

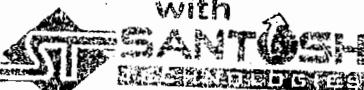
Name: _____

ADV JAVA

BY

Mr.santosh

ameerpetmaterials.blogspot.in

now
Mr. Santosh
with


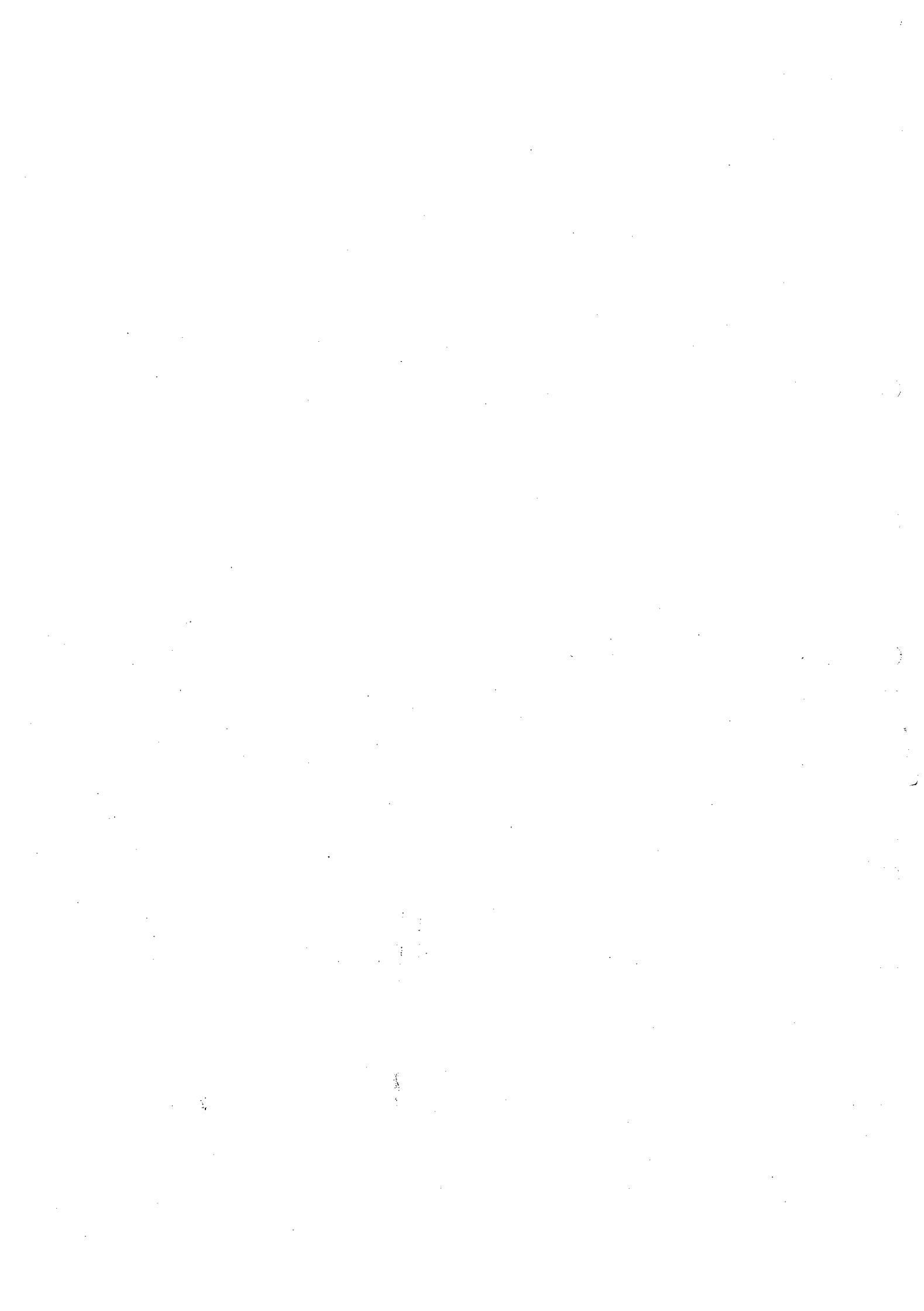
#2nd Floor, Saraswati Nilayam, Opp:DCB Bank
SR Nagar Police Station Lane, Ameerpet, Hyd.


NARESH 
technologies
SOFTWARE TRAINING & DEVELOPMENT

Opp. Satyam Theatre, Ameerpet, Hyderabad - 500 016.

Ph : 23746666, 23734842

MANOJ ENTERPRISES **XEROX**
Plot No: 40, Gayathri Nagar, Behind HUDA,
Ameerpet, Hyd. Cell: 98666 84838



February 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28					

003362, Week 01

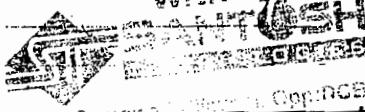
Priority

Mr. SANTOSH CLASS NOTES
Naresh I Technologies now
Ph 23734842, 23746666

MON

3

JAN



JDBC

Copyright B2005
2nd Floor, Garuda Building, Anna Nagar, Chennai, Tamil Nadu, India
SR Nagar Police Station, Anna Nagar, Chennai, Tamil Nadu, India

8am

DBMS:

→ Database management system which manages the data, i.e. allows us to store some data and retrieve the data when required.

→ It is also an application.

→ The requirement :-

* Front end application (C, C++, JAVA...) wants to interact with the DBMS (to store, update or get the data) dynamically at runtime.

→ To meet the above requirement

we can use the vendor specific API and interact with the DBMS.

Mr. SANTOSH

CLASS NOTES

Naresh I Technologies

Ph 23734842, 23746666

problems with above approach :-

1) The front end application designed becomes vendor specific since it is using vendor specific API.

2) front application may not be able to collaborate with the DBMS.

(like problem may be with the lang used to develop front end application and the DBMS abstraction, and the protocols used).



Birthday / Anniversary

Mr. SANTOSH CLASS NOTES
Naresh I Technologies
Ph 23734842, 23746666

Priority

January 2005

S	M	T	W	T	F	S
30	31					1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

095-361, Week 01

February 2005

S	M	T	W	T	F	S
	1	2	3	4		
6	7	8	9	10	11	
13	14	15	16	17	18	
20	21	22	23	24	25	
27	28					

095-360, Week 01

8/7/2005

8am

9

10

11

12

1pm

2

3

4

5

6

7

8

9

10

11

12

13

14

15

</OPEN>

→ is an open community (i.e independent of vendor) which provides some standard abstraction to the resources (like database)

These are X/open and XA interface.

→ These abstractions are implemented by the resources providers (i.e database vendors).

→ These abstractions is a low level abstractions, NO using this X/open abstraction and technologies the application was difficult.

Mr. SANTOSH
Naresh T. Technologies
Ph: 23734842, 23736666

→ 3rd party vendors (DB vendors) Started giving some call level interfaces which used to provide a high level application abstraction for the X/open interface.

Like for example:

for oracle we have oracle call level interface (oci)
Most of the call level interfaces given in native lang.

⇒ Now if a front end application wants to interact with DB if has to be programmed to call level interfaces provided by 3rd party vendors.

problems with above approach:

- 1) Application developed will be vendor specific.
- 2) Migrating the application from one DB to other is difficult.

February 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28					

Priority

Mr. SANTOSH CLASS NOTES
Naresh I Technologies
Ph: 23734842, 23746666

WED

5

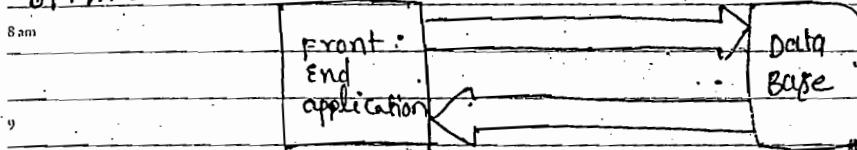
JAN

now
with
Mr. Santosh

SANTOSH

#2nd Floor, Saraswati Nilayam, Opp: DCB Bank,
SR Nagar Police Station Lane, Amravati, Hyd.

8/7/2005



10

→ To make the front end applications interact with any DB in a vendor independent way

12 Microsoft has started developing the arch for this type abstraction.

1 pm

This architecture was taken by open community which includes Microsoft, IBM, Oracle...
i.e ODBC community.

3 → This community given ODBC Specification (where ODBC stands for open database connectivity)

5 → ODBC API is used to interact with any DB (which has a support for ODBC) i.e. If a common abstraction through which we communicate with any DB.

7 → ODBC architecture :-

8 → These ODBC Specification can be implemented by any 3rd party vendor.

Birthday / Anniversary

THU

Priority:

6

JAN

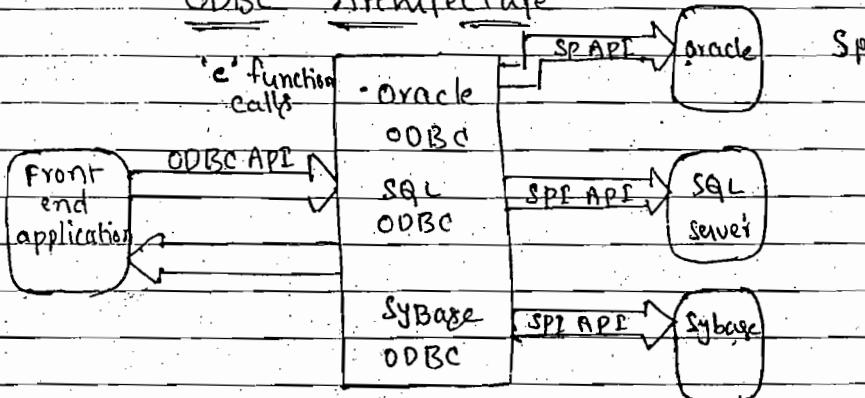
2005

January 2005

S	M	T	F	S
30	31			1
2	3	4	5	6
9	10	11	12	13
16	17	18	19	20
23	24	25	26	27
			28	29

06-12-05, Week 01

ODBC Architecture



- ODBC is designed based on x/open Standards.
- ODBC can take SQL queries.
- ODBC drivers were initially given for windows.
Now ODBC implementation is given for Linux (The complete implementation of ODBC drivers on non-windows platforms is being difficult since the ODBC architecture was designed specific to windows).
- ODBC API calls are 'c' function calls.

problems with this drivers:

- 1) Initially it was almost platform specific (i.e. drivers were available only for windows).

Advantages:

- 1) A single API (protocol) is used to interact with different DB.

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

007-358, Week 01

Priority

Mr. SANTOSH CLAS: NOTES
Naresh Technologies
PM 23734842, 23746666

FRI

7

JAN

2005

→ Migrating the application from one DB to other DB does requires any changes in the application.

If JAVA application wants to interact with DB

→ It can use DB vendor specific call level interfaces

If the above approach is followed then the problems are :-

→ Java application becomes vendor dependent

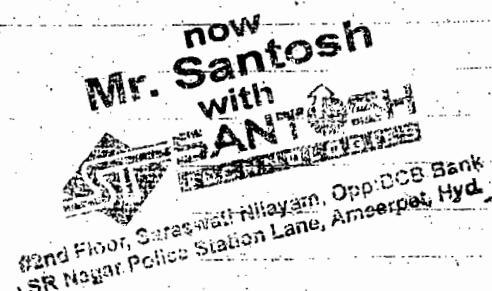
→ The DB specific call level interface may not be provided in Java (may be given in C or C++)
 So Java application has to use JNI tech and interact with the C++.

which may make our application platform dependent.

→ Can use JDBC API to interact with DB

→ Problem with this approach is JAVA application has to use JNI and make a request to JDBC Driver (i.e. using 'c' function calls).

To solve the above problem we require a common (general) Java abstraction to interact with DB.



SAT

Priority:

8

JAN

2005

January 2005

S.	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19	M20	M21	M22	M23	M24	M25	M26	M27	M28	M29	M30	M31
1																															
2																															
3																															
4																															
5																															
6																															
7																															
8																															
9																															
10																															
11																															
12																															
13																															
14																															
15																															
16																															
17																															
18																															
19																															
20																															
21																															
22																															
23																															
24																															
25																															
26																															
27																															
28																															
29																															
30																															
31																															

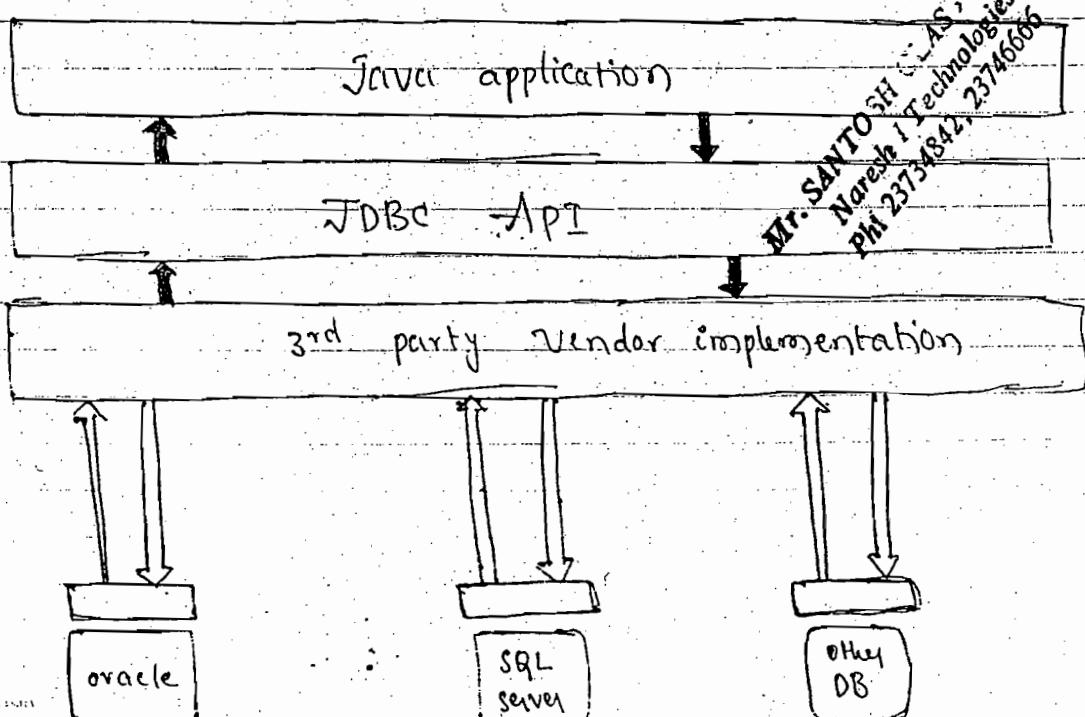
⇒ Want a Java abstraction to interact with data sources (like database)

⇒ JDBC is the solution for the above problem.

⇒ JDBC is standard API through which Java applications can interact with any data source - i.e. JDBC drivers can be designed to interact with some files (containing tabular data), Data Base.

⇒ JDBC is following X/Open standards and SQL Standards.

⇒ JDBC API allows Java applications to send SQL queries to the data source.



Mr. SANTOSH KASAS, NOTES
Mr. Narein, Technologies
Ph: 2373842, 2374666

February 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28					

Priority

S
See
Sathay.

SUN

9

JAN

2005

009-356, Week #1

8am * Sun Micro Systems given the JDBC API application

→ JDBC is a specification given by sun

JDBC 3.0 specification is included into J2SE

And these specifications has to be implemented by 3rd party vendors.

JDBC Drivers

Mr. SANTOSH CLASS NOTES
Naresh Technologies
Ph: 23734842, 23746666

→ is - a set of classes and interfaces which implements JDBC Specification.

→ which can drive our request to data source.

→ currently there are 220 drivers registered with sun i.e. 220 different implementations from different vendors are there for java JDBC Specification.

Sun has given 4 categories
i.e. 4 types of drivers

where all the drivers will fall into any one of the type of drivers.

These types describes the approach of the driver to drive the request to data source (dataBase).

Mr. SANTOSH CLASS NOTES

Naresh Technologies

Ph: 23734842, 23746666

Birthday / Anniversary

MON Priority
10 JAN 2005

January 2005						
S	M	T	W	F	S	S
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

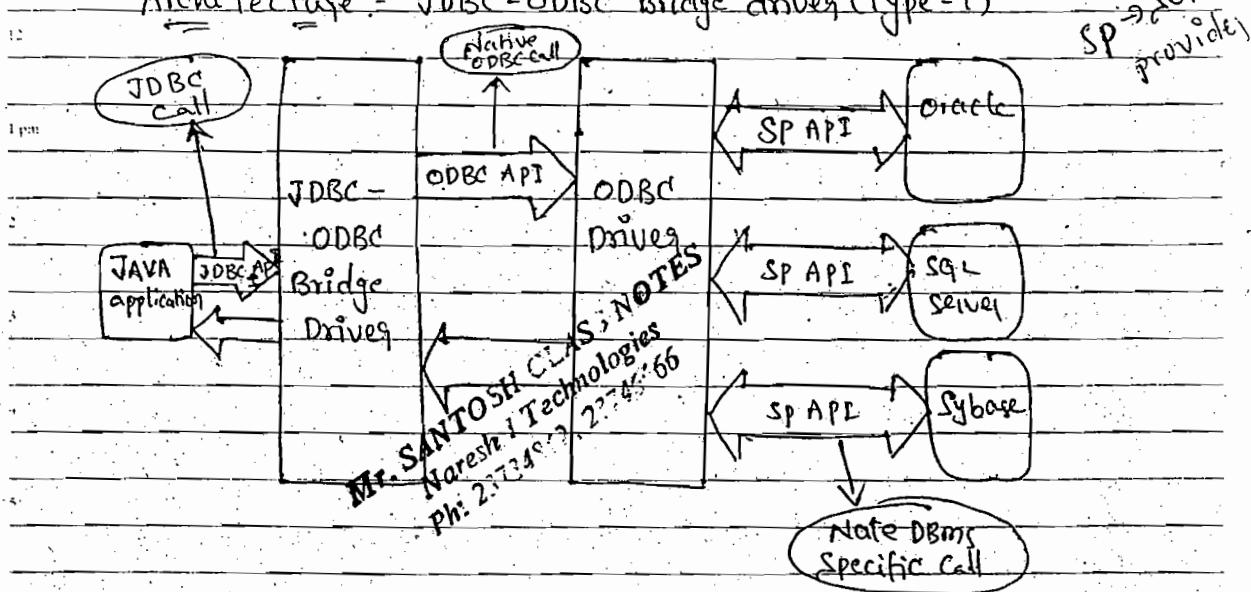
010,555, Week 02

Type-1 driver :-

JDBC - ODBC Bridge driver.

→ This driver converts the JDBC call into ODBC call and interact with ODBC driver (i.e here JDBC driver is responsible to drive the request to ODBC driver).

Architecture:- JDBC - ODBC Bridge driver (Type-1)



Advantages of system:-

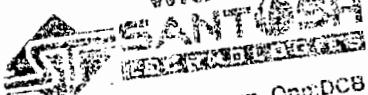
- 1) A single driver implementation can be used to interact with different DB.
- 2) If a DataBase has ODBC Support (i.e ODBC driver implementation) then using this driver from Java application we can interact with the Data Source.

February 2005						
S	M	T	W	T	F	S
1	2	3	4	5		
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28					

Priority:

011-3541 Week 02

now
Mr. Santosh
with



#2nd Floor, Saravalli Nilayam, Opp: DCB Bank
SR Nagar, Police Station Lane, Ameerpet, Hyd.

TUE

11

JAN

2005

Diseadvantages :-

8am

- 1) Due to number of conversions the performance of the application is reduced.
- 2) This driver depends on the ODBC i.e. indirectly JAVA applications are depending on ODBC.
- 3) ODBC client library has to be installed into the client machine.
- 4) In some cases database specific functionality may be required.

Mr. SANTOSH CLAS,
Naresh Technologies.
Ph: 23734542, 23746666

1pm

Uses:-

- This driver is preferable to be used when our data source doesn't have any other driver JDBC drivers and has support for ODBC.
- Not recommended to be used when Java applications are required with auto installation (like not recommended to be used in Applets).

Birthday / Anniversary

WED
12
JAN
2005

Priority

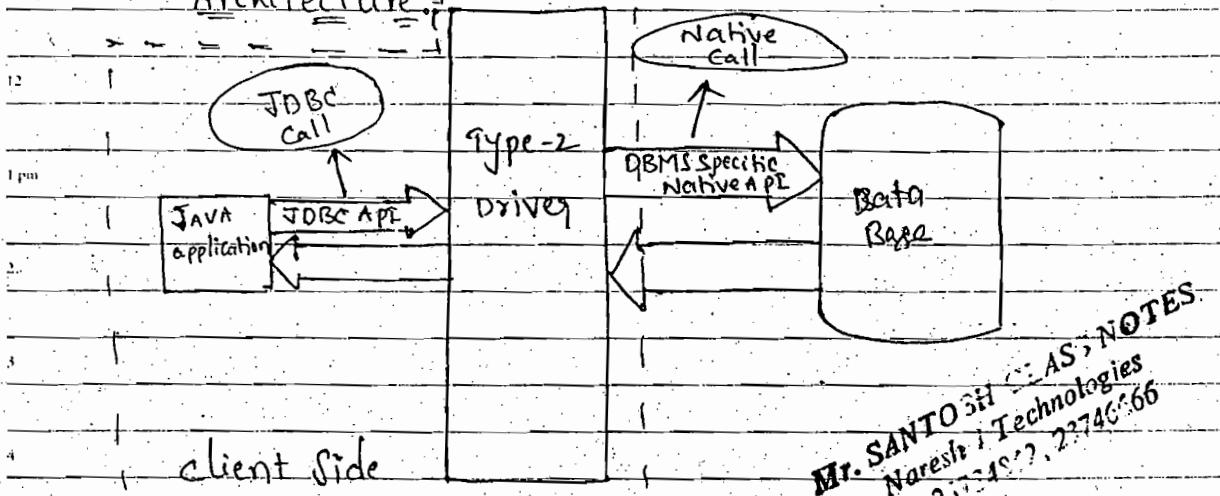
January 2005						
S	M	T	W	T	F	S
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	012-353, Week 02			

Type-II Drivers

8 am

- ⇒ The drivers which converts JDBC call into DB specific native call.
- ⇒ These drivers use JNI to convert the request into DB specific native call.

Architecture of



- ⇒ It follows 2-tier model (i.e. client and database server).
- ⇒ With these drivers the conversion from JDBC call to DB Native call is done in client system.

Disadvantages:

- ⇒ Driver implementation is specific to one DB.
- ⇒ (So if we want shift from one DB to other then we require to do some environmental changes in client No changes with our Java application).

February 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28					

013-352, Week 02

Priority _____

THU

13

JAN

2005

- 8am 2) Native library provided by DB vendor has to be installed into the client system.
- 9 3) Since native libraries r used, if there any bugs in the native libraries then it may result to crash the JVM:

10 Advantages :-

Mr. SANTOSH CLASS NOTES
Narender Technologies
Ph: 23734812, 23746666

- 11 1) These drivers are more faster than the other type of drivers.

- 1pm 2) USES :- It is used with **Server Side applications** i.e an application executing in the server to provide services to the client.

2 Note :-

- 3 It is recommended to the user only after properly testing the driver and if it is 100% bug free.

Native API - part Java implementation
(part java and part Native Implementation)
examples of these drivers,

- 4 → **oci8** driver from oracle
which can interact with oracle DB.
oci8 driver is JDBC compliant driver which takes the JDBC request convert into oracle native call and interact with the oracle DB.

- 5 Note :- this type of driver can be implemented by any vendor need not to be the DB vendor.
But in most of the cases it is implemented by DB Vendors.

FRI Priority

14

JAN

2005

January 2005

S	M	T	W	T	F	S
30	31				1	
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

014-351, Week 02

⇒ weblogicoci: driver from DEA Weblogic.

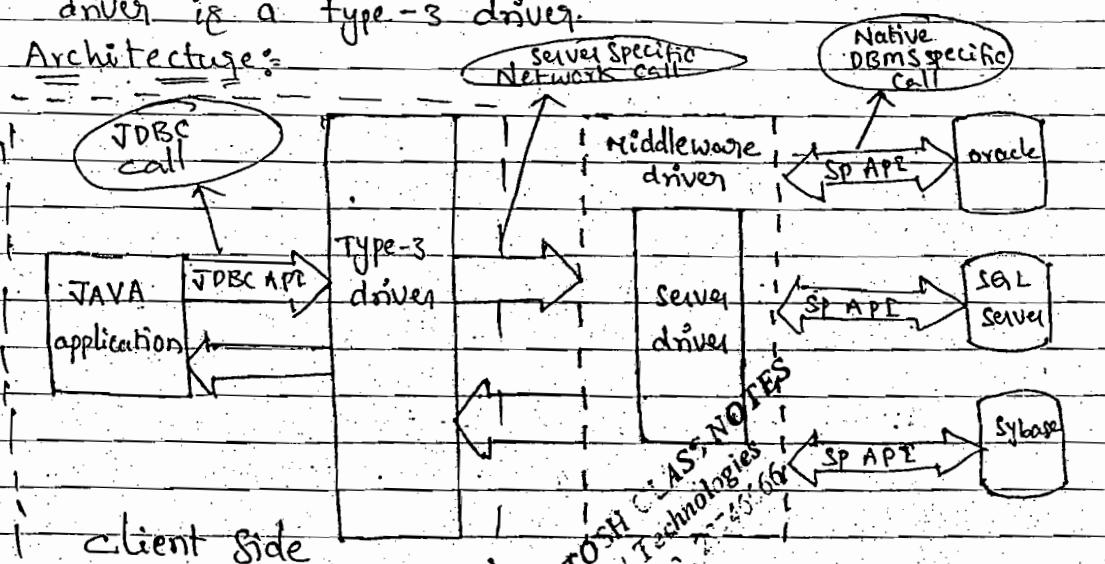
which can interact with Oracle DB.

* Type 1 and Type 2 both drivers 2-tier models.

* Type-3 Driver:

The JDBC driver which converts the JDBC call in server specific Network call and interacts with the server driver is a type-3 driver.

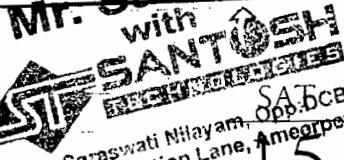
Architecture:



Advantages:

- 1) Follows 3-tier model.
- 2) Database details are not required to be distributed to all the Java applications (clients) like database address, DB username, DB password.
- 3) Resources can be managed perfectly i.e. Middleware takes the responsibility to manage.

now
Mr. Santosh



#2nd Floor, Saraswati Nilayam, Opp.BCB Bank
SR Nagar Police Station Lane, Ameerpet, Hyd.

15

JAN

2005

Priority						
S	M	T	W	T	F	S
	1	2	3	4	5	6
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28					

015,350, Week 02

the resources like the sessions to DB.

- 8 am 1) In Java applications (i.e. client) only type-3 driver implementations classes are required to be installed.
2) Any changes in database configurations doesn't require to be informed to the clients.
10 3) Single driver implementation can be used to interact with any DB.

42 disadv:

Mr. SANTOSH LAS NO:
Nareddy Technologies
ph: 23734542, 23746666

- 1 pm 1) The number of calls are increased compared to the other drivers.
2) Requires Middleware Server which may increase the cost of the application.

Use & :-

- can be used with client side executable applications. Since this driver is implemented purely in Java auto installation may not be a problem.
→ It is recommended to be used with server side applications since the resources can be managed properly. performance and availability of the application is increased.
Support for distributed transactions.

Name for this driver is call **Network protocol - pure Java Driver** (or Net protocol driver).

SUN Priority

16

JAN

2005

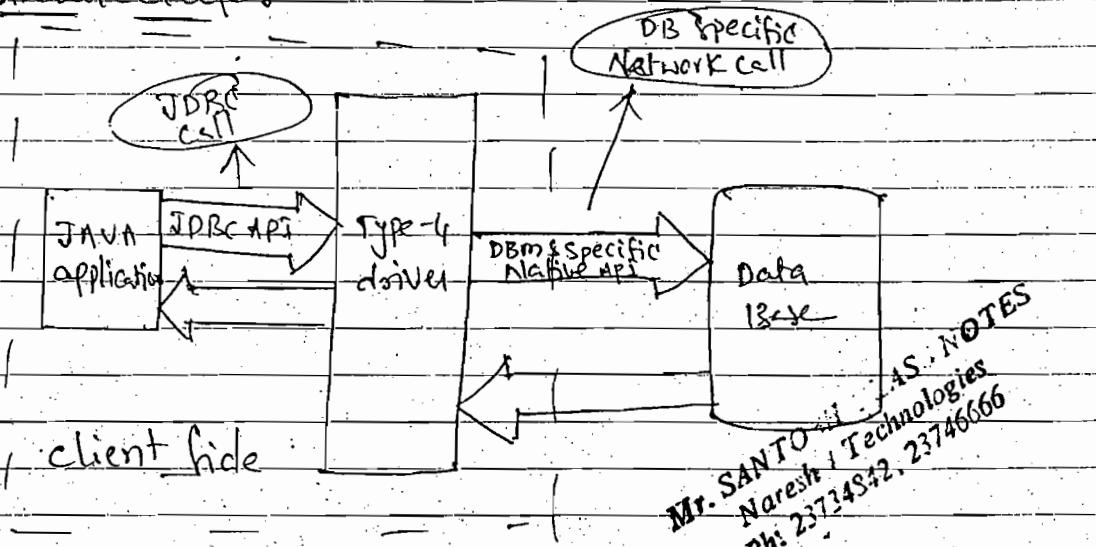
January 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

01631 Work 02

Type-4 Driver:

- ⇒ Driver which converts JDBC calls into database specific network call is a Type-4 driver.
- ⇒ It follows 2-tier model.

Architecture:



Adv:

- 1) This type of driver is pure Java driver.
- 2) Uses database specific network protocol and because of this it is safe to transfer the data compared to other formats since the DB specific protocols are not properly documented.
- 3) Compared to type-3 driver it costs less and the speed of operation is faster.

February 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28					

01^2-3^8, Week 03

MON

17

JAN

2005

Driver:

⇒ Compared to type-3 driver

Resources may not be able to be managed properly.

⇒ Compared to Type-2 driver if it is slower.

⇒ This type of driver is implemented specific to the DB.

Native protocol - pure Java driver.

Examples:

⇒ Thin Driver from oracle for interacting with Oracle DB

⇒ JdbcDriver from BEA weblogic for interacting with SQL Server.

Examples for Type-3 driver:

⇒ IDS driver.

⇒ RMI driver from BEA weblogic

pool driver and JTS Driver from BEA weblogic.

Uses:

⇒ It is preferable to be used with client side and server side applications.

⇒ JDBC Drivers can follow 2-tier model or 3-Tier model

TUE

Priority

18

JAN

2005

January 2005						
S	M	T	W	T	F	S
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29				

018-347, Week 03

- ⇒ Type-1 and Type-3 are database independent.
- ⇒ Type-2 and Type-4 are implementation specific DB.
- ⇒ Type-3 follows 3-tier model and rest of the other type of drivers follows 2-tier model.
- ⇒ Type-1 and Type-2 uses native library whereas Type-3 and Type-4 are pure java drivers.

JDBC Specifications:

- ⇒ It is a Service API

(i.e. This API allows Java applications to connect to the DB and execute some SQL queries in a DB independent format)

Steps to establish the connection to the DB and execute SQL queries:

- 1) Register the driver implementation class instance with `java.sql.DriverManager`

Note:

JDBC API has **2 parts**

- ⇒ JDBC Core API (`java.sql` package)

- ⇒ JDBC Ext API (`javax.sql` package)

- 2) get the connection from Driver manager using **get Connection method**.

- 3) Using the connection get Statement which can be used to execute the queries and get the results.

Mr. SANTOSH AS, NOTES
Narender Technologies
Ph: 23-34562, 27746666



JDBC Example 1:

Using Type-I driver

```
import java.sql.*; public class JDBCEx1 { public static void main(String s[]) throws Exception {
```

Step 1:

```
DriverManager.registerDriver(new Sun.jdbc.odbc.JdbcOdbcDriver());
```

/ where Sun.jdbc.odbc.JdbcOdbcDriver is an implementation
class of java.sql.Driver */*

/ given by the JDBC Driver provider (here it is from Sun) */*

Step 2: Connection

Connection

```
con = DriverManager.getConnection("jdbc:odbc:mydsn",  
        "scott", "tiger");
```

/* where the first argument of the above method is the JDBC URL

JDBC URL Format:-

jdb: <subprotocol>: driver specific info>

<Sub protocol> is given by the JDBC driver provider

<driver specific info> is used by the driver implementation to locate the DB or Middleware Server.

Second argument → DB user name.

3rd " → D13 password

Example :-

JDBC · ODBC · mySQL

where myden is Data source name which is configured in ODBC driver with the DB default like DB to which

THU
20
JAN
2005

Priority

January 2005						
S	M	T	W	T	F	S
30	31				1	
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

020-345, Week 03

Feb						
S	M	T	W	T	F	S
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26

021-3

it was interact, DB location) */

Step 3:

```
Statement st = con.createStatement();
st.executeUpdate("create table mytab(first Varchar(2
(20) .col2 number);
```

System.out.println("Table created");

3 //main

3 // class.

=

To run JDBC application

1> Create mydbcn configuration in ODBC

2> Control panel → Administrative tools → Datasource (ODBC)

3> Add Button

4> Select the required driver implementation
in this case to interact with oracle select

Microsoft Oracle ODBC for Oracle
(or)

Oracle ODBC driver

-OK-

5> give the DB details like

Database name = any unique name which
is used in the application

Service name :- is required if the oracle DB is
running in the different system.

(in this case ODBC library and oracle ODBC library has
to be installed in the client system)

OK-

6> finish

Priority						
S	M	T	W	T	F	S
1	2	3	4	5		
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28					

February 2005
021 311, Week 03

FRI

21

JAN

2005

Java-Sql driver

- is an interface given under the JDBC Specification
- has to be implemented by the JDBC's driver provider.

Mr. SANTOSH
Name : Mr. Technologies
Ph: 23734512, 23743366

Methods:

Boolean accept (URL s)

- checks whether the JDBC URL passed as an argument is valid or not (return true if it is valid).

java-Sql Connection connect (....)

- Takes the DB and driver properties as an argument
- prepares a connection type of object and returns

java-Sql Connection:

is an interface given under JDBC specification and implemented by JDBC driver provider.

This type of instance represents the session established to the DB.

java-Sql Driver manager:

is an non abstract class given under JDBC Specification, but can't be initialized outside the Driver manager class. i.e this is following

= Singleton design pattern.

is a factory class which is used to get a connection. (i.e to establish the session to DB and get the reference in the form of connection type of object).

SAT Priority

22

JAN

2005

Mr. SANTOSH CLASS NOTES
Naresh Technologies
Ph: 23734842, 23746666

January 2005						
S	M	T	W	T	F	S
30	31				1	
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

022-343, Week 03

Feb		
S	M	T
1	2	3
4	5	6
11	12	13
18	19	20
25	26	27
28	29	30

To get a connection using driver manager

- 1> The driver implement class has to be used to establish the session DB and get the connection has to be registered to DriverManager in Register Driver Method.

- 2> find whether this driver is already registered or not if it is not registered.
- 2> create an instance of driver into class (which holds driver type instance) driver class instance and driver name.
- 3> Store the instance (ie driver information) into the class variable (vector type).

when get Connection method is called

- ① gets driver information instances added into class variable (vector)

Note: driver manager can be registered with any number of drivers.

- ② in a loop (i.e the loop is iterated for each of the driver information instance in vector), it uses the driver information instance in the order in which they are registered).

Mr. SANTOSH
Naresh Technologies
Ph: 23734842, 23746666

Get the driverinformation instance & extract Driver instance from it

- 3> call accept URL method on the Driver instance passing the URL (which has been passed by java application as a first argument to get connection). If this method returns false then go to the next iteration.

Today Anniversary

Birthday

now
Mr. Santosh



SUN

Mr. SANTOSH #2nd Floor, Savitribai Nilayam, Opp: DCB Bank
Naresht Technologies
Ph: 23734842, 23746666

23

JAN
2005

Priority						
1	2	3	4	5	6	7
8	9	10	11	12		
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28					
29	30	31	1	2	3	4
Week 03						

ii) call connect method on the driver instance (Driver instance which has accepted the URL) get the connection and return it to the Java application.

If all the Drivers registered with the DriverManager rejects the URL, then it throws an SQL Exception saying No suitable driver found.

Note :- Register Driver and get Connection methods are synchronized.

Connection:

→ It is an interface given under JDBC Specification will be implemented by the JDBC Driver providers (3rd party vendor) and an instance of this type will represent the session established to the Datastore (data source). i.e this type of object represents the Session to DB to JDBC format.
→ This provides some methods to get Statement, prepared Statement and Callable Statement.

Type 1:

Mr. SANTOSH CLASS, NOTES
Naresht Technologies
Ph: 23734842, 23746666

Driver class: Sun.jdbc.Odbc.JdbcOdbcDriver

JDBC URL: jdbc:odbc:<db name>

Type 2: Driver is used to get Connection/session to DB-oci8 driver from oracle Corporation this driver is used to interact with oracle.

Driver class: Oracle.jdbc.driver.OracleDriver

JDBC URL: jdbc:oracle:oci8:@<database name> JDBC 1.0

class path: <Oracle name>\orasi\jdbc\lib\classes111.zip

or classes12.zip

~ JDBC 2.0

MON
24
JAN
2005

Priorities

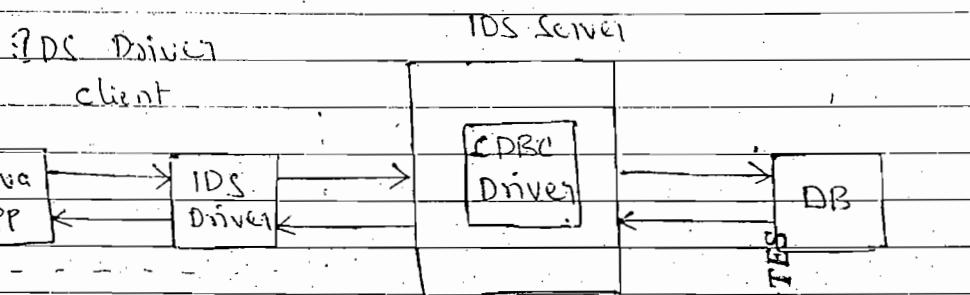
January 2005						
S	M	T	W	T	F	S
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	02-03-04, Week 01			

February						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	02-03-04, Week 01				05-06-07	

path = <oracle home>\orai8i\bin.

Where <database name> is the Oracle listener name configured in client system. This can be found in tnsnames.ora file.

Type 3:



Driver class: idcsrvl.IDS.Driver

JDBC URL: jdbc:idcs://<host name>:<port>
dsn = '<clsn name>' & uid = '<db username>
& pwd = '<db password>'

Here <host name> → IDS Server host

<port> → IDS Server port (default 12)

Mr. SANTOSH NARESH TATHAWADE
Ph: 27345223/22345666

Type 4:

This driver from Oracle Corporation

Driver class: oracle.jdbc.driver.OracleDriver

JDBC URL: jdbc:oracle:thin:@<host name>:<port no>
<sid>

<host name> → DB Server host name

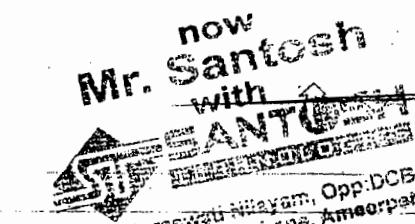
<port> → DB Server port number (default 1521)

<sid> → Oracle Service id (Oracle instance name)

February 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28					

Priority:

025-340, Week 04



#2nd Floor, Saraswati Niwasam, Opp:DCB Bank,
SR Nagar Police Station Lane, Ameerpet, Hyd.

