A Developer Diary

{about:"code learn and share"}

September 1, 2015 By Abhisek Jana — 2 Comments (Edit)

Steps for implementing DynaCache in Liberty and Websphere

DynaCache is maintained by the server, means you don't have to deal with concurrency, threading etc. The downside is, it will allocate memory in the application JVM Heap, increasing the Heap beyond a certain limit might impact the performance negatively, specially during Garbage Collection.

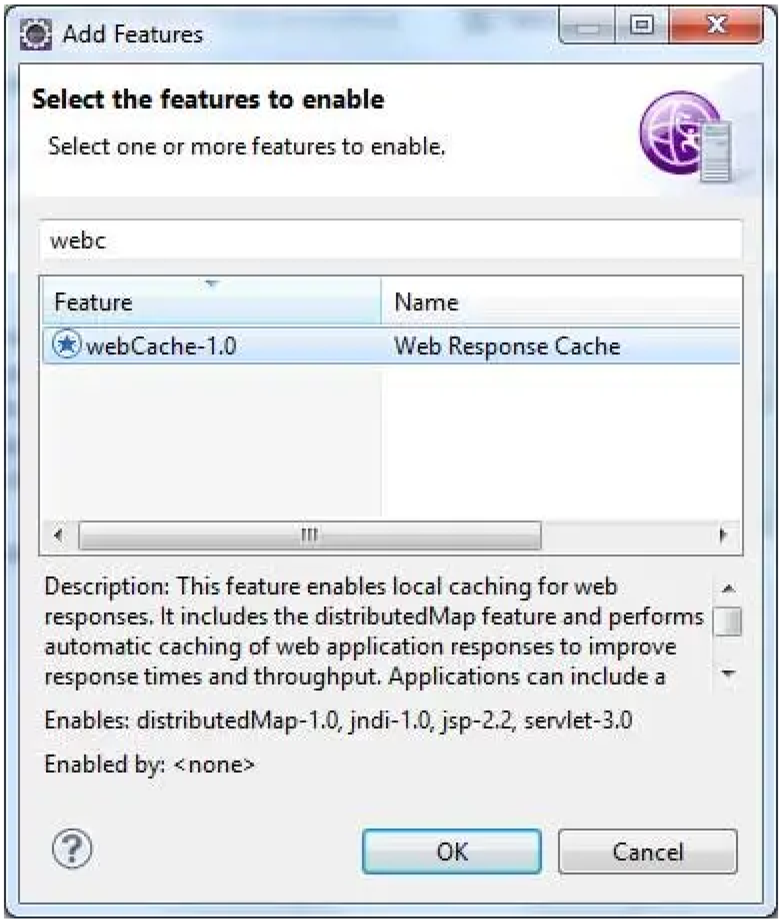
In this article, we will go over how to setup DynaCache in both Liberty & Websphere then use it in our code. DynaCache can also be persistent if you choose the option to use the disk.

Here are the steps for implementing DynaCache in Liberty and Websphere.

Liberty Server Setup :

If you want to find out how to setup Liberty you can follow my previous post on How to Setup Liberty.

Expand the Server and open the `Server Configuration`. Select `Feature Manager` in the design view. Click on Add. Select `webCache` and click on OK.


Add Features

Select the features to enable

Select one or more features to enable.

webc

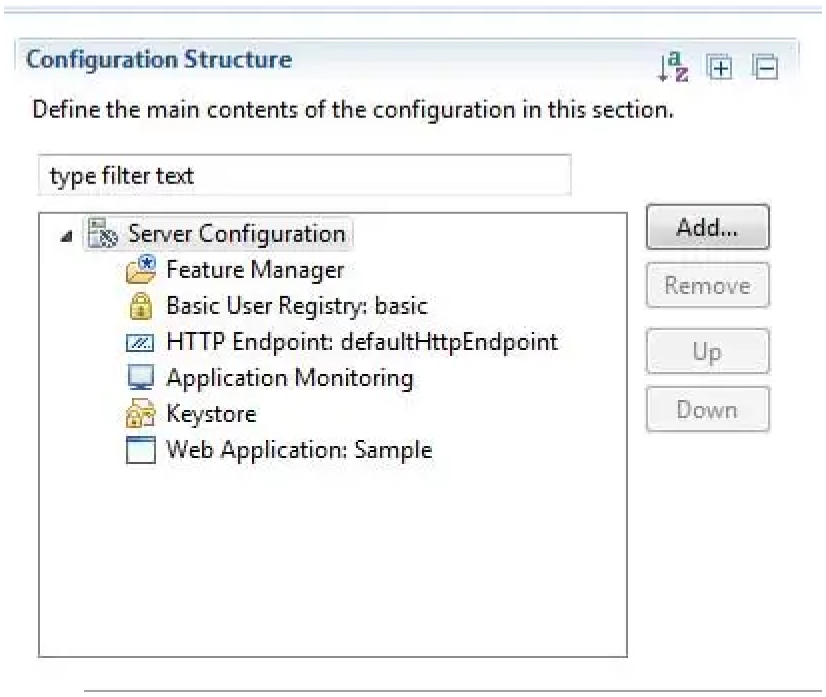
Feature	Name
webCache-1.0	Web Response Cache

