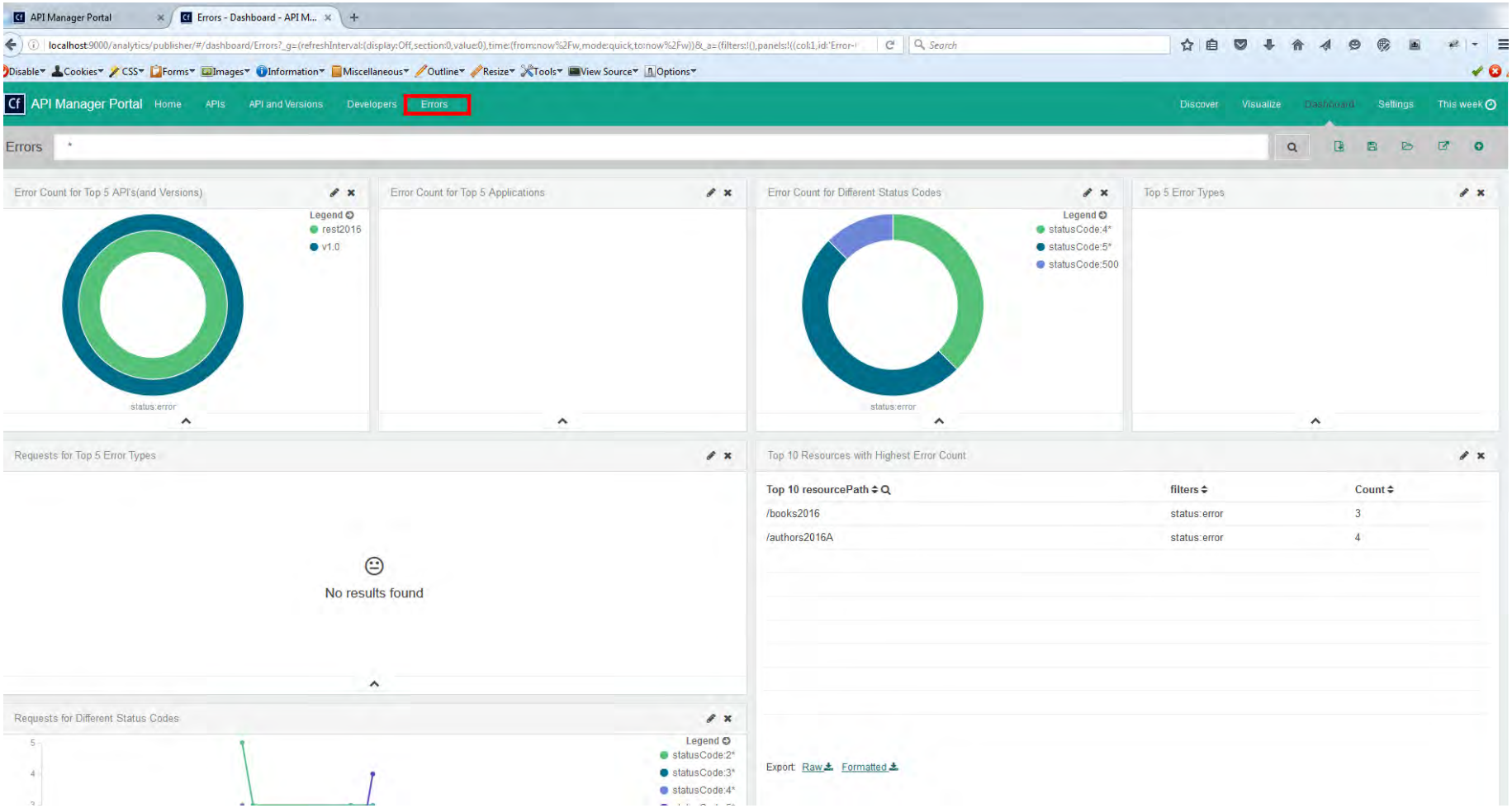
Analytics Dashboard - Home



Analytics Dashboard – Setting Time Frame



Analytics Dashboard - Errors



Helpful Links

REST RESOURCES

Getting started with RESTful web services in ColdFusion

REST Web Services In ColdFusion

Generating Swagger Docs

Swagger Docs CFML

Enable CORS Support

Getting CORS To Work With Apache

SOFTWARE

Adobe ColdFusion

SoapUI

Postman

Boomerang

Swagger UI

API MANAGER RESOURCES

ColdFusion API Manager Overview

API Manager Features Summary

Adobe ColdFusion API Manager Administrator

Introduction to Adobe ColdFusion API Managerl

API Administrator Portal

API Manager-Metrics and Logging

API Manager Analytics Dashboard

API Manager Authentication Types

APPLICATION DEMO

This Is The End

QUESTIONS?