TUE

25

JAN

2005

classpath:<oracle-home>\ora8i\jdbc\lib\classes111.zip

8am

(or)

classes 12.zip

OTES

Mr. SANTOSH
Nareesh Technologies
Ph: 23734842, 23746666

Statement

- It is an interface which is implemented by driverprovider and an instance of this type allows up to execute one (or) more SQL statements and get same results
- Statement is associated with one open Connection i.e., Statement sits on top of Connection and allows up to execute (i.e., pass) some SQL queries to DB

Methods

- 1> **executeUpdate**: → Used to execute some SQL Queries which does not results with the table data i.e., like select queries
If such a query are given it raises SQL exception taking an argument of SQL query (i.e., type String) and return int value (i.e., the number of rows affected by the query)

Mr. SANTOSH CLAS NOTES
Nareesh Technologies
Ph: 23734842, 23746666

- 2> **executeQuery**:

- Which is used to execute some SQL queries which return some table data
- Argument is String returns ResultSet

- 3> **Execute**:

- If can take any SQL query

Birthday / Anniversary

WED Priority

26

JAN

2005

January 2005						
S	M	T	W	T	F	S
31						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

026-339, Week 01

S	M
6	7
13	14
20	21
27	28

02

Returning boolean IMPORATNT

if it returns true then query has generated Resultset

if it returns false then the query has resulted with number of rows effected

4) getResultSet()

→ returning the Resultset

This is used with execute method

if it returns true

5) getUpdateCount

→ returning int (i.e., the number of rows effected)

This has to be used with execute method.

if it returns false

// Example to insert a record into DB and how to
Configure and use of ojdbc driver

```
import java.sql.*;  
public class JDBCEx {  
{
```

```
    public static void main (String s[]) throws Exception  
    {
```

```
        DriverManager.registerDriver (new oracle.jdbc.Oracle.  
        Driver());
```

```
        Connection con = DriverManager.getConnection
```

```
            ("jdbc:oracle:thin:@oci8:@nit", "scott", "tiger");
```

```
        Statement st = con.createStatement();
```

```
        int i = st.executeUpdate ("insert into mytab values  
            ('abc', 100)");
```

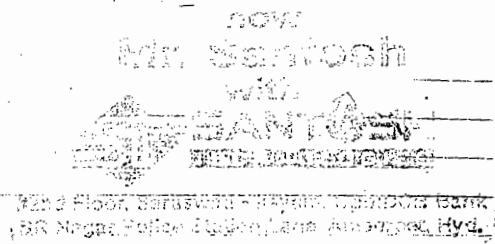
Birthday / Anniversary

Birthday

MR. SANTOSH KUMAR
NARESH TECHNOLOGIES
phi 23734842, 23740696

February 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28					

Purity



THU

27

JAN

2005

System.out.println ("number of rows affected
by the query");

{ //main
} //class

10 To find <DB name> to be used with oracle driver or
<oracle home>\ora8i\network\ADMIN\tnsnames.ora

11

12

1pm

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

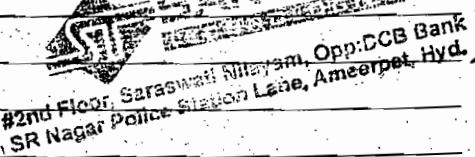
28

29

30

31

Birthday, Anniversary



February 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28					

Priority

MON

31

JAN

2005

031-334, Week 05

20/07/2005

Resultset :-

* Represents the selected tabular content provides some methods which are used to retrieve the selected data.

* It follows iterator design pattern.

Note:- java.util.iterator is different from Resultset

To get the values from Resultset :-

1) Move the Resultset Cursor to the row whose the data has to retrieved.

2) This can be done with help methods like

next()

previous()

relative(int)

absolute(int)

first()

last()

This methods are introduced in JDBC 2.0

Mr. SANTOSH CLASS NOTES
Naresh I Technologies
Ph: 23734542, 23746666

The above methods except next() are new from JDBC 2.0 and can be used only if the resultset is scrollable type.

Note:-

When a resultset is obtained the cursor position initially will be at before first.

2) Use column index (starts from 1) or the column name to get the value.

TUE

Priority _____

1

FEB

2005

February 2005						
S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28					

032-333, Week 05

S	M
3	4
10	11
17	18
24	25
31	

063.

Xxx getXxx(int columnindex)

Xxx getXxx(String columnName)

where Xxx is int, double, float, String, Object, Date...

Depending upon the column type we can use the respective getXXX method.

ResultSet gives the data not only in Simple Text format instead it performs Type Conversion and returns the value.

/*

Example to show how to use ResultSet,

* to get data.

```
import java.sql.*;  
public class ResultSetEx {
```

Mr. SANTOSH CLASS NOTES
Naresh Technologies
Phi 23734842, 23746666

```
    public static void main(String s[]) throws Exception,  
        Class.forName("oracle.jdbc.driver.OracleDriver");
```

/*

Note: Every driver implementation class provided with one static block in the class which registers the class instance with driver manager

so we can load the driver implementation class so that static block is executed and the driver is registered into driver manager

⇒ with some JVM's on unix and solaris, they are not executing the static blocks when client is loaded instead it is being executed when the first instance of the class is created

Birthday / Anniversary

Birt

April 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Priority _____

FRI

4

MAR

2005

063-302, Week 09

Because of this it is recommended to use
Class.forName("<driver implementation class>").newInstance();
*/

9 Connection con = DriverManager.getConnection("jdbc:oracle:oci8:
@nint", "scott", "tiger");

10 Statement st = con.createStatement();

11 ResultSet rs = st.executeQuery("Select * from bank");

12 /* Note : Now the ResultSet cursor points to before
first

1pm So before using ~~getXXX~~ methods the cursor has
to be moved to the valid record.

*/

2 while (rs.next())

MR. SANTOSH KAS, NOTES
Naresh I Technologies
Ph: 23734842, 23746666

3 // next method returns boolean : whether the resultSet cursor
after the requested operation is pointing to a valid
record or not //

4 int accno = rs.getInt(1);

5 // to get the first column value of the record to which the
current is pointer if pointing //

String name = rs.getString(2);

6 double bal = rs.getDouble("bal");

System.out.println(accno + " " + name + " " + bal);

7 } while (

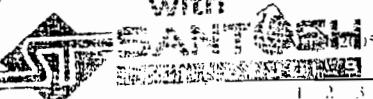
8 // the above loop is breached after reading the last
record i.e., when we try to move the pointer from
last record to the next //

9 con.close();

10 } // main

Birthday / Anniversary

now
Mr. Santosh



#2nd Floor, Sparshwati Nilayam, Opp: DCB Bank
SR Nagar Police Station, Laddu Jhimpeta, Hyd. 500085
20 21 22 23 24 25 26
27 28 29 30 31
06-4-2005, Week 09

SAT

Priority

5

MAR

2005

R. Vamgi Krishna

Result Set is associated with one open Statement i.e if Statement is closed when the Result set prepared by the Statement is automatically closed.

⇒ only one Result Set can be opened at a time of one Statement.

on a connection we can maximum obtain open ResultSet's equal to the number of statements created on it.

Note: from JDBC 3.0 we have an option where we can set the configurations to open Result Set with one Statement.

1/ My SQL Editor

1/ This application showing the importance of execute method

with Result Set getXXX method the column index used is the resulted table column index not the actual index Table.

MR. SANTOSH
Narenji Technologies
23734842, 23746600

Birthday / Anniversary

Birth

April 2005						
S	M	T	W	T	F	S
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
065-500, Week 09						

Priority

SUN

6

MAR

2005

Shreetha Vahana Reddy, Degan

// My SQL Editor

/-- This application shows the importance of execute method.

```
import java.sql.*;
import java.io.*;
public class MySQL_Editor {
```

Mr. SANTOSH CLAS, NOTES
Naresh I Technologies
Ph: 23734842, 23746666

```
    public static void main(String s[]) throws Exception {
```

```
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection("jdbc:oracle:
oci8:@nit", "scott", "tiger");
        Statement st = con.createStatement();
        BufferedReader br = new BufferedReader(System.in);
```

```
        while(true) {
```

```
            System.out.println("SQL>");
```

```
            String query = br.readLine();
```

```
            boolean flag = st.execute(query);
```

```
            if(flag)
```

```
{}
```

```
            ResultSet rs = st.getResultSet();
```

```
            printResultSet(rs);
```

```
        } //if
```

```
        else { System.out.println(st.getUpdateCount() + " number of records has
been effected"); }
```

```
} //else
```

```
} //while
```

```
con.close();
```

```
} //main
```

MON

Priority

7

MAR

2005

March 2005

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

066-299, Week 10

S	M
3	4
10	11
17	18
24	25

067

8 am private static void printResultSet (ResultSet) throws Exception
{
 Statement rs = con.createStatement ();
 ResultSetMetaData rsmd = rs.getMetaData ();
 int count = rsmd.getColumnCount ();

 while (rs.next ())
 {
 for (int i = 1; i <= count; i++)
 {
 System.out.print (rs.getString (i) + " ");
 }
 System.out.println ();
 }
 rs.close ();
}
4. print ResultSet
5. // class

Mr. SANTOSH CLAS^N NOTES
Naresh Technologies
Ph: 23734842, 23740666

Birthday / Anniversary

Birthday

April 2005						
S	M	T	W	T	F	S
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
067-298. Week 10						

Priority

TUE

8

MAR
2005

Agenda for 8th Mar 2005

8am Statement :-

when a statement is used to execute a query, then it

submits the query to DBMS where it verifies the query,
compiles it and prepares the execution plane before it
executes the query.

if we want to execute the same query with different values
for more than one time like

insert into myemp values (100, 'emp1')

insert into myemp values(101, 'emp2')

MR. SANTOSH CLAS NOTES
Naresh I Technologies
ph: 23734842, 23746666

Then in this case if they are individually executed, all the
above 4 steps are performed by the DBMS for both the queries,
but the first 3 steps performed are same i.e. the execution
plane prepared for both the queries is same. So in this
case we want the 3 steps not be repeated for 2nd query.

To avoid the same step repeatedly alone we can go for
precompiled queries (an option supported by most of the DB).

where DBMS provides a unique identity for the execution
plane prepared which can be used by the client while
executing the same query for the next time.

By this DBMS locates the existing execution plane, substitutes
the values and executes the query.

WED

Priority

9

MAR

2005

now
Mr. Santosh
with



#2nd Floor, Saraswati 21 22 23 24 25 26
SR Nagar Police Station Lane, Ameerpet, Hyd.
068-297 Week 10

In JDBC Specification using **Prepared Statement** we can request DBMS to prepare an execution plane and then use this Prepared Statement to execute the plane for any number of times.

i.e Prepared Statement object represents one precompiled query

⇒ Java. Sql.PreparedStatement

* is an interface which extends java. Sql. Statement

to get PreparedStatement object

use

PreparedStatement prepareStatement (String) method declared in **Connection**.

Here while getting the PreparedStatement object we have to pass the query which will be compiled and can be executed with different values using the PreparedStatement object obtained.

prepareStatement (String query)

where query can be included with ?

which will be filled before the query is executed

These ? are known as **indexed parameters**

Example :-

insert into myemp values (?,?)

Mr. SANTOSH CLASS NOTES
Naresh I Technologies
Ph: 23734842, 23746666

Priority

April 2005						
S	M	T	W	T	F	S
			1	2		
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

069-296, Week 10

THU

10

MAR

2005

Prepared Statement

void setXXX(int,XXX)

Like

void setString(int, String)

void setDouble(int, double) . . .

int executeUpdate()

ResultSet executeQuery()

boolean execute()

Mr. SANTOSH C-S AS, NOTES
Naresh Technologies
Ph: 93734842, 2374666

1. get a PreparedStatement object passing a query with indexed parameters.

2. Use setXXX method on PreparedStatement object to fill the parameters (in query).

3. executes the query.

4. Use steps 2 & 3 for execution of query more than once.

PreparedStatement Example:

```
import java.sql.*;
```

```
public class PreparedStatement Extends I
```

```
public static void main (String s[]) throws Exception
```

// to register the driver we can set the driver class into the system properties.

FRI Priority

11

MAR
2005

Vijay Krishna prasad

March 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

070-295, Week 10

S	M	T	W
3	4	5	6
10	11	12	13
17	18	19	20
24	25	26	27

071-294

8 am System::SetProperty ("jdbc.driver", "oracle.jdbc.driver.OracleDriver");

Connection con = DriverManager.getConnection("jdbc:oracle:@nit", "scott", "tiger");

PreparedStatement ps = con.prepareStatement("insert into myemp values (?,?)");

ps.setInt(1, 101);

ps.setString(2, "emp1");

int i = ps.executeUpdate();

System.out.println("i:" + i);

ps.setInt(1, 102);

ps.setString(2, "emp2");

int j = ps.executeUpdate();

s.o.p("j:" + j);

con.close();

} //main

} // class:

10/15 → The number of statements which can be opened on a connection is limited, depends on the driver which we are using.

And some drivers may allow only one statement to be opened at a time.

→ The query which has to be compiled and which is given as an argument to prepareStatement method can contain indexed parameters (i.e. ?) but ? cannot be placed in the place of SQL keywords, table names, column names, etc.

Mr. SANTOSH CLASS NOTES
Naresh 1 Technologies
Ph: 23734842, 23746666

Birthday / At

April 2005						
S	M	T	W	T	F	S
1	2					
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

071-294, Week 10

Priority

SAT

12

MAR

2005

S

We can have '0' or more parameters within the query.

Example :-

If we want to execute "Select * from myemp" for more than one line we can go for PreparedStatement with parameters.

Advantages :-

- One time compilation and the compiled query can be executed from any number of times.
- Dealing with datatypes is easy compared to the query written for the statement.
- Some SQL99 data types may not be able to be inserted through the statement (i.e. we have to take the help of some procedures or function). But using them with Prepared Statement is easy.

Example :- BLOB, CLOB

Dif adv or limitations of Prepared Statement :-

→ Prepared Statement object can represent only one query which can be executed for any number of times but if we want to execute different query we have to go for one more Prepared Statement.

If we want to execute multiple queries then it is recommended to go for Statement.

But if the Multiple query queries has to execute for more number of times then we have to go for a Prepared Statement object for each of the query.

When we deal with some complex datatypes then it is recommended to go for PS.

SUN
13
MAR
2005

Priority

March 2005	
S	F
6	7
13	14
20	21
27	28
3	4
10	11
17	18
24	25
31	
072-293, Week 10	

S	M	T	Δp
3	4	5	
10	11	12	
17	18	19	
24	25	26	
			073-29.

(even if it is required to be executed only one time)

// Example Using PreparedStatement for select queries and
// uses type-3 driver (JDBC Driver)

```
import java.sql.*;
public class Prepared Statement Ex {
    public static void main (String s[]) throws Exception {
        Connection con = DriverManager.getConnection ("jdbc:ids:  
//localhost:12/conn?dbname='mydsn' & uid='scott'  
& pwd='tiger'");
        PreparedStatement ps = con.prepareStatement ("select * from  
myemp where empno=?");
        ps.setInt (1, Integer.parseInt (s[0]));
        ResultSet rs = ps.executeQuery ();
        if (rs.next ()) {
            System.out.println ("Emp Found in Name :" + rs.getString ());
        } else {
            ps.setInt (1, Integer.parseInt (s[1]));
            if (rs.next ()) {
                System.out.println ("Emp Found : Name :" + rs.getString (2));
            } else {
                System.out.println ("No Record Found");
            }
        }
        con.close ();
    }
}
```

iv Anniversary

April 2005						
S	M	T	W	T	F	S
1	2					
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

073-292, Week 11

Priority

WHO R U?

WHAT NEXT?

MON

14

MAR

2005

To use TDS Driver:

8am

⇒ In the server we have to install TDS driver and ODBC library has to be installed into the system where TDS Server is installed.

10

⇒ Create System DSN (i.e. ODBC Data Source).

11

This has to be created in the system where TDS Server is running.

12

Set classpath = C:\TDSserver\classes\tdk14_dry.jar (in client)

1pm

java -Djdbc.driver=ids.sql.TDSDriver Prepared Statement

NOTES
Mr. SANTOSH CLAS
Naresh I Technologies
23734942, 2-4-65

2

procedure:

⇒ Holds one or more script

3

We can perform some small logic within a procedure.

4

⇒ Is the other approach of presenting the Precompiled query by the DB.

When a procedure is given to the DB engine

⇒ Reads the procedure

⇒ Verify the procedure (including the query in the procedure)

⇒ Compile the procedure and prepare an execution plan for the procedure

⇒ Attach the execution plan to the procedure and store it into the DB.

5

To prepare

Birthday / Anniversary

TUE

Priority

15

MAR

2005

March 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

S	M
3	4
10	11
17	18
24	25

07:1291, Week: 11

075-24

to prepare a procedure:

Create or replace procedure <procedure name> (~~<arg1>~~)

[IN|OUT] arg-type) as

BEGIN

<SQL queries>

END;

Mr. SANTOSH CLAS; NO
Naresht Technologies
Phi 23734842, 23746666

now
Mr. Santosh

to use/execute the procedure from java application

1) Get a Callable Statement type of object using

~~Callable Statement~~ as interface

prepareCall method of Connection

Callable Statement

CS = con.prepareCall(" { call <procedure-name> (?...)}");

2) Set all the IN and INOUT parameters using
setXXX method.

void setXXX (int index, XXX value)

in JDBC 3.0 we can set the parameter values using
the parameters names also.

3) Register OUT parameters using
registerOutParameter (int index, <Type>)

4) Execute the procedure using execute () method

5) If we have any out parameters we can get the values
using getXXX methods.

Priority						
S.	M.	W.	T.	F.	S.	
		1	2			
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

07-03-2000, Week 11

ప్రాంగం మనస్సు నా మనస్సు వీకారి
వీకారి స్కాస్సు నా మనియ అభిమంద

WED

16

MAR

2005

CallableStatement is an interface which has some methods, allows us to execute Stored Procedures and Functions.

Procedure 1:

create or replace procedure addEmp(eno, number, ename, vencher) as

BEGIN

insert into my.emp values(eno, ename);

END;

{ call addEmp(?,?) }

Mr. SANTOSH CLAS, NOT
Naresh Technologies
Phi 23734842, 23746066

// Callable Statement Example

// executing a procedure with all IN parameters

```
import java.sql.*;
import java.util.*;
public class CallableStatementEx1
{
    public static void main (String s[]) throws Exception
    {
        Class.forName ("oracle.jdbc.driver.OracleDriver").newInstance();
        properties p = new Properties();
        // Properties is one of implementation of Map.
        // Where it can hold name-value pairs.
        p.put ("user", "scott");
        p.put ("password", "tiger");
        Connection con = DriverManager.getConnection ("jdbc:oracle:
thin:@localhost:1521:xit", p);
        CallableStatement cs = con.prepareCall ("{call addEmp(?,?)}");
    }
}
```

THU

Priority

17

MAR

2005

March 2005

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

076-289, Week 11

Sam
9
10
11
12

```
cs.setInt(1, Integer.parseInt(s[0]));
cs.setString(2, s[1]);
cs.execute();
cs.setInt(1, Integer.parseInt(s[0]) + 1);
cs.setString(2, s[1]);
cs.execute();
con.close();
} // main
} // class
```

1 pm ~~Set classpath to Oracle~~
Procedure 2:
Create or replace procedure getEmpName(eno number,
ename OUT varchar2) as
BEGIN
select name into ename from myemp where empno=eno;
END;

Note :-

The argument names should not match with the DB table
name or table column names.

6 // Executing the procedure with one In and one Out parameter.

```
import java.sql.*;
import java.util.*;
public class CBSE {
    public static void main (String s[]) throws Exception
    {
```

Birthday / Anniversary

April 2005						
S	M	T	W	T	F	S
			1	2		
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Priority

GOD

FRI

18

MAR

2005

077-288, Week 11

```

8 am Class.forName("oracle.jdbc.driver.OracleDriver");
Properties p = new Properties();
p.put("user", "scott");
p.put("password", "tiger");
Connection con = DriverManager.getConnection("jdbc:oracle:
        thin:@localhost:1521:xit", p);
CallableStatement cs = con.prepareCall("{call getEmpName(?,?)}");
cs.setInt(1, Integer.parseInt(sr[1]));
cs.registerOutParameter(2, Types.LONGVARCHAR);
cs.execute();
String name = cs.getString(2);
System.out.println("Name :" + name);
con.close();
} //main
} // class

```

Procedure 3 :-

Mr. SANTOSH CLAS NOTES
 Naresh Technologies
 Ph: 23734542, 23746666

Birthday / Anniversary

SAT Priority

19

MAR

2005

March 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

078-287, Week 11

Function :-

8 am

create or replace function <function name> (<arg1> <arg type>, ...)
returns <type> as <variable declaration>

begin

<Statements>

8

return <return value>;
end.

Function 1 :-

Create or replace function getEmpSal (eno number) returning number

number as

emp.Sal number;

begin

Select Sal into emp.Sal from my.emp where empno = eno;

return emp.Sal;

end;

To invoke the function from Java application using JDBC

CallableStatement cs = new CallableStatement ("&call
cs = con.prepareCall ("& call ?=<function name> (?...);");

Birthday / Anniversary

April 2005						
S	M	T	W	T	F	S
			1	2		
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

079-286, Week 11

Priorities

SUN

20

MAR

2005

// Example to invoke function

8am

```
import java.sql.*;  
import java.util.*;  
import java.io.*;
```

10

public class InvokeFunction1

11

```
{ public static void main(String s[]) throws Exception  
{  
    oracle.jdbc.driver.OracleDriver od=new oracle.jdbc.driver.  
        OracleDriver();
```

12

```
Properties p = new Properties();
```

1pm

```
p.load(new FileInputStream("myproperties.properties"));
```

2

Connection

3

```
con=od.connect("jdbc:oracle:thin:@localhost:1521:INIT", p);
```

4

Callable Statement

5

```
cs=con.prepareCall("{call ? := getEmpSal(?)}");
```

6

```
cs.registerOutParameter(1, Types.NUMBER);
```

7

```
cs.setInt(2, Integer.parseInt(s[0]));
```

8

```
cs.execute();
```

9

```
double sal=cs.getDouble(1);
```

10

```
s.o.p("Sal:" +sal);
```

11

```
con.close();
```

12

```
}
```

13

→ myproperties.properties → properties file

14

```
username: scott
```

15

```
password: tiger
```

Mr. SANTOSH CLAS NOTES
Naresh Technologies
Ph: 23734842, 23746666

MON Priority
21
MAR
2005

March 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

080-285, Week 12

Cursor

CURrent Set of Records

⇒ It is a temporary execution area where we can catch the results into the cursor.

To select one or more records with one or more columns we can explicitly start a cursor and let the query be executed in the cursor environment so that the cursor can be returned.

to create a cursor type :-

Create or replace package <package name>
TYPE <type name> is REF CURSOR;

end;

Example :-

Create or replace package mypack

as

TYPE mycursor is REF CURSOR;

end;

Mr. SANTOSH CLAS,
Naresh Technologies
Phi 23734842, 23746666

Create or replace function getEmpNos (amt number)

return mypack.mycursor

as

across mypack.mycursor;

begin

Priority						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	081-284, Week 12				

April 2005

TUE

22

MAR

2005

open emp for

```
select accno from myemp where sal > amt;
return accnos;
end;
```

// Example to cursor

```
import java.sql.*;
import java.util.*;
import java.io.*;
public class InvokeFunction2
```

```
{ public static void main (String s[]) throws Exception
```

```
{ Driver d=(Driver) Class.forName ("oracle.jdbc.driver.Oracle
Driver").newInstance ();
```

```
Properties p= new Properties();
```

```
p.load (new FileInputStream ("myproperties.properties"));
```

Connection

```
con=d.connect ("jdbc:oracle:thin:@localhost:1521:xit",p);
```

Callable Statement

```
cs=con.prepareCall ("{call ?:= getEmpNos (?)}");
```

```
cs.registerOutParameter (1, oracle.jdbc.driver.OracleTypes.CURSOR);
```

```
cs.setDouble (2, Double.parseDouble (s[0]));
```

```
cs.execute();
```

```
Object o=cs.getObject (1);
```

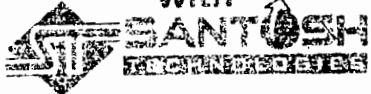
```
ResultSet rs= (ResultSet)o; while (rs.next ()) {
```

```
s.o.p ("rs.getInt ()");
```

MR. SANTOSH CLASS NOTES
Naresh IT Technologies
Ph: 23734842, 23746666

now
Mr. Santosh

with



WED

Priority

23

MAR

2005

#2nd Floor, Saraswati Nilayam, Oppo OCB Bank,
SR Nagar Police Station Lane, Amravati, Hyd. S

1	2	3	4	5
6	7	8	9	10
13	14	15	16	17
20	21	22	23	24
27	28	29	30	31

082-283, Week 12

→ Difference b/w function & procedure.

1) Function has a return value (i.e. it can return one value) whereas procedure doesn't have a return type but we can get the result using OUT parameters.

2) Function can be used within a DML query whereas procedure cannot be used.

If we want pack some DML queries in to set, compile it and prepare a execution plan for the statements so that we can execute the set of statements further then use a Procedure where as function is used for performing some small calculations which can be used within a DML query.

Mr. SANTOSH CLAS, NOTES
Nareste Technologies
Ph: 23734842, 23746666.

It is not recommended to use DML queries within the function.

Closing Connection or Statement's or Result Set.

The close method in Connection, Statement's and Result Set releases the resources opened to the DB.

Like resources may be the Native DB resource objects or N/W resources.

If we don't close resources then the DBMS closes the resources after the ideal time out.

Birthday / Anniversary

April 2005							Priority
S	M	T	W	T	F	S	
				1	2		
3	4	5	6	7	8	9	
10	11	12	13	14	15	16	
17	18	19	20	21	22	23	
24	25	26	27	28	29	30	

08:30-28:30, Week 12
2005

THU

24

MAR

2005

It is recommended to close the resource after completing the usage of resource.

so that it can be available for the other clients.

Note :-

If a Connection is closed then all the statements prepared on a connection will be automatically closed.

And if a statement is closed then the resultset if it is opened on a statement will be closed.

If we have a short term connection then we can directly close the connection which results in closing all the statements and the resultsets opened on those statements.

But if we want use connection for a long period then we can close the statement, resultset after completing the operation.

close()

This method may throw SQL Exception i.e., if the resource has been already closed or it is illegal to close resource.

MetaData

→ Is an information of the Information.

Two types :→ DatabaseMetaData

→ Result Set MetaData.

Mr. SANTOSH CLAS;
Naresh Technologies
Ph: 23734842, 23746666

DatabaseMetaData

→ Is used to get driver information like (driver name, driver version) database details, can find whether the JDBC driver being used was a support for the JDBC specification and can

FRI

Priority

25

MAR

2005

March 2005

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

984-281, Week 12

find whether a feature given in JDBC Specification is supported by the driver or not.

⇒ So an interface which has to be implemented by JDBC driver provider

to get DatabaseMetaData type of object

DatabaseMetaData dbmd = con.getMetaData();

where con is the Connection type of object.

// Example of using DatabaseMetaData

```
import java.sql.*;
import java.util.*;
import java.io.*;
public class DBMDEx
```

```
{
```

```
    public static void main(String s[]) throws Exception
```

```
    Driver d = (Driver) Class.forName(s[0]).newInstance();
    Properties p = new Properties();
    p.load(new FileInputStream("my properties.properties"));
    Connection con = d.connect(s[1], p);
    DatabaseMetaData dm = con.getMetaData();
    System.out.println("DataBase Name :" + dm.getDatabaseProductName());
    System.out.println(" DataBase version :" + dm.getDatabaseProductVersion());
    System.out.println(" Driver Name :" + dm.getDriverName());
    System.out.println(" Driver Version :" + dm.getDriverVersion());
    con.close();
}
```

Mr. SANTOSH CLAS, NOTES
Naresh Technologies
Ph. 23734842, 23746666

April 2005						
S	M	T	W	T	F	S
	1	2				
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Prin...

SAT

26

MAR

2005

085-280, Week 12
8/07/2005

Result Set MetaData :-

8am

⇒ Gives some information about the result obtained (i.e., ResultSet) like

the number of columns, each of the column type their column names, the table name from which the column has been selected

To get ResultSetMetaData object use

getMetadata() method on ResultSet object whose information is required.

Java.sql.ResultSetMetaData

⇒ int getColumnCount()

⇒ String getColumnName (int column_index)

⇒ ~~int~~ ^{int} getColumnType (int index)

⇒ String getColumnTypeName (int index)

getColumnType returns the type id as given in

java.sql.Type class static final fields.

Mr. SANTOSH CLASS NOTES
Naren 17 Technologies
Ph: 23736842, 23740566

// ResultSetMetaData Example

```
import java.sql.*;
```

```
import java.util.*;
```

```
public class RSMDEX,
```

```
{
```

```
    public static void main (String s[]) throws Exception
```

```
{
```

SUN Priority

27

MAR

2005

March 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

086-279, Week 12

String tablename = s[0];

Driver d = (Driver) Class.forName ("oracle.jdbc.driver.OracleDriver")

.new Instance ();

Properties p = new Properties ();

p.load(new java.io.FileInputStream ("my properties.properties"));

Connection con = d.connect ("jdbc:Oracle:thin:@localhost:1521: init", p);

Statement st = con.createStatement ();

ResultSet rs = st.executeQuery ("Select * FROM " + tablename);

print ResultSet (rs);

con.close ();

Mr. SANTOSH CLAS
Naresh Technologies
Ph: 23734842, 23746666

3//main

private static void ResultSet (ResultSet rs) throws Exception

{

ResultSetMetaData rsmd = rs.getMetaData ();

int colCount = rsmd.getColumnCount ();

for (int i=1; i<=colCount; i++)

{

s.o.println (rsmd.getColumnName (i) + "\t");

3//for

System.out.println ();

System.out.println ("-----");

while (rs.next ())

{

for (int i=1; i<=colCount; i++)

{

int type = rsmd.getColumnType (i); }

if (type == Types.INTEGER)

{

s.o.println (rs.getInt (i) + "\t");

Birthday Anniversary

Rej.

April 2005						
S	M	T	W	T	F	S
				1	2	
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
08:37:58, Week 13						

MON
28
MAR
2005

3/11/05

else {

System.out.println(rs.getString(i)+"\t");

}else

/* Note: we can use switch case and identify the column type
so that we can use the corresponding getXXX method to get the
column value */

System.out.println();

}while

}printResultSet

}class

JDBC P.O:

1> Scrollable ResultSet

It provides a feature to scroll within the results in any direction and for any number of times.

Before 2.0 we used to have only FORWARD ONLY resultset which allows to fetch the results only in one direction i.e. first to last (i.e. once if we reach to a record then we cannot come back.)

This type of resultset is required for performing some complex search operations from selected data or if we want to use the repeat Select data for more than one time within the application.

TUE Priority

29

MAR

2005

March 2005

S	M	T	W	T	F	S
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

088-277, Week 13

Example:

If we want to print the manager name of an emp
whose id is given by **the client**.

To support this type resultset new methods has been added in
ResultSet interface in JDBC 2.0

beforeFirst()

afterLast()

first()

last()

previous()

next() // available from JDBC 2.0

absolute(int row-number)

relative (int row-number)

isAfterLast()

isBeforeFirst()

If we want to use the above method (except **next()**)

the ResultSet obtained should be of type scrollable.

ResultSet Types

1) FORWARD-ONLY

2) Scroll Sensitive

3) Scroll Insensitive.

April 2005						
S	M	T	W	T	F	S
1	2					
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

089-276, Week 13

Priority

WED

30

MAR.

2005

the ResultSet type has to be set while creating the Statement or PreparedStatement ...

Using

Statement st = con.createStatement(int, int)

where the first int argument takes the ResultSet type we

Static final fields in ResultSet.

TYPE_FORWARD_ONLY (default)

TYPE_SCROLL_SENSITIVE

TYPE_SCROLL_INSENSITIVE

if scroll sensitive or insensitive type is set for the Statement then all the ResultSet objects obtained using the Statement will be scrollable type and can use all the above listed methods.

Note: For forward only resultset if we use the above listed methods (except next) it throws SQL Exception (saying unsupported operations for forward only resultset).

// Scrollable Result Set Example -

```
import java.sql.*;
import java.util.*;
import java.io.*;

public class SRSEx
```

```
{
```

```
    public static void main (String ss[]) throws Exception
```

```
{
```

```
        Driver d=(Driver)Class.forName("oracle.jdbc.driver.Oracle Driver")
```

```
.newInstance();
```

```
        Properties p = new Properties();
```

Mr. SANTOSH CLASS NOTE
Naresh Technologies
Ph: 23734842, 23746666

THU
31
MAR
2005

Priority

March 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	6
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

090-275, Week 13

S M T
1 2 3
8 9
15 16
22 23 24
29 30 31
091-2

p.load(new FileInputStream ("myproperties.properties"));

Connection

con = d.connect(pi.getProperty("myurl"), p);

Statement st = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_READ_ONLY);

ResultSet rs = st.executeQuery("select * from mybank");

while (rs.next())

{

s.o.p(rs.getString(1));

} //while

s.o.p("-----");

//now the Resultset cursor is at afterLast

s.o.p("AfterLast :" + rs.getAbsolute(1));

while (rs.previous())

{

s.o.p(rs.getString(1));

} //while

rs.first();

rs.next();

if (rs.absolute(3)) {

s.o.p("Record 3 Account No:" + rs.getString(1));

} //if

} //main

} //class

my properties file

myurl = jdbc:oracle:thin:@localhost:1521:nit

mydriver = oracle.jdbc.driver.OracleDriver

username = scott

password = tiger

Mr. SANTOSH CLAS, NOTES
Naresh Technologies
Ph: 23734842, 23746666

8am
9
10
11
12
1pm
2
3
4
5
6
7
8
Birthdays
Anniversary

May 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

091-274, Week 13

Priority

FRI

1

APR
2005

Scollable Result Set :

Can be Sensitive or InSensitive difference.

⇒ with Sensitive mode if any changes are done to the data in database (i.e., the data which was selected) will be available with the Resultset.

Where as with InSensitive mode the changes done to the data are not available into the Resultset.

We can use refreshRow() method on Resultset object if we want to get the new data.

⇒ If the data selected is not being used by multiple users and if there is no possibility to change the data after selecting the data and the time we read the data then it is recommended to use InSensitive mode.

Mr. SANTOSH CLAS. NOTES

Naresh Technologies

Ph: 23734842, 23746666

Programmatic Updates :

i.e., with Updatable ResultSet

⇒ Using ResultSet we can update, delete or insert a record (new in JDBC 2.0)

If the Resultset Concurrency mode is set to updatable i.e., CONCUR_UPDATABLE

then we can perform the above operations on Resultset.

⇒ To update the record using ResultSet.

→ move the resultset cursor to the record whose data has to be updated

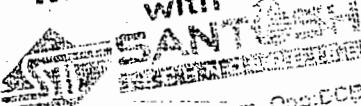
⇒ use updateXXX methods to update the values.

updateXXX (int col_index, XXX new-value)

updateXXX (String col-name, XXX new-value)

SAT Priority
2 APR 2005

now
Mr. Santosh



ESTABLISHED 1988

SR Nagar Police Station Lane, Ameerpet, Hyd.

2nd Floor Saraswati Vilayam, Opp ECIL Bank

April 2005						
S	M	T	W	T	F	S
1	2					
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

092-273, Week 13

S	V
1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30

⇒ call **updateRow()** method on the resultset object.

Note is:

The changes which are done to the resultset using update XXX methods are updated into the DB when update Row() method is called

Here the row which is pointed by the resultset cursor is updated.

// Example to update the empSal whose emp no is given as an argument.

```
import java.sql.*;
import java.util.*;
public class URSEEx
```

```
    {
        public static void main (String s[])
            throws Exception
```

```
        Properties p = new Properties();
```

```
        p.load (new java.io.InputStream ("my.properties.properties"));
```

```
        Driver d = (Driver) Class.forName (p.getProperty ("my.driver")).
```

```
            .newInstance ();
```

```
        Connection con = d.connect (p.getProperty ("my.url"));
```

```
        Statement st = con.createStatement (ResultSet.TYPE_SCROLL
```

```
            - SENSITIVE, ResultSet.CONCUR_UPDATABLE);
```

```
        Resultset rs = st.executeQuery ("Select empno, name, sal
```

```
            from myemp");
```

```
        while (rs.next ())
```

```
        {
```

```
            int empno = rs.getInt (1);
```

```
            if (empno == Integer.parseInt (s[0]))
```

Priority						
May 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

093-272, Week 13

SUN

3

APR

2005

```

1    double sal = rs.getDouble(3);
2    sal += sal * 2;
3    rs.updateDouble(3, sal);
4    rs.updateRow();
5    s.o.p("Emp found and sal changed");
6    } //if
7    } //while
8    rs.beforeFirst();
9    while (rs.next())
10   {
11     s.o.print(rs.getInt(1) + "\t");
12     s.o.print(rs.getString(2) + "\t");
13     s.o.p(rs.getDouble(3));
14   } //while
15   con.close();
16 } //main
17 } //class

```

Mr. SANTOSH CLASS NOTES
Naresh Technologies
PM 23734842, 23746666

2> To delete the row

- ⇒ move the cursor to the record which has to be deleted.
- ⇒ call deleteRow();

3> To insert a new record

- ⇒ move the cursor to the new empty record i.e., using moveToInsertRow();

- ⇒ use updateXXX methods to set the values
- ⇒ call insertRow() method on resultSet object.

MON

Priority

4

APR

2005

April 2005

S	M	T	W	T	F	S
				1	2	3
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

094-271, Week 14

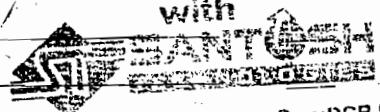
0*

// To insert a Record using Resultset.

```
import java.sql.*;
import java.util.*;
import java.io.*;
public class URSEXP3
{
    public static void main (String s[])
    {
        Properties p = new Properties();
        p.load (new FileInputStream ("my Properties.properties"));
        Driver d = (Driver) Class.forName (p.getProperty ("mydriver"))
                    .newInstance ();
        Connection con = d.connect (p.getProperty ("myurl"), p);
        Statement st = con.createStatement (ResultSet.TYPE
            - SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);
        ResultSet rs = st.executeQuery ("Select empno, name,
            sal from myemp");
        rs.moveToInsertRow ();
        rs.updateInt (1, Integer.parseInt (s[0]));
        rs.updateString (2, s[1]);
        rs.updateDouble (3, Double.parseDouble (s[2]));
        rs.insertRow ();
        System.out.println ("New Record created");
        con.close ();
    } // main
} // class
```

Mr. SANTOSH CLASS NOTES
Naresh I Technologies
Ph: 23734842, 23746666

#2nd Floor, Saraswati Niyam, Opp:DCB Bank
SR Nagar Police Station Lane, Ameerpet, Hyd.



Mr. SANTOSH CLASS NOTES
Naresh I Technologies
Ph: 23734842, 23746666

Priority						
May 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

095-270, Week 14

Priority

TUE

5

APR

2005

1/08/2005

Batch Update :-

8 am

Batch processing

⇒ If we want to execute multiple queries instead of submitting each query individually, we can prepare a batch in the client environment and submit the batch (i.e., set of queries) at a single request to the DB.

Support for preparing a batch and submitting it is provided from JDBC 2.0 onwards.

Note :-

Batch can hold update, insert, delete... but should not add a Select query to the batch.

This can be used with Statement and Prepared Statement Statement :-

→ addBatch (String SQL - query)

Mr. SANTOSH CLASS NOTE
Naresh Technologies
PM 23734842, 23746666

→ int[] executeBatch()

to use batch update

⇒ i) add all the queries in to batch using addBatch(<query>) i.e., here query is added into the batch which has to be submitted to the DB.

Note :-

For one statement only one batch can be prepared at a time.

By after adding all the queries into the batch to submit and execute it in DB call executeBatch().

Note :-

→ This method takes the batch and executes the queries in the order in which they have been added.

→ This method throws BatchUpdateException if it finds

WED Priority

6

APR

2005

April 2005						
S	M	T	W	T	F	S
				1	2	
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

096-269, Week 14

S M T
1 2
3 4
5 6
7 8
9 10
11 12
13 14
15 16
17 18
19 20
21 22
23 24
25 26
27 28
29 30
31

any exception while processing the queries.