Description: This feature enables local caching for web responses. It includes the distributedMap feature and performs automatic caching of web application responses to improve response times and throughput. Applications can include a

Enables: distributedMap-1.0, jndi-1.0, jsp-2.2, servlet-3.0

Enabled by: <none>

Select `Server Configuration` and click on Add.

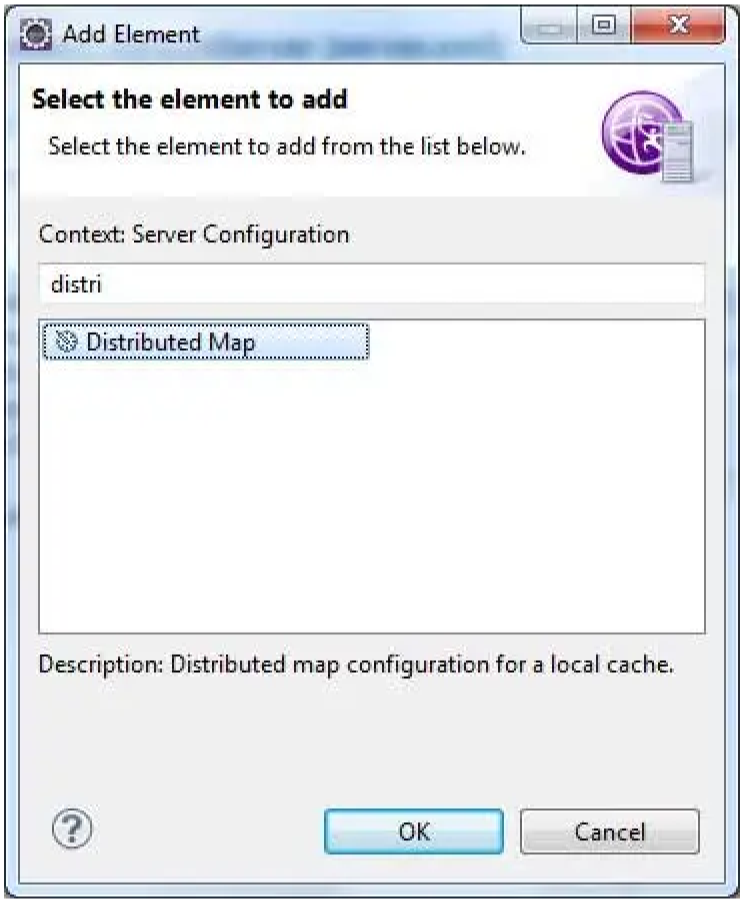

Configuration Structure

Define the main contents of the configuration in this section.

type filter text

- Server Configuration
  - Feature Manager
  - Basic User Registry: basic
  - HTTP Endpoint: defaultHttpEndpoint
  - Application Monitoring
  - Keystore
  - Web Application: Sample

Select `Distributed Map` and click on OK.



Select the element to add

Select the element to add from the list below.

