Technical Blog

Contents

[Reusable master template 1](#_Toc415417399)

[How this word document template is created 1](#_Toc415417400)

[Oracle Date - Do you know this 1](#_Toc415417401)

[Scalability rule – Communicate asynchronously as much as possible 2](#_Toc415417402)

[Jason - Java 2](#_Toc415417403)

[Using Annotations 2](#_Toc415417404)

[Hamcrest Matchers 3](#_Toc415417405)

[What is Restful Architecture 3](#_Toc415417406)

[Resource in Jersey 3](#_Toc415417407)

[Reusable master template 4](#_Toc415417408)

# Reusable master template

This is reusable master template that can be copied and reused for this entire document. All content in this document need to be as per this template.

# How this word document template is created

Reference >> Table of contents to generate table of contents.

Home >> small arrow under change styles to select headings which are used for table content generation.

<https://www.youtube.com/watch?v=awRxezHJpSM>

# Oracle Date - Do you know this

Oracle date is internally represented by using seven comma separated numbers like – 200,115, 03, 02, 01, 01, and 01. No time zone is stored. Dump function can be used to get internal representation.

To get actual date from internal representation we subtract 100 from first two numbers and 1 from last three numbers.

Date operations – Add days

DATE + NUMTODSINTERVAL(n , ‘second’)

ADD\_MONTHS(SYSDATE , 7) --- No exception and behaves in a special way during month ends

NUMTOYMINTERVAL – Dont use and not recomended

Date operations – Subtract days

D2-D1>> No of days

TO\_DATE(‘ ’ , ‘xxx’) - XXX= YYYY MM DD HH HH12 HH24 MI SS

TO\_CHAR(dt , ‘YYYY-MM-DD’)

TRUNC(dt) – This will remove hours , minutes and seconds to zero

# Scalability rule – Communicate asynchronously as much as possible

There is one problem with synchronous communication . A -> B-> C -> D. If there is problem it will lead to failure in all down line services though the problem is not in those services. To avoid this problem we should plan to take up asynchronous calls as much as possible with **proper call back calls**.

In general for below scenarios we go with asynchronous calls

* External system calls (Aon)
* Long time processes (Auto Research)
* Error prone code or one thats is changed frequently
* Temporal constraints like Email

# Jason - Java

You should know how every java object converts to Jason format {} – one java object and [] – list or array of objects or set

{ a : b , c:d } or {a:[{x :y} , { x:y} ] , b:d}

ObjectMapper mapper = new ObjectMapper(); mapper.writeValuesAsString(object..); mapper.writeWithDefaultPrettyFormat

# Using Annotations

@MyAnnotaion(single,single,many)

@MyAnnotation(value=”anil” , a=”b” , many={“x” , “y”})

# Hamcrest Matchers

Hamcrest matchers are used to verify whether a given object meets particular criteria or not.

String – startsWith,endsWith,containsString,equalsToIgnoreCase,equalsToIgnoreWhiteSpace

Number – closeTo , lessThan , greaterThan , lessThanOrEqualTo , greaterThanOrEqualTp

Logical – anyOf , allOf , not

# What is Restful Architecture

Elements in restful architecture

* Client , server
* Resources / Resource identifier
* Uniform interface for resources
* Representation and Representation with linked resources
* Stateless
* Cacheable

# Resource in Jersey

**Resource**

@Path(“\shgh\sdjbb\”) – forward slash at beginning and end are optional @Path(“/user/{id}”) @Path(“/user/{id : regular exp}”)

**Interface**

@GET @POST @PUT @DELETE

**@produces** – based on Accept header

**@consumes** – based on content type header

Based on above four parameters relevant method is called

# Database scalability

* Normal forms (Use and Discard) – **Explore**
* Design your database to split
* SQL query performance – **Explore**
* View / Materialized view – **Explore**
* On java side - **Explore**

# Unit testing - Terminology

Workers , Managers , SUT , DOC , Direct input , Direct output , Indirect input , Indirect output , state testing , interaction testing

# Velocity framework - Template

There are two symbols used in velocity template - $ #

$ - **Place holder** for data. Data gets pulled from velocity context object

# - **Directives** which have special purpose in template

#set($name=”anil”) - Put value in velocity context

#if( $value > 5 )

bigger

#elseif( $value < 5 )

smaller

#else

just right

#end

#foreach($item in $itemList)

#end

# Velocity Framework - Architecture

There are three components that work together to form velocity framework

* Velocity engine
* Template – File or string
* Velocity context – Holds data that need to be used by template

**Template as file:**

* Create velocity engine and call init method.
* Create velocity context object and put data in it.
* Call getTemplate method on velocity engine to get template
* Call merge method on template with velocity context and string writer

**Template as string:**

* Velocity.init
* Create velocity context
* Velocity.evaluate

[Tutorial](http://www.javaworld.com/article/2075966/core-java/start-up-the-velocity-template-engine.html)

[Velocity Examples](https://github.com/asakala/git/tree/master/Velocity)(https://github.com/asakala/git/tree/master/Velocity)

# Reusable master template

This is reusable master template that can be copied and reused for this entire document. All content in this document need to be as per this template.

# Reusable master template

This is reusable master template that can be copied and reused for this entire document. All content in this document need to be as per this template.