- In the sequence of processing the batch if an exception is raised while executing a query it stops the batch processing from that point and throws exception (i.e., BatchUpdateException)

Batch Update Exception :-

→ extends SQLException

→ has a method which gives the information about the query at which the exception has been raised and give the status of the queries which has been executed within batch.

int[] getUpdateCounts()

- This method gives the update counts of the queries which are successfully executed and whose batch has been terminated without completion (abnormal termination)

// A Factory Class used to get Connection

```
import java.sql.*;  
import java.util.*;  
public class MyConnection Factory  
{  
    private MyConnection Factory () {}  
  
    private static MyConnection Factory mcf;  
    public static MyConnection factory getInstance()  
    {  
        if (mcf == null)
```

Mr. SANTOSH CLASS NOTES
Naresh I Technologies
PM 23734842, 23746666

Mr. SANTOSH CLASS NOTES
Naresh I Technologies
PM 23734842, 23746666

May 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

097-268, Week 11

Priority:

THU

7

APR
2005

మాధ్వ మానుషులు

```

mcf = new MyConnection Factory ();
8am
    return mcf;
} //getInstance
private Connection con;
public Connection getConnection(String file) throws Connection
10 Exception {
try {
11     if (con == null) && !con.isClosed()
12     {
Properties p = new Properties();
13         p.load(new java.io.FileInputStream(file));
1pm
Driver d = (Driver) Class.forName(p.getProperty("mydriver"))
14             .newInstance();
15     con = d.connect(p.getProperty("myurl"), p); //if
16     else if (con.isClosed())
17     {
Properties p = new Properties();
18         p.load(new java.io.FileInputStream(file));
19     Driver d = (Driver) Class.forName(p.getProperty("my driver"))
20         .newInstance();
21     con = d.connect(p.getProperty("my url"), p);
22     } //else if
23     return con;
24 } //try
25 catch(Exception e) {
26     throw new ConnectionException(e.getMessage());
27 } //catch
28 } //getConnection
29 } //class

```

Mr. SANTOSH CLASS NOTES
Nareesh Technologies
Ph: 23734842, 23746666
PM 23734842, 23746666

Mr. SANTOSH CLASS NOTES
Nareesh Technologies
Ph: 23734842, 23746666

FRI

Priority

8

APR

2005

now
Mr. Santosh
with



April 2005

#2nd Floor, Sarovar Nilayam, Opposite Matka
SR Nagar Police Station Lane, Amravati, Hyd.
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30

098-267, Week 14

// Connection Exception

```
public class ConnectionException extends RuntimeException
```

```
{  
    public ConnectionException (String s)  
    { super (s); }  
}
```

```
//class  
import java.sql.*;  
import java.util.*;  
public class Batch
```

```
{  
    public static void main (String s[]) throws Exception
```

```
{  
    My Connection Factory mc = My Connection Factory .get  
    instance ();
```

```
Connection con = mc .get Connection ("my Properties . properties");  
Statement st = con .create Statement ();
```

```
st .add Batch ("insert into myemp values ("+Integer  
    .parseInt (s[0]) + ", 'emp name')");
```

```
st .add Batch ("update myemp set name = " + s[1] + " where  
    emp no = " + s[0]);
```

```
st .add Batch ("update myemp set name = @ 'defaultname' where  
    emp no = 100");
```

```
try {
```

```
    int [] i = st .execute Batch ();
```

```
    for (int k = 0; k < i.length; k++)
```

```
{  
    System.out.println ("Query " +(k+1) +" Update Count is : "+ i[k]);  
}
```

Birthday / Anniversary

Mr. SANTOSH CLAS'S NOTES
Naresh Technologies
Ph 23734842, 23746666

May 2005						
S	M	T	W	T.	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

099-266. Week 14

Priority

SAT

9

APR

2005

catch (Batch Update Exception e) {

8am s.o.p ("Batch processing Terminated with Exception");

int [] i = e.getUpdateCounts();

for (int k=0; k < i.length; k++)

s.o.p ("Query " +(k+1) +" Update Count is :" +i[k]);

} // for

} // catch

c.out.close();

} // main

} // class

Mr. SANTOSH CLAS NOTES
Naresh Technologies
Ph: 23734542, 23746666

1pm // Batch 2

import java.sql.*;

2 public class Batch2

{

3 public static void main (String s[]) throws Exception

{

4 MyConnection factory mcf = MyConnection Factory . getInstance ();

5 Connection con = mcf . getConnection ("my properties . properties ");

6 Prepared Statement ps = con . prepareStatement ("insert into
myemp values (?,?)");

7 ps . setInt (1, Integer . parseInt (s [0]));

8 ps . setString (2, s [1]);

ps . addBatch ();

9 ps . setInt (1, Integer . parseInt (s [0]) + 1);

10 ps . setString (2, "emp1");

11 ps . addBatch ();

12 try

{

Burdwan Anniversary

13 int i [] = ps . executeBatch ();

SUN Priority

10

APR

2005

April 2005						
S	M	T	W	T	F	S
				1	2	
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

100-265, Week 1+

for(int k=0; k<i.length; k++)

s.o.p ("Query" +(k+1) +"update count :" +i[k]);

}//for

//try

catch(Batch Update Exception e)

{

s.o.p ("Batch Processing Terminated with Exception");

int i[] = e.getUpdateCounts();

for(int k=0; k<i.length; k++)

s.o.p ("Query" +(k+1) +";" +i[k]);

} //Catch

con.close();

} //main

} // class

Mr. SANTOSH CLASS NOTES
Naresh Technologies
Ph 23734842, 23746666

May 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	"
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Diary

MON

11

APR

2005

10:25 AM, Week 15

SQL 99 Data Types (SQL 3 Types)

8 AM

BLOB

CLOB

Array

Struct

ref

BLOB :-

Binary Large Object

→ It allows to store some binary data upto 4 GB.

CLOB :-

Character Large Object

→ It allows to store some character of size upto 4 GB.

To create a table with these types

```
create table myempl (empno number, name varchar2(20)
                     ing BLOB)
```

To insert a Record with BLOB type col from java app.

In prepared Statement

```
setBinaryStream (int index, InputStream inputStream, int size)
setBLOB (int, BLOB)
```

To use this method we have to create / get BLOB type of object (BLOB is an interface), we have to use 3rd party vendor implement class.

Birthday : 25/04/1984

TUE
12
APR
2005

Priority

April 2005						
S	M	T	W	T	F	S
					1	2
					3	4
					5	6
					7	8
					9	10
					11	12
					13	14
					15	16
					17	18
					19	20
					21	22
					23	24
					25	26
					27	28
					29	30
					31	

102-263, Week 15

// Insert Image

now
Mr. SANTOSH
with
SANTOSH
#2nd Floor, Chaitanya Bhawan, Opp. DCCB Bank,
SR Nagar, Police Station Lane, Amravati, Hyd.

```
import java.sql.*;  
public class InsertImage  
{  
    public static void main (String s[]) throws Exception  
    {  
        MyConnectionFactory mcf = MyConnectionFactory.getInstance();  
        Connection con = mcf.getConnection ("myproperties.properties");  
        PreparedStatement ps = con.prepareStatement ("insert into  
            myemp1 values (?, ?, ?)");  
        ps.setInt (1, Integer.parseInt (s[0]));  
        ps.setString (2, s[1]);  
        java.io.File f = new java.io.File ("my img.gif");  
        java.io.InputStream fis = new java.io.InputStream (f);  
        ps.setBinaryStream (3, (int)f.length());  
        ps.executeUpdate ();  
        System.out.println ("Record Inserted");  
        con.close ();  
    } // main  
} // class
```

To read BLOB type column in java application.

in ResultSet

InputStream getBinaryStream (int index);

May 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
403-262, Week 15						

Prerna

WED

13

APR

2005

InputStream gerBinaryStream (String col-name)

// SelectImage

```

import java.sql.*;
import java.io.*;
public class SelectImage
{
    public static void main(String s[])
    {
        MyConnectionFactory mcf = MyConnectionFactory.getInstance();
        Connection con = mcf.getConnection("my_properties.properties");
        Statement st = con.createStatement();
        ResultSet rs = st.executeQuery("Select * from myempl");
        rs.next();
        System.out.println("Emp" + rs.getInt(1));
        InputStream is = rs.getBinaryStream(3);
        FileOutputStream fos = new FileOutputStream("myimg1.gif");
        int i = is.read();
        while (i != -1)
        {
            fos.write(i);
            i = is.read();
        }
    }
}

```

MR. SANTOSH CLASS NOTES
Naresh Technologies
Ph 23734842, 23746666

THU

Priority _____

14

APR

2005

April 2005						
S	M	T	W	T	F	S
1	2					
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

104-261, Week 15

Array :=

8 am

→ This is an SQL 99 data type which allows to prepare a sequence of values of one type and save into a single column (i.e., of Array type)

To Create an array in the database

Create TYPE <array-name> as VARRY(<length>) of <type>;

Where <type> is the data type of the elements in the array.

Mr. SANTOSH CLASS NOTES

Naresh Technologies

Ph 23734842, 23746666

create TYPE Viga nos as VARRY(10) of varchar2(20);

Create Table myemp2 (name varchar2(20), sal number(10,2), viga decimal(8,2) Viga nos);

To Use an Array type in java application using JDBC

1) To insert a row with array type column.

→ Using Statement

use executeUpdate or execute method with the following query.

insert into <table name> values(---'<array-name>'
(list of elements))

→ Using Prepared Statement

Birthday / Anniversary

May 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	-
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Priority _____

FRI

15

APR

2005

105-260. Week 15

Example :-

8am insert into myemp2 values(101, 'emp101', visaNos('v₁', v₂, v₃...))

9 ~~Using Prepared Statement~~

10 // Inserting a row which has a array type of column, using Prepared Statement

```
11 import java.sql.*;
12 import oracle.sql.*;
13 public class InsertArray
14 {
```

```
15 public static void main(String s[])
16 {
```

2 My Connection factory mcf = MyConnectionFactory.get
Instance();

```
3 Connection con = mcf.getConnection("myproperties.properties");
4 Prepared Statement ps = con.prepareStatement("insert into
5 myemp2 values(?, ?, ?, ?)");
6 ps.setInt(1, Integer.parseInt(s[0]));
7 ps.setString(2, s[1]);
8 ps.setDouble(3, 1000);
```

6 // to set Array

```
String s[] = {"v1", "v2", "v3", "v4", "v5", "v6", "v7", "v8", "v9", "v10"}
```

Mr. SANTOSH CLASS NO.
Naresh Technologies
PHM 23734842, 23746666

7 Array Descriptor ad = ArrayDescriptor.createDescriptor("VISA_NOS",
8 con);

```
9 ARRAY a = new ARRAY(ad, con, s);
10 ps.setArray(4, a);
```

```
11 int i = ps.executeUpdate();
```

Birthday / Anniversary

SAT Priority

16

APR

2005

April 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				
Total 30 days, Week 15						

```
System.out.println("Row Inserted, Update Count: "+i);
con.close();
```

```
}/main
```

```
}/class
```

=> Using Prepared Statement

=> to select array type

Mr. SANTOSH CLASS NOTES
Naresh I Technologies
PM 23734842, 23746666

ResultSet:

can use

java.sql.Array getArray (int)

java.sql.Array getArray (String)

in Array Interface

ResultSet getResultSet ()

The above methods return a ResultSet with 2 columns
and row equivalent to the array size.

Where first column will be the index no ~~second~~ column
will be the element value.

// Select Array type column

```
import java.sql.*;
```

```
public class Select Array
```

```
{
```

May 2015						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Prin...

SUN

17

APR

2005

10-258, Week 15

1 public static void main (String s[]) throws Exception

2 {

3 MyConnection factory mcf = MyConnectionFactory.getInstance();

4 Connection con = mcf.getConnection ("myproperties.properties");

5 Statement st = con.createStatement ();

6 ResultSet rs = st.executeQuery ("Select * from myemp2");

7 while (rs.next ())

8 {

9 System.out.println (rs.getInt (1) + "\t");

10 Array a = rs.getArray (4);

11 ResultSet rs1 = a.getResultSet ();

12 while (rs1.next ())

13 {

14 System.out.println (rs1.getString (2) + " ");

15 } //while

16 s.o.p ();

17 } //while

18 con.close ();

19 } //main

20 } //class

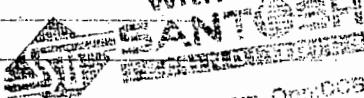
Mr. SANTOSH CLAS, NOTES
Naresh Technologies
Ph 23734842, 23746666

MON
18
APR
2005

Priority

now
Mr. Santosh

with



#2nd Floor, Sarojini Naidu, Opp DODS Bank
SR Nagar Police Station Lake, Ameerpet, Hyd.

April 2005

S	W	T	F	S	S
3	4	5	6	7	8
9	10	11	12	13	14
15	16	17	18	19	20
22	23	24	25	26	27
28	29	30	31		

Struct

8 am

Object type

→ This, i.e. used to define a type which can hold one or more other types of elements (and all the element types need not be same) i.e., defining a complex type in SQL.

11

To create an object

Create type <type name> as object (<element-name><type>,...)

1 pm

Example:-

Create type emp-add as object (flat-no number,

Street varchar(20), city varchar(20), pin number);

Create table myemp3 (empno number, name varchar(20),

sal number(10,2), address emp-add);

Mr. SANTOSH CLAS, NOTES

Naresh Technologies

Ph: 23734842, 23746666

To insert a record

insert into myemp3 (101, emp101, 10000, emp-add(10,

'street1', 'hyd', 500049));

To insert a record with object type using Prepared Statement

→ Should write a class which can mapped to the object type defined in SQL

User defined type (UDT)

// UDT

```
import java.sql.*;  
public class EMPAddress implements SQLData  
{  
    public EMPAddress() {}
```

May 2005

Priorities

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Week 16

TUE

19

APR

2005

public EMPAddress (int i, String s, String s, int n, String sr)

{

fno = i;

Street = s;

city = s; pin = n;

SQL-typeName = sr;

} // constructor

public String getSQLTypeNname ()

{

return SQL-typeName;

} // getSN

public void readSQL(SQL-Input si, String n) throws SQLException

{

fno = si.readInt();

Street = si.readString();

city = si.readString();

pin = si.readInt();

SQL-typeName = sn;

} // read SQL

public void writeSQL(SQL-Output so) throws SQLException

{

so.writeInt(fno);

so.writeString(Street);

so.writeString(city);

so.writeInt(pin);

} // SW

String Street, city, SQL-typeName;

int fno, pin;

} // class

Mr. SANTOSH CLAS: NOTES
Naresh Technologies
PMA 23734842, 23746666

WED
20
APR
2005

Priority

April 2005						
S	M	T	W	T	F	S
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
110-255, Week 16						115-2

//Inserting object

```
import java.sql.*;
public class InsertObject
{
    public static void main (String s[])
        throws Exception
    {
        MyConnectionFactory mcf = MyConnectionFactory.getInstance();
        Connection con = mcf.getConnection ("myproperties.properties");
        PreparedStatement ps = con.prepareStatement ("? ?, ?, ?");
        ps.setInt (1, 101);
        ps.setString (2, "empl01");
        ps.setDouble (3, 10000);
        EMPAddress ea = new EMPAddress (100, "Street", "hyd", 500049,
                                         "Emp-ADDRESS");
        ps.setObject (4, ea);
        int i = ps.executeUpdate ();
        System.out.println (i);
        con.close ();
    }
}
```

Mr. SANTOSH CLASS NOTES
Naresh Technologies
Ph 23734842, 23746666

When executeUpdate method is called on PreparedStatement object

- driver reads the SQLType name by calling getSQLType
- Name() on the object passed as an argument to setObject method
- prepare a suitable SQLOutput Object
- call writeSQL method passing the SQLOutput type of Object as an argument so that in writeSQL method we

May 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	-
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

111-254, Week 16

Priority:

THU

21

APR

2005

have to write the values of the elements in the Object
 → Driver takes the values given (i.e., using SQL Output) and send it to the DB

// Selecting the data (Object type)

```
10 import java.sql.*;
public class selectObject
11 {
12     public static void main(String s[])
13     {
14         MyConnectionfactory mcf = MyConnectionfactory.getInstance();
15         Connection con = mcf.getConnection("myproperties.properties");
16         Statement st = con.createStatement();
17         Resultset rs = st.executeQuery("select * from myemp3");
18         if (rs.next())
19         {
20             System.out.println(rs.getInt(1) + "lt");
21             Object o = rs.getObject(2);
22             Struct st = (Struct)o;
23             Object o[] = st.getAttribute();
24             for (int i = 0; i < o.length; i++)
25                 System.out.println(o[i] + "lt");
26         }
27         con.close();
28     }
29 }
```

Mr. SANTOSH CLAS³ NOTES
 Naresh Technologies
 Ph: 23734842, 23746566

Birthday / Anniversary

FRI

Priority _____

22

APR

2005

now
Mr. Santosh

April 2005

S	M	T	W	T	F	S
				1	2	
			3	4	5	6
			7	8	9	
			10	11	12	13
			14	15	16	17
			18	19	20	21
			22	23	24	25
			26	27	28	29
			30			

#2nd Floor, Saraswati Niyyam, Opp. P.G. Block

SR Nagar Police Station Lane, Ameerpet, Hyderabad

112-253, Week 16

Using Type Mapping in JDBC

8 am

If an SQL server defined type (Object type) is defined in the DB and if we want to map it the type defined in Java environment (using JDBC Standard)

10

In Connection

11

Map getTypeMap()

12

- returning the current set of data type mappings

void setTypeMap (Map m)

1 pm

- allows us to set a Map object with undefined type mappings

These type mappings are used by driver provider when selecting the data

(or)

Mr. SANTOSH CLASS NOTES

Naresh Technologies

Ph 23734842, 23746666

Object getObject (int index, Map m)

Object getObject (String name, Map m)

The map type of object passed as argument to the Connection's setTypeMap or Resultset's getObject method should hold the mappings of SQL type to the Java type.

Example :-

```
Map m = new HashMap();
mput ("EMP-ADDRESS", class.forName ("EMPAddress"));
```


SUN

Priority

24

APR

2005

April 2005

S	M	T	W	T	F	S
				1	2	
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

11-251, Week 16

getObject (int i, map m)
8am { //read the data from DB
Object o = m.get("<SQL type of DB col>");
if (o1 == null) (or)
Class c = (Class)o;
Object o1 = c.newInstance();
SQLData cd = (SQLData)o1;
//prepare an SQL-Input type of object
cd.readSQL(8i, "<SQL type names>");
} return cd;
1pm

Mr. SANTOSH CLASS NOTES
Naresh Technologies
PMT 23734842, 23746666



Birthday / Anniversary

THU

Priority

28

APR

2005

April 2005						
S	M	T	W	F	S	S
				1	2	
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

118-247, Week 17

Transaction

Sam

→ set of statements executed onto resources/resource's following ACID properties if said to be a transaction

A - (Atomicity) All the statements in the transaction should work as single unit

C - (Consistency) If the data used in the transaction is consistent before starting the transaction then it will be left in a consistent state even after the Transaction (TX)

I - (Isolation) is an ability of the TX to hide the data from other TX's

D - (Durability) After giving a commit instruction to the transaction manager, then if any ^{possible} ^{Note} error occurs in the network or in resource of ^{DBSHC Technologies} Nareesh Patel 23740566

Birthday / Anniversary

FRI

29

APR

2005

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

119-216, Week 1"

Local Transaction

→ A transaction statements are executed on a single resource using a single resource object is known as local transaction.

Example:-

If all the statements in transaction are executed using a single connection. **Mr. SANTOSH**
Naresh Technologies
Ph: 23734842, 23746666

In this type of transaction all the statements are under the control of a single Resource Manager (Under one session)

Here in this case Resource Manager can manage the transaction since it is known about all the statements of the transaction and has a control on them.

In a Java application if we want some set of statements to be in a single transaction then JDBC allows us to give an instruction to the Resource Manager about the transaction.

in Connection Interface

setAutoCommit (boolean)

commit ()

rollback ()

1> get a Connection

2> set auto commit mode to false

3> get the statements (i.e. statement or prepared Statement . . .)

SAT

Priority

30

APR

2005

April 2005

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

120-245, Week 17

↳ execute queries

8 am ↳ end the transaction by calling commit() or rollback()

9 //Local TX

10 import java.sql.*;

11 public class Local TX

12 public static void main (String s[]) throws Exception

1 pm MyConnection factory mcf = My Connection Factory . getInstance();

Connection con = mcf . getConnection ("myproperties . properties");
con . setAutoCommit (false);

Statement st = con . createStatement ();

int i = st . executeUpdate ("update myemp set sal = 1000

where empid = " + s [0]);

int j = st . executeUpdate ("update myemp set sal = 1000

where empid = " + s [i]);

if (i == 1 && j == 1)

{

con . commit ();

s . c . p ("commit");

& return;

} else

con . rollback ();

s . c . p ("Rollback");

} //main

} //class

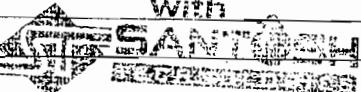
Birthday / Anniversary

Mr. SANTOSH CLAS; NOTES
Naresh I Technologies
Ph: 23734842, 23746666

June 2005						
S	M	T	W	T	F	S
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		
121-244, Week 1						

Priority

now
Mr. Santosh



#2nd Floor, Saraswati Nilayam, Opp:DCB Bank
SR Nagar Police Station Lane, Amerspet, Hyd.

SUN

1

MAY
2005

when execute method is called

- finds whether there is any TX currently associated with the Connection(session) or not
- If it is not available then start a new TX
- If not continue with the existing TX
- execute the query
- find the auto commit mode of the connection if it is true then end the TX
- if not hold the TX in the same state
- returns the response to client.

Note :-

At a time only one TX can be associated with the connection.
i.e., a statement can be involved into only one TX at a time

In JDBC 3.0 Savepoints has been introduced:-

- Using save points we can have a mark at a point within the TX and we can rollback the statements upto mark from the current position if we want

In JDBC 3.0 the new methods introduced to support this feature

In Connection :-

SavePoint setSavePoint (String)

where Savepoint is an interface (new in JDBC 3.0)

MON Priority

2

MAY

2005

May 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	6
	7	8	9	10	11	12
	13	14	15	16	17	18
	19	20	21	22	23	24
	25	26	27	28	29	30
	22-24-3, Week 18					

S N
5 1
12 2
19 3
26 4

Rollback (SavePoint)

releaseSavePoints()

In DataBase MetaData

boolean supportsSavePoints()

// Save Point Example

// JDBC 3.0

import java.sql.*;

public class SavePointsEx

{

public static void main (String s[]) throws Exception

{

Driver d=(Driver) class.forName ("Sun. jdbc. odbc. JDBC ODBC
Driver"). newInstance();

java.util.Properties p = new java.util.Properties();

p.put("User", "scott");

p.put("password", "tiger");

Connection con=d.connect ("jdbc:odbc: mydsn", p);
con.setAutoCommit (false);

Statement st= con.createStatement();

int i= st.executeUpdate ("update myemp set sal=1000 where
empid = "+s[0]);

SavePoint sp=con.setSavePoint ("SP1");

int j= st.executeUpdate ("update myemp set sal=2000 where
empid = "+s[1]);

con.rollback(sp);

System.out.println ("i:"+i+" j:"+j);

Mr. SANTOSH CLAS. NOTES
Naresh Technologies
Ph 23734842, 23746666

Priority						
S	M	T	W	T	F	S
	1	2	3	4	5	6
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

12-2-2 Week

June 2005

TUE

3

MAY

2005

Con::commit();

8 am

Con::close();

} // main

9

} // class.

Isolation Levels

problems which multiple transactions reading the same data.

1) Dirty Read Problem

2) non repeatable read

3) phantom read.

1) reading an uncommitted data is known as dirty read.
i.e. if the data which has been modified by the TX and
has not been committed and if any other TX reads
that data then it is known as dirty read.

2) In this case if the data is selected and being used
in a TX and meanwhile if the data is changed by
other TX (i.e. within the TX if we read the data
for more than once and if we get different results)
Such a read problem is known as nonrepeatable
read problem.

This problem can be avoided by maintaining
a lock on the data which is being used in the TX
This has to be done by resource manager.

WED

Priority

4

MAY

2005

May 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

124-241, Week 18

3) In this case if we get different number of records satisfying the condition when the query is executed for more than once in a TX i.e., other TX might insert a new record into the table.

This problem can be avoided by maintaining lock on the table.

Mr. SANTOSH CLAS NOTES
Naresh I Technologies
PM 23734842, 23746666

levels:

TRANSACTION-READ-UNCOMMITTED

will not avoid dirty read or nonrepeatable read or phantom read problems.

TRANSACTION-READ-COMMITTED

avoids dirty read i.e., it allows only to read committed data.

But still nonrepeatable read and phantom reads may occur.

TRANSACTION-REPEATABLE-READ

avoids dirty read and nonrepeatable read problems but phantom read may occur.

TRANSACTION-SEARIALIZABLE

avoids all the 3 problems

TRANSACTION-NONE

is not a isolation level but it indicates that TX isolation levels are not supported by the resource.

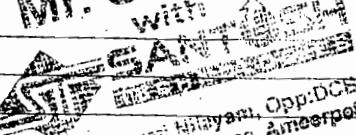
S	June 2005						
	S	M	T	W	T	F	S
1		1	2	3	4		
2	5	6	7	8	9	10	11
3	12	13	14	15	16	17	18
4	19	20	21	22	23	24	25
5	26	27	28	29	30		

125-240. Week 18

Priority _____

Mr. SANTOSH

with



42nd Floor, Samruddhi Mahayani, Opp.DCS Bank
SR Nagar Police Station Lane, Ameerpet, Hyd.

THU

5

MAY

2005

Note :-

8am As we come from top to bottom the TX Isolation level increases but performance decreases.

To set these isolation levels use setTransactionIsolation(int) method declared in Connection.

11 RowSet :- It is actually introduced in JDBC 2.0

12 Provides a java bean based approach to deal with the data selected or to update, insert the data into database.

1pm

2 types of rowsets

↳ Connected Rowset

↳ DisConnected Rowset

MR. SANTOSH CLASS
Naresh Technologies
PH 23734842, 23746666
OTE

↳ Connected Rowset :-

→ This type of Rowset wraps the Resultset and the Resultset associated with the Rowset should not be closed until the rowset is in use.

↳ Jdbc Rowset

↳ DisConnected Rowset :-

→ with this type rowset's jdbc connection is used to just load the data into the rowset or to update the data into the database.

i.e., Once after populating the data from resultset into the rowset resultset can be closed.

Birthday / Anniversary

FRI

Priority ..

6

MAY

2005

May 2005

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

126-239, Week 18

- Cached RowSet

: Web RowSet

- Join RowSet

: Filtered RowSet

These type of RowSet's are used

- When the data has to be send for long time
- if we want to transfer the data in N/w

from J2SE 5.0 Sun has included the reference implementation of the RowSet into j2sdk (TAKS CLASS NOTE)

Mr. SANTOSH
Naresh I Technologies
Ph: 23734842, 23740666

// Connected RowSet

```
import java.sql.*;  
import javax.sql.rowset.*;  
import com.sun.rowset.*;  
// holds the reference implementation of rowset
```

```
public class Connected RS
```

```
{
```

```
    public static void main (String s[]) throws Exception
```

```
{
```

```
    Driver d = (Driver) Class.forName ("oracle.jdbc.driver.Oracle  
Driver").newInstance ();  
    java.util.Properties p = new java.util.Properties ();  
    p.put ("user", "scott");  
    p.put ("password", "tiger");
```

```
    Connection con = d.connect ("jdbc:oracle:thin:@localhost:  
1521:nik");
```

```
    Statement st = con.createStatement ();
```

Birthday Anniversary

June 2005						
S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

Priority _____

SAT

7

MAY

2005

127-238. Week 18

```

8am   ResultSet rs = st.executeQuery("select * from myemp");
      JdbcRowSet jrs = new JdbcRowSet();
      while (jrs.next())
      {
        System.out.println(jrs.getString(1) + " " + jrs.getDouble(2));
      } //while
    } //main
  } //class

```

// DisConnected RowSet

// Here we are using the RowSet implementation
provided by Sun under the reference implementation

```

import java.sql.*;
import javax.sql.rowset.*;
import com.sun.rowset.*;

```

MR. SANTOSH CLAS, NO:
Naresh Technologies
PH 23734842, 23746666

```

public class DisConnected RS
{

```

```

  public static void main (String s[])
  throws Exception
  {

```

```

    Driver d = (Driver) Class.forName ("oracle.jdbc.driver.
    OracleDriver").newInstance ();
  
```

Connection Conn =

```

    java.util.Properties p = new java.util.Properties ();
    p.put ("user", "scott");
    p.put ("password", "tiger");
  
```

```

  Connection Con = d.connect ("jdbc:oracle:thin:@localhost:1521:
    @nit", p);

```

```

  Statement st = Con.createStatement ();

```

Birthday Anniversary

SUN

Priority _____

8

MAY

2005

May 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

128-237, Week 18

Result Set rs = st.executeQuery ("Select * from myemp");

8 am

Cached RowSet cr = new CachedRowSetImpl();

9

cr.populate (rs);

10

cr.close ();

{

11 s.o.p (cr.getString (1) + " " + rs.getString (2));

12 //while

13 //main

14 //close

1 pm

Mr. SANTOSH CLAS, NOT

Naresh Technologies

PM 23734842, 23746666

Jdbc RowSet jrs = new JdbcRowSetImpl();

2

jrs.setUrl ("jdbc:oracle:thin:@localhost:1521:xit");

3

jrs.setUsername ("scott");

4

jrs.setPassword ("tiger");

5

jrs.setCommand ("Select * from myemp");

6

jrs.execute ();

7

while (jrs.next ())

8

{ -- }

Getting Auto generated Keys :-

In JDBC 3.0 new methods in Statement has been introduced to obtain an auto increment column value.

Statement:

1) execute (String query, int i)

where int argument takes

Statement.NO_GENERATED_KEYS

Statement.RETURN_GENERATED_KEYS

Birthday / Anniversary

June 2005						
S	M	T	W	T	F	S
1	2	3	4			
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

Priority _____

MON

9

MAY

2005

129-236, Week 19

8am

3) execute (String query, int[] col-indexes)

take the query and the column index where auto increment values has to returned
i.e. make available for application

3) execute (String query, String[] col-names)

4) ResultSet getGeneratedKeys()

The above methods are introduced in JDBC 3.0

Mr. SANTOSH CHAVAS NOTES

Naresh Technologies

Ph 23734842, 23746666

In My SQL

create table myemp (empid int NOT NULL AUTO_INCREMENT
name char(30), Sal double, primary key(empid));

insert into myemp(name, Sal)

values ('emp1', 10000);

// JDBC 3.0

// Getting auto generated col values

import java.sql.*;

public class Auto increment col

{

public static void main(String ss) throws Exception

{

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost/testdb");

Statement st = con.createStatement();

Birthday / Anniversary

TUE

Priority _____

10

MAY

2005

May 2005

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

130-235, Week 19

8 am boolean b = st.execute("insert into myemp(name, sal) values ('emp2', 20000)", Statement.RETURN_GENERATED_KEYS);

9 ResultSet rs = st.getGeneratedKeys();
if (rs.next())

10 {

System.out.println(rs.getInt(1));

11 //if

con.close();

12 } // main.

13 } // class

1 pm

mysql connector/j driver

2 is a Type-4 driver

3 Set classpath = C:\mysql-connector-java-3.1.8\mysql-connector-jar-3.1.8-bin.jar;%classpath%

oracle:

4 Create Sequence <sequence-name> Start with 1 increment by 1;

5 Create table myemp(empid number, name varchar2(20),
sal number(10,2));

6 Create trigger <trigger-name> before insert on myemp
begin

7 select next val into

8 select <sequence-name>.next val into: new.empid from dual;
end;

Birthday Anniversary

June 2005						
S	M	T	W	T	F	S
		1	2	3	4	
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

131-234, Week 19

Priority.....

GS
w

now
Mr. Santosh
with



WED

11

MAY

2005

insert into myemp (name, sal) values ('empl', 10000);

Sam

or

insert into myemp value (<sequence-name>. nextVal, 'empl', 10000);

this can be used without trigger

OTES

Named parameters:

Mr. SANTOSH
Naresh Technologies
Ph: 23734842, 23746666

New in JDBC 3.0

⇒ with CallableStatement we used to set the values and register the out parameters using the parameter index.

where as from 3.0 we can set and register using the parameter names

i.e., setXXX methods has been overloaded

setXXX (int parameter-name, XXX value)

and registerOutParameter is also overloaded

registerOutParameter (int parameter-index, int type)

registerOutParameter (String parameter-name, int type)

In Supporting to this java.sql.ParameterMetaData has been given which can be used to get the information about the parameters.

THU Priority

12

MAY

2005

May 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

132-233, Week 19.

Connection pooling :-

⇒ Using the normal DataSource to obtain the Connection gives the physical Connection reference directly to the client.

By once if the Client completes the usage of Connection and when he calls Connection.close() the physical session opened to the DB will be closed.

If we require a Connection again or if any other client requires the Connection Driver(dataSource) has to establish a physical session to DB again.

Mr. SANTOSH C. AS / NOTE

Naresh I Technological

1 pm Connection Pool allows us to maintain physical connection and logical connection to the client so that if Client closes the Connection the physical Connection associated with the logical Connection will be placed into pool instead of closing the Connection (physical) so that when other client makes a request for the Connection it just prepares one logical Connection attach it to the existing physical Connection and give it to client.

Connection Pool implementation can be used in 2-tier or 3 tier architecture

with 2-tier model :-

JDBC Driver provides implements the Connection pool data source and pooled connections i.e., javax.sql.ConnectionPoolDataSource
javax.sql.PooledConnection

S						S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Priority

June 2005

133-232, Week 19

FRI

13

MAY

2005

Example :-

8 am Oracle implementation

oracle.jdbc.pool.OracleConnectionPoolDataSource

i.e., ConnectionPoolDataSource can cache the physical connection reference to the client and give the logical connection reference to the client.

Mr. SANTOSH CLAS; NOTE:
Naresh I Technologies
Ph: 23734842, 23746666

// Connection Pool (2-tier)

```
import java.sql.*;
import javax.sql.*;
import oracle.jdbc.pool.*;
public class ConnectionPoolEx
{
```

```
public static void main ( String s[] ) throws Exception
{
```

```
    OracleConnectionPoolDataSource ocp = new OracleConnection
        PoolDataSource ();
```

```
    ocp.setUrl ("jdbc:oracle:thin:@localhost:1521:xit");
```

```
    ocp.setUser ("scott");
```

```
    ocp.setPassword ("tiger");
```

```
PooledConnection pc = ocp.getPooledConnection ();
```

```
Connection con = pc.getConnection ()
```

// con is the reference to logical connection reference to the client

Note:- only one logical connection can be opened at a time and the latest logical connection will be the valid connection of multiple logical connection are obtained on the same Pooled Connection

SAT Priority _____
4 _____
MAY _____
005 _____

May 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

134-231, Week 19

```

* Statement st = con.createStatement();
/* Prepared Statement ps = con.prepareStatement ("insert into myemp
values (?,?)"); */
ResultSet rs = st.executeQuery ("Select * from myemp");
while (rs.next ())
{
    System.out.println (rs.getInt (1));
}
//while
con.close ();
pc.close ();
}
//main
}
//class

```

Mr. SANTOSH CLASS NOTE
Naresh Technologies
Pm 23734842, 23746666
(1)):

In JDBC 3.0 Statement Cache is introduced with the connection pool i.e, Connection pool can hold the statements (PreparedStatement, CallableStatement) and reuse it for multiple clients (one after the other).

This can be done in implicit way by taking the query as a unique identity or explicitly assigning some unique identity to the Statement

ResultSet Holdability (New in 3.0)

The resultset which is open and can be used even after the completion of the transaction is known as holdable resultset

set holdability (`:inf`)

June 2005						
M	T	W	T	F	S	S
			1	2		
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			
135-230, Week						

S am Rej
Rej

Note

at)

JDF

$$12 \Rightarrow 1$$

1 pm \Rightarrow 10

\Rightarrow

3

1

三

6
W

三

卷之三

— 3 —

Birthday / Anniversary

三

10

June 2005						
S	M	T	W	T	F	S
1	2	3	4			
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

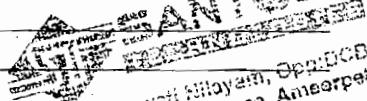
Priority _____

135-230, Week 19

now
Mr. Santosh
with

SUN

15



#2nd Floor, Goparawati Niloyam, Opp DCB Bank,
SR Nagar Police Station Lane, Ameerpet, Hyd.
2005

It takes

ResultSet.HOLD_CURSORS_OVER_COMMIT (or)

ResultSet.CLOSE_CURSORS_AT_COMMIT

Note :-

In JDBC 3.0 we can have multiple ResultSet open at a time on a statement.

JDBC 4.0 Draft 2

Mr. SANTOSH CLASS NOTES
Naresh Technologies
Ph: 23734842, 23740000

⇒ includes annotation (to obtain the Connection, to prepare and execute query...)

⇒ DataSet has been introduced

⇒ SQL 2003 Support

⇒ XML/SQL Support

⇒ Wrapper classes has been given so we can use some vendor specific implemented Service

(i.e., using the non-standard service implemented by a specific can be used without making the application vendor specification)

JDBC is not using individually. It is for use including with the Servlets, JSP and etc.

MON Priority

16

MAY

2005

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

18-22 Week 20

2-tier Architecture :-

Client tier: is responsible for ^{the} presentation and business logic

Server tier - in most of the cases it will be the data store (like database server) and may hold some part of processing (i.e., like procedures and functions in DB)

Example :-

An AWT Application (java) which takes the Country and State details and displays the weather forecast Report of the selected state and the weightage available in DB

Mr. SANTOSH CHAVHATE
Naresh Patel 23/05/06
Ph: 23734842, 23734843

Problems with this architecture

1) Client tier application is included with presentation logic and business logic so

If we want to have different presentations for the same business process then we may require to rewrite the business process (implement Business rules)

2) The resources required to run the presentation logic and business logic has to be set in the client system (and it may not be possible to set all the required resources in all client environment)

3) Since the Business process is in the client tier it less secure

4) The same business process will be running in different client environment accessing the data and in some cases where the process requires to be synchronized and requires the data sharing between some set of client, may not be possible with this type of arch

June 2005						
S	M	T	W	T	F	S
		1	2	3	4	
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

137-228, Week 20

Priority

TUE

17

MAY

2005

To solve the above problem, 3-tier architecture has been introduced.

3-tier Architecture

- Here Middleware Server is introduced in between Client and Datastore (Backend Server)

Middleware Server takes the responsibility to execute the business process and interact with the Backend Server to complete the process (on the client's request)

Mr. SANTOSH CLASS NOTES

The advantages of 3-tier arch

- 1) The business process can be reused i.e., we can support different types of clients (different presentation formats)
- 2) Business Process is placed and executed in the middle ware server so:

- The resources required to execute the business process has to be set in the middle ware server
- It is secure since the business process is abstracted from the clients

- The business process can be synchronized and we can have a control on the request/response (since the same business process is not running on each of the client)

i.e., we can have an application level isolation

WED
18
MAY
2005

Priority _____

May 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

138-227, Week 20

In the basic 3-tier architecture we require the client software to be installed into the client system to browse the service provided by the Middleware Server.

And a client software may be intelligent to interact with one or 2 middleware server and present the services.

We require a generic client software which can interact with different middleware servers (i.e., at the time of developing client software we don't require to know about the middleware).

Mr. SANTOSH CLASSTIME NOTES

Naresh / Technologies

Ph 23734842, 23740066

Requirements to design such an architecture

- 1) Requires a standard protocol to exchange the messages between the client and server.
- 2) Requires some standard format which can be used by the Middleware server to explain the client software what and how to present the data.

The client software is a Browser.

HTTP (Hyper Text Transfer Protocol) is a standard protocol which can be used to exchange the data (messages) between the client and server (i.e. Request/Response).

HTML (Hyper Text Markup Lang) is a standard markup lang which can explain the browser about the data to be presented and the format in

Priority						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

139-226, Week 20

THU
19
MAY
2005

which it should be presented

8am

And This type arch (i.e, with a generic client S/W (browser) and using some standard protocol like http) is known as Web Based Arch (Web Arch)

Want to generate Dynamic content

Dynamic Content = a content which may be changed from one request to other or one client other

Static Content = which doesn't change from one request to others.

(The content to be displayed, is hand coded into the page while developing the page) **Mr. SANTOSH** Naresu Technical Services
P.M 23734842, 2374066

CGI : Common Gateway Interface

⇒ is a standard abstraction between Http Adapter (Server) and a small application (written in lang like C, C++ ...) which is responsible for generating the dynamic content which is said to be CGI Application

In this case

when client makes a request for the Dynamic page

⇒ Http Adapter finds that the request is for Dynamic page

— locate the page (ie, CGI Application)

— Start a new process and invoke the application within the process

— Collect the response from CGI Application

— destroy the process and prepare http response and send it to the client.

Bonday Anniversary

FRI

Priority

20

MAY

2005

May 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

140-225 Week 20

In this approach Server used to create and destroy the process for every request which used to increase the load on the server.

To reduce the load of the server Fast CGI was introduced where in this case Server used to maintain pool of processes and recycle the processes for processing multiple requests one after another.

With fast CGI

Mr. SANTOSH NARASHI TECHNOLOGIES
Naresh Technologies
P.M. 23734842, 23746666

→ A process created may handle one or more request within the lifetime of the process but can handle only one request at a time.

And Http Server maintaining pool of processes
Problems :-

→ Use a heavy weight process to handle the client request

i.e., it requires one process to handle one request

→ Interacting with the http server using CGI standards and getting the request data, submitting the response content was difficult (was in low level)

June 2005						
S	M	T	W	T	F	S
5		1	2	3	4	
14	5	6	7	8	9	10
21	12	13	14	15	16	17
28	19	20	21	22	23	24
	25	26	27	28	29	30

141-224, Week 20

Priority

SAT

21

MAY

2005

Sree ~~now~~ Mr. Santosh



#2nd Floor, Saravalli Millay Bldg, Opp: DCC Bank
SR Nagar Police Station Lane, Ameerpet, Hyd.