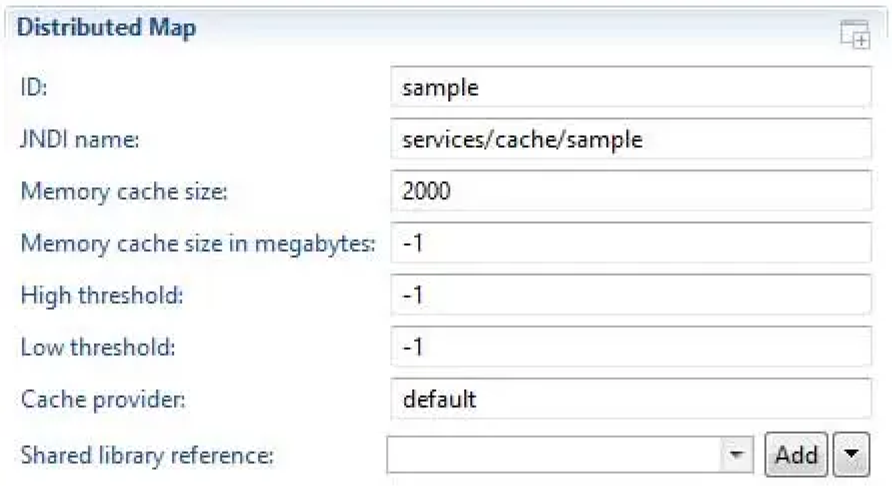
Context: Server Configuration

distri

Distributed Map

Description: Distributed map configuration for a local cache.

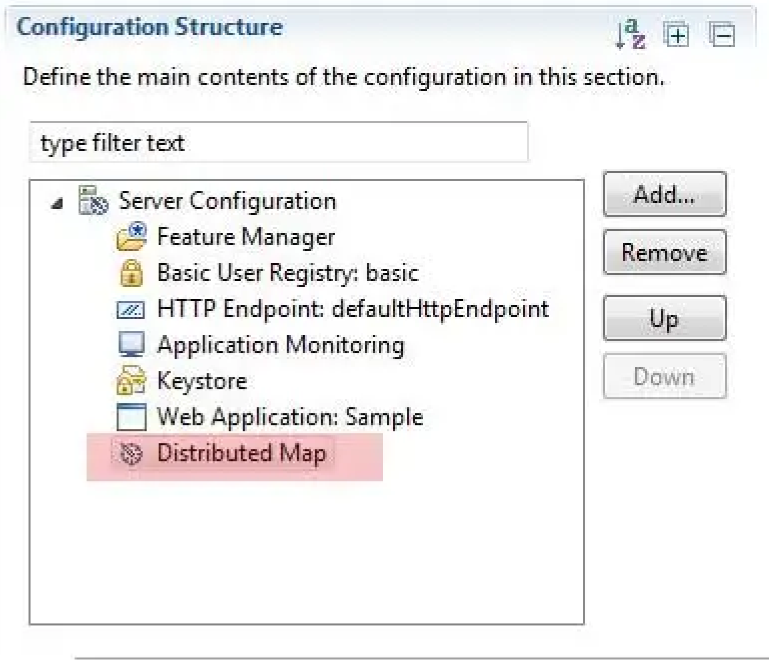
Add the JNDI Name as `services/cache/samplecache` and id as `sample`.



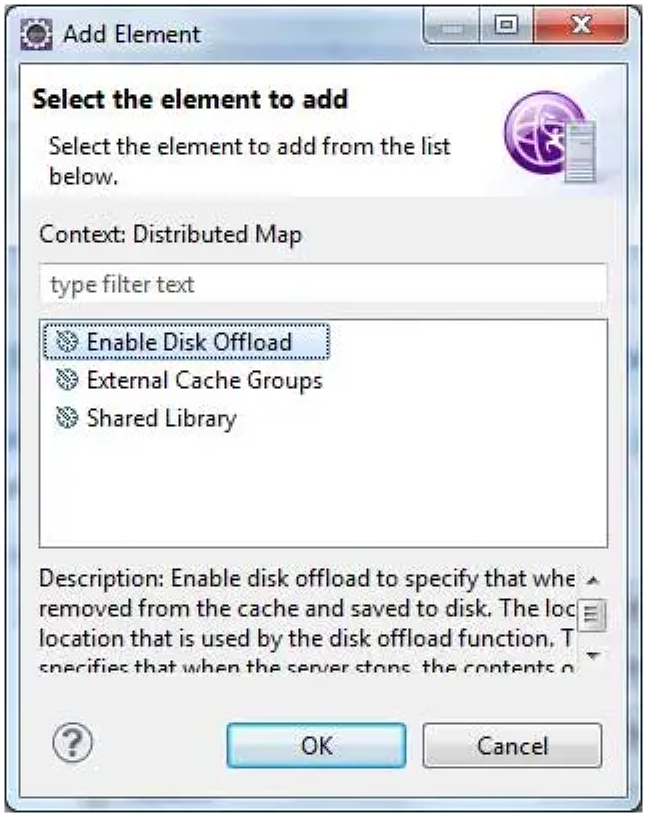
Distributed Map

ID:	sample
JNDI name:	services/cache/sample
Memory cache size:	2000
Memory cache size in megabytes:	-1
High threshold:	-1
Low threshold:	-1
Cache provider:	default
Shared library reference:	

Select `Distributed Map` and Click on Add.

