
Article

[Istvan Hahn](#) · Sep 23, 2016 6m read

Creating a RESTful Service using Ensemble

This is a detailed guide to develop RESTful services using InterSystems Ensemble. The goal of this guide is to make you understanding the basic concept and building blocks of a RESTful service. The service is going to provide a very basic functionality (a “ Hello world! ”).

You will learn how to create required components as Ensemble classes, configure the run-time as an Ensemble Production and create a service configuration as a web application.

The Ensemble documentation library explains basically two ways to implement RESTful web service using Ensemble. The first alternative is creating a direct definition of the service by building a subclass of `EnsLib.REST.Service` class. This is the all-in-one solution. A single class does everything: serving HTTP request, map service URL to Ensemble production item, process service requests. The second approach is developing an indirect solution. In the indirect solution the standard CSP infrastructure serves HTTP requests and there are two separate classes implementing the service. One holds the resource map which links the service URL to the actual Ensemble production item and another class holds a standard (adapter agnostic) Ensemble service implementation. This guide follows the second approach.

Before you read

You need

- An Ensemble server running version 2016.2 or later
- An Ensemble client or an Atelier with compatible version to the server
- An Ensemble enable namespace
- Sufficient level of authorization to perform the development, administration and test

Experience required

- Basic Caché Object Script programming
- Basic Ensemble concept
- Basic system administration
- Understanding the REST concept

Building project with Ensemble Studio

The following section explains how to create the project when you are using Ensemble Studio.

Create a new Studio project (optional)

To organize the components we need to have a Studio Project. To create a new project launch the Ensemble Studio.

Use the File menu and call the New Project menu point. Right after this call the Save Project As menu point.

Create new Resource Map

Resource map is responsible to translate the resource name included by the URL to Ensemble configuration item name. The Resource Map is an Ensemble class created as a subclass of %CSP.REST.

The class can be created by the New Class wizard. The wizard starts from the File à New menu.

Make sure, that the General / Caché Class Definition is selected on the first page of the wizard.

Please use any package and class name on your preference.

Complete the wizard by selecting the Subclass of on the class type page and entering %CSP-REST as the superclass.

Create a service class

The service class is the implementation of your business logic. You create it by using the New wizard. Use the Production / Business Service wizard.

Please use any package and class name on your preference.

Leave the Adapter class field <undefined>.

Building project with Atelier

The following section explains how to create the project when you are using Atelier.

Create a new Atelier project

To create a new project launch Atelier, and in the File menu call the New à Project menu point. Please remember, that Atelier requires a project to work on.

Please select the server connection you created prior and the namespace you want to work in.

Create new Resource Map

Resource map is responsible to translate the resource name included by the URL to Ensemble configuration item name. The Resource Map is an Ensemble class created as a subclass of %CSP.REST.

The class can be created by the New Class wizard. The wizard starts from the File à New menu. Please use the General/ Empty class template.

Please use any package and class name on your preference.

Add %CSP.REST to the list of superclasses.

Create new Service class

The next step is to create the service implementation class. It can be created using the File à New wizard. The wizard should use the Ensemble Business Service template.

Please note! The current version of Atelier requires an adapter class selected. You can pick up any, but please make sure that after creating the class the ADAPTER static parameter is reset to empty.

Deploy project using Ensemble Management Portal

Once the project has been finished, it is time to create a deployment configuration using the Ensemble Management Portal. You should log-on to your namespace. The first step is to create a Production. The Production is the container running your service.

When you first time launch the Production configuration page, you ought to start the “ New ” wizard. It pops up a form. Please select “ Generic production ” .

When the new production has been created start the Add Service wizard and add the service class to the production. Please use a short name as the service name. This service name will be part of your RESTful service URL, therefore it must be short, self-explaining and unique to the production.

Once the service add successfully and enabled the production is ready to start.

The last step in the deployment is the service configuration set-up. It could be done by the Management Portal, Security settings section in the Web Application menu.

You should add a new Web Application to the list.

When you create the application make sure, that the following information are filled consistently:

- The name of the application does not exist yet.
- The application namespace points to your production namespace.
- The Dispatch class is pointing to your ResourceMap class.

Implementation

The subsequent sections are containing the implementation of the two classes developed during the course.

ResourceMap class

Here is the implementation of the ResourceMap class.

```
Class h2.createrestfulservice.ResourceMap extends %CSP.REST {  
  ///  
  /// The UrlMap determines how a Url should map to a HTTP Method and a Target ClassMet  
  hod  
  /// indicated by the 'Call' attribute. The call attribute is either the name of a met  
  hod  
  /// or the name of a class and method seperated by a ':'. Parameters within the URL p  
  receded  
  /// by a ':' will be extracted from the supplied URL and passed as arguments to the n  
  amed method.  
  ///  
  /// In this Route Entry GET requests to /:service will call the InvokeEnsembleService  
  method.  
  /// You see three variants. One without any additional parameter, and the others with  
  one and  
  /// two positional parameters.  
  ///
```

```

///
XData UrlMap [ XMLNamespace = "http://www.intersystems.com/urlmap" ]
{
<Routes>
<Route Url="/:service" Method="GET" Call="InvokeEnsembleService"/>
<Route Url="/:service/:p1" Method="GET" Call="InvokeEnsembleService"/>
<Route Url="/:service/:p1/:p2" Method="GET" Call="InvokeEnsembleService"/>
</Routes>
}

/// Please note that the positional parameters are passed by the argv argument vector
.
ClassMethod InvokeEnsembleService(service, argv...) As %Status
{
    set status = ##class(Ens.Director).CreateBusinessService(service, .instace)
    if $$$ISOK(status) {
        #dim response as %DynamicObject
        set status = instace.OnProcessInput(.argv, .response)
        if $isObject(response) {
            write response.%ToJSON()
        }
    }
    quit status
}
}

```

Service class

Here is the implementation of the Service class.

```

Class h2.createrestfulservice.Service extends Ens.BusinessService {

Parameter ADAPTER = "";

Method OnProcessInput(pInput As %String, Output pOutput As %DynamicObject) As %Status
{
    set pOutput = {"responseText":"Hello world!"}
    Quit $$$OK
}

}

```

Trying out

The easiest way to try the service is to invoke it from a browser. In order to do that, you just have to type in the navigator input the service URL. Assuming that the service runs on my local machine on port 57774 in web application "/csp/eh2/restful" and the production item name of the service is "echo" the URL would be:

<http://localhost:57774/csp/eh2/restful/echo>.

If you see the JSON string as a returning page, than congratulations, you completed your first RESTful service

implementation.

Stay tuned, I ' ll be back soon with further reading on Ensemble RESTful web services. Next is “ Enabling Cross-origin Resource Sharing (CORS) for RESTful Services ” .

[#Beginner](#) [#Ensemble](#) [#JSON](#) [#REST API](#)

Source URL: <https://community.intersystems.com/post/creating-restful-service-using-ensemble>