Servlet:

8am

⇒ A Java application developed following Java Servlet Specification, placed on top of Servlet (Web) Container to provide server side extensions

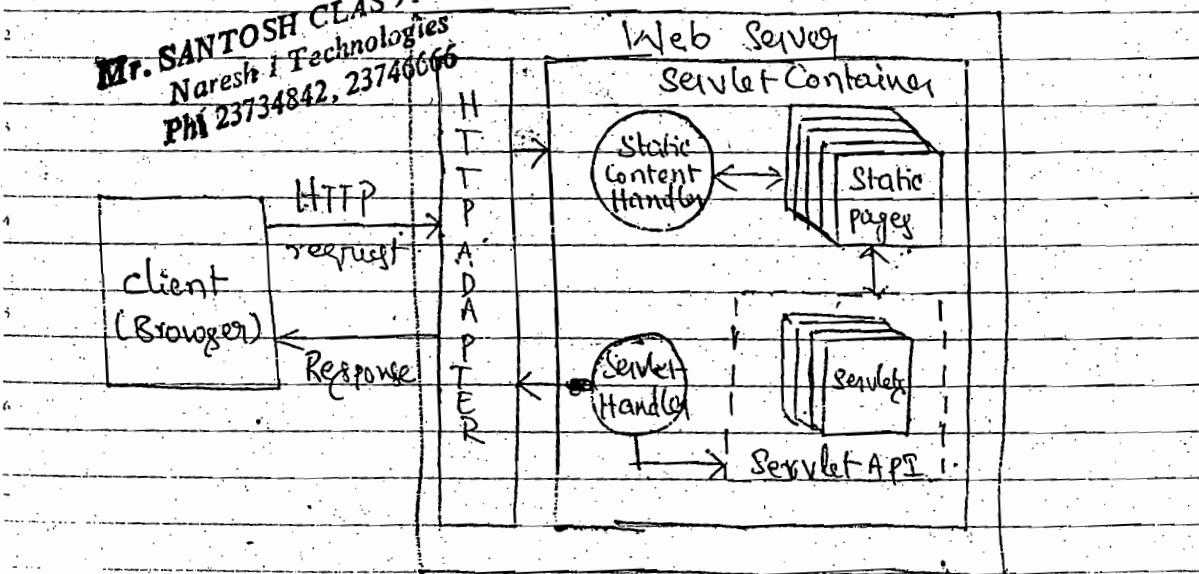
⇒ If a Server side executable java application

Servlet Container:

⇒ provides a runtime environment for the servlets
(where Servlet is application written)

1pm Architecture :- Standalone Servlet Container

Mr. SANTOSH CLASS NOTES
Naresh I Technologies
Ph: 23734842, 23746666



SUN

Priority

22

MAY

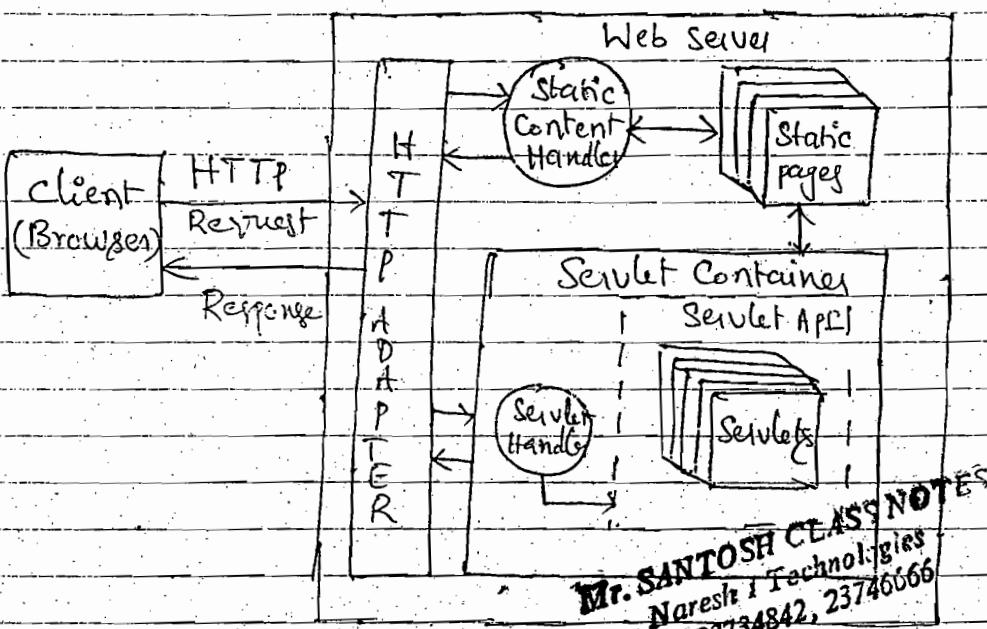
2005

May 2005

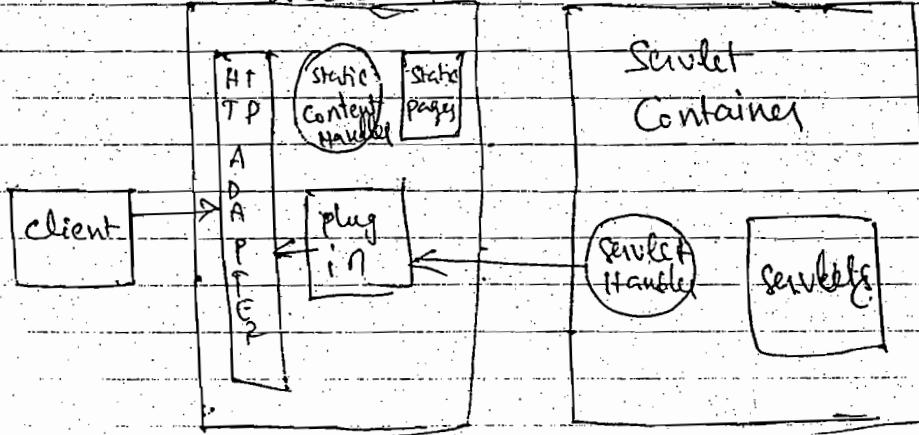
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

142-223, Week 20

In Process Servlet Container



out process Container



S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Priority

June 2005

143-222, Week 21

MON

23

MAY

2005

Standalone Containers:

Example: Tomcat Server

These Type of Containers are built provided by the Web Server

- In this type of Servers Http Adapter receives the request (Http Request)

- Unpacks the request

- Submit the request and request data to the Servlet Container (Web container)

- Container identifies whether the request is for the static resources or Dynamic resources

- If it is for static resources then Static Content Handler processes the request (where it locates the resource and generates the response whose content will be the content of the file located)

- If it is for dynamic resource page then the request is handled by the Servlet Handler (it locates the servlet and completes the request processing)

- The Response Content is submitted to the Http Adapter so that it can be transferred to the client

Mr. SANTOSH CLASS NOTES

Naresh I Technologies

Ph: 23734842, 23740066

In process

In this case Servlet Container and Web Server can be provided by different vendors

i.e., This helps us to get a container from different vendor compatible to the existing web server and extend the application of the server

Example: Microsoft IIS Server (web server) added with the web server container plug-in for IIS Server.

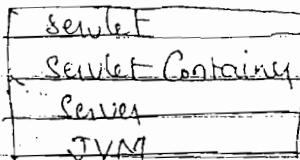
TUE

Priority

24

MAY

2005



May 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

14-221, Week 21

In this Server When a request is given by web server identifies whether the request can be handled by the web server or if for the Servlet Container (i.e plugin)

out process Container:

⇒ In this case Servlet Container is attached to the Web Server externally (i.e, it runs outside the Web Server process)

Example:

An Apache Web Server Configured to redirect the request for servlets to the Tomcat Server which is running ~~separately~~ separately (may be in the ~~same~~ system or different)

Mr. SANTOSH CLASSTIME
Naresh Technologies

23734842, 23746666

Servlet API:

⇒ Is given under Servlet Specification By Sun
Current version is 2.4

It provides Standard abstraction in b/w of Servlet Container and Servlet Component

Servlet Specification provides a facility to use the HTTP protocol specific information which processing the client request

i.e, every server which is used to handle the servlet request should have a HTTP Adapter

And even we can develop the application to process non http clients also

June 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11		
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

Priority:

29-30 June

now
Mr. Santosh
with

Siri Siri muriva

MR. SANTOSH WED

#2nd Floor, Saraswati Nilayam, Opp:DCB Bank,
Sri Nageswaran's Station Lane, Amedpet, Hyderabad 500025

MAY

2005

javax.servlet.* → This package defines some set of interfaces and classes which helps up to write Servlets which can process the client request (i.e., HTTP Client and Non-HTTP Clients)

javax.servlet.http.* → Some additional interfaces and classes which helps up to program Servlets which can take the HTTP protocol specification information and generate HTTP specific response content (i.e., generate some HTTP headers...).

Mr. SANTOSH CHAVADE
Naresh Technologies
PM 23734842, 23746666

Java Servlet.Servlet = It is an interface which provides standard abstraction between the Container and the Servlet Component.

i.e., Container try to understand and interact with the Servlet Component through the Servlet interface.

Every Servlet has to be the subtype of this type (i.e., this interface)

in javax.servlet.Servlet interface

```
init(Servlet Configuration)
Service(Servlet Request, Servlet Response) } they are lifecycle
destroy()
```

} methods of Servlets

String getServletInfo()

Servlet Config getServletConfig()

THU
26
MAY
2005

Priority

May 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

146-219, Week 21

init:

8 am This method is called by the Servlet Container when a new instance of a Servlet is created (i.e., the servlet where this init method is implemented)

Service:

This method is called on the initialized Servlet Instance for every request from the client for the Servlet

destroy:

is called before destroying the servlet

Mr. SANTOSH CLASS, NO. 5
Naresh Technologies
Ph: 23734842, 23746566

Example:

// Step 1

// HelloServlet

package com.nit.servlets.hello;

import javax.servlet.*;

import java.io.*;

public class HelloServlet implements Servlet

{

 System.out.println("In Init method");

}init

 public void service (ServletRequest req, ServletResponse res)

 throws ServletException, IOException

{

 System.out.println("In Service");

 PrintWriter out = res.getWriter();

Birthday / Anniversary

June 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Priority

FRI

27

MAY

2005

```

out.println("<b>Hello from first Servlet </b>");

}

public void destroy()
{
    System.out.println("Destroy");
}

public String getServletInfo()
{
    return "HelloServlet";
}

public ServletConfig getServletConfig()
{
    return null;
}

}


```

Mr. SANTOSH CLASS NOTES
 Naresh Technologies
 P.M. 23734842, 23746666

Deployment Descriptor (DD)

→ This file holds some information and instructions which are used by the container to manage the component's at runtime.

This is an XML format

Step 2: web.xml:

```

<Web-app>
    <Servlet>
        <Servlet-name> HS </Servlet-name>
        <Servlet-class>
            com.nit.servlets.HelloServlet
        </Servlet-class>
    </Servlet>
</Web-app>

```

SAT

Priority

28

MAY

2005

May 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

148-217 / Week 21

</Servlet>

8 am <Servlet-mapping>

<Servlet-name> HS </Servlet-name>

<url-pattern> /sayHello </url-pattern>

</Servlet-mapping>

</web-app>

Step 3 :-

11 Arrange the files in a directory structure given by Sun

12 1) place all the static resources into the <work folder> or
into any subdirectories of <work folder> (other than WEB-INF)

1 pm 2) place web.xml file into WEB-INF folder (where WEB-INF
should be in the <work folder>)

2 3) place all the Servlet class and other class files into the
WEB-INF\classes folder with the pack structure

3 4) If we have any .zip or .jar files containing the resources
then place them into WEB-INF\lib

<work folder>*.html....

WEB-INF\web.xml

WEB-INF\classes

• class files with the package dir structure

WEB-INF\lib*.zip, *.jar

To compile the Servlet

With Tomcat 5.x

Set class path : <tomcat home>\common\lib\Servlet-api.jar

Set class path = c:\tomcat 5\common\lib\Servlet-api.jar

June 2005						
S	M	T	W	T	F	S
	1	2	3	4		
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

Priority

SUN

29

MAY

2005

149-216: Week 21

To deploy the application into tomcat server

8am

Copy the <work folder> into <tomcat home>\webapps Folder

To start Tomcat Server

⇒ Set environmental variable

Name :- JAVA-HOME

value :- jdk home folder

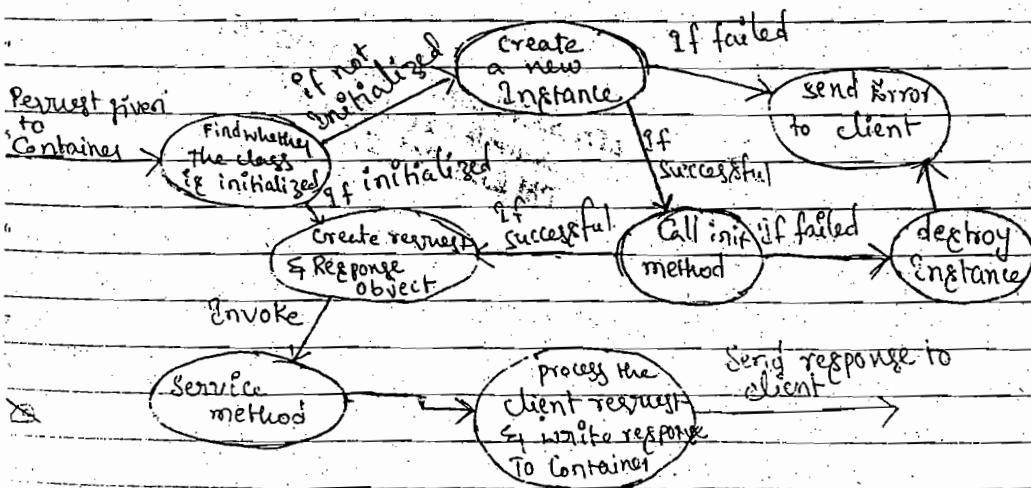
JAVA_HOME → c:\j2sdk 1.4.1

<tomcat home>\bin\startup.bat

1pm To call the Servlet

http://localhost:8080/hello/sayHello

<protocol>://<host name>:<port no>/<context root>/<url pattern>



MON
30
MAY
2005

Priority _____

May 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

150-215, Week 22

init

→ It's called after creating the new instance of the servlet and before making it available to process the client request (i.e., before calling the service)

→ This method carries a `ServletConfig` object i.e., Servlet Container tries to set the `ServletConfig` type of object into newly created `Servlet` instance.

Note :-

`ServletConfig` type of instance (which is created by the Container for the `Servlet` instance) will be passed into the `Servlet` instance only for one time throughout the lifetime of the `Servlet` instance.

So if this object has to be used with `service` method and destroy then we have hold it ⁱⁿ `init` method.

Service :-

→ This is called when a client makes a request for the `Servlet`.

It carries

`ServletRequest` and

`ServletResponse`.

Mr. SANTOSH CLAS
Naresh I Technologies
Ph: 23734842, 23740366

⇒ This method is not thread-safe

Note :-

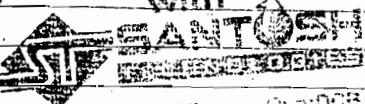
`ServletRequest` & `ServletResponse` objects are created separately for each of the request and are thread-safe.

June 2005
 S M T W T F S
 1 2 3 4
 5 6 7 8 9 10 11
 12 13 14 15 16 17 18
 19 20 21 22 23 24 25
 26 27 28 29 30
 151-214, Week 22

Priority _____

now
Mr. Santosh

with



SANTOSH
 TECHNOLOGIES

#2nd Floor, Samayati Hillayan, Opp: OCB Bank,
 SR Nagar Police Station Lane, Ameerpet, Hyd.

TUE

31

MAY

2005

Servlet Request

- 8am → Is an interface which allows us to get all the request related information (and is provided by the container)

Servlet Response

- 10 → Is used to set/configure the response content and the response related information

Destroy :

12 → Is used to perform some finalizations

Note:

1pm finalize method is not recommended to be used

javax.servlet.GenericServlet

2 → Is a adapter class

3 → Is an abstract class with one abstract method i.e., service method

4 → This class implements

javax.servlet.Servlet and

javax.servlet.ServletConfig interfaces

// Servlet (GenericServlet)

package com.nit.servletex;gs;

import javax.servlet.*;

import java.io.*;

public class M extends GenericServlet {

public void service(ServletRequest req, ServletResponse res) throws ServletException, IOException

Mr. SANTOSH CLASS NOTES
 Naresh / Technologies
 BM 23734842, 23746666

WED Priority

1

JUN

2005

June 2005						
S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

152-213, Week 22

{
8 am
printWriter out = res.getWriter();
out.println("Hello from GenericServlet Example ");
9 } // Service.
10 } // class
11

<!--web.xml-->
< web-app>
< servlet>
< servlet-name>gs</servlet-name>
< servlet-class>
com.nit.servlets.g.s.MyServlet
</servlet-class>
</servlet>
< servlet-mapping>
< servlet-name>gs </servlet-name>
< url-pattern>/myser</url-pattern>
</servlet-mapping>
</web-app>

GenericServlet

↳ init (ServletConfig)

⇒ takes the ServletConfig type of object passed as an argument to this method into an instance variable (private)

↳ call init with no argument method on the same instance (i.e., this)

Note:-

In GenericServlet init method is overloaded i.e. one declared in Servlet int init(ServletConfig) and init

Mr. SANTOSH
Naresh I.T. Technologies
Ph: 93734812, 23746550

July 2005						
S	M	T	W	T	F	S
1			2			
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Priority _____

THU

2

JUN

2005

- 2) Implementing all the methods declared in `ServletConfig` interface
 ↳ It uses the `ServletConfig` instance variable to implement the `ServletConfig` methods i.e. for example:-

```
public String getServletName()
{ }
```

```
return servletConfig.getServletName();
```

//Where `servletConfig` is the instance variable of type `ServletConfig`.

Mr. SANTOSH CLASS NOTES

Naresh Technologies

PH 23734842, 237406666

- 3) If we want to use the `ServletConfig` method implementation provided in `GenericServlet` then we should allow `init(ServletConfig)` method on `GenericServlet` to be executed

- 3) `destroy()` method is given with null implementation
 4) `getServletInfo()` method returning the `ServletName`
 5) `getServletConfig()` method returning the `ServletConfig` instance variable initialized in `init()` method

- To get the request data from the client's

`ServletContainer` takes the data from the protocol adapter and set them into the `ServletRequest` object

→ `ServletRequest` object represents the request related data

(i.e., using this object we can interact with Container and get some request related information)

FRI

Priority _____

3

JUN

2005

June 2005						
S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

154-211, Week 22

S	M
31	
1	2
10	11
17	18
24	25

155-2

Methods :-

- String getParameter (String)
- Enumeration getParameterNames()
- String[] getParameterValues (String)

Request Parameters :-

→ If a name-value pair send from the client

Note :-

- ⇒ we can pass any number of request level parameters depending on the capability of the protocol
- ⇒ we can associate more than one value for the same name (i.e., parameter name need not be unique)

```
<!--login.html-->
```

```
<html>
<body>
<form action="login">
<pre>
User name : <input type="text" name="uname"/>
password : <input type="password" name="pass"/>
<input type="Submit" value="LogIN"/>
</pre> </form>
</body> </html>
```

Mr. SANTOSH CLAS: NOTES
Naresh I Technologies
Ph: 23734842, 23746666

S	July 2005						
	S	M	T	W	T	F	S
1					1	2	
2	3	4	5	6	7	8	9
3	10	11	12	13	14	15	16
4	17	18	19	20	21	22	23
5	24	25	26	27	28	29	30

Priority

now
Mr. Santosh

SAT

4

JUN

2005

155-210, Week 22



// LoginServlet

```

8 am package com.nit.Servlets.myapp;
import java.sql.*;
import javax.servlet.*;
import java.io.*;

10 public class LoginServlet extends GenericServlet
{
    public void service(ServletRequest req, ServletResponse res)
        throws ServletException, IOException
    {
        String un = req.getParameter("uname");
        String pw = req.getParameter("pass");
        PrintWriter out = req.getWriter();
        if(un == null || " ".equals(un))
        {
            out.println("Username field should not be empty
<br/>");

            out.println("<a href = " + login.htm + ">Try again </a>");

            return;
        }
        if(pw == null || " ".equals(pw))
        {
            out.println("Password Field should not be empty
<br/>");

            out.println("<a href = " + login.htm + ">Try again </a>");

            return;
        }
        Connection con = null;
        Statement st = null;
        Result Set rs = null;
    }
}

```

SUN

Priority

5

JUN

2005

```
try {
    Con = DriverManager.getConnection("jdbc:oracle:thin:
        @localhost:1521:xit", "scott", "tiger");
```

```
St = Con.createStatement();
```

```
rs = St.executeQuery("Select * from User_details where
    UserID = '" + un + "' and pass = '" + pw + "');
```

```
if (rs.next())
```

```
{
```

```
    out.println("Valid User");
```

```
    return;
```

```
} //if
```

```
} //try
```

```
Catch (Exception e)
```

```
{}
```

```
finally
```

```
{
```

```
    try {
```

```
        Con.close();
```

```
} //by
```

```
Catch (Exception e) {}
```

```
} //finally
```

```
    out.println("Invalid User Details");
```

```
} //Service
```

```
public void init() throws ServletException
```

```
{
```

```
    DriverManager.registerDriver(new oracle.jdbc.driver.
        OracleDriver());
```

June 2005

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

156-209, Week 22

S	M	T
31		
1		
10	11	
17	18	
24	25	26

8am

1pm

Birthday / Anniversary

*Mr. SANTOSH CLASS NOTES
Naresh Technologies
Ph: 23734842, 23746666*

S	M	T	W	T	F	S
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

157-208, Week 23

Priority

do sigeypuwa configurations
with the available
DB for supernet
of photo etc...

MON

6

JUN

2005

8/try

8am catch (Exception e)

{

9 throw new ServletException ("Error while registering the drive");

10 } //catch

11 } //init

12 } //class

13 <!-- web.xml -->

14 <web-app>

15 <Servlet>

16 <Servlet-name> ls </Servlet-name>

17 <Servlet-class>

18 com.nit.Servlets.myapp.LoginServlet

19 </Servlet-class>

20 <load-on-startup>0</load-on-startup>

21 <!-->

22 This tag is used to give instruction to the servlet
Container to load and initialize the servlet at the time
when the context is loaded into the container
and the value it takes is order in which the
servlets has to be initialized.

23 -->

24 </Servlet>

25 <Servlet-mapping>

26 <Servlet-name> ls </Servlet-name>

27 <url-pattern> /login </url-pattern>

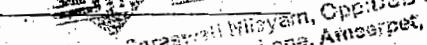
28 </Servlet-mapping>

Birthday Anniversary

<web-app>

Mr. SANTOSH CLASS NOTES
Naresh Technologies
FM 23734842, 23746666

now
with
SANTOSH



#2nd Floor, Saravalli Bypass, Opp DCB Bank
SR Nagar Police Station Lane, Amravati, Hyd.

TUE Priority

7

JUN

2005

June 2005						
S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

158-207; Week 2.3

To run this application in
<tomcat home>\bin\setClasspath.bat

edit the file

Find the line starting with

Set CLASSPATH =

for the existing classpath add the classpath required
for the driver i.e. for oracle oracledriver

1 pm ora 8.1

d:\oracle\8.1.7\jdbc\lib\classes12.zip

and save the file

Create table user_details (user_id varchar(20),
pass varchar(20)) ;

insert into user_details values ("user1", "user1 pass");

request parameters:

→ are read only

Contains sets the parameters which are passed from the
client

before giving the ServletRequest type of object to the
client

These are set at the moment the request is given and
will be destroyed after the response is sent to
the client

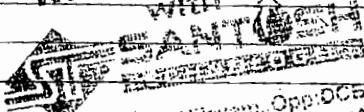
Mr. SANTOSH CLASS NOTES
Naresh Technologies
Ph. 23734542, 23746666

	S	M	T	W	T	F	S
1						2	
2						3	4
3						5	6
4						7	8
5						9	10
6						11	12
7						13	14
8						15	16
9						17	18
10						19	20
11						21	22
12						23	24
13						25	26
14						27	28
15						29	30

Priority _____

now
Mr. Santosh

with



#2nd Floor, Sanketam, Nizamuddin, Opp DCS Bank
ISR Nagar Police Station Lane, Ameerpet, Hyd.

WED

8

JUN

2005

⇒ every parameter has a name (i.e., without name parameter is not constructed)

⇒ for the same name more than one value can be attached

in this case use getParameterValues method to get all the values

and if we use getParameter method returns the first value of the given name

New method related to request parameters

in ServletRequest (P-3)

java.util.Map getParameterMap()

Mr. SANTOSH CLASS NOTES
Naresh | Technologies
Ph 93734842, 23746006

Map returned holds the name value pairs of the request parameter

The Map returned is immutable (read only)

Initialization parameters :-

⇒ the name-value pairs (parameters) which are given to the servlet at the time of initializing the servlet are known as Initialization parameters

Requirement of these type of parameters

⇒ If we have any data which is not known at the time of developing the application and known at the time of deploying the application (and the information is same for processing all the request for the servlet)

THU

Priority

9

JUN

2005

June 2005						
S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

160-265, Week 23

for example DB username and password

→ When the data is being frequently changed

In the above conditions we configured the init parameters
(this can be done at the time deploying the application)
and try to access the parameter values in the servlet

These parameters are given to the servlet at time when
the servlet is being initialized

And ServletConfig makes this initialization ~~parameters~~ ^{class notes}
available to the servlet

in javax.servlet.ServletConfig
interface

String getInitParameter(String)

Enumeration getInitParameterNames()

Note :-

Init parameter names should be unique
i.e., More than one values cannot be assigned to the
same name.

To → Init parameters are configured separately for each of the
servlet

To Configure Initialization parameters :-

in web.xml

Bengaluru Anniversary

Mr. SANTOSH CLAS NOTES
Naresh I Technologies
Ph: 23734842, 23746666

Birthday

July 2005

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

FRI

10

JUN

2005

161-204, Week 23

in <Servlet> tag after the <Servlet-class> tag

<init-param>

<!-- This tag declares one init parameter --> now
<param-name> </param-name> **Mr. Santosh**
<param-value> </param-value> with
</init-param>

<!--

any number of init params can be declared but
init param should be unique within the declaration

-->

Mr. SANTOSH CLAS, NOTES
Nare�i Technologies
98437334842, 23746666

Context Parameters:

⇒ If we want to configure the parameters which have to be used with all the web components (Servlets) within the context (Web module/web application)
Then these parameters are used

javax.servlet.ServletContext

⇒ It is an interface which allows us to interact with the Container to get some details related to the context i.e., the environment details where the Servlets are running.

⇒ This is implemented by the Container provider and initialized at time of loading the context into the server.

i.e. before making the context resources available

SAT Priority

11

JUN

2005

June 2005						
S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

162-203, Week 23

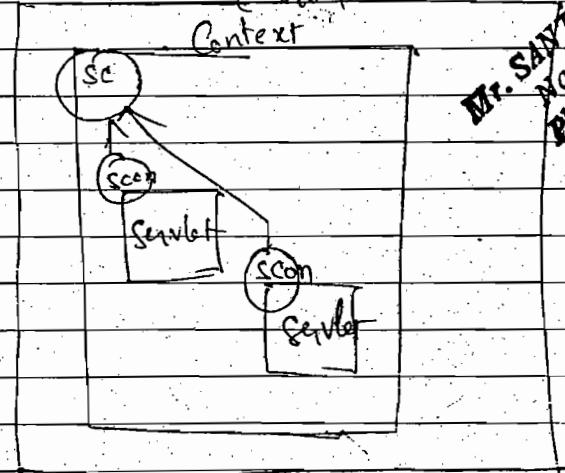
to the clients

This object reference is given to all the servlet instances at the time of initializing the servlet i.e., through the `ServletConfig` object

in `ServletConfig`

`ServletContext getServletContext()` method which can give the reference of the `ServletContext` i.e. `ServletContext` object which is initialized for the context in which the servlet is being initialized

`ServletContext` object can be used to get parameters



Mr. SAVITOSH CLAS NOTES
Naresh I Technologies, Gurgaon
Ph: 23734842, 23746666

Sc → Servlet Context
(Object)

Sc → Servlet Config

➤ `String getInitParameters (String)`

Enumeration

`getInitParameterNames()`

July 2005							
S	M	T	W	T	F	S	S
31					1	2	
1	2	3	4	5	6	7	8
8	9	10	11	12	13	14	15
15	16	17	18	19	20	21	22
22	23	24	25	26	27	28	29
29	30	163-202, Week 23					

Priorit Enumeration is available from JDK 1.6
it is an object

SUN

12

JUN

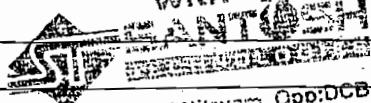
2005

Note :-

- Context parameters are read only
 - names should be unique within the context (web.xml)

To configure context parameter
in Web.xml

now
Mr. Santosh
with



<Context-param>
<param-name> </param-name>
<param-value> </param-value>
</Context-param>

Should be in web-app and before starting the servlet declarations

Servlet 1

```
package com.nil.servletex.params;  
import javax.servlet.*;  
import java.io.*;  
import java.util.*;
```

Mr. SANTOSH CLASS NOTES
Naresh I Technologies
Ph 23734842, 23746666

My public class ContextServlet1 extends GenericServlet

```
public void init (ServletConfig sc) throws ServletException  
{
```

```
System.out.println ("In init");
```

```
System.out.println ("Initialization Parameters for "+sc.get  
ServletName());
```

`java.util.Enumeration enum = sc.getInitParameterNames();`

```
while( enum.hasMoreElements() )
```

5

Birthday / Anniversary

```
Object o = enum.nextElement();
```

MON

13

JUN

2005

Priority

June 2005

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

164-297, Agha 24

String name = (String) o;

String value = sc.getParameter(name);

System.out.println(name + ":" + value);

//while

System.out.println("Context Parameters for " + sc.getServletName());

//init

ServletContext ctxt = sc.getServletContext();

Enumeration enum

enum = ctxt.getInitParameterNames();

while (enum.hasMoreElements())

{

String name = (String) enum.nextElement();

String value = ctxt.getInitParameter(name);

System.out.println(name + ":" + value);

//while

//init

public void service(ServletRequest req, ServletResponse res)

throws ServletException, IOException

{

PrintWriter out = res.getWriter();

System.out.println("In Service");

out.println("<html><body>");

out.println("<table border=2>");

out.println("<tr><th colspan=2>Request Parameters</th></tr>");

out.println("<tr><th> Name </th><th> Value </th></tr>");

Mr. SANTOSH CLAS, NOTE
Naresh Technologies
PH: 23734842, 23740666

July 2005						
S	M	T	W	T	F	S
31			1	2		
1	2	3	4	5	6	7
8	9					
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

165-200, Week 24

Priority

TUE

14

JUN

2005

8am
java.util.Enumeration enum = req.getParameterNames();
while (enum.hasMoreElements()) {
 {

String name = (String) enum.nextElement();
String values [] = req.getParameterValues(name);
out.println("<tr><td>" + name + "</td>" + "<td>");
for(int i=0; i<values.length; i++)
{
 out.print(values[i] + ",");
} // for
out.println("</td>");
out // while
out.println("</table></body></html>");
} // Service
} // class

<!-- web.xml -->

<web-app>

<context-param>

<param-name>cparam1</param-name>

<param-value>cparam1Value</param-value>

</context-param>

<context-param>

<param-name>cparam2</param-name>

<param-value>cparam2Value</param-value>

</context-param>

<servlet>

<servlet-name>MyServlet11</servlet-name>

<servlet-class>

Birthday / Anniversary

Mr. Santosh
now
with
SANTOSH
#2nd Floor, Suresh Bhawan, Opp: DCB Bank,
SR Nagar, Pimpri Chinchwad, Pune, Maharashtra, India.

MR. SANTOSH CLASS NOTES
Naresh J Technologies
Ph 23734842, 23740666

WED

Priority

15

JUN

2005

June 2005

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

166-199, Week 2:

com.nit.servletex.params.MyServlet
8am </Servlet-class>
<init-param>
<param-name>param1</param-name>
<param-value>param1Value</param-value>
</init-param>
</Servlet>
<Servlet>
<Servlet-name>MyServlet1</Servlet-name>
<Servlet-class>
com.nit.servletex.params.MyServlet2</Servlet-class>
1pm </Servlet>
<init-param>
<param-name>param2</param-name>
<param-value>param2</param-value>
</init-param>
</Servlet>
<init-param>
<param-name>param3</param-name>
<param-value>param3</param-value>
</init-param>
</Servlet>
<Servlet-mapping>
<Servlet-name>MyServlet1</Servlet-name>
<Url-pattern>/MyServlet1</Url-pattern>
</Servlet-mapping>

Mr. SANTOSH CLASS NOTES
Naresh I Technologies
P.M 23734842, 23740666

><Servlet-mapping>
<Servlet-name>MyServlet1</Servlet-name>
<Url-pattern>/MyServlet1</Url-pattern>
</Servlet-mapping>
</web-app>

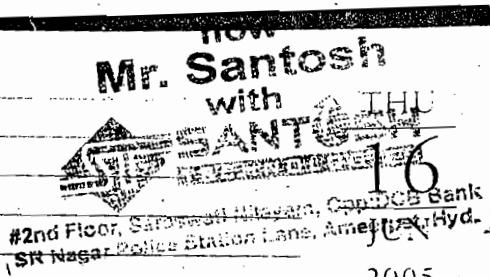
Jan 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

167-198, Week 2)

Priority

Swetha Vahns Reddy Degan

B.Tech



2005

<http://localhost:8080/param/MySeq11?reqparam=param1Value>

Sam

Dispatching the request from one Servlet to other:

Servlets 2 & onwards

Can use RequestDispatcher to dispatch the request from the Servlet to other resource

java.servlet.RequestDispatcher

=> is an interface with 2 methods include

forward (ServletRequest, ServletResponse)

forward (ServletRequest, ServletResponse)

In ServletContext

RequestDispatcher

getRequestDispatcher(String Uri)

RequestDispatcher getNamedDispatcher(String Name)

In ServletRequest

RequestDispatcher getRequestDispatcher(String Uri)

The getRequestDispatcher method declared in ServletContext takes the relative Uri to the context and it should start with /

And the method declared in ServletRequest takes the relative Uri to the request path or context path (if started with / it will be taken as relative to context path)

Birthday / Anniversary

FRI

Priority

17

JUN

2005

June 2005						
S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

168-197, Week 24

To make a RequestDispatcher object pointing to the resource in some other context then we have to get the ServletContext of the other context and then use getRequestDispatcher method on that context object

Note :-

It is not recommended and may not be supported by all the servers

→ Include :-

This is used to include the resource generated response into the requesting resource response

⇒ forward

This is used to forward the request and give the response generation responsibility to the other resource

Here with include the response and request processing is shared i.e., both the servlets can be involved whereas with forward request processing is shared but response generation has to be done, only by the resource to which the request has been forwarded

// Servlet 1

// Request Dispatcher Example

package com.nit.servlets;

```
import javax.servlet.*;
```

```
import java.io.*;
```

```
public class MyServlet1 extends GenericServlet
```

Mr. SANTOSH CHAS, NIT
Naresh Technologies
Ph: 23734842, 23740666

Birthday / At

S							
3	4						
10	11						
17	18						
24	25						

July 2005

Priority _____

31

1 2

3 4 5 6 7 8 9

10 11 12 13 14 15 16

17 18 19 20 21 22 23

24 25 26 27 28 29 30

169-196, Week 21

SAT

18

JUN

2005

public void service (ServletRequest req, ServletResponse res)
throws ServletException, IOException

{

String path = req.getParameter("path");

PrintWriter out = res.getWriter();

out.println("<i>Path

if (path == null || path.equals(""))

{

out.println("<i>Path should Not be empty</i>");

RequestDispatcher rd = req.getRequestDispatcher("Home.html");
rd.include(req, res);

return;

} // if

RequestDispatcher rd1 = req.getRequestDispatcher(path);

RequestDispatcher rd2 = getServletContext().getRequestDispatcher(path);

RequestDispatcher rd3 = getServletContext().getNamedDispatcher(path);

if (rd1 == null)

out.println("
 Path: " + path + " is not valid to locate the
resource using req.getRD()");

else

rd1.include(req, res);

if (rd2 == null)

out.println("
 Path: " + path + " is not valid to be
used with Context.getRD()");

else

rd2.include(req, res);

if (rd3 == null)

out.println("
 Path: " + path + " not valid to be

Birthday / Anniversary

Mr. SANTOSH CLAS NOTES
Naresh Technologies
PBM 23734842, 23746666

SUN

Priority

19

JUN

2005

June 2005						
S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

170-195, Week 24

used with Context.getND");

else

reqd.include(req, res);

//Service

//class

// MyServlet2

package com.nit.servletex.rd;

import javax.servlet.*;

import java.io.*;

public class MyServlet2 extends GenericServlet

{

public void service (ServletRequest req, ServletResponse res)

throws ServletException, IOException

{

PrintWriter out = req.getWriter();

out.println ("
");

out.println ("Response from MyServlet2," + "

//Service

//class

<html>

<body>

<form action = "/Servlets/Ser1">

<pre>

<input type = "text" name = "path"/>

<input type = "submit value = "Test"/>

</pre>

Path or Name : <input type = "text" name = "path"/>

Mr. SANTOSH C - AS
Naresh / Technol. Life
EM 23734842, 23740566

Birthday / Anniversary

Birthday

July 2005 Priority

S	M	T	W	T	F	S
31		1	2			
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

1-194, Week 25.

MON

20

JUN

2005

</pre>

Sam </body>

</html>

<!-- Web-xml -->

<web-app>

<servlet>

<servlet-name> mySrv1 </servlet-name>

<servlet-class>

com.nit.Servletex.rl.MyServlet1

</servlet-class>

</servlet>

<servlet>

<servlet-name> mySrv2 </servlet-name>

<servlet-class>

com.nit.Servletex.rl.MyServlet2

</servlet-class>

</servlet>

<servlet-mapping>

mySrv1 <!-- servlet-names -->

<servlet-name> /servlets /srv1 </servlet-name>

</servlet-mapping> <url-pattern> /servlets /srv1 </url-pattern>

<servlet-mapping>

<servlet-name> /servlets /mySrv2 /servlet-name>

<url-pattern> mySrv2 /servlets /srv2 </url-pattern>

</servlet-mapping>

http://localhost:8080/regDisp/Servlets/srv1

TUE

Priority:

21

JUN

2005

June 2005						
S	M	T	W	T	F	S
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

172 43 Week 25

July

Request Attributes :-

⇒ Are used to share the data between more than one
Servlets invoked under the same request

in ServletRequest

Mr. SANTOSH CLASS NOTES
Naresh Technologies
Ph: 23734542, 23746666

Mr. SANTOSH CLASS NOTES
Naresh Technologies
Ph: 23734542, 23746666

SetAttribute(String, Object)

ObjectgetAttribute(String)

removeAttribute(String)

Enumeration.getAttributeNames()

Note :-

- 1> attribute names should be unique
- 2> These attributes are thread-safe
- 3> These objects need not to be Serializable type

```
<!-- Home.html -->
<html>
<body>
<form action="getEmpDetails">
<pre>
Employee Number: <input type="text" name="eid" />
<input type="checkbox" name="view" value="table" />
Table <input type="checkbox" name="view"
      value="list" />. list
<input type="submit" value="Employee Details" />
</pre>
</form> </body>
</html>
```

July 2005						
S	M	T	W	T	F	S
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
173-192, Week 25						

Priority

WED

22

JUN

2005

//GetEmpDetails

```
8am package com.nit.servlets;
import javax.servlet.*;
import java.io.*;
import java.sql.*;
public class GetEmpDetails extends GenericServlet
{
    public void service(ServletRequest req, ServletResponse res)
        throws ServletException, IOException
    {
        Connection con = null;
        Statement st = null;
        ResultSet rs = null;
        try
        {
            con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:nit", getInitParameter("dbuser"), getInitParameter("dbpass"));
            st = con.createStatement();
            String s = req.getParameter("eid");
            try
            {
                Integer.parseInt(s);
            } catch (Exception e)
            {
                PrintWriter out = res.getWriter();
                out.println("EmpID should be number<br/><br/>");
                ResultDispatcher rd = req.getRequestDispatcher("Home.html");
                rd.include(req, res);
                return;
            }
        }
    }
}
```

MR. SANTOSH CLAS, NOTES
Naresh I Technologies
PM 23734842, 23746666

```
Birthday / Anniversary
```

THU

23

JUN

2005

Priority

June 2005

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

174-191, Week 25

3 //catch

9 rs = st.executeQuery ("Select * from emp where empno = " + s);
if (rs.next ())

{

10 res.setAttribute ("status", "found");

11 res.setAttribute ("name", rs.getString (2));

12 res.setAttribute ("job", rs.getString (3));

13 res.setAttribute ("sal", rs.getDouble ("sal"));

14 res.setAttribute ("sal", new Double (rs.getDouble ("sal")));

15 res.setAttribute ("deptno", new Integer (rs.getInt ("deptno")));

16 } //if

17 else {

18 res.setAttribute ("status", "notfound");

19 } //else

20 String view = res.getAttribute ("view");

21 if (view.equals ("")) || view.equals ("table"))

22 {

23 RequestDispatcher rd = res.getRequestDispatcher ("tableview");

24 rd.forward (res, res);

25 } //if

26 else {

27 }

28 RequestDispatcher rd = res.getRequestDispatcher ("listView");

29 rd.forward (res, res);

30 } //else

31 } //try

32 catch (Exception e)

33 {

34 RequestDispatcher rd = res.getRequestDispatcher ("home.html");

35 rd.include (res, res);

Birthday Anniversary

M. SANTOSH C-15
Naresh Techno-Solutions
Ph: 23734842, 23740066

July 2005						
S	M	T	W	T	F	S
31			1	2		
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

175-190, Week 25

Priority _____

FRI

24

JUN

2005

```

8am    //catch
       finally
       {
         try {
           con.close();
         }
         catch (Exception e)
         {
           //Finally
         }
       }
       //Service
       public void init() throws ServletException
       {
         try {
           DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
         }
         catch (Exception e)
         {
           throw new ServletException();
         }
       }
       //Destory
       //Class
       // TableView Servlet
       package com.nit.servlets;

       import javax.servlet.*;
       import java.io.*;

       public class TableViewServlet extends GenericServlet
       {
         public void service (ServletRequest req, ServletResponse res)
           throws ServletException, IOException
       }
  
```

Birthday / Anniversary

Mr. SANTOSH CLAS NOTES
Naresh, Technical Gids
Ph: 23734512, 23746566

now
Mr. Santosh
with
SANTOSH

#2nd Floor, Samayam Millayam, Opp:OCB Bank
SP Nagar Police Station Lane, Amarapet, Hyd.