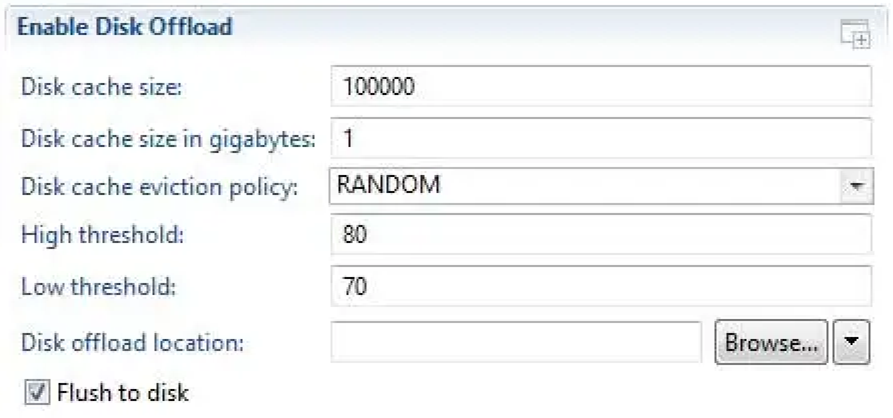


Select `Enable Disk Offload`, click OK.



Description: Enable disk offload to specify that whe
removed from the cache and saved to disk. The loc
location that is used by the disk offload function. T
specifies that when the server stops, the contents o

Here are the settings; I have just have set the `Flush to Disk`. You can change these settings as per requirement. This is an optional step.


Enable Disk Offload

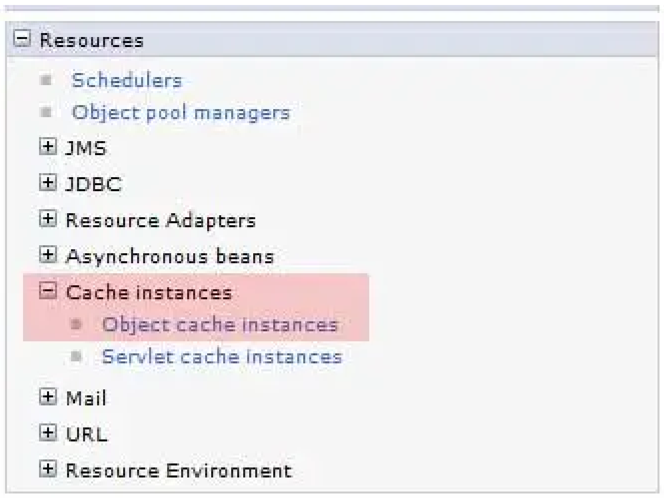
Disk cache size:	100000
Disk cache size in gigabytes:	1
Disk cache eviction policy:	RANDOM
High threshold:	80
Low threshold:	70
Disk offload location:	

[✓] Flush to disk

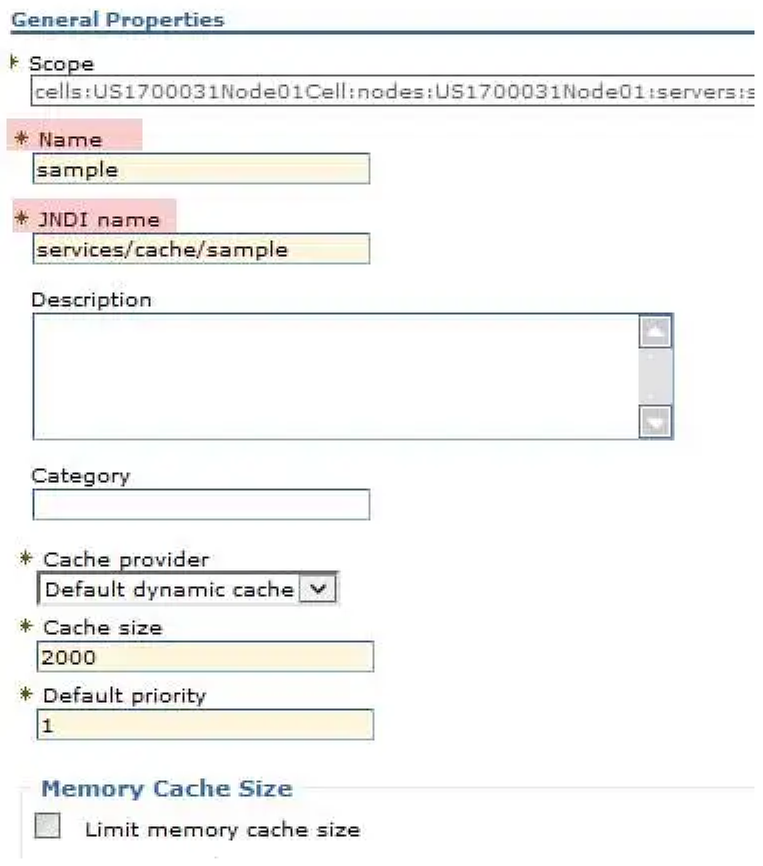
Save the settings and let’s jump into the code.

