

JOBS

- legacy jobs are deprecated.
- **Jobs** can use out-of-the-box system steps, or a developer can create custom **job** steps. You can run **jobs** manually as needed or schedule them to run at a specific time or on a recurring basis.
- **Job Flow** A flow controls the sequence in which **job** steps are executed, Every job must contain at least one flow. Each flow must contain at least one step.
- You can configure sequential flows, which execute one after the other. You can also configure sibling flows, which execute in parallel.
- **Job Flow Scope** can have two types of scope. Organization or site based
- A job step tells the job what to do. Each job flow must contain at least one step. The job framework includes some out-of-the box system job steps that require no coding for common processes. If there is no system step that does what you want to do, a developer must create a custom job step.
- step types: to define a custom job step you need to configure steptyps.json file that is located in the root of a cartridge. You have an example of that file in app hang cartridge.
That way you can create your custom job parameters in the job step

DW.System.Loggers

A log4j like logger instance. To obtain such an instance, use the [Logger.getRootLogger\(\)](#) or [Logger.getLogger\(String\)](#) or [Logger.getLogger\(String, String\)](#) methods.

Logger object has 5 levels. Fatal, Error, Warning, Info and Debug debug level can not be used on production instance. For every of this levels you have a method that you can call ex. `Logger.error(msg, arg)`, `Logger.debug(msg, arg)`...

Custom logger: you can configure custom logger in BM administration/operations/custom log settings. You define a category of the log file like Product, the product can have subcategories Product.import....

The ancestor category will include the errors of all enabled subcategories. When creating custom log you select cat name exp. onSession. you can log that like [Logger.getLogger\(onSession: cat\)](#) or [Logger.getLogger\(onSession: cat, onSession: prefix\)](#) and you will get log file with a name

custom-OnSession-blade7-4.mon.demandware.net-0-appserver-20200712.log if you use the prefix

There is a limit to number of log messages per realm per day that can be viewed in Log Center. A message appears when the limit is exceeded and the log file will be suspended until the end of the day.

Service framework

Salesforce B2C Commerce provides a web service framework, which helps you manage calls to web services and analyze service performance and configure limits.

For each web service, the framework requires a service configuration, a service profile configuration, and a service credential configuration. You

create these configurations in Business Manager. The service configuration generates a ServiceConfig object. The service credential is used for basic authentication. The service profile determines how B2C Commerce manages timeouts and rate limits for the web service.

Go to administration/Operations/services and configure a service object by creating a service, service profile and service credentials.

A web service script obtains a Service object by calling the LocalServiceRegistry.createService method. The method takes two arguments: the name of a service configuration, and a script object that contains callback code that you provide. Your script then invokes the service by calling Service.call()

When the call is complete, your script either passes the result of the successful call to the pipeline dictionary or throws an error.

if you experience reports of problems with a specific web service on your site, in addition to logging, you can also investigate the incident using the analytics for registered web services. The analytics are available at Business Manager: **Administration > Operations > Service Status**

How to config a service: go to Administration > Operations > Service service tab

name is mandatory field, select type of the service. There are 5 types http and httpForms are http sevicees, sftp, ftp and soap.

http type is mostly used. Enable it and the select the mode of the service. It can be mocked(lazni sa odgovorom koji sami pripremimo) or live. the select which profile and credentials will the service use(you create them). The service profiles are used to set the limits to the service call for exp timeout connection, Circuit Breaker(how many calls per ms is alowed), the Service credentials is where you set the name of credentials url of the service, username and the password. All of this informations are available in service object that is created

and you can use them in callback functions of the service or you can override them.

Calling the services and callback functions

in most cases you can services form controller or model. Require the `localServiceRegistry` script where service definitions will be exp.

`LocalServiceRegistry.nameOfServiceExportFun.call()` . In the `LocaleService`

```
var smAddingExternalEventSvc =  
LocalServiceRegistry.createService('name.of.the.service', {
```

```
  createRequest: function (svc, args) {  
    svc.addHeader('Content-Type', 'application/json');  
    svc.addHeader('charset', 'utf-8');
```

```
  return args;  
},
```

```
  parseResponse: function (svc, client) {  
    return client.text;  
  },
```

```
// Hide sensitive data in server request and response logs  
  getRequestLogMessage: function (reqObj) {  
    var requestJSON = JSON.parse(reqObj);
```

```
    return JSON.stringify(requestJSON);  
  }  
});
```

createRequest, parseResponse and getRequestMessage are the callback function of the service obj. if the service is type soap the the callback will be initServiceClient(Service), createRequest, execute(Service, Object), parseResponse and getRequestMessage(this is optional callback function). Most of the coding is done in createRequest as in that function we can access the values of configuration object(url,username...), set headers, adjust request body. if service is mocked the we use mockCall(Service, Object) instead of execute method.

The response of the service is in json format. If you use soap service you have to provide a wsdl document that is placed in the webreferences or webreferences2 folder

if in the question you see rpc, wsdl, webreferences that is regarding a soap service.