SAT Priority

25

JUN

2005

June 2005

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

176-189, Week 25

StringBuffer sb = new StringBuffer("<html> <body>");

8am String status = (String)req.getAttribute("status");
if (status.equals("not found"))
9 {

10 sb.append("Employee given is not Available");

11 } //if

12 else

13 {

14 sb.append("<table border='1'>");

15 sb.append("<tr><th> Name </th>");

16 sb.append("<th> Job </th>");

17 sb.append("<th> Sal </th>");

18 sb.append("<th> DeptNo </th></tr>");

19 sb.append("<tr>");

20 sb.append("<td>" + req.getAttribute("name") + "</td>");

21 sb.append("<td>" + req.getAttribute("job") + "</td>");

22 sb.append("<td>" + req.getAttribute("sal") + "</td>");

23 sb.append("<td>" + req.getAttribute("deptno") + "</td>");

24 sb.append("</tr> </table>");

25 } //else

26 sb.append("

");

27 sb.append("Get The emp details:
28 Get ");

29 PrintWriter out = res.getWriter();

30 out.println(sb);

31 } //service

32 } //class

July 2005						
S	M	T	W	T	F	S
31			1	2		
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

177-188, Week 25

Priority

SUN

26

JUN

2005

<Web-app>

Sam

<Servlet>

<Servlet-name> ge </Servlet-name>

<Servlet-class>

com.nit.servlets.revatt.GetEmpDetails

</Servlet-class>

<init-param>

11

12

1pm

2

3

5

6

7

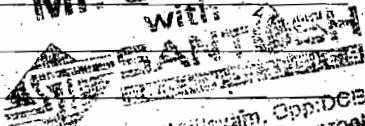
8

Birthday Anniversary

Mr. Santosh Class Notes
Mr. Santosh Tech Solutions
Ph: 9373515166

now
Mr. Santosh

with



#2nd Floor, Sankalpa Bhayyam, Oppides Bank
Shanagar Police Station Lane, Ameerpet, Hyd.

MON
27
JUN
2005

Priority

June 2005						
S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

178-187, W.C. 2

To get a ~~Context~~ ServletContext Object of the other context

Use getContext(String) method in ServletContext

It is not recommended to be used since all the Server may not support this feature

Http 1.1

- Hyper Text Transfer Protocol
- Is an application layer protocol
- Http is built on TCP/IP but is not mandatory to implement http on TCP only can use any transport layer protocol which can give the assurance of data transfer

The methods Supported by the Http 1.1 protocol

- Get
- Post
- Head
- Put
- Delete
- Trace
- Options

Get:

It is used to get data from the server i.e; resource located by the request-URI.

Post:

It is used to send some data to the resource located by the request-URI

It allows message-body content to pass, to the server

Mr. SANTOSH
Naraini
PHD 2013

July 2005						
S	M	T	W	T	F	S
31		1	2			
1	3	4	5	6	7	8
2	9	10	11	12	13	14
3	15	16	17	18	19	20
4	21	22	23	24	25	26
5	27	28	29	30		

Priority

High

17021806, Week 26

TUE

28

JUN

2005

(resource located by the request-uri)

Head:

It's same as get but it gets only the header information (like metadata) and will not get the message-body content

PUT:

It's used to update or put some resource into the server at a request-uri

If the resource is located using the given request-uri then it will be updated if not it creates a uri (given request-uri) pointing to the resource placed into the server.

Note:

We require privilege to perform this operation

Delete:

Remove the resource which is located by the request-uri

Trace:

It's used to get some information from the server about the loop back operation performed in the server to locate the resource

Options:

It's used to get the information like the options of the resource located by the request-uri

→ Http is assigned with 80 port number default but it's not mandatory to build http protocol to listen on 80 port only

Servlet Specification provides the support for http protocol

Servlet Container can understand http protocol and make all the http features available to the Servlet

WED

Priority _____

29

JUN

2005

SWETHA VAIHANA RELLY DEGARM

June 2005

S	M	T	W	T	F	S
		1	2	3	4	
		5	6	7	8	9
		10	11	12	13	14
		15	16	17	18	19
		20	21	22	23	24
		25	26	27	28	29
		30				

8-185, Week 26

javax.servlet.http

→ HttpServletRequest

- is an interface which extends ServletRequest

has some methods which can give some details about the http request

⇒ This is implemented by the Container provider

→ HttpServletResponse

- interface which extends ServletResponse

→ HttpServlet

- is an abstract class extending GenericServlet

- it doesn't have any abstract methods

- overrides the service method

protected void service(HttpServletRequest, HttpServletResponse)

throws ServletException, IOException

- defines do XXX methods for each of the http method like

doGet

doPost

public Service(HttpServletRequest, HttpServletResponse)

↓ if time uses http protocol

protocolService(HttpServletRequest, HttpServletResponse)

get method
doGet()

post method
doPost()

HttpServlet LifeCycle

July 2005						
S	M	T	W	T	F	S
31				1	2	
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
181-184, Week 26						

Priority _____

now
Mr. Santosh

with

SANTOSH

#2nd Floor, Samavedam Nilayam, Opp: DCB Bank
SR Nagar Police Station L, Interpol, Hyd.

THU

30

JUN

2005

// Example to test the lifecycle of HttpServlet

8am

```

package com.nit.Servletex;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

public class HttpServlet1 extends HttpServlet
{
    public void init(ServletConfig sc) throws ServletException
    {
        System.out.println("In init(sc)");
        super.init(sc);
    }
    public void init()
    {
        System.out.println("In init()");
    }
    public void service(ServletRequest req, ServletResponse resp)
        throws ServletException, IOException
    {
        System.out.println("In service (SReq, SRes)");
        super.service(req, res);
    }
    public void service(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
    {
        System.out.println("In service (HReq, HRes)");
        super.service(req, res);
    }
}

```

Birthday / Anniversary

FRI

Priority

1

JUL

2005

July 2005

S	M	T	W	T	F	S
31						
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

182-183, Week 24

8 am public void doGet(HttpServletRequest req, HttpServletResponse res)

throws ServletException, IOException

{

PrintWriter out = req.getWriter();

out.println("Response from doGet");

System.out.println("In doGet");

} //doGet

10 pm public void doPost(HttpServletRequest req, HttpServletResponse res)

throws ServletException, IOException {

System.out.println("In doPost");

PrintWriter out = req.getWriter();

out.println("Response from doPost");

} //doPost

1 pm <web-app>

<servlet>

<servlet-name>hs</servlet-name>

<servlet-class>

com.nit.servlets.HsHttpServlet

</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>hs</servlet-name>

<url-pattern>/myServlet</url-pattern>

</servlet-mapping>

</web-app>

Birthday / Anniversary

Birthday

MR. SANTOS
Narenkumar
phi 23734542
15/7/2005
66

August 2005						
S	M	T	W	T	F	S
2		1	2	3	4	5
9		8	9	10	11	12
16		14	15	16	17	18
23		21	22	23	24	25
30		28	29	30	31	

Priority _____

Swetha Venkata Reddy Duggay

SAT

2

JUL

2005

183-182, Week 26

<!-- Home.html -->

8 am <html>

<body>

<form action = "myServlets">

<pre>

10 <input type = "submit" value = "Make a Get Request" />

</pre>

</form>

11 <form action = "myServlets" method = "post">

<pre>

12 <input type = "submit" value = "Make a Post Request" /> </pre>

</form>

</pre>

</body>

</html>

Mr. SANTOSH
Naresh 11th ching
Ph: 23345123

ServletContainer calls service method with ServletRequest, ServletResponse arguments only if it doesn't call service or doXXX methods with HttpServletRequest and HttpServletResponse arguments.

Help Note :-

- in a html form if we don't use method attribute it takes get as default

- using hyperlinks we can make a get request only

SUN

5/3

JUL

2005

Priority _____

July 2005						
S	M	T	W	T	F	S
31				1	2	3
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

184-181, Week 26

HttpServletRequest :-

8 am

- `String getMethod()`

- returning the client request http method

- `String getHeader(String s)`

- returning the header value for the given name (gives the http request header information)

http request headers like → Accept-Language

→ Accept-Type

String getRequestedPath()

- returning the part of the request url

Example :-

http://nit10:8080/myctxt/mysen

Http/1.1/myctxt/mysen GET

returning /myctxt/mysen

MR. SANTOSH CHAUHAN'S NOTES
2009/10/2010/11/2011/2012/2013/2014/2015
PH: 23732522, 2140465

getRequestedStringBuffer getRequestedURI()

- it returning the complete URL which includes protocol, port number, host name, request uri

in the above example it returns

http://nit10:8080/myctxt/mysen

String getServletURI()

- returning the uri used to invoke the servlet

Example :-

for the above url it returns
/mysen

Birthday / Anniversary

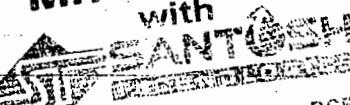
Birthday

S	August 2005						S
	S	M	T	W	T	F	S
1	2	3	4	5	6	7	8
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
23	24	25	26	27	28	29	30
29	30	31					

Priority

now
Mr. Santosh

MON



4

JUL
#2nd Floor, Saraswati Millayam, Opp:DCB Bank
SR Nagar Police Station Lane, Amaerpet, Hyd.
2005

HttpServlet Response :-

8am

setHeader(String, String)

-Used to set some http response headers
like refresh, location, content-type.....

refresh:

-It's a header used to give an instruction to the client side http adapter to make a request to the given url (or if url is not given then to the same current url) after the number of seconds given

refresh = <number of sec> ; URL = <URL>

if url is not given it takes the url which was generated by the response

// HttpServlet Example

package com.nit.servletex.hs;

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

public class MyServlet extends HttpServlet
```

```
public void doGet(HttpServletRequest req, HttpServletResponse res)
throws ServletException, IOException
```

/* we override the doxxx methods if we want to provide the service for the client specifically only for some http Methods or if we want provide the service in a different way for different http methods.

Mr. SANTOSH CLASS NOTES
Naresh Technologies
Ph: 23734842, 23745666

TUE Priority

5

JUL

2005

July 2005						
S	M	T	W	T	F	S
				1	2	
		3	4	5	6	7
		8	9	10	11	12
		13	14	15	16	17
		18	19	20	21	22
		23	24	25	26	27
		28	29	30		

186-179, Week 27

If we want to provide the service for all the http methods in a same way then override the protected service method of HttpServlet

*/

```
res.setContentType("text/html");
res.setHeader("refresh","15");
PrintWriter out = res.getWriter();
out.println("<b>");
out.println(new java.util.Date());
out.println("</b>");
```

}

&/class

Mr. SANTOSH C. A. NOTES
Naresh Techno Solutions
Ph: 23734842, 23740566

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
    <servlet>
        <servlet-name>hs</servlet-name>
        <servlet-class>
            com.naresh.servlets.HttpServletMyServlet
        </servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>hs</servlet-name>
        <url-pattern>/myse1</url-pattern>
    </servlet-mapping>
</web-app>
```

August 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	
8	9	10	11	12	13	
15	16	17	18	19	20	
22	23	24	25	26	27	
29	30	31				
187-178, Week 27						

Priority

WED

6

JUL

2005

Redirection the request :-

8am

⇒ Http response header 'Location' takes the url(relative or absolute)

When the client http adapter finds this header (i.e., in response) it uses the url and sends the request to the location.

With Servlets :-

→ in HttpServlet:Response

sendRedirect(String url)

- This method sets the location header of the http response with the given url

Example :-

/ctrl

/Ser1

/Ser2

in Servlet1 (i.e. /Ser1)

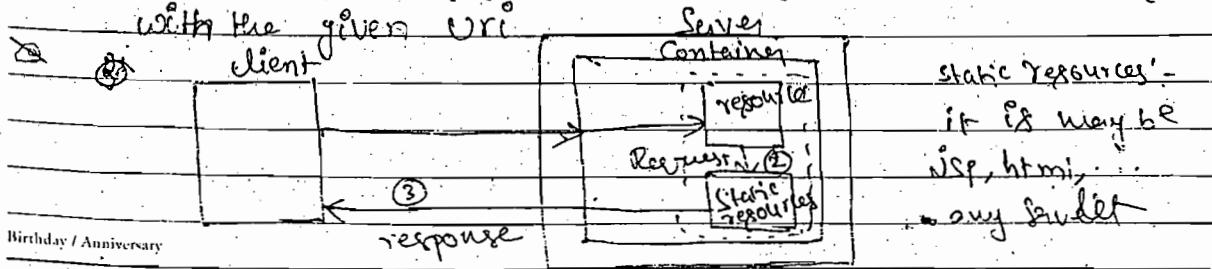
res.sendRedirect("Ser2");

- This Redirects the request given to the Servlet1 to the resource 'Ser2' relative to the request Uri

Mr. SANTOSH CLAS; NOTES
Nutan Technologies
PM 23734842, 23740666

Difference between forward & sendRedirect :-

> Request is shared with the forward option where as with sendRedirect it uses a new request to invoke the resource with the given Uri



Birthday / Anniversary

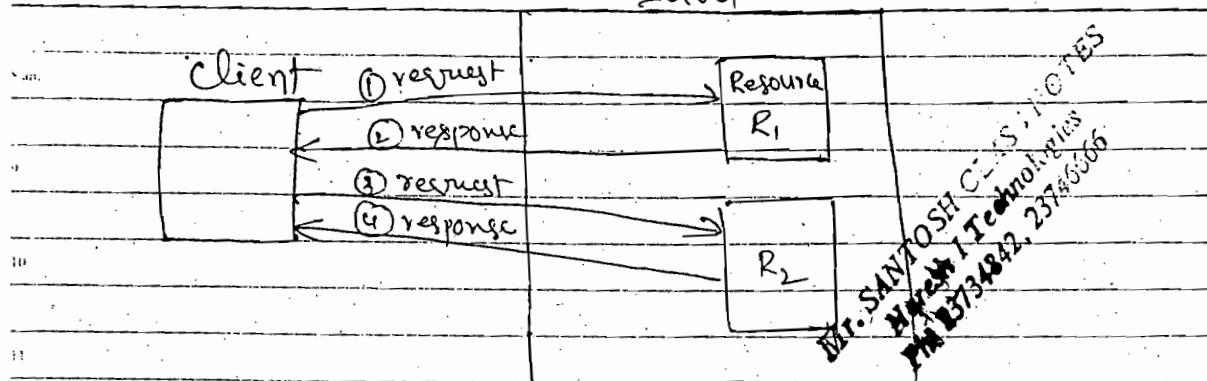
THU Priority ..

7 JUL
2005

July 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

188-177, Week 2nd

Server



② with sendRedirect 2 requests and 2 responses will be consumed where as with forward only one request and one response.

③ forward can be used with http and non http clients (it is a service provided by servlet container)

where as sendredirect can be used only with http clients (since it is an option provided by the http)

④ Using forward we can dispatch the request to resource (dynamic (servlet/JSP) or static) in the same context (env)

Note:- we have an option to dispatch the request to the resource in other context (but in the same server) but is not recommended to be used and may not be supported by all the servers

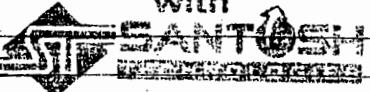
where as using sendRedirect we can redirect the request

to any resource (in the same environment or in a different environment i.e., it can be different server also)

But the destination resource should be allowed to be invoked using http

Mr. Santosh

with



FRI

8

JUL

2005

August 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Priority

#2nd Floor, Saraswati Nilayam, Opp. D.O.B. Bank
SR Nagar Police Station Lane, Ammaerpet, Hyd.

189-176, Week 27

forward instruction is executed in server

where as the redirect option is executed in client

- redirect option is set in the server

// Servlet 1

```
import com.nit.SimpleEx.Redirect;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class Servlet1 extends HttpServlet
{
    public void doGet(HttpServletRequest req, HttpServletResponse
req) throws ServletException, IOException
    {
        System.out.println("In Servlet");
        req.sendRedirect("Home.html");
```

15/09/05
Mr. SANTOSH CLAS
Naresh I Technolgies
PH 23734842, 23746366

Session Tracking

⇒ Session :-

- is uninterrupted set of requests and responses between the client server

- Session tracking deals with the technique used to maintain the client information (client given request data or may be the data obtained after processing the request data) within the session period)

Birthday / Anniversary

SAT

Priority _____

9

JUL

2005

July 2005						
S	M	T	W	T	F	S
31				1	2	
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	190-175, Week 2"				

8 am	1
9	2
10	3
11	4
12	5

8 am	1
9	2
10	3
11	4
12	5

1 pm	1
2	2
3	3
4	4
5	5

Birthday / Anniversary	1

Requirement

Want to hold the client data of one request and use it for the next requests from the client.

→ HTTP is a stateless protocol i.e., it cannot remember and identify the client and his request.

- 1) URL Rewriting
- 2) Hidden form fields
- 3) Cookies
- 4) HttpSession

Mr. SANTOSH CLAS,
Naresh Technologies
Ph: 23734842, 23746666

1) URL Rewriting :-

→ In this case whatever the data of client we want to maintain (i.e., use for the next requests)

We have set the data into URL's generated by the response and send it to the client.

So that when client uses the URL (i.e., to make a next request) the data whatever is required is sent as a request data.

Example :-

in LoginServlet

- take the user name / pass
validate it

if it is valid

 inbox "

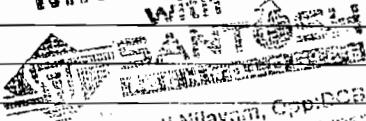
With http

http://<hostname>[:<portno>] /<request Uri> ?<query string>

August 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Priority _____

now
Mr. Santosh
with



SP Nagar Police Station, New Alipore, C.P. D.C.B. Bank
2nd Floor, 100ft Road, Alipore Lane, Amritsar, H.P.

SUN

10

JUL

2005

The above format is used for Get method of http

<query string> :-

<name>=<value> & <name>=<value>

Note :-

the query string with url should not be greater than 256 characters

Problems :-

1) Network traffic is increased

2) not secure

3) can be used Get method (and using it with the post method is difficult we have to use some generic patterns for the servlets)

4) cannot maintain huge amount of data (because of the limitation of the url length)

#

<!-- Login.html -->

<html><body><pre>

<form action = "login">

User name : <input type = "text" name = "uname" />

Password : <input type = "password" name = "pass" />

<input type = "Submit" value = "Login" />

</form></pre></body></html>

// LoginServlet

package com.nit.servlets; st;

import javax.servlet.*;

import javax.servlet.http.*;

Birthday / Anniversary

MON
11
JUL
2005

Priority

```
import java.io.*;
public class LoginServlet extends HttpServlet
{
    public void doGet(HttpServletRequest req, HttpServletResponse
    res) throws ServletException, IOException
    {
        LoginBean lb = new LoginBean("uname");
        String pass = req.getParameter("pass");
        if(lb.validate(uname, pass))
        {
            PrintWriter out = res.getWriter();
            out.println("<html><body><pre>");
            out.println("<a href='inbox?uname=" + uname + "'>");
            out.println("InBox </a>");
            out.println("<a href='compose?uname=" + uname + "'>");
            out.println("OutBox </a>");
            out.println("</pre></body></html>");
        }
        else
        {
            RequestDispatcher rd = req.getRequestDispatcher("Login.html");
            rd.forward(req, res);
        }
    }
    //doGet
}
```

5. // LoginBean

```
package com.nit.servlets;  
import java.sql.*;
```

public class LoginBean

Mr. SANTOSH CLASSES
Naresh I.Techinal 2-14566
Phi 23734542. 2-14566

August 2005						
S	M	T	W	T	F	S
	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Priority

TUE

12

JUL

2005

public boolean validate(String s₁, String s₂)

{

Connection con = null;

Statement st = null;

ResultSet rs = null;

try {

con = getCon();

st = con.createStatement();

rs = st.executeQuery("Select * from users where username = " + s₁ + " and pass = " + s₂);

return rs.next();

1 pm

}//try

Catch (Exception e)

{ Finally {

try {

con.close();

//try

Catch (Exception e) {} }

} Finally return false;

//validate

private Connection getCon() throws Exception

{

Class.forName("oracle.jdbc.driver.OracleDriver");

return

DriverManager.getConnection("jdbc:oracle:thin:

@localhost:1521:nit", "JES", "tiger");

//getCon

//class

Birthday / Anniversary

Mr. SANTOSH CLAS
Naresh IT Technologies
Ph: 23734842, 23746666

WED Priority

13

JUL

2005

July 2005						
S	M	T	W	T	F	S
31				1	2	
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

19-171, Week 28

//Inbox

```
package com.nit.servletex.st;
import javax.servlet.http.*;
import java.io.*;
public class InboxServlet extends HttpServlet
{
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
    {
```

```
    PrintWriter out = res.getWriter();
    out.println("Inbox Details of user");
    Parameter("uname");
}
```

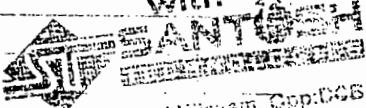
1/1/doGet
1/1/class

```
<!-- Web.xml -->
<!DOCTYPE...>
<web-app>
<servlet>
    <servlet-name>1s</servlet-name>
    <servlet-class>
    </servlet-class>
</servlet>
<servlet>
    <servlet-name>ib</servlet-name>
    <servlet-class>
        com.nit.servletex.InboxServlet
    </servlet-class>
</servlet>
<servlet-mapping>
```

Mr. SANTOSH CLAS. NOTES
Naresh J Techinal
PH: 23734522-24366

Mr. Santosh

with



THU

14

JUL

2005

#2nd floor, Saraswati Nilayam, Opp: DGB Bank
SR Nagar Police Station Lane, Amravati, Hyd.

August 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Priority

```
<Servlet-name> ls </Servlet-name>
<url-pattern> /login </url-pattern>
</Servlet-mapping>
<Servlet-mapping>
<Servlet-name> ib </Servlet-name>
<url-pattern> /inbox </url-pattern>
</Servlet-mapping>
</web-app>
```

12 To prepare a war
from work folder

```
<work folder> \ * [sub dir] *.html, *.gif ...
\WEB-INF\ web.xml
\WEB-INF\ classes\ * .class with pack
\WEB-INF\ lib\ * .jar, * .zip
```

work_folder >

jar -cvf <war name> . war *

Example :-

jar -cvf urlrewriting.war *

to deploy war into tomcat

> copy the war file into webapps

> deploy using ^{tomcat} manager console

in manager console

use Browse and select the war file

-deploy-

FRI

Priority

15

JUL

2005

July 2005

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				
Total Week 38.						

Create table users (uname varchar(20), pass varchar(10))

insert into users values('User1', 'User1pass');

To Configure Connection Pool in Tomcat Server

1) go to Tomcat Admin Console (admin/admin)
admin user

2) Select DataSource under the Resources

3) Select Create New DataSource

4) enter the required details

- save -

- commit changes -

Mr. SANTOSH
Nareswar
Phi 23734842
Date: 11/07/2005

To use the datasource and get the connection from the pool

```
javax.naming.InitialContext ic = new javax.naming.  
InitialContext();  
java.sql.DataSource ds = (java.sql.DataSource)  
ic.lookup("java:/myds");  
Connection con = ds.getConnection();
```

2) Hidden Form Fields

In this case we can set some html hidden form fields within a form so that when a form is submitted along with the other form fields these fields are also

submitted to the server

It is preferred to be used with the forms

```
<input type="hidden" name="" value="" />
```

Today's Anniversary

Birthday: April

Priority						
S	M	T	W	T	F	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

197-168, Week 28

August 2005

Svetha Vaishna Reddy (Descr)

SAT

16

JUL

2005

OTES

Mr. SANTOS
Narenthi
Ph: 23734842, 25746566

(10)):

// PrepareComposeServlet

```

package com.nit.servletex;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

public class PrepareCompose extends HttpServlet
{
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
    {
        String uname = req.getParameter("uname");
        res.setContentType("text/html");
        PrintWriter out = req.getWriter();
        out.println("<html><body>");
        out.println("<form action='SendMail' method='post'><per>");
        out.println("<input type='hidden' name='uname'"
                   " value='"+uname+"'/>");
        out.println("To : <input type='text' name='to' />");
        out.println("Subject : <input type='text' name='subject' />");
        out.println("Message : <input type='text' name='message' />");
        out.println("<input type='submit' value='Send' />");
        out.println("</pre></form>");
        out.println("</body></html>");
    }
}

```

//doGet

//SendMail Servlet

```

package com.nit.servletex;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

```

Birthday / Anniversary

SUN
17
JUL
2005

Priority

July 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

198 167 Week 28

August

public class MailServlet extends HttpServlet

{
 public void doPost(HttpServletRequest req, HttpServletResponse res)
 throws ServletException, IOException

 res.setContentType("text/html");

 PrintWriter out = res.getWriter();

 out.println("Mail Details:
");

 String to = req.getParameter("to");

 String sub = req.getParameter("subject");

 String message = req.getParameter("message");

 String from = req.getParameter("uname");

 out.println("To: " + to + "
");

 out.println("From: " + from + "
");

 out.println("Subject: " + subject + "
");

 out.println("Message: " + message + "
");

} // doPost

Mr. SANTOSH

Naresh Technologies

Ph: 23734842, 23740666

<Context docBase="d:\AdvJava_Proj\Servlets\urlrewriting">

 path="/mymail">

 <Resource name="myds1" type="javax.sql.DataSource"/>

 <ResourceParams name="myds1">

 <Parameter>

 <name>driverClass</name>

 <Value>

 oracle.jdbc.driver.OracleDriver

 </Value>

 </Parameter>

Today Anniversary:

Birthday Anniversary:

August 2005						
S	M	T	W	T	F	S
-	1	2	3	4	5	6
-	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Priority

కృష్ణారెడ్డి బస్తి

MON.

18

JUL

2005

199-166, Week 2nd

<parameters>

<name>url</name>

<value>

<idbc:oracle:thin:@localhost:1521:INIT

</value>

</parameter>

<parameters>

<name>User</name>

<value>scott</value>

</parameter>

<parameters>

<name>password</name>

<value>tiger</value>

</parameter>

</ResourceParams>

</Context>

copy the xml file (myctr.xml) into

<tomcat home>\conf\catalina\localhost

advantage over Url rewriting

1) We don't have a limit of 256 characters or we can use post method

2) Can be used with forms

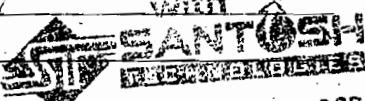
Problems

1) N/W traffic

2) Security

3) It is not recommended to maintain huge amount of data

Birthday / Anniversary

now
Mr. Santosh#2nd Floor, Saraswati Nilayam, Opp: EBC Bank,
SR Nagar Police Station Lane, Amserpet, Hyd.

MR. SANTOSH C-S NOTES
Naresh Technologies
Ph: 23734842, 2246666

TUE

Priority _____

19

JUL

2005

July 2005

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

2005-10-8 Week 2

To use the resource which is configured in the XML file

Sam

```
InitialContext ic = new InitialContext();
```

```
DataSource ds = (DataSource) ic.lookup("java:comp/env/myds");
```

```
Connection con = ds.getConnection();
```

10

Cookies:

→ is Small data unit which can be saved into the client machine

→ is Supported by http protocol

i.e. the client side adapter can take the instructions from the Server (i.e. in the response message) and save the data given (i.e. by the Server)

If we are programming the http ^{Servlet} for http protocol
then we can use this option

Cookie consist of

-- name

-- value

-- maximum age

-- path

-- is secure flag

-- domain name

Mr. SANTOSH CHATURVEDI
Mareesh I Tech 1st year 3rd sem
Ph: 23734542, 23734550

To present this cookie information to the servlet

Servlet Container uses

javax.servlet.http.Cookie

- is a non abstract class

Birthday / Anniversary

Birthday

August 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Priority _____

WED

20

JUL

2005

201-164, Week 29

HttpServletRequest

now
Mr. Santosh
with
SANTOSH

#2nd Floor, Saravalli Hillayan, Opp: IDBI Bank
SR Nagar Police Station Lane, Ameerpet, Hyd.

Cookie[] getCookies()

HttpServletResponse:

addCookie(Cookie)

Mr. SANTOSH CLAS. NOTES
Naresh I Technologies
Ph: 23734842, 276666

<!-- Home.html -->

<html><body>

<form action = "cookie">

<pre>

Name : <input type = "text" name = "cname" />

value : <input type = "text" name = "cvalue" />

<input type = "submit" value = "GetCookie" />

</pre> </form> </body></html>

//CookieServlet 1

package com.nit.servletex.st;

import javax.servlet.http.*;

 " " " " " ";

import java.io.*;

public class CookieServlet1 extends HttpServlet

{

 public void doGet(HttpServletRequest req, HttpServletResponse res)

 throws ServletException, IOException

{

 String name = req.getParameter("cname");

 String value = req.getParameter("cvalue");

 Cookie c = new Cookie(name, value);

Birthday Anniversary

THU

Priority

21

JUL

2005

July 2005

S	M	T	W	T	F	S
31				1	2	
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

202-168, Week 29

```
res.addCookie(c);
res.setContentType("text/html");
PrintWriter out = res.getWriter();
out.println("Cookie has been set");
}
// doGet
// class
```

// CookieServlet 2

```
package com.nit.servlets;
import javax.servlet.http.*;
import javax.servlet.*;
import java.io.*;
public class CookieServlet extends HttpServlet
{
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
    {
        Cookie c[] = req.getCookies();
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        for (int i=0; i<c.length; i++)
        {
            out.println("Name :" + c[i].getName());
            out.println("<br/>");
        }
    }
}
```

Mr. Santos
Narash
Phi 231252

August 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
203-162, Week 29						

Priority HashTable :- It is one of the map implementation of collection of API's

FRI
22
JUL
2005

Difference b/w the HashTable & HashMap is in HashTable Synchronizing the data.

<Context docBase = "d:\advjava\prg\Servlets\cookies" path = "/myCookies"/>

Advantages over URL rewriting and hidden form fields

- ⇒ The data is not required to be passed from the Server to Client in each of the response.
- ⇒ we don't require to follow the hyperlink or form submit to make the data available for the next request.
- ⇒ URL rewriting difficult to manage than compared to the cookies since it is more attached to the view.

DisAdvantages :-

Mr. SANTOSH
Naresh Technologies
ph: 23734842, 23746360

- ⇒ Client Side Support is required
- ⇒ Still Network traffic problem is reduced compared to the first 2 options.
- ⇒ Not Secure
- ⇒ Space is limited (depends on the browser and client settings)

To solve the above problems HttpSession can be used

⇒ Here the client data will be saved in the server and a unique identity is created for each of the client which has to be maintained using Cookies or URL rewriting.

Key is known as SessionID

The sessions are created and maintained by container and the identity for these sessions is also created by the container.

SAT

Priority _____

23

JUL

2005

July 2005

M	T	W	T	F	S
1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30

204-161, Week 29

- HttpServletRequest

8 am

HttpSession getSession(boolean)

HttpSession getSession()

Java.x.Servlet.HttpSession

is an interface which provides a standard abstraction between the Container and the Component to exchange the session data.

void setAttribute(String, Object)

Object getAttribute(String)

removeAttribute(String)

HttpServletRequest

getSession()

getSession(boolean)

When getSession method is called:

- 1) Container try to get the session id from the request it finds it in cookie if not found then try to get from the request uri

Note:-

The order (preference) of finding the session id can be changed (varies from one server to other)

If session id is found if container is able to locate the session then return the HttpSession type of object representing the state of the session

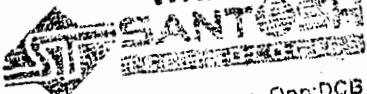
Birthday / Anniversary

August 2005						
F	S	S	M	T	W	TH
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				
2005-160, Weeks 2						

Priority

now
Mr. Santosh

with



#2nd Floor, Saraswati Nikayam, Opp:DCB Bank
SR Nagar Police Station Lane, Ameerpet, Hyd.

SUN

24

JUL

2005

if it is not found

- 8am \Rightarrow find whether the getSession method argument is true or false
- \Rightarrow If it is false then return null
- \Rightarrow If it is true then

- i) Create a new session
- ii) Create a unique identity for it (Session id)
- iii) Set the session id into cookie

Note: If the server/context is configured to use url rewriting for maintaining the sessionid then it may not set the cookie

iv) Return the HttpSession type of object representing the new session created

Note: if we use getSession with no argument then it is equivalent to using it with giving true as an argument

Note: ~~Bob~~

- 1) Session objects are not synchronized
- 2) Session attributes are recommended to be serializable type of objects

Mr. SANTOSH CLASS NOTES

Naresh i Technologies

Ph: 23734842, 23746360

Since HttpSession Object may be required to be serialized

1) With some servers when context is reloaded

2) With the servers maintained in cluster

Cluster is logical group of servers with an administrator

Server

If used for load balancing and to handle fail over conditions

MON

Priority

25

JUL

2005

July 2005						
S	M	T	W	T	F	S
31				1	2	
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

2005-159, Week 36

HttpSession

getting this type of object using getSession method of

HttpServletRequest

↳ boolean isNew()

returns true if the session is created in this request (or)

returns true if the session created is not yet accepted by the client

↳ getMaxInactiveInterval()

- returns the maximum time period for which the session will be maintained even if the client is inactive

setMaxInactiveInterval(int)

Note: if the argument value is less than '0' then the session will never expire (i.e., no inactive interval time)

↳ setAttribute(String, Object)

↳ Object getAttribute(String)

↳ removeAttribute(String)

↳ EnumerationgetAttributeNames()

putValue(String, Object)

Object getValue(String)

String[] getValueNames()

MR. SANTOSH
Notes 1/15/2005
PH 2/2/2005

The above 3 methods are deprecated and replaced with

setAttribute, getAttribute and getAttributeNames respectively (from Servlet 2.2 onwards)

ServletContext getServletContext()

Birthday / Anniversary ↗ return the ServletContext object to which the session

August 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Priority

TUE

26

JUL

2005

arounds 8th

is associated

8am

String getId()
- returning the Session ID

now
Mr. Santosh



10. <!-- Home.html -->
<html> <body> <pre>
<form action = "Servlet1">
Name: <input type = "text" name = "name" />
Value: <input type = "text" name = "value" />
<input type = "submit" value = "Add Session Attribute" />
</form> </pre> </body> </html>

// Servlet1

package com.nit.eibexample; hs;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

Mr. SANTOSH CLASS NOTES
Naresh I.Tutorial 11/11/06
Phi 23734S12, 231456

5. public class Servlet1 extends HttpServlet

6. public void doGet(HttpServletRequest req, HttpServletResponse res)
throws ServletException, IOException

7. {
HttpSession hs = req.getSession();
PrintWriter out = res.getWriter();
if(hs.isNew())

out.println("Welcome to the Session ID:" + hs.getId() + "
");

Birthday / Anniversary String name = req.getParameter("name");

String value = req.getParameter("value");

WED

Priority

27

JUL

2005

July 2005						
S	M	T	W	T	F	S
31					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

208-157, Week 30

```
hs.getAttribute(name, value); out.println("Attribute Added<br/>");  
out.println("<a href = " + name + ".html" > Add another attribute  
</a> <br/>");  
out.println("<a href = " + ser12 + "> View Session Attribute </a>  
<br/>");
```

311doge

class

// Servlet2

```
package com.mit.servlets;
```

```
import javax.servlet.*;
```

```
import javax.servlet.http.*;
```

```
import java.io.*;
```

```
public class Servlet2 extends HttpServlet
```

```
public void doGet(HttpServletRequest req, HttpServletResponse res)
    throws ServletException, IOException
```

```
PrintWriter out = res.getWriter(); HttpSession hs = req.getSession(false);  
String id = hs.getAttribute("Name");
```

if (hs == null)

1

```
RequestDispatcher rd = res.getRequestDispatcher("Home.html");  
rd.forward(req, res);
```

Mr. Williams (Rep), 117-118
and 910.

~~return;~~

11

while (enamel • hasMoreElements ())

[

```
out.println ("Name:");
```

Priority						
August 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

209-156, Week 30

THU

28

JUL

2005

```

String name=(String)names.nextElement();
out.println("name::Value:");
out.println(hs.getAttribute(name)+"<br/>");
} //while
out.println("<a href=\"Home.html\"> Add Another attribute</a>");
} //doGet
} // class

```

*Web application *

HttpServlet Response:

Mr. SANTOSH CLASS NOTES
Naresh Technologies
PM 23734842, 23746066

String encodeURL(String)

- finds whether the cookie is supported by the client or not
if it is supported then return the url which is given
as an argument

if not add the session id to the given url and return it

Note :-

if the session is New then it add the session id to the given
url and returning the url

Session ID has to be maintained

↳ Using cookies or

↳ URL Rewriting

to use it with URL Rewriting use encodeURL method

String encodeURL(String)

String encoduri (String) → deprecated

String ~~old~~ RedirectURL (String)

String Redirect Uri (String) → deprecated

FRI

Priority _____

29

JUL

2005

July 2005

S	M	T	W	T	F	S
31				1	2	
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

210-155 Week 30

// HomeServlet

```

package com.nit.servlets;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

public class HomeServlet extends HttpServlet
{
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
    {
        System.out.println("In Service");
        PrintWriter out = res.getWriter();
        out.println("<html><body>");
        String url = req.encodeURL("Ser/");
        out.println("<form action = \"" + url + "\">");
        out.println("Name:<input type = \"text\" name = \"name\"/>");
        out.println("<br/>Value:");
        out.println("<input type = \"text\" name = \"value\" /><br/>");
        out.println("<input type = \"submit\" />");
        out.println("</form></body></html>");
    }
}

```

Mr. SANTOSH CLAS: NO
Naresh Technologies
phi 23734542, 23746565

In Servlet (above program)

after

```

out.println("Attribute Added<br/>");
if(req.isRequestedSessionIdFromUrl())
{
}

```

```

out.println("<a href = \"" + req.encodeURL("home") + "\>")

```

Priority						
S	M	T	W	T	F	S
1	2	3	4	5		
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		
211-154, Week 30						

SAT

30

JUL

2005

Add another Attribute ``:

if

else

out.println(" Add Another Attribute
 ");

out.println("
 view Session Attributes ");

in Servlet2:-

after closing while loop
} //while

if (req.isRequestSessionIDFromURL())
{

url = resp.encodeURL("home");
} //else

url = "Home.html";

out.println("> Add Another Attribute ");

HttpSession

setInactiveInterval(int)

→ This is the time interval between the one request to another

→ invalidate()

This method invalidates the session

To set the session time out

in web.xml

Birthday / Anniversary After all the servlet-mapping

Mr. SANTOSH CLASS NOTES
Naresh Technologies
Ph: 23734842, 23746555

SUN
31
JUL
2005

July 2005						
S	M	T	W	T	F	S
31					1	2
	1	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	23	22	23
24	25	26	27	28	29	30

212-153, Week 30

<Session-Config>

< session-timeout > </session-timeout >

</session-config>

Session-timeout tag takes the number of minimum maximum it has to maintain the session in inactive state (i.e., the max period between the one request to the next request in a session)

=> This is applicable for all the sessions created in the context

Context Attributes

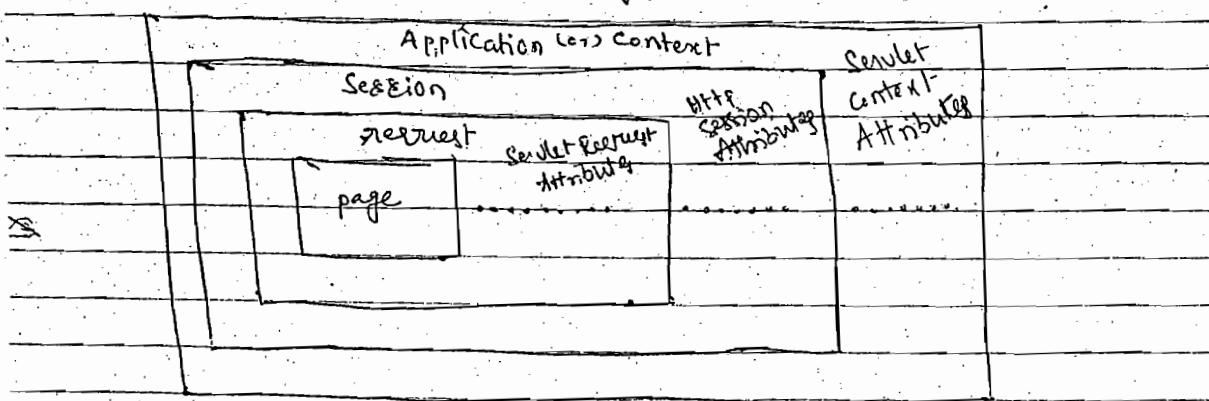
⇒ are used to maintain some objects which can be shared among all the servlet instances initialized under one context

javax.servlet.ServletContext

```
void setAttribute(String, Object)
```

Object.getAttribute(String)

void removeAttribute(String)



Birthday Anniversary

Birthday - Aug. 1

September 2005						
S	M	T	W	T	F	S
1	2	3				
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

213-152, Week 31.