Websphere Server Setup :

In the admin console, you need to go to `Resources -> Cache Instances -> Object cache instances`.



Add the JNDI Name as `services/cache/samplecache` and name as `sample`.


General Properties

* Scope
cells:US1700031Node01Cell:nodes:US1700031Node01:servers:s

* Name
sample

* JNDI name
services/cache/sample

Description

Category

* Cache provider
Default dynamic cache

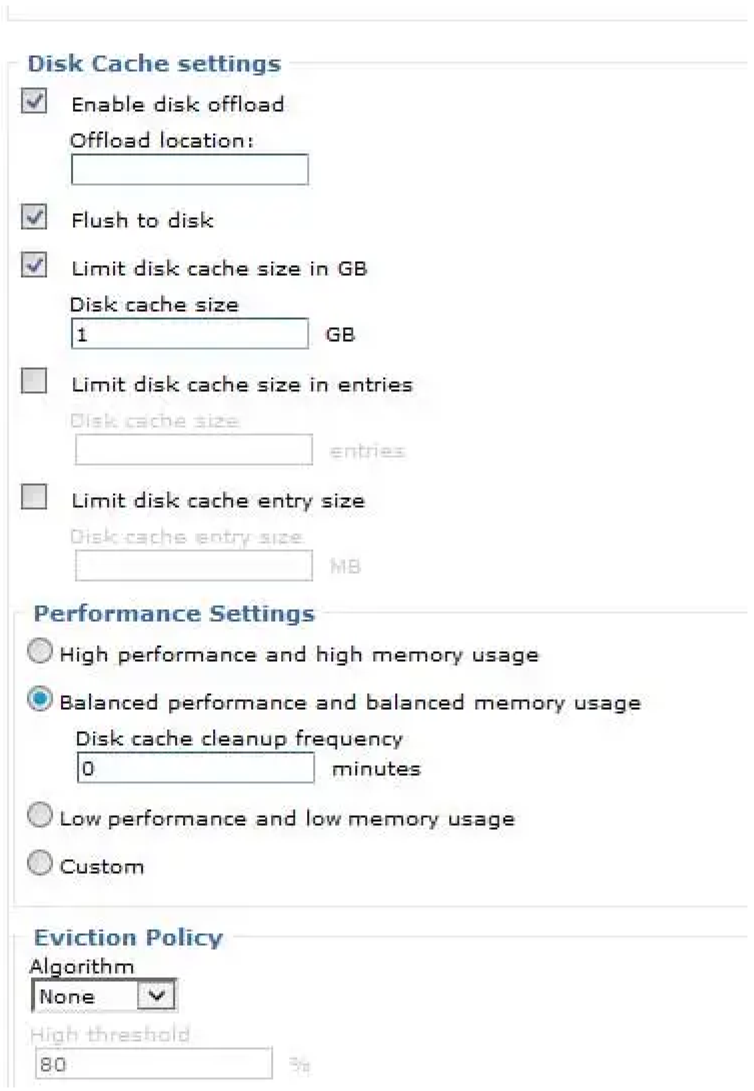
* Cache size
2000

* Default priority
1

Memory Cache Size

☐ Limit memory cache size

You can select the `Disk Cache Settings`, where you can update the following as needed.



Disk Cache settings

[✓] Enable disk offload

Offload location:

[✓] Flush to disk

[✓] Limit disk cache size in GB

Disk cache size: 1 GB

[ ] Limit disk cache size in entries

Disk cache size: entries

[ ] Limit disk cache entry size

Disk cache entry size: MB

Performance Settings

( ) High performance and high memory usage

(●) Balanced performance and balanced memory usage

Disk cache cleanup frequency: 0 minutes

( ) Low performance and low memory usage

( ) Custom

Eviction Policy

Algorithm: None

High threshold: 80 %

Now let’s use the cache from the code.