Priority

now
Mr. Santosh

MON



SANTOSH
TECHNICAL SERVICES

1

#2nd Floor, Saraswati Nilayam, Opp:DCB Bank,
SR Nagar Police Station Lane, Attapur, Hyd.
2005

AUG

⇒ These attributes need not be Serializable

Sam C.i.e. these attribute values are never serialized)

⇒ These [redacted] objects are not thread safe.

// Example on Scopes

// Servlet1

package com.nit.Servlets.Scope;

import javax.servlet.*;

import javax.servlet.http.*;

import java.io.*;

public class Servlet1 extends HttpServlet

{

 public void service(HttpServletRequest req, HttpServletResponse res)
 throws ServletException, IOException

{

 System.out.println("In Service of Servlet1");

 req.setAttribute("name", "Request Attribute Value");

 HttpSession hs = req.getSession();

 hs.setAttribute("name", "Session Attribute Value");

 ServletContext sc = getServletContext();

 sc.setAttribute("name", "Context Attribute Value");

 System.out.println("Value Set");

 PrintWriter out = res.getWriter();

 out.println("In Servlet1
");

 RequestDispatcher rd = req.getRequestDispatcher("Ser1");

 rd.include(req, res);

}

}

Birthday / Anniversary

TUE

Priority

2

AUG

2005

//Servlet 2

```
package com.nit.servlets.scope;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

public class Servlet2 extends HttpServlet
{
    public void service(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
    {
        System.out.println("In Service method of Servlet 2");
        String req_attr = (String)req.getAttribute("name");
        HttpSession hs = req.getSession();
        String session_attr = (String)hs.getAttribute("name");
        PrintWriter out = res.getWriter();
        out.println("Request Attribute : "+req_attr);
        out.println("<br/>Session Attribute : "+session_attr);
        out.println("<br/>Context Attribute : "+cxt.getAttribute("name"));
    }
}
```

Servlet OutputStream

⇒ ~~Object~~ this type of object reference can be obtained using

getOutputStream() method of ServletResponse

ServletOutputStream is used if we want to present some binary data.

MR. SANTOSH
Narotam
Ph: 27345144

August 2005						
M	T	W	T	F	S	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

10-151 Week 31

Notes
SANTOSH
Narotam
Ph: 27345144

Priority						
S	M	T	W	T	F	S
6			1	2	3	
13	4	5	6	7	8	9
20	11	12	13	14	15	16
27	18	19	20	21	22	23
	24	25	26	27	28	29
	30					
215-150, Week 34						

WED

3

AUG

2005

Where of PrintWriter (getWriter()) is used if we want to generate character based response

// Example to Use ServletOutputStream

```
package com.nit.servlet.SOS;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class Servlet extends HttpServlet
```

```
{ public void service (HttpServletRequest req, HttpServletResponse res)
    throws ServletException, IOException { }
```

```
res.setContentType ("image/gif");
```

```
// get the Connection Statement and Select the Block column
```

```
ServletOutputStream sos = res.getOutputStream();
```

```
if (qs.next ())
```

```
{
```

```
InputStream is = qs.getBinaryStream ("img");
```

```
int i = is.read();
```

```
while (i != -1)
```

```
{
```

```
sos.write (i);
```

```
i = is.read();
```

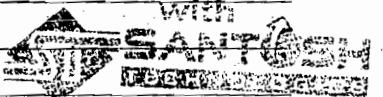
```
} // while
```

```
is.close();
```

```
} // if
```

Mr. Santosh

WITH



#2nd Floor, Saraswati Nilayam, Opp:DCB Bank
SR Nagar Police Station Lane, Ameerpet, Hyd.

THU

Priority

4

AUG

2005

August 2005

S	M	T	W	T	F	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

216-149, Week 31

8 am

9

10

11

12

1 pm

2

3

4

5

6

7

Birthday / Anniversary

Mr. SANTOSH C. S. S. I. T. C. E.
Naresh Patel, Technical
Phi 23734842, 23746566

```
Connection con = null;
Statement st = null;
ResultSet rs = null;
try {
    Con = DriverManager.getConnection ("jdbc:oracle:thin:@localhost:1521","scott","tiger");
    st = Con.createStatement ();
    rs = st.executeQuery ("Select * from emp1 where empno = "
        "+ nvl.getParameter ("empno"));
    if (rs.next ()) {
        rs.setContenType ("image/gif");
        "Before page";
    }
    nvl.setContentType ("text/html");
    PrintWriter out = nvl.getWriter ();
}
```

September 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Priority

FRI

5

AUG

2005

In a request we can use only `getOutputStream()` or `getWriter()` any one of them

SingleThreadModel :-

→ is an marker interface which tells the container to use the servlet instance by only one thread at a time
i.e., service method is invoked under the synchronization environment

Note :-

Is not recommended to be used
In Servlets 2.4 it is deprecated

Instead if we have to execute any statements in synchronize mode
then use synchronize blocks

// Single Thread Model Example

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
```

```
public class STM extends HttpServlet implements
SingleThreadModel
{
    public void service (HttpServletRequest req, HttpServletResponse
    res) throws ServletException, IOException
    {
```

```
        System.out.println ("In service method:" + inst_no);
        try {
```

```
            Thread.sleep(5000);
        } //try
```

Mr. SANTOSH CLAS, NOTES
Naresh I Technologies
PHI 23734842, 23746660

SAT

Priority

6

AUG

2005

August 2005

S	M	T	W	T	F	S
	1	2	3	4	5	
	6	7	8	9	10	11
	12	13	14	15	16	17
	18	19	20	21	22	23
	24	25	26	27	28	29
	30	31				

218-177, Week 31

catch (Exception e)

{ }

```
System.out.println("In service after sleep:" + inst_no);
req.getWriter().println("Response from servlet" + inst_no);
} // service
```

```
public void init()
```

{

no_inst++;

inst_no = no_inst;

```
System.out.println("In init:" + inst_no);
} // init
```

```
static int no_inst;
```

```
int inst_no;
```

} // class

Mr. SANTOSH C. AS
Naresh I Technol. Bios
P.M. 23734842, 23746566

Security :-

- JAAS (Java Authentication and Authorization Service)
- It provides a grained fine grained, user base security model

Authentication :-

- It is a process of finding whether the security details given by the user are valid or not.

Priority						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					
2005 Week 34						

SUN

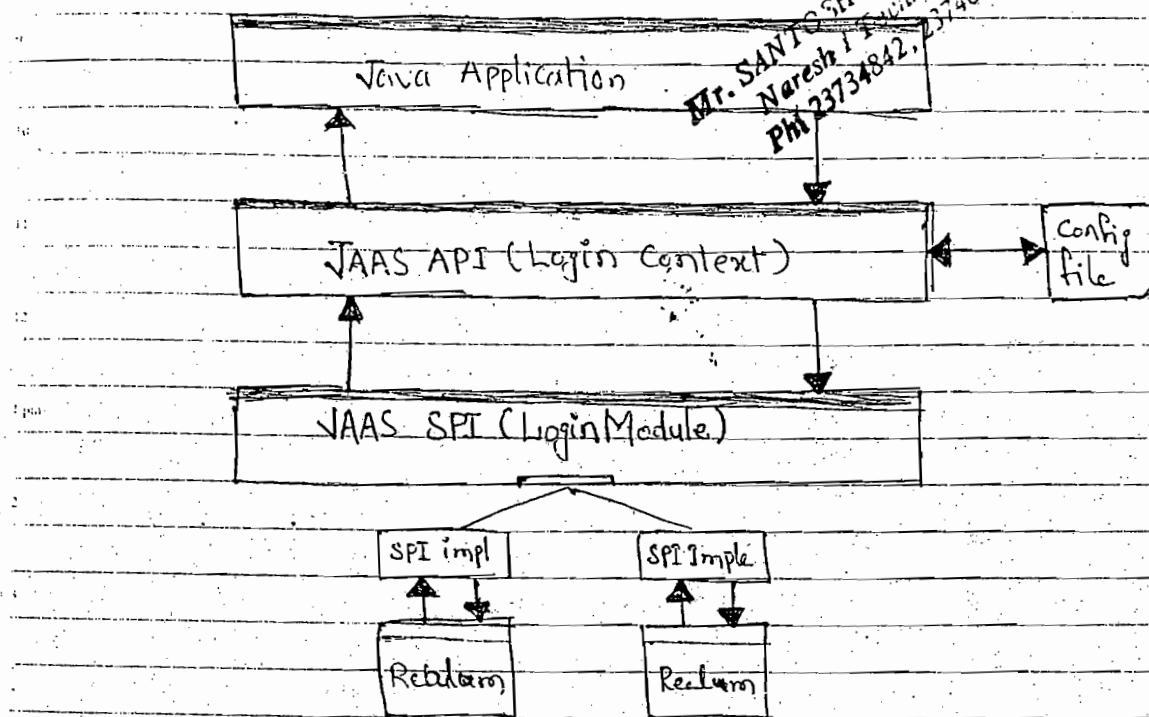
7

AUG

2005

pluggable Authentication Module (PAM) framework

Mr. SANTOSH CHAVAS
Naresh Iyer
Ph: 23734842, 23746366



Authorization

The process of finding whether the authenticated user has a privileges to access the resource or not.

Authorizer has to depend on Identity created by the authenticator.

Authorizer requires to know about the access privileges for the user i.e. it should be submitted with the ACE.

MON

Priority

8

AUG

2005

August 2005

S	M	T	W	T	F	S
	1	2	3	4	5	6
	7	8	9	10	11	12
	13	14	15	16	17	18
	19	20	21	22	23	24
	25	26	27	28	29	30
	31					

20-1-05, Week 32

Web Application

⇒ every server provides a Http Authenticator
which contains/Server can interact with the Authenticator
to authenticate the client

Authentication is performed when client makes a request for
the privileged services and if he is not authenticated

where as authorization is done for every request to the
privileged service

Mr. SANTOSH
Naresh, I Tech, 23-07-05
phi 23734842, 23-07-05

Web Application Security :-

⇒ Http Authenticator

- is an authenticator which can validate the client based on the
username/password or the client certificate

Http Authentication is used to take the details like username/password
from the client to validate

(i.e., JAAS (Authenticator) for web application uses this model)

1) BASIC

2) FORM

3) client cert

⇒ in web.xml

<Security-constraint>

<Web-resource-collection>

<Web-resource-name> myresources </web-resource-name>

<url-pattern> /* </url-pattern>

</Web-resource-collection>

<auth-constraint>

Priority						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

221-TUE, Week 32

September 2005

TUE

9

AUG

2005

now
Mr. Santosh
with

SANTOSH
TECHNOLOGIES

#2nd Floor, Sarastan Millayam, Opp:DCB Bank
Sri Nagar Police Station Lane, Ameerpet, Hyderabad

MR. SANTOSH CLAS NOTES
Narenji Technologies
Phi: 23734542, 23746566

<role-name> role1 </role-name>
</auth-constraint>
<login-config>
<auth-method> BASIC </auth-method>
</login-config>
<!-- declare a role -->
<security-role>
<role-name> role1 </role-name>
</security-role>

To assign user identifier

i) with tomcat

using console we have to create the roles in server with the names same as our security role names

ii) Create users and add them to the roles

write the role and user tags into the tomcat-users.xml file

2) with weblogic server

weblogic have to write weblogic.xml file (should be in web-int)

<security-role-assignment>

<role-name> role1 </role-name>

<principal-name> </principal-name>

<security-role-assignment>

Note: for each of the security-role tag declared in web.xml there should be one

<security-role-assignment>

WED *Priority*

10

AUGI

12005

August 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	
8	9	10	11	12	13	
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

122 133 Week 3

If we want to take user details using our own page (i.e. the login page wants to be in the format in which we want)

The login page should have one form with 2 input fields and one submit button.

Whege

forms action → "n: security-check"

Username field name → *i*-username

password field name → in password

in web.xml

<login-config>

<auth-method> FORM </auth-method>

<form=login-config>

<form = login - page> / Login.html </form = login - page>

<Form-error-page>/Error.html</Form-error-page>

</form-login-Config>

</login-Config>

```
<!-- login.html -->
```

<html>

<body>

<form action="a-security-check">

<pre>

UserName: <input type="text" name='username' />

PassWord: <input type="password" name="password"/>

```
<input type="Submit" value="Log In"/>
```

</pre><form> </body></html>

September 2005						
S	M	T	W	T	F	S
			1	2	3	
		4	5	6	7	8
		9	10	11	12	13
		14	15	16	17	18
		19	20	21	22	23
		24	25	26	27	28
		29	30			

22nd Week 32

Priority

THU

11

AUG

2005

<!-- Error.html -->

In Valid User

In HttpServletRequest

Principal java.security.Principal getUserPrincipal()

- returning the Principal object which is associated to the client

java.security.Principal String getName()

- returning the Principal name (Identity)

boolean isUserInRole(String roleName)

- takes the role name and returns true if the user is in the given role

* * * * *

Mr. SANTOSH
Nareswar Technologies
B.M. 23734842, 23740666

Applet To Servlet

→ With Applet if we want to connect to the different host (i.e., other than the host from where the applet is loaded) it requires security privileges (and may not be able to connect)

Here we can have a Servlet running in the host from where the applet is loaded and can behave as a proxy

in Applet

-- get the java.net.URL object pointing to the Servlet address

- use URLConnection (HttpURLConnection) to exchange the data i.e., UC.getOutputStream() & UC.getInputStream()

FRI Priority _____

12 AUG
2005

August 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	6
	7	8	9	10	11	12
	13	14	15	16	17	18
	19	20	21	22	23	24
	25	26	27	28	29	30
	31					

22-141, Week 32

// Applet

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class Myapplet extends Applet implements ActionListener
{
    Button b; TextField tf;
    public void init()
    {
        b=new Button ("Get Value");
        tf = new TextField();
        b.addActionListener(this);
        add(tf); add(b);
    }
    public void actionPerformed (ActionEvent ae)
    {
        try {
            java.net.URL u= new java.net.URL("http://nit:8080/mytxt/");
            String name = " "+tf.getText();
            URLConnection uc = u.openConnection();
            uc.setDoInput(true);
            DataInputStream dis= new DataInputStream(uc.getInputStream());
            message = dis.readLine();
        } catch {
            Catch (Exception e)
            {
                message = e.getMessage();
            } catch
            repaint();
        }
    }
}
```

Mr. SANTOSH CHS, NOTES
Naresh, Technical
Ph 23734842, 23745366

September 2005						
S	M	T	W	T	F	S
1	2	3				
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

225-140, Week 32

Priority _____

now
Mr. Santosh
with
SANTOSH

SAT

13

AUG

2005

#2nd Floor, Sarvodaya Nilayam, Opp. DCB Bank
SR Nagar Police Station Lane, Ameerpet, Hyd.

public void paint (Graphics g)

g.drawString (message, 100, 100);

} //paint

String message = " ";

} //class

Mr. SANTOSH CLASS NOTES
Naresh Technologies
Ph 23734842, 23746366

//Servlet

import javax.servlet.*;

import java.io.*;

public class MyServlet extends GenericServlet

{

public void service (ServletRequest req, ServletResponse res) throws
ServletException, IOException

{

String message = req.getParameter ("name");

System.out.println ("Message" + message);

PrintWriter out = res.getWriter();

out.println ("Response from Servlet Message : " + message);

}

<!-- Home.html -->

<applet code = "MyApplet" width = "300" height = "300">

→ Servlet to Servlet Communication

in ServletContext methods

Servlet getServlet (String)

Servlet[] getServlets ()

Birthday / Anniversary

can be used to get the other Servlet instance and call some

SUN

Priority

14

AUG

2005

August 2005

S	M	T	W	T	F	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

226-1391 Week 32

method on them

These 2 methods cause the performance effect and security problem, hence it is not recommended to be used and is deprecated from Servlet.

Now the Container returns null if we use the above methods.

Servlet changing:

With previous version of Servlet Container we used to give the sequence Servlet to a single URL-pattern

Example - /myself → Servlet, Servlet, JSP

Where Container used to invoke all the given Servlets in the same sequence and send the response

The above two can be achieved by using RequestDispatcher where RequestDispatcher was introduced in Servlet 2.1

See Servlets P.3

Mr. SANTOSH CHAVADE
Naresh Technologies
Ph: 23734842, 23746660

Filter:

→ A new component introduced in Servlet 2.3 Specification, which is used to intercept the request to a Servlet or JSP

Use of Filter

→ It can be used for pre processing

-- It can read the request and perform some validation like Session validation and Session data Validation

-- The request content type checking

i.e., finding whether the content type used by the client is supported by our application to process the request or not

-- Check character encoding

-- Locale Management (I18N)

September 2005						
S	M	T	W	T	F	S
	1	2	3			
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	
22	23	24	25	26	27	28
Week 33						

Priority

MON

15

AUG

2005

⇒ It can be used to perform some post processing i.e., after the Servlet/JSP completes the response generation.

-- Translating the response i.e., we can use XSLT and convert the response into different types.

Like a single servlet can be used to generate a plain XML output and a filter can translate (with the help of XSLT) into WML if it is a request from mobile clients and HTML if it is a request from browser based clients.

-- Response data compression (can improve the download capacity)

Note:-

Filters are not generally used for generating the response but they can manipulate the response generated by Servlet or JSP -- Can be used to attach the part of the response to the response generated by Servlet/JSP.

Javaee.Servlet.Filter

-- is an interface

-- init

⇒ is called when a Filter instance is created

-- doFilter

is called when a request is being made for a servlet/JSP to which the Filter is being attached

Note:-

one filter can attach to any number of servlets/JSP

one Servlet/JSP can be attached with any number of filters

-- destroy

is called before destroying the Filter (Before GC)

Mr. SANTOSH CLASS NOTES
Naresh Technologies
Ph: 23734842, 23740666

TUE

Priority

16

AUG

2005

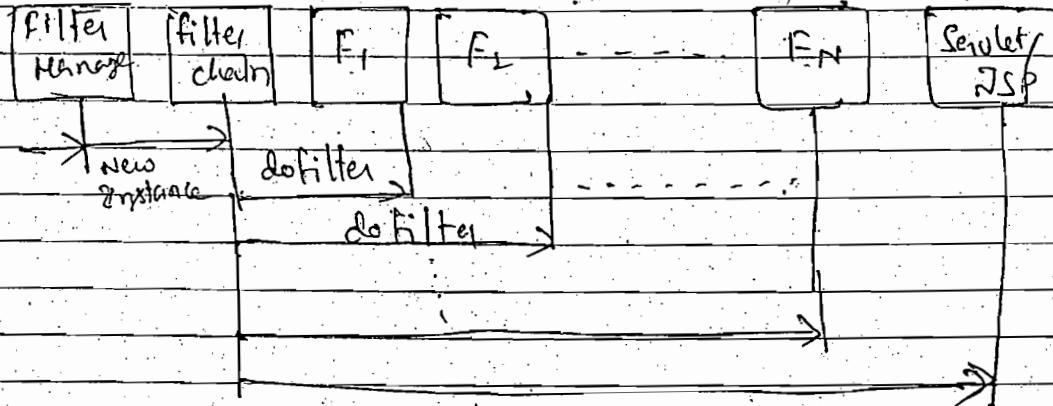
August 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

228-137; Week 33

public class MyFilter implements Filter

```
public void doFilter(ServletRequest req, ServletResponse res,  
                     FilterChain fc) {  
    // pre processing  
    fc.doFilter(req, res);  
    // post processing  
}
```

Mr. SANTOSH CLAS-107
Naresh Tech. ICS
PMI 23734S12, 2374066



In a Filter's doFilter method

if any statements has to be executed as a pre processing
then write all the statements before the fc.doFilter method
is called

If post operations are there then after the fc.doFilter

doFilter method on FilterChain instance is called
to dispatch the request to the next Filter/Server or

Birthday Anniversary JSP

September 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	229-130, Week 3.5			

Priority rules for writing a filter

1) Should be a subtype of javax.servlet.Filter

2) Should be a public non-abstract class

3) Should have a no argument constructor

Note: is not recommended to write the constructor

and to perform initialization use init method

WED

17

AUG

2005

// Filter

```

8. package com.nit.servletex.filter;
9. import javax.servlet.*;
10. public class MyFilter1 implements Filter {
11.     public void init(FilterConfig fc) {
12.         System.out.println("MyFilter1 : init");
13.     }
14.     public void doFilter(ServletRequest req, ServletResponse res)
15.             throws FilterException,
16.             java.io.IOException {
17.         System.out.println("Pre processing of MyFilter1");
18.         fc.doFilter(req, res);
19.         // The above statements dispatches the request to next component
20.         // i.e. Filter or Servlet
21.         System.out.println("Post Processing of MyFilter1");
22.         fc.doFilter();
23.     }
24.     public void destroy() {
25.         System.out.println("MyFilter1 : destroy");
26.     }
27. }
```

Mr. SANTOSH CLASS NOTES
Nareshta Technologies
FM 23734842, 23740666

// Servlet

```

28. package com.nit.servletex.filter;
29. import javax.servlet.*;
30. import java.io.*;
31. public class TestServlet extends GenericServlet {
32.     public void service(ServletRequest req, ServletResponse res)
33. }
```

THU Priority _____

18
AUG
2005

August 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	6
	7	8	9	10	11	12
	13	14	15	16	17	18
	19	20	21	22	23	24
	25	26	27	28	29	30
	31					

250-135, Week 3x

throws ServletException, IOException

```
    System.out.println("In Servlet Service");
```

```
    PrintWriter out = req.getPrintWriter();
```

```
    out.println("Response from Test Service");
```

```
}
```

```
}
```

```
<!-- web-app -->
```

```
<filter>
```

```
<filter-name>f1</filter-name>
```

```
<filter-class>com.nit.servlet.filter.MyFilter</filter-class>
```

```
</filter>
```

```
<filter>
```

```
<filter-name>f2</filter-name>
```

```
<filter-class>com.nit.servlet.filter.MyFilter</filter-class>
```

```
</filter>
```

```
<filter-mapping>
```

```
<filter-name>f1</filter-name>
```

```
<url-pattern>/s1</url-pattern>
```

```
</filter-mapping>
```

```
same f
```

```
<Servlet>
```

```
<Servlet-name>S1</Servlet-name>
```

```
<Servlet-class>com.nit.Servlet.filter.TestServlet
```

```
</Servlet-class>
```

```
<Servlet>
```

```
2 same
```

10th Anniversary

Birthday /

Mr. SANTOSH
Narashri
Phi 23734522274666

Mr. SANTOSH
Narashri
Phi 23734522274666

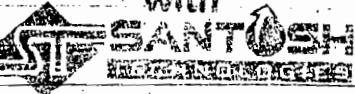
September 2005						
S	M	T	W	T	F	S
	1	2	3			
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

231-134, Week 33

Priority

now
Mr. Santosh

with



#2nd Floor, Saraswati Nilayam, Opp: DCS Bank
SR Nagar Police Station Lane, Ameerpet, Hyd.

FRI

19

AUG

2005

⇒ Filter Chain locates the filters which should be invoked for that request i.e., identify the filters which has to intercept the request. This is done by mapping the URL pattern of the servlet or the servlet name i.e. the filters are located with the help of the request URL or servlet name.

<filter-mapping>

<filter-name></>

<url-pattern></>

or

<servlet-name></>

<filter-mapping>

if servlet name is used then this filter is applied to the request made by any URL pattern of the servlet (whose name is given).

(i.e., if we have more than one mappings for the same servlet then for all the URL patterns of the servlet, this filter will be applicable.)

And determine the order in which the filters has to be invoked based on the order of these mappings.

In Servlet 2.3 Filters used to intercept only the direct request to the servlet/JSP from client, but it was not able to intercept the request which is forwarded or included.

In Servlet 2.4

we can tell the Filter chain/filter manager when

MR. SANTOSH'S NOTES
Naresh I.T. Technologies
Ph: 23734842, 23746666

SAT

Priority

20

AUG

2005

August 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

232-133 Week 33

September						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

233-132

Lip
= o

the filter has to be invoked like
REQUEST, FORWARD, INCLUDE, ERROR

the above 4 types can be given in <dispatcher> tag which
can be used in <filter-mapping>

this tag can be used for 0 or upto 4 times

and if dispatcher is not given then it considers the filter key
to intercept only the direct request from client

→ we can configure init parameters for the filter while
declaring the filter

```
<filter>
<filter-name>f1</>
<filter-class>Filter1</>
<init-param>
<param-name>myparam</>
<param-value>value</>
</init-param>
...
</filter>
```

Mr. SANTOSH
Narayan
A.S. - 1101
Ph: 23734512, 23760666

To get the values of init parameters use

```
String getInitParameter(String)
Enumeration getParameterNames()
```

declared in FilterConfig

Example :-

```
String s = filterConfig.getInitParameter("paramname");
```

September 2005

Priority

TUE

23

AUG

2005

Sneha Valang Reddy DExam

SessionDestroyed

- is called when session is destroyed

argument → HttpSessionEvent

has one method which can give the session which is created or destroyed

To configure these listeners to the application

in web.xml

<listener>

<listener-class> </listener-class> CLASS NOTES

</listener>

Mr. SANJAY
Naresh
phi: 25

this tag can be used for '0' or many times just above the servlet declaration.

<listener-class> takes a qualified class name, should be a non-abstract class and sub type of any one of the listeners

// ServletContextListener Implementation

import javax.servlet.*;

public class SCLImpl implements ServletContextListener

{

System.out.println("In Context Initialized");

ServletContext sc = sce.getServletContext();

sc.setAttribute("myatt", "My Ctxt Att Value");

} // ContentInitialized

public void contextDestroyed(ServletContextEvent sce)

{

System.out.println("Context Destroyed");

} // Destroyed

+ class

WED

Priority

24

AUG

2005

//SCAIImpl

8.am import javax.servlet.*;

public class SCAIImpl implements ServletContextAttributeListener

{

10 public void attributeAdded (ServletContextAttributeEvent scae)

{

11 System.out.println("Context Attribute added");

12 System.out.print("Name :" + scae.getName());

13 System.out.print("Value :" + scae.getValue());

14 } //aa

15 public void attributeRemoved (ServletContextAttributeEvent scae)

{

16 System.out.println("In Attribute Removed");

17 System.out.print("Name :" + scae.getName());

18 System.out.print("Value :" + scae.getValue());

19 } //ar

20 public void attributeReplaced (SCAIImpl scae)

{

21 System.out.println("In attribute replaced");

22 ServletContext sc = scae.getServletContext();

23 System.out.print("Name :" + scae.getName());

24 System.out.print("New Value :" + sc.getAttribute(scae.getName()));

25 System.out.print("Value :" + scae.getValue());

26 } //ar

27 } //class

August 2005						
S	M	T	W	T	F	S
-	1	2	3	4	5	6
-	7	8	9	10	11	12
-	13	14	15	16	17	18
-	19	20	21	22	23	24
-	25	26	27	28	29	30
-	31					

236-129, Week 34

MR. SANTOSH CLAS-NOTE
Naresh 1 Technical 5/5
PM 23734842, 23/4/2005

September 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	233-132, Week 33			

Priority

SUN

21

AUG

2005

Mr. SANTOSH
Naresh
Ph: 21345678

NOTES

66

21345678
Naresh

Listeners :-

= are new in Servlet 2.3

and some new listeners have been added in Servlet 2.4

1) ServletContextListener :-

Can be used to listen the lifecycle events of ServletContext instance

ContextInitialized

- is called by the container when the ServletContext instance is created and before it is given to the servlets.

ContextDestroyed

- is called by the container while destroying the ServletContext

- The above 2 methods takes the argument of ServletContextEvent

This has one method which gives the ServletContext instance for which the event has been generated (i.e. the ServletContext instance which is created or destroyed respectively.)

ServletContext getServletContext()

- These methods can be used if we want to perform some initialization with the new Context object, like setting some attributes in to the Context... and closing the resources when Context is being destroyed.

2) ServletContextAttributeListener

- it can be used to listen the events related to the context attributes like attributeAdded

- is called when a context attribute is added

MON

Priority

22

AUG

2(M)5

August 2005

S	M	T	W	T	F	S
	1	2	3	4	5	6
	7	8	9	10	11	12
	13	14	15	16	17	18
	19	20	21	22	23	24
	25	26	27	28	29	30
	31					

234-131, Week 31

attributeReplaced

- is called after replacing the attribute.

attributeRemoved

- after removing the context attribute

Argument of the above 3 methods is:

ServletContextAttributeEvent

it extends ServletContextEvent class

it has 2 methods and one method inherited from
ServletContextEvent

String getName()

Object getValue()

Note:-

- getValue() and getName() returning the attribute value
and name on which the event has been generated

- ServletContextAttributeListener methods are not thread safe

- ServletContextListener methods are thread safe

3) HttpSessionListener

- is used to listen the basic lifecycle events of HttpSession

SessionCreated

- is called when a new session is created within the context and before the reference is given to the servlet

September 2005						
S	M	T	W	T	F	S
			1	2	3	
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

23-128, Week 3)

Priority _____

THU

25

AUG

2005

Wrappers

ServletRequestWrapper
ServletResponseWrapper

Mr. SANTOSH - 45 NOTES
Narender Technical Services
Ph: 23734872, 27746666

Decorator Pattern :-

With this pattern an object/type is wrapped into some object/type wrapped with some pre/post operations

In Servlets 2.3 for request and response types we have been given with wrappers where these can be used to add some pre/post operations for the service provided by the request/response (implemented by the container provider)

// Wrapper example

```
import javax.servlet.*;
public MyResponseWrapper extends ServletResponseWrapper
{
    public MyResponseWrapper(ServletResponse res)
    {
        Super(res);
    }
    public void setContentLength(String s)
    {
        System.out.println("In setContentLength: " + s);
        super.setContentLength(s);
    }
}
```

FRI Priority _____

26

AUG

2005

August 2005

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

238-127, Week 31

//TestServlet

```
import javax.servlet.*;
```

```
import java.io.*;
```

```
public class TestServlet extends GenericServlet
```

```
{
```

```
    public void service(ServletRequest req, ServletResponse res)
```

```
        throws ServletException, IOException
```

```
{
```

```
    res.setContentType("text/html");
```

```
    PrintWriter out = res.getWriter();
```

```
    out.println("Hello from TestServlet");
```

```
} //Service
```

```
} //class
```

//Filter

```
import javax.servlet.*;
```

```
public class MyFilter implements Filter
```

```
{
```

```
    public void init(FilterConfig fc) { }
```

```
    public void destroy() { }
```

```
    public void doFilter(ServletRequest req, ServletResponse res,
```

```
        FilterChain fc) throws ServletException, IOException
```

```
    {
        MyResponseWrapper w = new MyResponseWrapper(res);
```

```
        fc.doFilter(req, w);
    }
}
```

```
MyResponseWrapper w = new MyResponseWrapper(res);
```

```
fc.doFilter(req, w);
```

Mr. SANTOSH CLASS NOTES
Mr. Maresh Technol. PVT LTD
Ph: 23134542, 231

September 2005						
S	M	T	W	T	F	S
			1	2	3	
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

239-126, Week 34

Priority

SAT

27

AUG

2005

<!-- Web.xml -->

```

<web-app>
  <filter>
    <filter-name>mf</filter-name>
    <filter-class>MyFilter</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>mf</filter-name>
    <url-pattern>/sel1</url-pattern>
  </filter-mapping>
  <servlet>
    <servlet-name>ts</servlet-name>
    <servlet-class>TestServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>ts</servlet-name>
    <url-pattern>/sel1</url-pattern>
  <servlet-mapping>
  <servlet-mapping>
    <servlet-name>ds</servlet-name>
    <url-pattern>/sel2</url-pattern>
  </servlet-mapping>
</web-app>

```

Mr. SANTOSH CLAS NOTES
 Naresh I Technologies
 PH 23734872, 274666

- In Servlet 2.4 XML, Schema is given for defining the web.xml rather than DTD

SUN

Priority

28

AUG

2005

JSP

Java Server pages

August 2005

S	M	T	W	T	F	S
	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

2005-125, Week 34

S	M
1	2
8	9
15	16
22	23
29	30
5	6
12	13
19	20
26	27
3	4
10	11
17	18
24	25
31	

JSP (Java Server pages)

versions → 1.0 / 1.1 / 1.2 / 2.0 / 2.1 (draft)

→ provides a tag based approach to develop the server side executable web application which is used to generate dynamic content

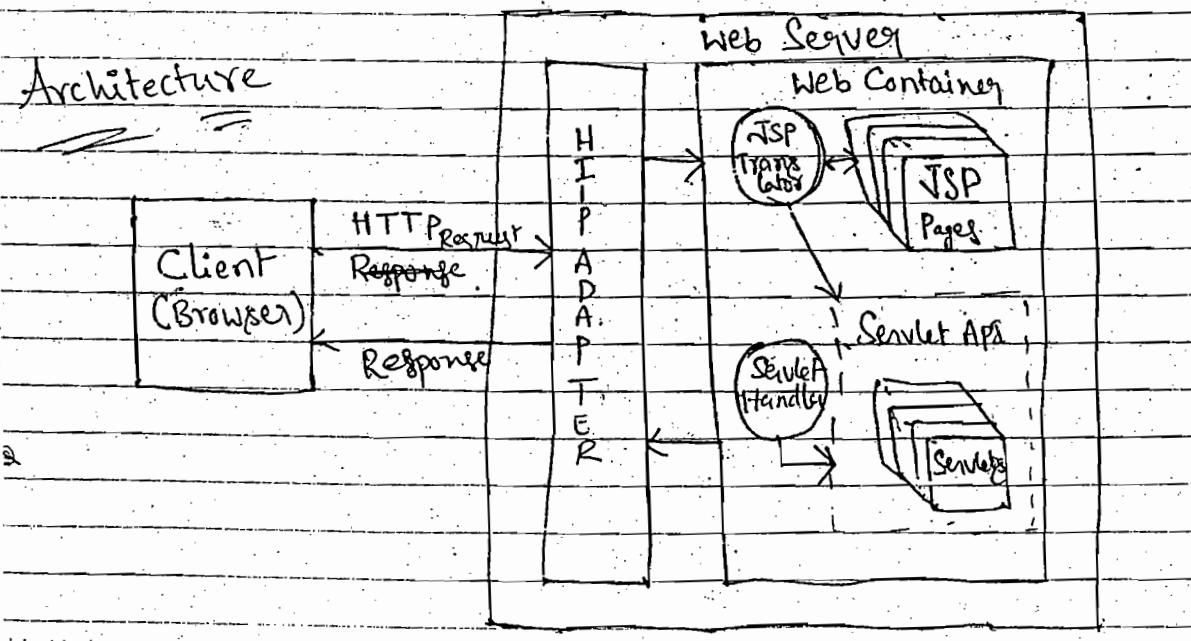
problem with Servlet

Mr. SANTOSH S. S. NOTE
Naresh Technologies
Ph: 23734842, 23740666

If we want to generate a page which consist of some dynamic content then we have to include the entire static content into the servlet (i.e., out.println method) along with the dynamic content

i.e., when a huge view has to be generated then there was a problem with the application development and modifications (view/application)

Architecture

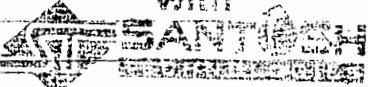


September 2005						
S	M	T	W	T	F	S
1	2	3				
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Priority _____

NOW
Mr. Santosh

with



#2nd Floor, Chhatrapati Nilayam, Opp:DCB Bank,
SR Nagar Police Station Lane, Amravati, Hyd.

MON

29

AUG

2005

Advantages of JSP

- 8am → provides a better mechanism to develop and maintain huge view compared to Servlet
- problem (or disadvantage) of JSP over Servlet
- The execution time required (i.e., the time required for the respective response generation) is more compared to Servlet

Architecture::

Mr. SANTOSH CLASS NOTES

Naresh Technologies

PM 23734842, 23746566

- JSP pages / JSP Document
- consist of some static content (Output Markup language tags like html tags and static text to be presented)
- Some dynamic tags (i.e., JSP tags)
- (which are understood and resolved by the JSP Translator / JSP Engine / JSP Container)
- Can have some scripting code (like Java code) (and from JSP 1.2/2.0 onwards we can use JSP EL (Expression language))

when the request is made for the JSP page / document

==== (like http request) ===

- request given to the JSP Translator
- find whether the requested JSP document is translated previously or not
- if not → then read the JSP document and translate the document into the Servlet equivalent (Translation Phase)
- Compile the Servlet equivalent java file (to get the class file (byteCode)) (is done by JSP Translator)
- (Compilation phase)
- request is dispatched to the servlet handler

Birthday / Anniversary

and the next process performed by the servlet handler is

TUE

Priority

30

AUG

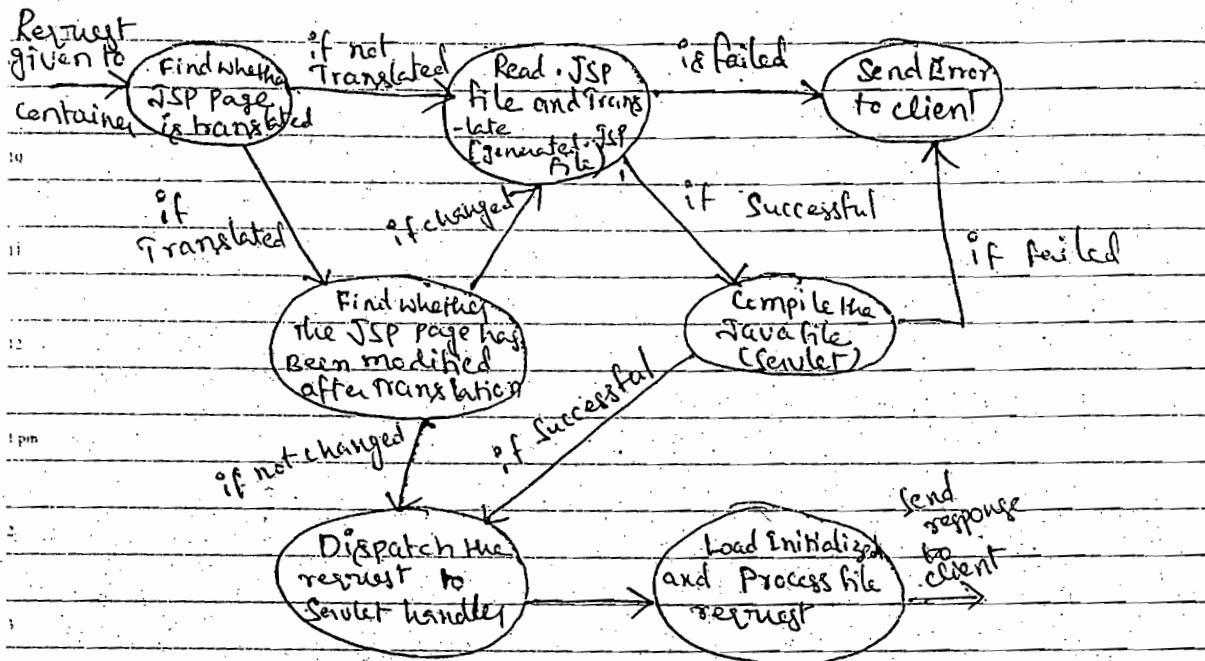
2005

August 2005

S	M	T	W	T	F	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

242-123, Week 35

Same as what it does with Servlet which we have written



JSP
=

- JSP Tags
- JSP API
 - (Used to develop custom tags)
- Expression language (EL)
- JSTL (JSP Standard Tag library)

Mr. SANTOSH CLAS - 10TF
Naresh | Tech. I. Eng.
Ph: 23734842, 23760660

September 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				
21st-22nd Week '05						

Priority

WED

31

AUG

2005

JSP Tags :-

→ These are the instructions used to give some information & instructions to the JSP Translator which are used by the JSP Translator at the time translation phase i.e. These are resolved at the time of translating the page

» Scripting Tags

- Scriptlet
- Declaration
- Expression

Mr. SANTOSH'S NOTES

Narensh Patel

Ph: 23734842, 23740000

Scripting (Scriptlet)

- is used to include some script (i.e., Java)
i.e. it can hold one or more Java instructions
which are placed directly into the _JspService method
(i.e. while translating the page)

~~Scriptlet~~

Syntax

<% java code %>

XML equivalent Syntax

<jsp:scriptlet>

java code

</jsp:scriptlet>

Expression tag :-

- is used to give some Java expression (one expression within one tag)

THU Priority _____
1
SEP
2005

September 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

244-121, Week 35

- The expression given in the expression tag is resolved and then the result will be printed to output

Syntax

```
<% = expression %>  
<jsp:expression>  
expression  
</jsp:expression>
```

Note:-

The Expression should not be ended with the ;

Declarative

is used to place some java code which has to be copied into the class and outside the - jsp:service method

i.e. it can be used for the class, instance variable declarations and writing some methods

<% !

declarations (can have multiple declarations)

%>

```
<jsp:declaration>  
</jsp:declaration>
```

Mr. SANTOSH CLASS NOTES
Naresh Technologies
phi 23734842, 23746666

Note:-

Scripting Tags should not be written inside the other Scripting tag.