Code :

First, create a `Web Project`. We will be testing the DynaCache using two REST (JAX-RS) services, you want to find out more on setting up the REST (JAX-RS) in liberty, you can follow my other post on How to configure REST Service

Let’s create an adaptor class named `DynaCacheAdaptor` to hold the cache object using a singleton pattern.

```
package com.ajana;
```

```
import javax.naming.InitialContext;
import javax.naming.NamingException;

import com.ibm.websphere.cache.DistributedMap;

public class DynaCacheAdaptor {

        private static DistributedMap cacheObject;

        protected DynaCacheAdaptor() {

        }

        public static DistributedMap getCacheMapObj()
throws NamingException {
                if (cacheObject == null) {
                        InitialContext ctx = new
InitialContext();
                        cacheObject = (DistributedMap)
ctx

.lookup("services/cache/sample");
                }

                return cacheObject;
        }

}
```

The `EmployeeService` class would have service named `createEmployee()` to create an employee using the name that was passed and store the `EmployeeTO` object in DynaCache. Then we using `getAllEmployees()` method we will retrieve the list of employees from the DynaCache.

```
package com.ajana;

import javax.naming.NamingException;
import javax.ws.rs.ApplicationPath;
import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;
import javax.ws.rs.QueryParam;
import javax.ws.rs.core.Application;

@ApplicationPath("app")
@Path("employee")
public class EmployeeService extends Application {

        @GET
        @Path("getAllEmployees")
        @Produces("application/json")
        public Object[] getAllEmployees() throws
NamingException {

                return
DynaCacheAdaptor.getCacheMapObj().entrySet().toArray();
        }

        @GET
        @Path("createEmployee")
        @Produces("application/json")
        public boolean createEmployee(@QueryParam("name")
String name)
                        throws NamingException {

                boolean returnCode = false;
                EmployeeTO employee = new EmployeeTO();

                employee.setName(name);

employee.setId(String.valueOf(name.hashCode()));


DynaCacheAdaptor.getCacheMapObj().put(employee.getId(),
employee);
                returnCode = true;

                return returnCode;
        }
}

class EmployeeTO {

        private String id;
        private String name;

        public String getId() {
                return id;
        }

        public void setId(String id) {
                this.id = id;
        }

        public String getName() {
                return name;
        }

        public void setName(String name) {
                this.name = name;
        }
}
```

Now add the project to the server and start the server.

Invoke the following URLs in the browser to create two employees, named John & Robert.

`http://localhost:9080/Sample/app/employee/createEmployee?name=John`
`http://localhost:9080/Sample/app/employee/createEmployee?name=Robert`

In both cases, the above URLs should return true if the the employee has been added to the DynaCache.

Then retrieve the employees from DynaCache using the following URL:

`http://localhost:9080/Sample/app/employee/getAllEmployees`

Here is the JSON response from the server with the list of employees. Since I am returning the `entrySet()` from the Map, you can see the whole object here.

`[{"value":{"name":"John","id":"108952"},"key":"108952"},{"value":{"name":"Robert","id":"108986"},"key":"108986"}]`

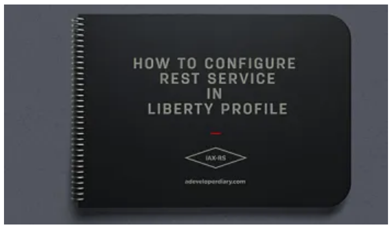
Related

How to setup IBM Liberty Profile

In "Server"

How to deploy Spring Boot application in IBM Liberty and WAS 8.5

In "Spring Boot"



How to configure REST Service (JAX-RS) in IBM Liberty Profile

In "REST"

Filed Under: Cache | Tagged With: Cache, DistributedMap, DynaCache, Java, liberty, Liberty Profile, WebService

Subscribe to stay in loop

* indicates required

Email Address *

Subscribe

Comments

Parth hirpara says

September 22, 2017 at 8:51 am

(Edit)

This class is not available in IBM Liberty run time
com.ibm.websphere.cache.DistributedMap

Reply

Omer Zahid says
December 2, 2022 at 8:07 pm

(Edit)

Hi did you ever get resolve the issue?

I am getting:
fails with a ClassCastException:
java.lang.ClassCastException:
com.ibm.ws.cache.CacheServiceImpl cannot be cast to
com.ibm.websphere.cache.DistributedMap

Reply

Leave a Reply

Logged in as Abhisek Jana. Edit your profile. Log out? Required fields are marked *

Comment *

Post Comment