If it is written then it raises translation phase error

October 2005						
S	M	T	W	T	F	S
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	215-120, Week 35			

Priority

now
Mr. Santosh

FRI

2

SEP
2005

#2nd Floor, Sarawati Niwas, Opp. DCB Bank
SR Nagar Police Station Lane, Ameerpet, Hyd.

Implicit Objects

Type Name

HttpServletRequest -- request

HttpServletResponse -- response

ServletConfig -- config

ServletContext -- application

HttpSession -- session

Object -- page (this object act as 'this')

javax.servlet.jsp.PageContext -- pageContext

JSPWriter -- out

all the above implicit objects can be used within the Scriptlet and expression tag

Life Cycle methods of JSP

- The for JSP's servlet equivalent file generated by the JSP Translator should be sub type of javax.servlet.jsp.JspPage

Where JspPage is an interface and extends javax.servlet.Servelt

in JspPage

→ jspInit()

→ - jspService()

→ destroy() inherited from Servlet interface

jspInit is called at time of initializing the servlet instance (i.e., the instance of a servlet equivalent for the JSP which is generated by the Translator)

SAT

Priority

3

SEP

2005

September 2005

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

JspService is called for every request

destroy is called before destroying the service instance

<!--First JSP Application-->

First.jsp

-->

<%!

//This tag can be used to declare the instance/class variable

int i;

%>

<html>

<body>

instance variable count value :

<%

i++;

out.println(i);

%>

<% int j=0; %>

Local Variable count Value :

<%= ++j %>

</body>

</html>

Mr. SANTOSH C. AS
Naresh Techinal
EMR 83734842, 25744366

http://localhost:8080/scripting/First.jsp

Oct 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				
Ch. 35						
27-31st, Week 35						

Priority _____

SUN

4

SEP

2005

To Set the Initialization parameters for the JSP document

- we have to Configure the JSP (declare & map) in web.xml file

web.xml

<Servlet>

<Servlet-name> </Servlet-name>

<jsp-file></jsp-file>

<init-param>

<param-name> </param-name>

<param-value> </param-value>

</init-param>

</Servlet>

1pm

Jsp file is mandatory to be declared in web.xml it is done in the following cases

1) if Init parameters are required to be configured/attached to the JSP

2) If we want the jsp file to be accessed using logical path name (url-pattern) instead of the file name.

(and this is mandatory if we want to place the jsp files into the private folder i.e, inside WEB-INF)

<!--InitParameter Example InitParamEx.jsp -->
<%!

public void jspInit()

{

System.out.println("In jspInit");

ServletConfig config = page.getServletConfig();

System.out.println(config.getInitParameter("myparam"));

} // jspInit

Birthday : Anniversary

Mr. SANTOSH CLAS NOTES
Naresh Technologies
Ph: 23731842, 23746566

N Priority

P
15

```
public void jspDestroy()
```

```
{  
    System.out.println("In jspDestroy");  
    } // jspDestroy
```

```
    public void mymethod()
```

```
{  
    System.out.println("In mymethod");  
    } // mymethod
```

```
%>
```

```
<%
```

```
String s = config.getInitParameter("myparam");
```

```
%>  
<b>Init Param Value : </b> <% = s %> <br />  
<b>Init Param Value : </b>  
<%  
    out.println(s);  
%>
```

```
<!-- web.xml -->
```

```
<web-app>
```

```
    <Servlet>
```

```
        <Servlet-name>abc</Servlet-name>
```

```
        <jsp-file>/WEB-INF/myJSPs/initParamEx.jsp
```

```
    </jsp-file>
```

```
    <init-param>
```

```
        <param-name>myparam</param-name>
```

```
        <param-value>MyParamValue</param-value>
```

```
    </init-param>
```

```
    </Servlet>
```

```
    <Servlet-mapping>
```

September 2005						
S	M	T	W	T	F	S
				1	2	3
	4	5	6	7	8	9
	11	12	13	14	15	16
	18	19	20	21	22	23
	25	26	27	28	29	30
	248-117, Week 36					

October 2005						
S	M	T	W	T	F	S
				1	2	3
	4	5	6	7	8	9
	11	12	13	14	15	16
	18	19	20	21	22	23
	24	25	26	27	28	29
	249-116, Week 37					

<Sep>

CUI

</Se>

</we>

Scru

JSP

But

the f

to th

E>D

page

to c

which

<

@@ att

=

(i)

def

↔

i

b

v

c

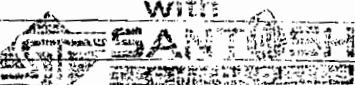
Mr. SANTOSH CHAUHAN
Naresh, Technical Gies
Ph: 23734842, 23745566
NOTES

S	October 2005						
	S	M	T	W	T	F	S
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	
16	17	18	19	20	21	22	
23	24	25	26	27	28	29	
	249-116, Week 36						

Priority _____

now
Mr. Santosh

TUE



6

SEP
2005

#2nd Floor, Saraswati Nilayam, Opp.DCB Bank
SR Nagar Police Station Lane, Amsarpet, Hyd.

```
<Servlet-name>abc </Servlet-mapping>
<url-pattern>/initex</url-pattern>
</Servlet-mapping>
</web-app>
```

Scripting tags allow us to include some scripting code into the JSP document (i.e., Java code)

But it is not recommended to use these tags, since it combines the presentation code again into the JSP document it may result to the same problem identified in Servlets.

2) Directives :-

- 1) page
- 2) include
- 3) taglibrary

MR. SANTOSH CLASS NOTES
Naresh Technologies
ph 23734842, 23740666

page :-

This directive holds the information which is required to customize the page translation i.e., this holds some instructions which helps the translator while translating the page into servlet.

Syntax:-

<%@page attributes%>

<jsp:directive.page attributes/>

⇒ attributes :-

⇒ language - takes the language which is used in the JSP document (i.e., Scripting tags)

⇒ default is java and most of the servers allows only java

⇒ import - takes one or more packages/qualified class names which are included into the import statements list

While the page is being translated 'Wher', 'char' is used as a separator

WED

Priority

7

SEP

20015

September 2005						
S	M	T	W	T	F	S
			1	2	3	
			4	5	6	7
			8	9	10	
			11	12	13	14
			15	16	17	
			18	19	20	21
			22	23	24	
			25	26	27	28
			29	30		
						250-115, Week 36

or we can use this import for any number of times
example :-

```
<%@page = "java.util.Collection.java.util.Hashtable"%>
```

→ extends - takes the complete qualified class name, this class
should be the subtype of JSPPage
* is not recommended to be used

Since the Document may not be portable i.e., such a document
may not be execute properly in the all the servers

→ session - takes true/false (default is ~~session~~ true)

→ buffer - takes the buffer size like 8kb, 16kb, 32kb

We can Specify OKb, if it is then there is no buffer
maintained instead the output generated will flushed
directly to the client

→ autoFlush - true/false (default is true)

→ errorPage - takes the path of the page to which the request
has to be dispatched when an error (exception) is NOTE
raised in the document

Mr. SANTOSH
Naresh IT Solutions
Ph: 23734842, 23740000

→ isErrorPage - takes true/false (default is false)

If this is true then we are available with one implicit
object exception of type Throwable

Notes

page directive can be used in the JSP document for any
number of times but the same attributes except import
should not be repeated

```
<%@ page import = "mypack.*" errorPage = "abc.jsp"%>
```

```
<%@ page import = "java.util.*" language = "java"%>
```

Birthday: Aanil

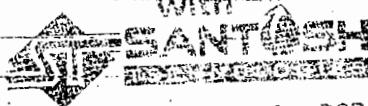
October 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

251-114, Week 36

Priority

now
Mr. Santosh

with



THU

8

SEP

#2nd Floor, Saraswati Nilayam, Opp DCC Bank
SR Nagar Police Station Lane, Ameerpet, Hyd.

2005

The above tags are invalid to be written

2) include:

is used to include some other page into the JSP document

This is known as static include since the target page is placed located and included at the time of page compilation (i.e. at the time of translating the page)

Syntax

<%@include file = " "%>

It takes the valid path of the target page which has to be included and this path has to be used and located at the time of translation

3) taglib:

This is used to attach some tag lib (i.e., custom tags) to the JSP element.

Eg: <!-- Example for directive: Header.jsp -->

<%@ page language = "java" %>
Header Part

<!-- Home.jsp -->

<%@ page import = "java.util.*" errorPage = "MyError.jsp" %>

<%@ include file = "/Header.jsp" %>

Date : <i> <%= new Date() %> </i>

<% if (request.getParameter("text").equals("error"))

throw new IOException ("Exception in Home.jsp, text : " + text);

Mr. SANTOSH C-157, 1st floor
Naresh Tech, 2, 2nd floor
PM 237345, 2, 2nd floor

FRI

Priority _____

9

SEP

2005

September 2005

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

252-1143 Work 56

<!-- My error.jsp -->

<%@ page isErrorPage = "true" %>

<i>Error:</i>

<%= exception.getMessage() %>

<% exception.printStackTrace() %>

<Web-app/>

*Mr. SANTOSH
Naresh Patel
Phi 23734542, 2124565
NOTES*

12 Some other attributes in page directive

1 pm - info -- takes a string which is returned when getServletInfo method is being used

- isThreadSafe -- takes true or false (default is false) if it is set to true then the jsp page can be used by only one request at a time

i.e, the servlet equivalent instance will be synchronized (Like Single Thread Model)

It is not recommended to make it true

- contentType -- takes the content type which has to be set for the response

Action Tag

- These are new in JSP 1.1

1>useBean

2>setProperty

3>getProperty

4>include

5>forward

October 2005						
S	M	T	W	T	F	S
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Priority _____

SAT

10

SEP

2005

6> param

7> plugin param

8> plugin

1> Use Bean

10 - is an action tag which is used to locate/create an instance of a java bean (non-GUI) (i.e., a non abstract java class)

Java Bean

- Should be a public non abstract class

- implement java.io.Serializable

(It is not mandatory in all the cases, if we want to use this component within the network or if we want to persist the object then it is required to implement.)

- Recommended to declare all the instance variables as private

- for each of the property declared there should one set/get/set and get methods

i.e., if the property is for read and write then provide set and get methods

i.e., if the property is for read and write then provide set and get method

if it is read only then only get method

- set method should take our argument where return type should be void

- get method should not take any argument (i.e., no argument) and return type should be the property type.

- recommended to have a No Argument constructor and in some cases it is mandatory

SUN Priority

11

SEP

2005

September 2005						
S	M	T	W	T	F	S
				1	2	3
	4	5	6	7	8	9
	11	12	13	14	15	16
	18	19	20	21	22	23
	25	26	27	28	29	30

25-11-11, Week 36

Syntax of useBean

<jsp:useBean attribute/>

where attributes are

Mr. SANTOSH
Naresht Technologies
Ph: 23734542, 23745566

id -- This is mandatory

takes the string value which is taken as reference variable name to point the instance created or located

class -- takes the complete qualified class name of the java bean / java class which has to be initialized
(This Should be a non-abstract class)

scope -- page / request / session / application

-- takes the scope from which the object has to be located or into which the object has to be stored after initialization

type -- takes the complete qualified type name of the reference (This has to be super type of the class given in class attribute)

beanName -- takes the java bean class name

(Some of class attribute but if this attribute is used then the Java bean will be initialized using the Bean API)

Note :-

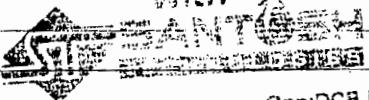
in one useBean tag either class or beanName any one of the attribute has to be used

October 2005						
S	M	T	W	T	F	S
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	255-110, Week 37			

Priority _____

now
Mr. Santosh

with



#2nd Floor, Savashetti Main Road, Opp:DCB Bank,
SR Nagar Police Station, Lumbra, Ameerpet, Hyd.

MON

12

SEP

2005

- and type optional

- scope is also optional if not given takes page as a default scope
- among type, class & beanName at least one of them has to be used then it can be used individually or combinedly with class or beanName

Example :-

```
<jsp:useBean id="myinstance" class="mypack.Myclass"
scope="request"/>
```

When use Bean tag is used

Mr. SANTOSH CLAS: NOTES

Naresh Technologies

Ph: 23734842, 25746666

- 1) Locate the bean instance in the given scope. If it is null and if class attribute is used with the useBean Create a new instance and take it into the reference variable name will be the value given in id attribute and the reference type will be of the type given in type attribute of useBean if not it takes it into the class type of reference

If instance is created successfully then set the instance into the given scope where ever the attribute name will be value given id

- 2) Make the reference available to the page.

Note:-

If there is any exception raised while initializing the instance it skips the page execution and throws an exception to client

TUE

Priority

13

SEP

2005

September 2005						
S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Octo						
S	M	T	W	T	F	S
				30	31	
				1	2	3
				4	5	6
				7	8	9
				10	11	12
				13	14	15
				16	17	18
				19	20	21
				22	23	24
				25	26	27

256-109 Wages 5%

257-38

2) SetProperty

- This tag is used to set the bean properties

< JSP : SetProperty attribute />

Attributes

↳ name -- taking the reference variable name on which set the set method has to be called

↳ property -- taking the property name whose setXXX method has to be called

i.e. here the method name excluding the set in it has to be given

↳ Value -- taking the direct value which has to be set to the property

↳ param -- taking the request parameter name whose value has to be set to the property

Note :-

- value and param are not mandatory but both of them should not be used in a single SetProperty tag

- if both of them are not given then the property name itself is taken as the request parameter name

Mr. SANTOSH CLAS NOTES

Naresh Technologies

Ph: 23734842, 23740366

property can be set *

if it is done then it means that all the setXXX method of the class bean are called

If this is used then we should not use value or param attribute

3) GetProperty :-

- is used to get the property value and print it to the output

✓ Anniversary

Birthday Anniv:

October 2005						
S	M	T	W	T	F	S
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Priority _____

now
Mr. Santosh

with

SANTOSH

#2nd Floor, Sarvawati Mahalayam, Opp. DCB Bank,
K Nagar, Police Station Lane, Ameerpet, Hyd.

WED

14

SEP

2005

<jsp: getProperty name = " " property = " " >

// LoginBean

package com.nit.ijspx;

public class LoginBean implements java.io.Serializable

{

 public LoginBean() { }

 public void setUserName(String s)

 {

 uid = s;

 }

 public String getUserName()

 {

 return uid;

 }

 public void setPassword(String s)

 {

 pwd = s;

 }

 public String getPassword()

 {

 return pwd;

 }

 private uid, pwd;

} //LoginBean

<!-- Login.html -->

<html> <body>

<form action = "LoginValidate.jsp">

<pre>

 UserName : <input type = "text" name = "UserName" />

 Password : <input type = "password" name = "password" />

Birthday / Anniversary

THU
15
SEP
2005

Priority _____

| September 2005 | | | | | | |
|----------------|----|----|----|----|----|----|
| S | M | T | W | T | F | S |
| | | | 1 | 2 | 3 | |
| 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| 25 | 26 | 27 | 28 | 29 | 30 | |

258-107, Week 37

| Oct | | | | | | |
|-----|----|----|----|----|----|----|
| S | M | T | W | T | F | S |
| 30 | 31 | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 22 | 23 | 24 | 25 | 26 | 27 | 28 |

259.10p

<input type = "submit" value = "Login" />
</pre>
</form> </body>
</html>

Mr. SANTOSH CLAS, NOTES
Naresh I Technologies
PM 23734842, 23745666

<!-- LoginValidate.jsp -->
<jsp:useBean id = "loginBean" class = "com.nit.jspex.LoginBean"
scope = "session" />
<jsp:setProperty name = "loginBean" property = "userName"
param = "userName" />
<jsp:setProperty name = "loginBean" property = "password" />

<%@include file = "Home.jsp"%>
<jsp:include page = "Home.jsp" />
<!-- Home.jsp -->
<jsp:useBean id = "loginBean" class = "com.nit.jspex.LoginBean"
scope = "session" />
<html><body>
UserName:
<jsp:getProperty name = "loginBean"
property = "userName" />

 Next page

<web-app />

Mr. SANTOSH CLAS, NOTES
Naresh I Technologies
PM 23734842, 23745666

Birthday / Anniversary

Birthday / Anniversary

October 2005						
S	M	T	W	T	F	S
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

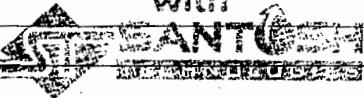
259-100, Week 37

Priority _____

now

Mr. Santosh

with



2nd Floor, Sureswari Nityam, Opp: BCE Bank
SR Nagar Police Station Lane, Amaarpur, Hyd.

FRI

16

SEP

2005

4> include

- is used to include some other page into this page
- this is done at time of execution
- this is known as dynamic including

<jsp:include page = " " />

This difference between the directive include and action include

directive include is static whereas action include

is dynamic

i.e., directive include locates the page and includes it at the time of translating the page whereas with action it locates the page and includes at the time of executing the page

2> with Static include we cannot use an expression to give the file path
whereas with dynamic include we can use (an expression)

Mr. SANTOSH

Name: Naresh T

Ph: 23734842, 244456

OTES

5> forward :

- is used to forward the request to some other page
- this is performed at the time of execution

<jsp:forward page = "relative Uri" />

6> param:

- is used to set request parameters while including the page (dynamic) or forwarding the page

7> is used to set the applet parameters

<jsp:param name = " " value = " " />

Birthday / Anniversary

SAT

Priority _____

17

SEP

2005

September 2005

S	M	T	W	T	F	S
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

260-105, Week 3*

Note :-

8 am This tag should be a child tag for include/forward/param

9 <!-- Page.jsp -->
 10 <% int i = Integer.parseInt(request.getParameter("field1"));
 11 int j = Integer.parseInt(request.getParameter("field2"));
 12 int sum = i + j;
 13 %>

14 <%@ page errorPage="MyErrors.jsp" %>
 15 <jsp:include page="Result.jsp">
 16 <jsp:param name="sum" value="<% =sum %>" />
 17 </jsp:include>

18 <!-- Result.jsp -->
 19 <html> <body>
 20 Result of <% =request.getParameter("field1") %> and
 21 <% =request.getParameter("field2") %> is
 22 <% =request.getParameter("sum") %>
 23 </html> </body>

24 <!-- Home.html -->
 25 <html> <body>
 26 <form action="Page1.jsp">
 27 <pre>
 28 Field 1 : <input type="text" name="field1" />
 29 Field 2 : <input type="text" name="field2" />
 30 <input type="submit" value="GetSum" />
 31 </pre> </form>
 32 </body> </html>

Birthday / Anniversary

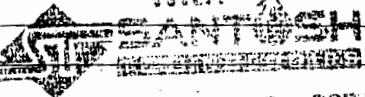
Mr. SANTOSH CHAUHAN
 Naresh IT 2nd year
 Ph: 23734542, 2440066

October 2005
 S M T W T F S
 30 31 1
 2 3 4 5 6 7 8
 9 10 11 12 13 14 15
 16 17 18 19 20 21 22
 23 24 25 26 27 28 29
 261-104, Week 37

Priority _____

now
Mr. Santosh

with



#2nd Floor, Sarabwadi Bhayani, DGB Bank
 SR Nagar Police Station Lane, Kompeta, Hyd.

SUN
 18
 SEP
 2005

<!-- MyError.jsp -->
 8 am <%@ page isErrorPage = "true" %>
 <i><% = exception.getMessage() %></i>

 <%@ include file = "Home.html" %>

10 <!-- web.xml -->
 11 <web-app>

12 > plugin

→ This tag is used to include an Applet or JavaBean (GWT) to the output

i.e., it is resolved into an object or embed tag of HTML it

is equivalent to applet tag

<jsp:plugin attributes/>

Attributes

Type → takes Applet or Bean

Code → takes the applet class name without (complete qualified name) which has to be initialized

codeBase → takes the base URL from which the applet and other required resources can be loaded (if optional)

(If this page is not given it takes the base URL of the page which has generated this tag)

Width → width of the applet

height → height of the applet

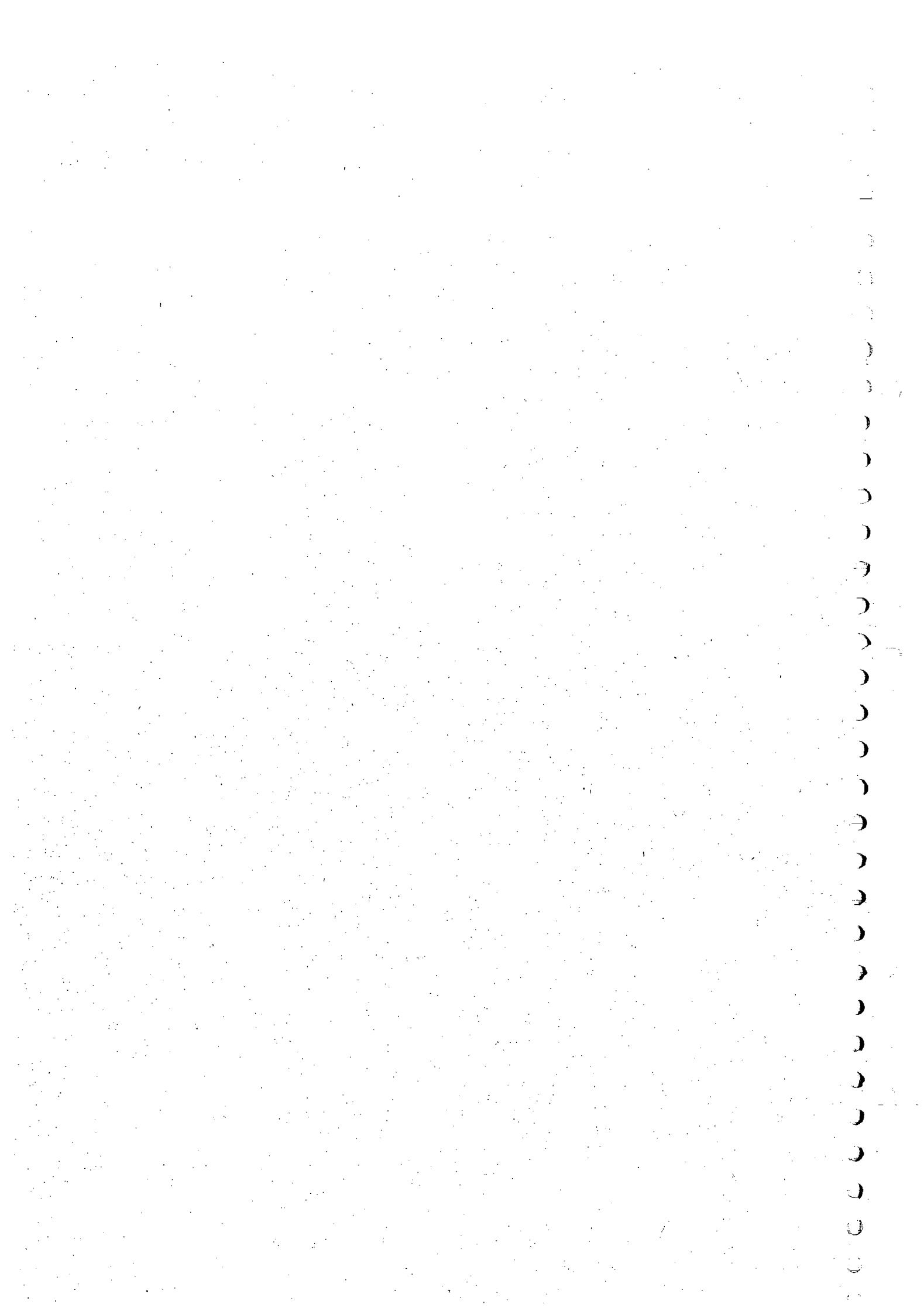
MR. SANTOSH CLASS NOTES
 Naresh Technologies
 PM 23734842, 23746666

VSpace

hSpace

Birthday / Anniversary

JspVersion → takes the JSP version which has to be used



ON
9
EP
005

Priority _____

September 2005						
S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

26-2-103, Week 38

October						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Cl

Sam

F

archive → taking the archive (zip, jar, ...) which has the resources required for the applet execution (Same of of applet tag archive attribute)

iplugin → takes the plugin url for IE

nplugin → for netscape

The above url are used to load the java plugin into the browser if the plugin is not available

8) param

This tag is used to enclose all the param tags within the plugin tag

<jsp:plugin ...>

<jsp:param>

<jsp:param name = " " value = " " /> NOTES

</jsp:param>

</jsp:plugin>

Mr. SANTOSH CLAS⁷ NOTES
Naresh Technol. gies
Ph 23734842, 23746666

9) fallback

It is used within the plugin tag which takes some content which is displayed in the place of applet if it is unable to be initialized

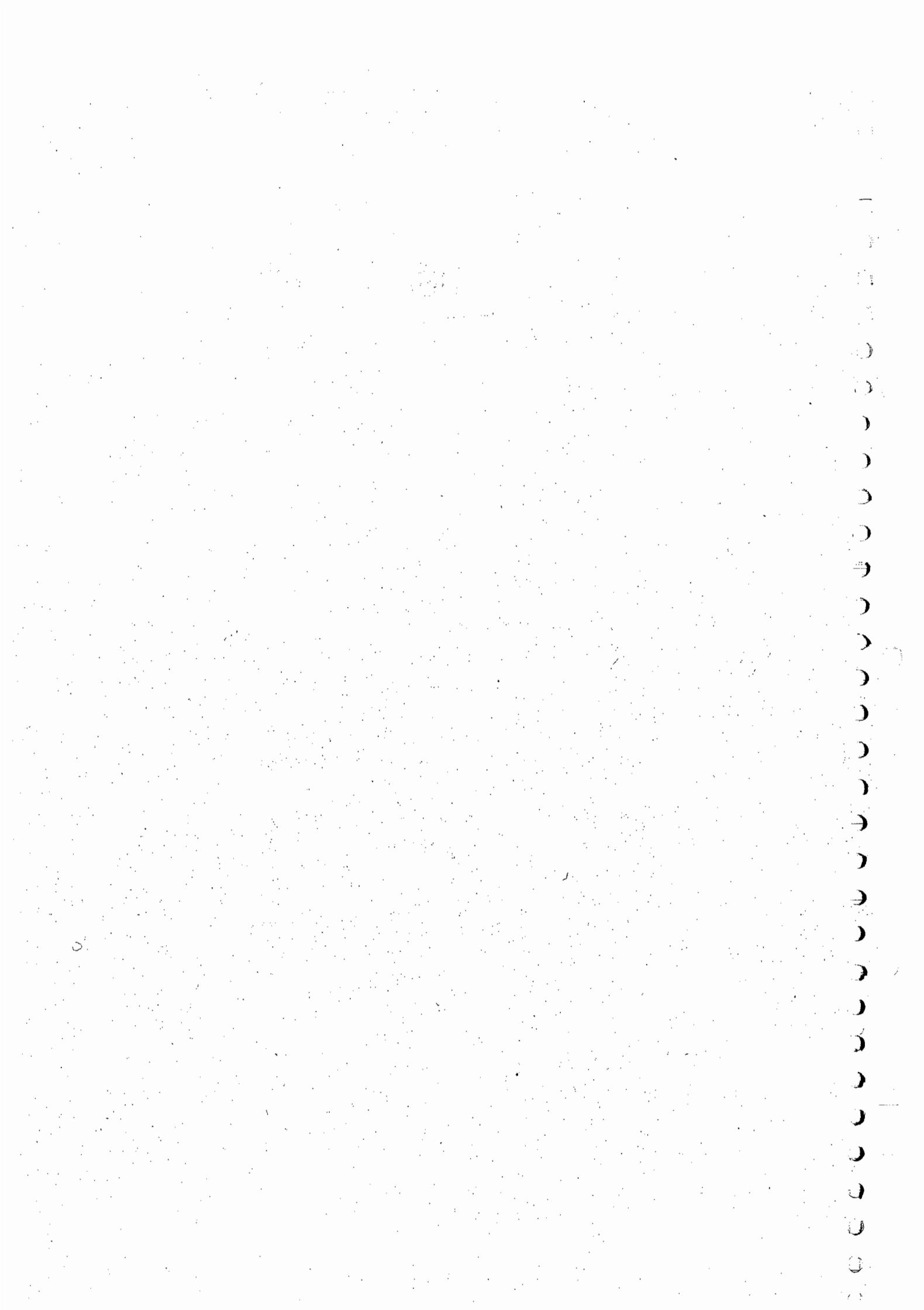
Example :-

<jsp:plugin ...>

<jsp:fallback>Unable to Load Applet

</jsp:fallback>

</jsp:plugin>



S
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
263-102, Week 38

Priority _____

October 2005

now
Mr. Santosh

with



SANTOSH
TECHNOLOGIES

#2nd Floor, Samarth Nilayam, Opp:DCB Bank,
SR Nagar Police Station Lane, Ameerpet, Hyd.

TUE

20

SEP

2005

Custom Tags

Under the JSP Specifications we have been given with the set of tags like Scripting, Directive and Action Tags which are standard and has to be recognized by error JSP Translator i.e., JSP Translator knowing what it has to write in the place of the tag used while translating the document.

But these tags are not enough to build the view or it may be difficult to build the view using these limited tags. So we may require to use the Scripting tags and write some Java code to meet the requirements.

And this approach is not recommended since it is combining the application logic and presentation logic.

To solve the above problem

We require to enhance set of tags and this can be done by custom tags.

Under JSP Specifications Sun has given set of standards which helps us to develop our own tags with our required functionality.

i.e. we can explain the Translator that what are the tags it has to understand and explain it what it has to perform when a particular tag is found in JSP document AS: ^{NOTES}

This option is from JSP 1.1.

Mr. SANTOSH Technologies
Nare�i, 1 Technologies
PH 23734842, 23746666

1) Tag Handler:

- is a java class (non abstract) which has the functionality which has to be executed at the location where the tag is found.

WED
21 SEP 2005

Priority _____

September 2005						
S	M	T	W	T	F	S
1	2	3				
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

S	M
30	31
1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	

26-10-1, Week 38

3) Tag Library declaration (tld)

- It is an XML document which is used to declare the tags (i.e. custom tags) and attach the tag handler to the tag.

3) In web.xml we have to attach these tag library to our application (so that the library will available for the JSP documents under the web application)

4) declare the tag library in the JSP document and then can make use of the tag's declared in it within the document

// Tag Handler

- 1) Should be a non-abstract class
- 2) Should be a sub type of javax.servlet.jsp.tagext.Tag interface

We can use some adapter classes like javax.servlet.jsp.tagext.TagSupport implements

// Tag Handler

package com.nit.jspex;

```
import javax.servlet.*;  
import javax.servlet.jsp.*;
```

```
public class MyTagHandler extends TagSupport {
```

Mr. SANTOSH CLAS: NOTES
Naresh / Technol. ages
Ph: 23734842, 23746666

October 2005
 S M T W T F S
 30 31 1 2 3 4 5 6 7 8
 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
 265,100, Week 38

Priority _____

THU

22

SEP

2005

```

public int doStartTag()
{
    try {
        JspWriter out = pageContext.getOut();
        out.println("<b> Hello From Custom Tag </b>");
    } //try
    catch (Exception e) {
    }
    return SKIP_BODY;
} //doStartTag
public int doEndTag()
{
    System.out.println("In doEndTag");
    return EVAL_PAGE;
} //doEndTag
} //class
<!-- TLD MyTag.tld --&gt;
&lt;!DOCTYPE ...&gt;</pre>

```

MR. SANTOSH CLASS NOTES
 Mr. Naresh Patel Technologies
 Ph: 93734842, 23746666

```

<taglib>
    <tlib-version>1.1 </tlib-version>
    <jsp-version>1.2 </jsp-version>
    <short-name> MyTag </short-name>
    <info> Hello Tag To test Custom Tag </info>
```

```

<tag>
    <name> sayHello </name>
    <tag-class> com.nit.jspex.MyTagHandler </tag-class>
    <body-content> empty </body-content>
</tag>
```

RI Priority
3
EP
105

Priority

September 2005						
S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

260-009, Week 38

S	M	T	W	T	F	S
36	37					
2						
9						
16						
23	24	25	26			

260-008

Tag lib
8am
1
2
3
4
5
6
7
8
9
10
11
12
1pm
2
3
4
5
6
7
8
9
10
11
12
Birthday Anniversary

<tag>
<name> getMessage </name>
<tag-class> com.nit.jspex.MyTagHandler </tag-class>
<body-content> empty </body-content>
</tag>
</taglib>
<!-- Web.xml -->
<web-app>
<taglib>
<taglib-uri> http://www.nit.com/mytags </taglib-uri>
<taglib-location> %WEB-INF/MyTags.tld
</taglib-location>
</taglib>
</web-app>
<!-- Text.jsp -->
<%@ taglib uri="http://www.nit.com/mytags"
prefix="mytags"%>
<html>
<body>
<i> Message : </i> <pre>
<mytags:sayHello></mytags:sayHello>

<mytags:sayHello />

<mytags:getMessage />
</pre>
</body>
</html>

Mr. SANTOSH CHAVES
Naresh, Technical Officer
Ph: 23734842, 23746166

October 2005						
S	M	T	W	T	F	S
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

26-098, Week 38

Priority

now
Mr. Santosh
with

SAT

24

SEP.

2005

2nd Floor, Saraswati Nilayam, Opp: DCG Bank
SR Nagar, Police Station Lane, Anna Nagar, Hyd.

Tag Handler instance life cycle for empty tag

8 am

- 1) Container maintaining tag pool i.e all the TagHandler instances in an application are maintained in a pool so that they can be reused
 - Create or locate the tag handler instance
 - Set the pageContext instance by calling setPageContext method on the taghandler instance
 - if existing set all its attributes
 - Call doStartTag
 - depending on the return value given by the doStartTag (i.e. for empty element it can be only SKIP_BODY)
 - invoke doEndTag
 - if doEndTag returning EVAL_PAGE then continue
 - if it is returning SKIP_PAGE then return i.e end the service (end of request processing)

To add attributes for the element

1) we have to declare the attribute while instantiating the tag (i.e. in tld)

2) for each of the attribute declared there should be one set and get method in TagHandler i.e. Hello class attached to tld

3) we can use them in the JSP ~~simply~~ within the tag

in MyTags.tld

inside <tag> with attribute name sayHello, after body-content tag

<attribute>

<name> user </name>

Birthday / Anniversary

SUN

Priority _____

25
SEP
2005

September 2005						
S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

268-097, Week 38

<required> true </required>
<rtepr-value> true <rtepr-value>
</attribute>

in MyTagHandler.java

```
public void setUser(String s) { user = s; }  
public String getUser() { return user; }  
String user;
```

public int doStart(CTag c)

```
{  
try {  
    JspWriter out = pageContext.getOut();  
    out.println("Hello from Custom Tag to " + user);  
} catch (Exception e)  
{  
    return SKIP_BODY;  
}  
//do starting
```

in test.jsp

in the place of <mytags:sayHello> use
<mytags:sayHello user="testUser"/>

Birthday Anniversary

8am

10am

11am

1pm

4pm

8pm

Birthday

October 2005
 S M T W T F S
 30 31 1
 2 3 4 5 6 7 8
 9 10 11 12 13 14 15
 16 17 18 19 20 21 22
 23 24 25 26 27 28 29
 269-096, Week 39

Priority _____

now
Mr. Santosh
 with

MON

26

SEP

#2nd Floor, Saraswati Niwas, Opp:DCB Bank
 SR Nagar Police Station, Ameerpet, Hyd.

2005

Custom tags with body content

8am

- TagHandler

Should be a sub type of BodyTag
 i.e., can extend BodyTag Support

doStartTag

doInitBody

doAfterBody

doEndTag

Mr. SANTOSH CLAS. NOTES
 Naresh I Technologies
 Ph: 23734842, 23746666

SKIP_BODY

EVAL_BODY_INCLUDE

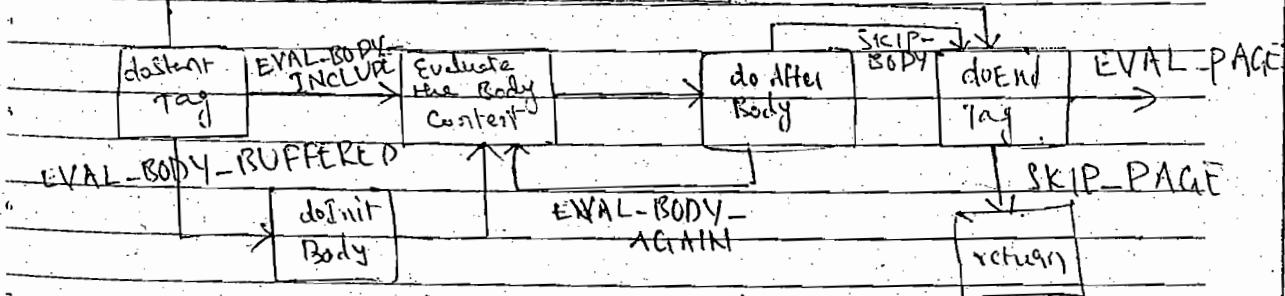
EVAL_BODY_BUFFERED

EVAL_BODY_AGAIN

SKIP_PAGE

EVAL_PAGE

SKIP_BODY



Life cycle of TagHandler

TUE

27

SEP

2005

Priority _____

September 2005						
S	M	T	W	T	F	S
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

270-095, Week 39

// Get Message TagHandler // package com.nit.jspex;
 import javax.servlet.jsp.*; package org.apache.jsp;
 import javax.servlet.jsp.tagext.*;
 import java.io.*;

10 public class GetMessage extends BodyTagSupport
 {
 11 public String getBody()
 12 {
 13 return user;
 14 }
 15 public void setBody(String s)
 16 {
 17 user = s;
 18 }
 19 public int doStartTag()
 20 {
 21 Sop("In GetMessage doStartTag");
 22 return EVAL_BODY_BUFFERED;
 23 }
 24 public void doInitBody()
 25 {
 26 Sop("In GetMessage doInitBody");
 27 }
 28 public int doAfterBody()
 29 {
 30 message = getBodyContent().getString();
 31 message += message + "User: " + getUser();
 32 return SKIP_BODY;
 33 }
 34 }

MR. SANTOSH K. NEESH, Technologies
 Ph: 23734842, 2374636

October 2005
 S M T W T F S
 30 31 1
 2 3 4 5 6 7 8
 9 10 11 12 13 14 15
 16 17 18 19 20 21 22
 23 24 25 26 27 28 29

Priority _____

WED

28

SEP

2005

271-094, Week 39.

public int doEndtag ()

8am

System.out.println("In GetMessage doEndtag");

9

try {
 JspWriter out = pageContext.getOut();
 out.println(message);
} // try

10

catch (Exception e) {}

11

return EVAL_PAGE;

12

} // doEndtag

1pm

String message;

String user;

2

} // class

To declare getmessage tag in TLD

<tag> <name>getmessage</name>
 <tag-class>GetMessage</name tag-class>
 <tag-class>

<body-content>JSP</body-content>

<attribute>

<name> user</name>

<required>true</required>

<expr-value>true</expr-value>

</attribute>

<tag>

use <taglib> in web.xml file to include the tld to the application

MR. SANTOSH CLASS NOTES
 Naresh Technologies
 Ph: 23734842, 23746666

THU

29

SEP

2005

Priority _____

September 2005					
S	M	T	W	T	F
			1	2	3
4	5	6	7	8	9
11	12	13	14	15	16
18	19	20	21	22	23
25	26	27	28	29	30

272-093, Week 39

October 2005					
S	M	T	W	T	F
			30	31	
			2	3	4
			9	10	11
			16	17	
			23	24	25

273-093

<--! Test.jsp -->

```

8 am <%@taglib uri="http://www.nit.com/mytags" prefix="mytags"%>
<html>
  <body>
    <mytags:getMessage type="1" />
    <b>Hello message for </b>
    <mytags:getMessage>
  </body></html>

```

MR. SANTOSH CLAS NOTES
 Naresh Technolgies
 Ph: 23734842, 23746666

Iterative Tag

- Tag whose body content can be evaluated for more than 1 time

This can be done by returning EVAL_BODY_AGAIN from doAfterBody

```
<iterate name="" scope="" id="" />
```

```
<getValue name="" />
```

```
//Iterate TagHandler
```

```
package com.nit.jspex;
```

```
import javax.servlet.jsp.*;
```

```
import javax.servlet.jsp.tagext.*;
```

```
import javax.servlet.http.*;
```

```
import java.util.*;
```

```
public class IterateTag extends BodyTagSupport
```

Birthday / Anniversary

Birthday / Anniv

October 2005						
S	M	T	W	T	F	S
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29				

Priority _____

FRI

30

SEP

2005

```

public void setId(String s) { id=s; }
" String getId() { return id; }
" void setName(String s) { name=s; }
" String getName() { return name; }
" void setScope(String s) { scope=s; }
" String getScope() { return scope; }
public int doString
public int doStartTag()
{
try {
    if (getScope() == null || getScope().equals(""))
    {
        c = (Collection) getPageContext().getAttribute(getName());
    }
    else if (getScope().equals("page"))
    {
        i = getPageContext().PAGE_SCOPE;
    }
    else if (getScope().equals("request"))
    {
        i = getPageContext().REQUEST_SCOPE;
    }
    else if (getScope().equals("session"))
    {
        i = getPageContext().SESSION_SCOPE;
    }
    else if (getScope().equals("application"))
    {
        i = getPageContext().APPLICATION_SCOPE;
    }
}

```

Mr. SANTOSH CLAS. NOTES
Naresh Technologies
Phi 23734842, 23746666

SAT

Priority _____

1

OCT

2005

else {

```
pageContext.getOut().println("Invalid scope");
return SKIP_BODY;
```

}

```
if(c==null)
c(Collection) pageContext.getAttribute(getName(), i);
if(c==null)
```

{

```
pageContext.getOut().println("Attribute not found");
return Tag.SKIP_BODY;
}
```

```
elements = c.iterator();
if(element.hasNext())
{
```

Object o = elements.next();

```
pageContext.setAttribute(getId(), o);
return EVAL_BODY_INCLUDE;
}
if
```

```
return SKIP_BODY;
}
by
```

```
catch(Exception e) {}
```

```
return SKIP_BODY;
}
doStartTag
public int doAfterBody()
```

try {

```
if(element.hasNext())
{
```

Object o = elements.next();

```
pageContext.setAttribute(getId(), o);
```

Mr. SANTOSH NARESH JAIN
Ph: 23723842, 23745660
CLAS NOTES

November 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Priority _____

SUN

2

OCT

2005

275-090. Week 39

now
Mr. Santosh
with

MR. SANTOSH

#2nd Floor, Sangam Valleyam, Cognica Bank
SR Nagar Police Station Lane, Ameerpet, Hyd.

return EVAL_BODY_AGAIN

11:15

{//try

catch (Exception e) { } }

return SKIP_BODY;

{//doAfterBody

public int doEndTag ()

{ }

if (c == null)

return SKIP_PAGE;

c = null;

elements = null;

i = 0;

pageContext.removeAttribute(getIdc ());

return EVAL_PAGE;

{//doEndTag

Collection c;

Iterator elements;

int i;

String name, id, scope; private PageContext getPageContext ()

{@Override} { return pageContext; }

{//class

//GetValue TagHandler

package com.nit.jspex;

import javax.servlet.jsp.*;

import javax.servlet.jsp.tagext.*;

public class GetValueTag extends TagSupport

{

public void setName (String s) { name = s; }

public void getName () { return name; }

Birthday / Anniversary

MON

Priority _____

3

OCT

2005

October 2005						
S	M	T	W	T	F	S
30	31				1	
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

276-089, Week 40

8 am public int doStartTag()
 try {
 pageContext.getOut().println(pageContext.getAttribute(
 getName()));
 } // try
 catch (Exception e) {}
 return SKIP_BODY;
} // doStartTag

1 pm public int doEndTag()
 return EVAL_PAGE;
 String name;
} // class

Mr. SANTOSH NARAYAN JAGAT
 Ph: 2373842, 23740556
 E-mail: jn@technichronics.com
 OTES

<!DOCTYPE....>

4 <taglib>
 5 <lib-version>1.1</lib-version>
 6 <isp-version>1.0</isp-version>
 7 <short-name>mytag</short-name>
 <info>MyTags Declaration</info>

8 <tag>
 9 <name>iterate</name>
 10 <tag-class>com.nit.jsp.IterateTag</tag-class>
 11 <body-content>JSP</body-content>
 <attribute>
 <name>id</name>
 <required>true</required>

Birthday / Anniversary

Birthday / Anniversary

November 2005						
S	M	T	W	T	F	S
1	2	3	4	5		
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

277-088, Week -0

Priority :

TUE

4

OCT

2005

<@texpr -Value > false </@texpr -Value >

</attribute>

<name attribute>

<name> name </name >

<required> true </required >

<@texpr -Value > true </@texpr -Value >

</attribute>

<attribute>

<name> Scope </name >

<required> false </required >

<@texpr -Value > false </@texpr -Value >

</attribute>

</bag>

<tag>

<name> getValue </name >

<tag-class> com:it . jspex . GetValueTag </tag-class >

<body -content> empty </body -content >

<attribute>

<@texpr <name> name </name >

<required> true </required >

<@texpr -Value > false </@texpr -Value >

</attribute> </tag > </taglib >

<!-- Text -->

<% Vector v = new Vector (); %>

v.add ("Element 1");

v.add ("Element 2");

v.add ("Element 3");

pageContext.setAttribute ("my elements", v, pageContext.
SESSION_SCOPE);

%>

Mr. SANTOSH : AS : NOTES

Naresh Technologies

PM 23734842, 23740566

WED

Priority

5

OCT

2005

October 2005

S	M	T	W	T	F	S
30	31				1	
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
278-087, Week 40						27

<%@ page import = "java.util.*"%>
<%@include page = "Text.jsp" />

<!-- Text.jsp -->
<%@ taglib uri = "http://www.mit.com/mytags"
prefix = "mytags" %>
<html><body> <table border = "1">
<tr><th> Elements of the Given Collection </th></tr>
<tr> ^{iterate} <td> mytags: name = "myelements" <td>
id = "myelement" scope = "myelement" <td>

<td>
<mytags:getValue name = "myelement" />
</td> </td>
</tr> </table> </body></html>

Mr. SANTOSH KUMAR
Naresh Technologies
Ph: 23734842, 23746165
NOTE:

- EVAL BODY TAG

is replaced with EVAL_BODY_INCLUDE and
EVAL_BODY_BUFFERED i.e., EVAL_BODY_TAG is
deprecated

in Tag interface

-Tag getParent()

return parent tag instance. (i.e., the instance of the tagHandler
of the parent element)

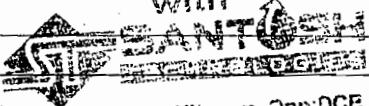
This is set into the current tagHandler before it
invokes doStartTag

November 2005						
S	M	T	W	T	F	S
1	2	3	4	5		
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Priority _____

now
Mr. Santos

with



#2nd Floor, Sarovar Niwas, Opp: DCE Bank,
SR Nagar Police Station Lane, Ameerpet, Hyd.

THU

6

OCT

2005

Example:

8:20am
<mytags:iterate ...>
<mytags:value ... />
</mytags:iterate>

in the above case getValue tag handler can get the iterate tag handler instance

11 class GetValueTag ... {
12 public int doStartTag() {
13 IterateTag it = (IterateTag) getParent();
14 // can use this instance to get the attributes of the parent
15 // element and to get some other properties of the parent element
16 }
17 }

Mr. SANTOSH CLASS NOTES
Naresh Technologies
Ph: 23734842, 23740666

JSP 1.2/2.0

- in 1.2 we have been given with XML equivalent tags for scripting and directive tags
i.e 1.2 JSP pages can be designed according to the XML standards

Birthday / Anniversary

FRI *Priority*

7

OCT
2005

JSTL

(JSP Standard Tag Library)

Supported with JSP 1.2 and Standardized with JSP 2.0
- it is a specification from sun and is implemented by 3rd party vendor

under JSTL

- Core Tags
 - Formatting Tags
 - XML Tags
 - SQL Tags

Core Tags

- namespace under which these tags is `http://www.w3.org/2001/XMLSchema`
and the preferred prefix is 'c'

```
<%@ Cxt = "http://www.sun.com/jstl/core" prefix="c" %>
```

四

```
<jsp:root xmlns:c="http://www.sun.com/list1/core" ...>
```

but

- If a tag used to display some output
`<C:out value="|<value>" />`

The given expression is resolved and then the value is placed into the output.

See

- is used to set an attribute

```
<C:SET var=" " value=" " scope=" " />
```

Birthday / Anniversary:

Value can take an expression

November 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

283-082, Week 41

Priority _____

MON

10

OCT.

2005

Eh implicit Objects

07/10/05

8am

pageScope

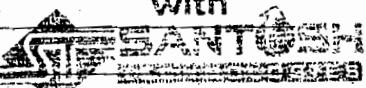
requestScope

sessionScope

applicationScope

Mr. SANTOSH Technologies
Naresh 1 Technologies
Pvt Ltd 23734842, 23740666

now
Mr. Santosh
with
SANTOSH



2nd Floor, Baraswami Villawarm, Opp:DCB Bank
SR Nagar Police Station Lane, Ameerpet, Hyd.

param

header - used to get the request header

Example :-

To get a value from the session scope and print on to the output (where attribute name is uname)

`$ {sessionScope.uname}` or
`$ {uname}`

`<c:out value = "$ {uname}" />`

To get from the request parameters

`<c:out value = "$ {param.uname}" />`

`<!-- JSTL and Eh example -->`

`<%@taglib uri="http://java.sun.com/jstl/core" prefix="c"%>`

UserName from request param : `<c:out value = "$ {param.uname}" />
`

`<c:set value="$ {param.uname}" var="user" scope="Session" />`

UserName from the sessionScope : `<c:out value = "$ {user}" />`

(or) `<c:out value = "$ {SessionScope.user}" />`

FUE

Priority

11

DCT

.005

October 2005						
S	M	T	W	T	F	S
30	31					1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

28, 2005, Week 11

November 2005						
S	M	T	W	T	F	S

28, 2005, Week 11

<!-- Home.html -->

<html> <body> <form action = "Test.jsp">

<pre>

User Name : <input type = "text" name = "username"/>
<input type = "submit"/>
</pre> </form> </body> </html>

<!-- Test.jsp -->

<%@ page import = "java.util.*" %>

Vector v = new Vector();
v.add("ele1");
v.add("ele2");
v.add("ele3");

session.setAttribute("elements", v);

%>

<jsp:include page = "Test2.jsp"/>

<!-- Test2.jsp -->

<%@ taglib uri = "http://java.sun.com/jstl/core" prefix = "c"%>

<html> <body>

<table border = "1">

<tr> <th>Elements </th></tr>

<c:forEach items = "\${elements}" var = "element">

<tr> <td>

<c:out value = "\${element}"/>

</td> </tr>

</c:forEach>

</table> </body> </html>

Mr. SANTOSH CLASS NOTES

Naresh I Technologies
Ph: 23734842, 23746666

46666

November 2005						
S	M	T	W	T	F	S
1	2	3	4	5		
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

281-084, Week 40

Priority _____

SAT

8

OCT

2005

if

- is used for conditional
`<c:if test = "<expression>">`

`</c:if>`

The expression given in the test should be resolved into a boolean value.

We can use var and scope attributes so that we can save the expression result in the given scope.

choose

- The child tags of this element should be when and otherwise

where we can have any number of when tags and 0 or 1 otherwise tag

`<c:choose>`

`<c:when test = " " > ... </c:when>`

`<c:when test = " " > ... </c:when>`

...

`<c:otherwise> ... </c:otherwise>`

`</c:choose>`

forEach

- Used to iterate through the collection

var - taking the attribute name

scope - page/request/session/application

start, end which also takes the starting and ending index

Mr. SANTOSH CLASS NOTES
 Naresh Technologies
 Ph: 23734842, 23746666

SUN Priority

9

OCT
2005

October 2005

S	M	T	W	T	F	S
30	31					1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

282-083, Week 40

for loops

- used to iterate for each of the token in the given string (i.e. uses StringTokenizer)

Formatting Tags :-

used for internationalization (I18N)

XML :-

- Some tags used to parse the given XML document and navigate through the node's

Expression Language (EL)

Mr. SANTOSH KUMAR
Naresh / Technical Lead
Ph: 23734542, 23140566

- Helps us to specify some expression's which are understood and resolved by the JSP engine.

is supported from JSP 1.2

- Is a scoped variable expression language
i.e., this language can locate only scoped variables it cannot refer to the page variables (Scripting variable)

expression written using EL Syntax has to enclosed into

`${<expression>}`

Note :-

`<expression>` in the above syntax is not enclosed using Scripting code rather it is designed using EL code

November 2005
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29

Priority

now
Mr. Santosh

WED

12

OCT

2005

#2nd Floor, Saraswati Niwas, Opp. SBI Bank
SR Nagar Police Station Lane, Amravati, Hyd.

<c:foreach items="\${\${yscr}}" var="user">
<c:if test="\${user.uid == param.uid}">
</c:if>
</c:foreach>

sql tag

query - takes the sql query which generates some results
var attributes

sql - holds the sql query to be executed (is an optional attribute)

We can give the query within the body of the query tag

var - the attribute name with which the result type of object key to be stored

Scope - page [request] session [application]

dataSource - takes the datasource object

Example :-

<sql:query dataSource="\${myds}" var="result"
sql="select * from emp"/>

or

<sql:query dataSource="\${myds}" var="result">
select * from emp
</sql:query>

param :-

can be used inside query or update

<sql:param value=""/>

THU

Priority

13

OCT

2005

October 2005

S	M	T	W	T	F	S
31					1	2
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

286-079, Week +1

3) dataparam

<sql: dataparam value = " " />

Example :-

```
<sql:query dataSource = "${applicationScope.myds}">
    var = "result"> select * from emp where empno =?
    <sql:query>
        <sql:param value = "${param.empno}" />
    </sql:query>
```

Result

is an interface given under the JSTL specification

```
String[] getColumnNames()
Map[] getRows()
int getCount()
```

Example :-

```
<sql:query dataSource = "${myds}" var = "result">
    select * from emp
</sql:query>
```

<table>

```
<xsl:foreach var = "row" items = "${result.rows}">
    <br> <c:if test = "${row.sal > 10000}">
        <tr><t:d><c:out value = "${row.empno}" /></td>
        <t:d><c:out value = "${row.empname}" /></td>
        <t:d><c:out value = "${row.sal}" /></td>
    </c:if> </xsl:foreach>
```

</table>

Birthday / Anniversary

Mr. SANTOSH CLASS NOTES
Naresh I Technologies
Ph: 23734812, 23740066

November 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

18 078, Week 41

Priority

FRI

14

OCT

2005

4) Update

takes a query which doesn't results with a result-set

```
<sql:uprlite dataSource = "${myds}" var = "count">
    update emp set sal = sal + 1000 where empno = ?
    <sql:param value = "${param.empno}" />
</sql:update>
```

5) SetDataSource

can be used to set the DataSource using the `id` name or the `DriverManager` parameters

<sql:setDataSource attributes>

attribute

= `dataSource`

var

Scope

driver - driver class name

url

user

password

<sql:setDataSource var = "myds" scope = "application"

driver = "oracle.jdbc.driver.OracleDriver"

url = "jdbc:oracle:thin:@init:1521:nit"

user = "Scott" password = "Tiger" />

MR. SANTOSH CLASS NOTES
Naresh I Technologies
PM 23734842, 23746666

transaction

used to maintain one or more queries within a transaction

<sql:transaction>

SAT

Priority

15

OCT

2005

Pramati

October 2005						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

288-077, Week +1

Version 4.1

- Java application server
is J2EE 1.3 compliant server and supports all the J2EE 1.4 specifications.

default port number

http port - 8181

Naming port - 9191

To deploy the web application in to the server

using auto deploy option copy the war file or the whole directory into

C:\PServer\Pramati\Server\nodes\default\archives\autoDeploy

(or)

→ using console:

c:\>localhost:8181/admin/login

username: root

password: root

Select Configuration

under it select

Web Application

MR. SANTOSH CLAS, NOTES
Naresh IT Technologies
Ph: 23734542, 23746666

InitialContext ic = new InitialContext();

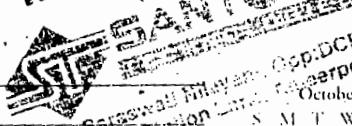
DataSource ds = (DataSource) ic.lookup("MyDataSource");

Connection con = ds.getConnection();

Birthday / Anniversary

Birthday

Mr. Santosh
now
with
SANTOSH



#2nd Floor, Sankarwadi Main Road, Opp. DCB Bank
SR Nagar Police Station, Latur, Maharashtra, India

October 2005

S	M	T	W	T	F	S
30	31					1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

200-075, Week 42

MON

Priority

17

OCT

2005

Struts

Srinivasan Reddy @ Diggatji

#2nd Floor, Sankarwadi Main Road, Opp. DCB Bank

SR Nagar Police Station, Latur, Maharashtra, India

October 2005

S M T W T F S

30 31

1

2 3 4 5 6 7 8

9 10 11 12 13 14 15

16 17 18 19 20 21 22

23 24 25 26 27 28 29

200-075, Week 42

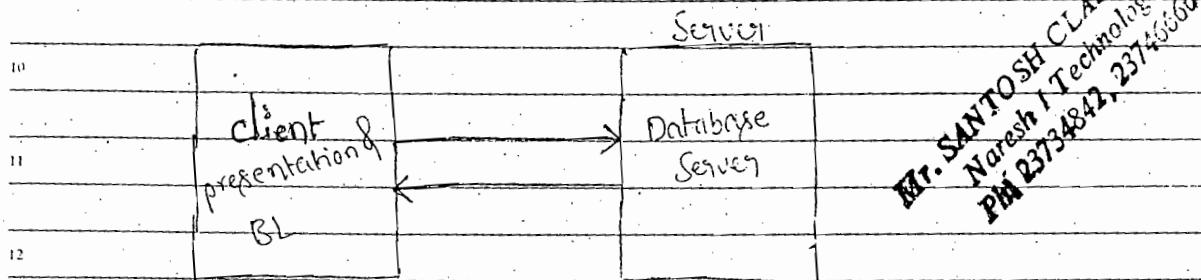
Struts

Architecture

2-Tier Architecture

BL - Business Logic

MR. SANTOSH CLASS NOTES
Naresh IT Technologies
PM 23734842, 23746600



- Here client tier is include with the presentation logic, Business Logic and its integration logic to interact with the backend server
- problems
 - presentation logic, Business logic are required to be installed into the client tier
 - if not secure
 - the resources required to execute presentation and business logic has to be installed into the client tier thus increases the cost of the application and maintenance
 - presentation and business logic are tightly coupled into a single tier so we cannot reuse the business logic with different types of clients
 - The business process runs independently in each of the client separately which results in more dependency on the backend system for maintaining the data consistency

Birthday / Anniversary

Birthda

November 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

291,079, Week 42

Priority _____

TUE

18

OCT

2005

3-Tier Architecture

- Here a middleware server is introduced in between of client and backend server (database)

where client tier is responsible for presentation and communicate with the middleware server

Middleware Server is responsible for the Business logic i.e., to provide the business services

In this Architecture the Business Service are separated from the client tier which provides the following advantages

- Business Logic can be reused

- The resources required to execute the Business Services are not required to be installed into each of the client instead they have to be provided only in the middleware Server

which decreases the application cost

Mr. SANTOSH CLAS NOTES
Naresh I Technologies
23734842, 23740666

N-Tier Architecture

- Here the Application logic is divided into n number of parts and can be separated physically or logically

Here in this Architecture as compared to the 3-tier Architecture middleware services are divided into more than one tier

Advantages

1> Code reusability

Birthday / Anniversary 2> Low maintenance

Model View Controller, OpenOffice.org, Apache Calc
Version 2.3.0, 2005-07-12, Build 1000, Apache-2.0, Apache-2.0, Apache-2.0, Apache-2.0

WED

Priority _____

19

OCT

2005

October 2005

S	M	T	W	T	F	S
30	31					1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

292-073, Week 42

- 8 am - Feature enhancements can be done without disturbing the existing services
9 - New types of clients and Backend systems can be attached

Mr. SANTOSH CLASS NOTES

MVC

Naresh Technologies

Ph 23734842, 23745666

= = Model view Controller

- 10 - i.e pattern which describe how and know into how many tiers the middleware logic has to be divided (i.e., to form N-tier Architecture) and it describes the clear behaviour/ responsibilities of each of the tier
11 - According to this pattern the middleware logic is divided into 3 tier Model, view & controller

Model

- 3 - responsible for the business process and to represent business data.

View

- 4 - takes the responsibility to build view i.e, presentation

Controller

- 5 - is responsible for taking the request from the client tier

6 - Understand the request - if required perform some validations on the request data

7 - Locate the model components and dispatch the request which can generate the presentation for the client (i.e., response)

November 2005						
S	M	T	W	T	F	S
1	2	3	4	5		
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

293-072, Week 42

THU

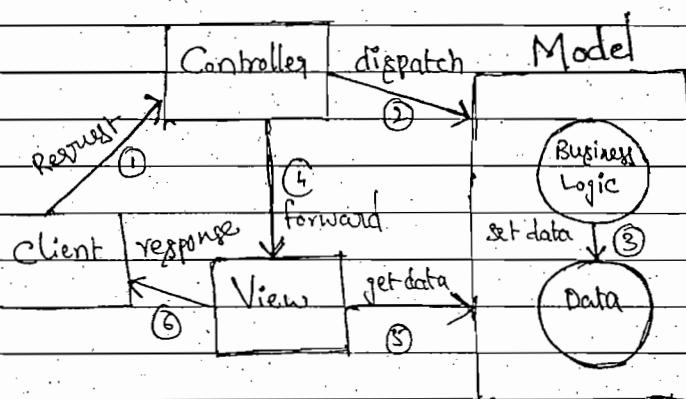
20

OCT

2005

MVC Architecture

8 am



1 pm To design a web based n-tier Architecture in J2EE using MVC architecture

Mode - 1

-explaining the controller and view to implemented using JSP and Model as a JavaBean or EJB

Mode - 2

-Controller - Servlet

view - JSP

Model - JavaBean or EJB

where these Mode-1 and 2 are the architecture recommended given by Sun to implement web based n-tier application in J2ee using MVC

Mr. SANTOSH CLASS NOTES
Naresh I Technologies
PM 23734842, 23746666

Controller

According to the Model-> driven architecture Controller functionalities can be implemented using Servlets.

FRI

Priority _____

21

OCT

2005

now
Mr. Santosh
with

October 2005

SUN	MON	TUE	WED	THU	FRI	SAT
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

#2nd Floor, Saraswati Nilayam, Opp: DCCB Bank
SR Nagar Police Station Lane, Amravati, Haryanava

29+071, Week 42

- Q) How many number of Servlets we have to use to implement it?
- Ans) Using Single Servlet to implement the controller functionality of all the Services
- if we have few number of Services then it may be used
 - if the number of services increased then problem with this approach is
 - Development is difficult since one single component should know about all the services and it should be intelligent to locate the respective model components for all the services
 - Testing the component is difficult
 - Feature enhancements are difficult since it is difficult to identify the exact location where the code has to be placed and may require modifications to the existing code.
 - can be implemented using more than one Servlet
 - The problem of this approach is
 - we don't have a centralized access to implement some common operations which has to be applied for all the requests (i.e., requests for all the different services)

To solve the above problem we can use Front Controller Design Pattern

Mr. SANTOSH CLASS NOTES
Narender Technologies
Ph: 23734842, 73746666

November 2005						
S	M	T	W	T	F	S
1	2	3	4	5		
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

295-070, Week +2

Priority

27606965

27790601

Sneha Venkatesh Reddy
Sneha Venkatesh Reddy
Sneha Venkatesh Reddy

SAT

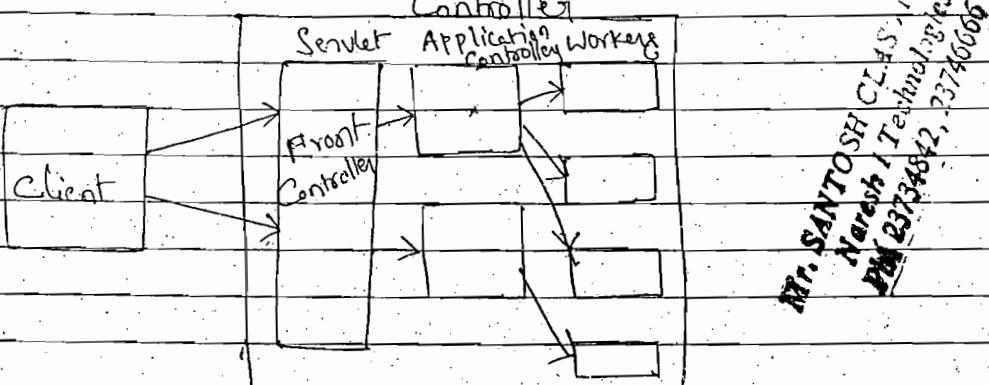
22

OCT

2005

Front Controller

- is a web design pattern
- it which provide a centralized access to the web clients
- It is responsible for the protocol handling and then understand the request given by the client, locate the worker which can process the request and dispatch



Mr. SANTOSH K. Narendran
Techno Solutions
Ph: 03134612, 23746666

As the number of workers are increased (as the number of services are increased) front controller logic is being complicated and it requires to remember about all the workers. This way causing the maintenance and application development difficult.

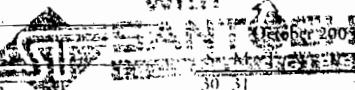
To overcome the above problem Application Controller Design pattern can be used

Application Controller

- It allows us to divide the application controller functionality into number of modules where Application Controller will be responsible to manage the workers under the module

Mr. Santosh

with



SANTOSH OCTOBER 2001

Narash

SUN

23

OCT

2005

Priority

Struts App

Struts Framework

Web Container

Server

30 31 1

2nd Floor, Srinivasan, Opp DGB Bank

SR Nagar Police Station, Hyderabad, Hyd.

16 17 18 19 20 21 22

23 24 25 26 27 28 29

296-069, Week 42

Now

S M

6 7 8

13 14 15

20 21 22

27 28 29

29 30

Note:

a Single application may have one or more Applications
Controllers and are managed by Front Controller

- Application Controller is responsible for
Action Management

View Management

Action Management :-

- is a process where it has to identify the request and
find which worker can process the request, then dispatch
the request to the worker and collect the response
from the worker

View Management

- depending upon the response given by the worker, it has to
locate the view which can prepare the presentation for
the response and then dispatch the request to the view

Struts

- It is an open source web application framework from apache group
- It follows MVC (Model 2 driven architecture)
- The framework is designed completely based on the Servlets and JSP Specification

MR. SANTOSH CTA
Narash Technologies
PM 23734842, 23746566

November 2005

S	M	T	W	T	F	S
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

297-068, Week 43

Priority _____

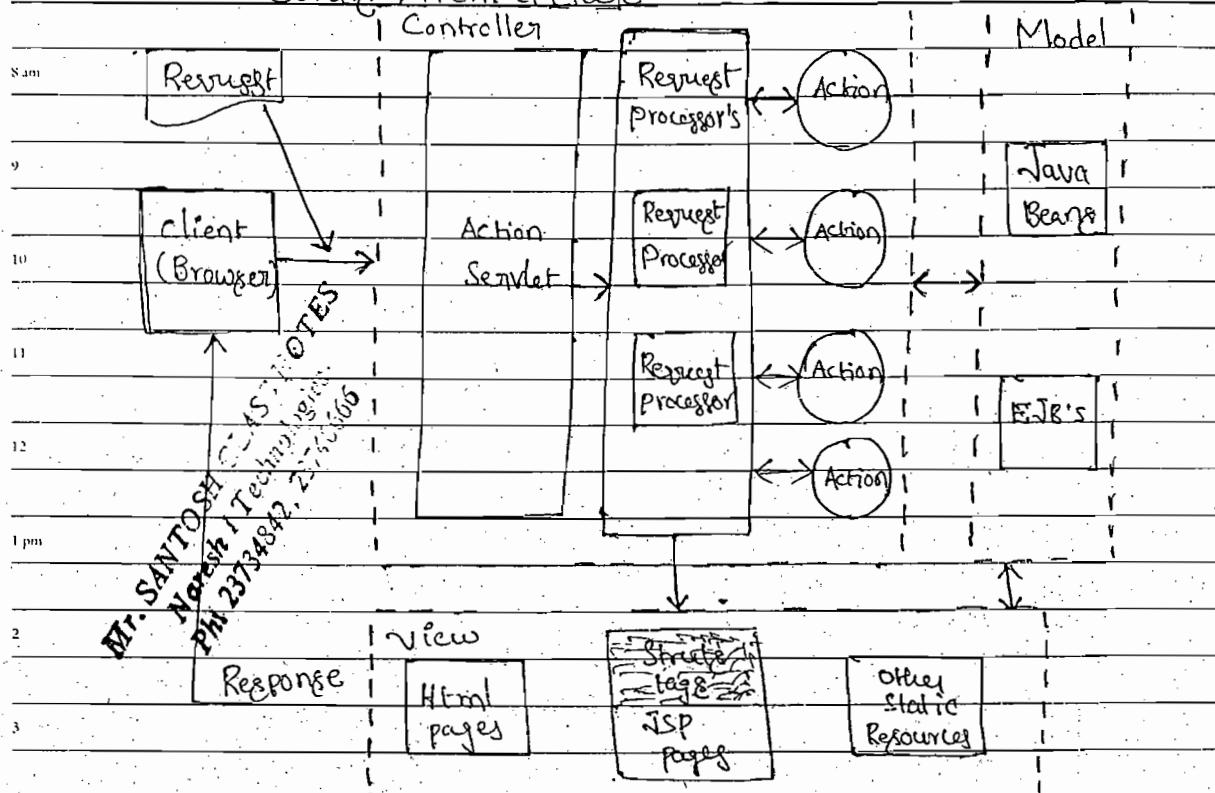
MON

24

OCT

2005

Struts Architecture



// Action

- 1) Should be a subtype of `org.apache.struts.action.ActionBase`
- 2) Should not be an abstract class
- 3) Implement `execute` method

*/

```
package com.nit.StrutsEx;
import javax.servlet.http.*;
import org.apache.struts.action.*;
public class LoginAction extends Action
{
```

```
    public ActionForward execute(ActionMapping am,
        ActionForm af, HttpServletRequest req, HttpServletResponse
        res) throws Exception
```

Birthday / Anniversary

TUE

Priority Medium

25

OCT

2005

October 2005

S	M	T	W	T	F	S
30	31					
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

298/467 Week 43

{

// in Struts 1.0 we have to override perform method from 1.1 onwards execute method is introduced

// This supports up to throw any exception, i.e which provides declarative exception handling support

String uname = req.getParameter("uname");
String pass = req.getParameter("pass");
LoginModel lm = new LoginModel();
if(lm.validate(uname, pass))

{

HttpSession hs = req.getSession();
hs.setAttribute("uname", uname);
return am.findForward("ValidUser");

} else

return am.findForward("InvalidUser");

} execute

} class

// LoginModel

package com.nit.struts.ex1;
public class LoginModel

{ public boolean validate (String s1, String s2)

} // can connect to the backend database to get the valid user list, so that we can compare the entered user details to validate

if(s1.equals("MyUser") && s2.equals("MyPassword"))

Birthday / Anniversary

Mr. SANTOSH CLAS NOTES
Naresh Technologies
PH 23734842, 23745666

November 2005						
S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

299-066, Week -43

Priority _____

WED

26

OCT

2005

8 am return true;

3

return false;

9 //validate

10 //class

11 <!-- Login.html -->

12 <html> <body>

<form action = "login.do">

13 <pre>

14 userName : <input type = "text" name = "uname" />

15 Password : <input type = "text" name = "pass" />

16 <input type = "Submit" value = "LogIN" />

17 </pre> </form>

18 </html> </body>

19 <!-- UserHome.jsp -->

20 <html> <body>

21 Welcome to User : <% = session.getAttribute("uname") %>

22 </body> </html>

23 <!-- Web.xml -->

24 <web-app>

25 <Servlet>

26 <Servlet-name>af </Servlet-name>

27 <Servlet-class>

28 org.apache.struts.action.ActionServlet

29 </Servlet-class>

30 <load-on-startup>1</load-on-startup>

31 </Servlet>

Mr. SANTOSH CLASS NOTES
Naresh I Technologies
PM 23734842, 23746666

THU
27
OCT
2005

Priority _____

October 2005					
S	M	T	W	T	F
30	31			1	
2	3	4	5	6	7
9	10	11	12	13	14
16	17	18	19	20	21
23	24	25	26	27	28
29					

No.	
S	M
6	7
13	14
20	21
27	28

300-065, Week 45

30-4

<Servlet-mapping>

8am <Servlet-name> is </Servlet-name>
<url-pattern> *.do </url-pattern>
</Servlet-mapping>
</web-app>

10 <!-- Struts configuration file -->

11 <!-- Struts-config.xml -->

12 <!DOCTYPE Struts-config PUBLIC

13 " -//Apache Software Foundation//DTD Struts Configuration
1.2//EN" or 1.0

1pm "http://struts.apache.org/dtds/struts-config_1_2.dtd." or

2 <Struts-config>

3 <action-mappings>

4 <action path="/login" type="com.nit.struts.ex.
LoginAction">

5 <forward path="/login.html" name="InvalidUser" />

6 <forward name="ValidUser" path="/UserHome.jsp" />

7 </action>

8 </action-mapping>

9 </Struts-config>

To Compile LoginAction

Set classpath to struts.jar file

Mr. SANTOSH CLASS NOTES
Naresh Technologies
Ph: 23734542, 23746666

Set classpath = d:\struts-1.2.6\lib\struts.jar;%classpath%

Birthday / Anniversary

Birthday

November 2005
 S M T W T F S
 1 2 3 4 5
 6 7 8 9 10 11 12
 13 14 15 16 17 18 19
 20 21 22 23 24 25 26
 27 28 29 30
 301-064, Week 43

Priority _____

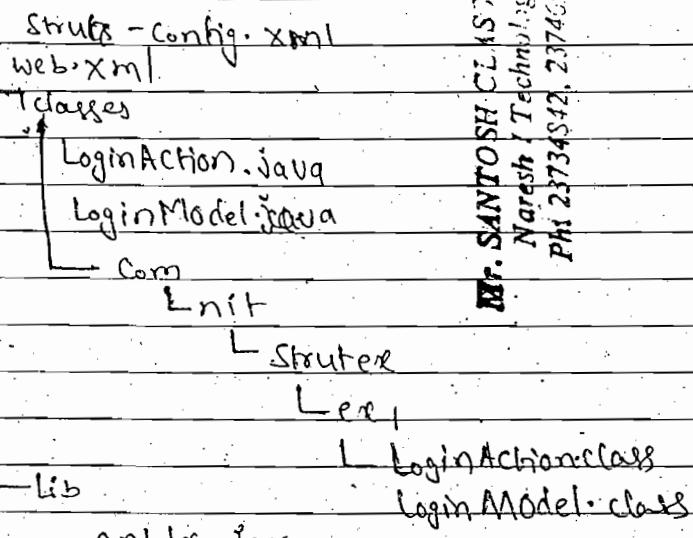
FRI

28

OCT

2005

WEB-INF



MR. SANTOSH CLASSTIMES

Nareesh / Technologies
 Ph: 23734542, 2374566

Next page

- When client makes a request to the ActionServlet

Example -

If ActionServlet is mapped to *.do then the request

uri with /login.do makes your request reach ActionServlet -

ActionServlet is a Subclass of HttpServlet and override
 doGet and doPost methods.

ActionServlet with the help of the request uri locate
 the RequestProcessor (i.e., which can process the request)
 and dispatch the request to it.

Birthday / Anniversary

SAT

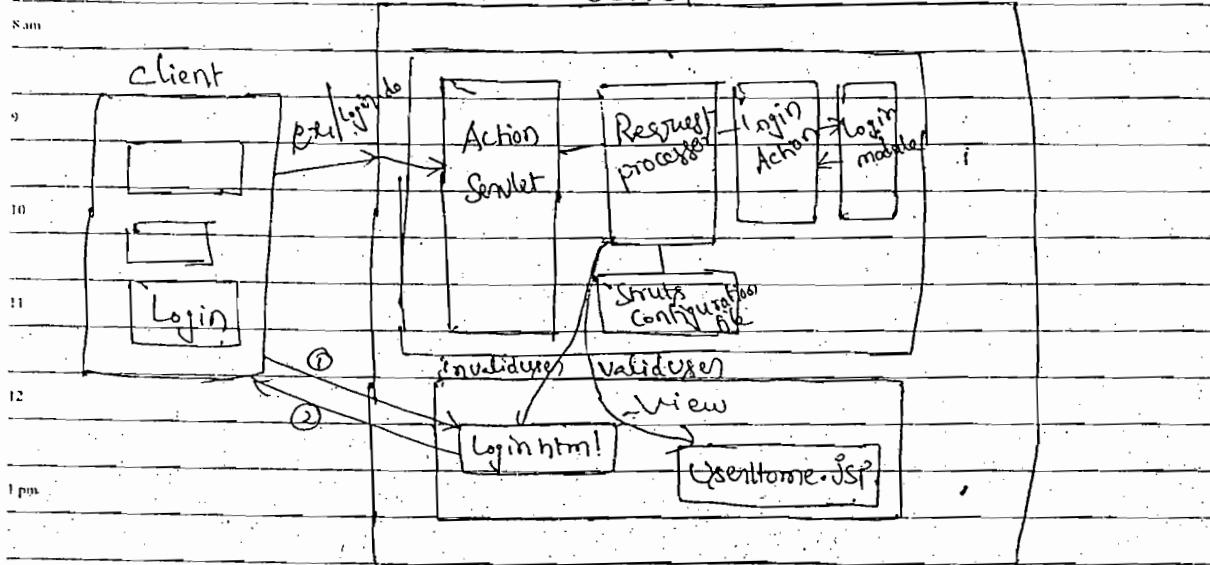
Priority _____

29
OCT
2005

AS NOTES
Mr. SANTOSH
Naresh I Technologies
P.M 23734842, 23746666

October 2005
S M T W T F S
30 31 1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
302-063, Week 43

Server



- it calls the process method on the located RequestProcessor instance
- RequestProcessor performs all the pre processing operations like local processing, security, formbean processing, forward mapping, action processing ...
- locate the action with help of the action mappings and the request uri which can process the request
- invoke the execute method on the located/created action instance
- collect the response from the action i.e., if it is an ActionForward object then locate the view with the help of ActionForward instance and forward the request to the view
- which takes the response to prepare the response and submit to the client, i.e., include into the

November 2005						
M	T	W	T	F	S	S
1	2	3	4	5		
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

30.3.062, Week -03

Priority

SUN

30

OCT

2005

output

Form Beans

- want to separate the logic required to retrieve the data from the request object using the parameter names,

Data Conversion responsibility from the action and if required performing small validation (pre validations) before the data is submitted to the model

To meet the above requirements Struts framework provides a support with the help of FormBeans

i.e., for one Action we can map 0 or 1 form bean (i.e., mapping formbean to the action is not mandatory)

Form Bean

- is a simple public, non-abstract Java bean (java class)
- this should be a subtype of org.apache.struts.action.ActionForm
- should hold set and get method for each of the parameters (i.e., req param)
- optionally we can override validate method if we want to perform some validations

⇒ <!-- Registration.html -->

```
<html> <body>
<form action = "reg.do" >
<pre>
```

UserName;<input type = "text" name = "uname" />

password;<input type = "password" name = "pass" />

Repassword;<input type = "password" name = "repass" />

Mr. SANTOSH CLASS NOTES
Naresh IT Technologies
Ph. 23734842, 23734843

MON

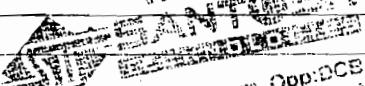
Priority

31

OCT

2005

now
with
Mr. Santosh



#2nd Floor, Saraswati Niyyam, Opp: DCB Bank
SR Nager Police Station Lane, Ameerpet, Hyd.

October 2005

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

304-061, Week 44

Address : <input type="text" name="address" />

PIN : <input type="text" name="pin" />

<input type="submit" value="Register" />

</pre> </form></html> </body></html>

10 => //Form Bean

//Reg Bean

11 package com.nit.strutsex.ex2;

import org.apache.struts.action.*;

12 public class RegBean extends ActionForm

{

1 public void setUname(String s)

{ un=s;

? }

3 public String getUname()

{

4 return un;

}

5 public void setPass(String s){ p=s;}

6 public String getPass(){ return p;}

7 public void setRepass(String s){ rep=p=s;}

8 public String getRepass(){ return rep;}

9 public String getAddress(){ return a;}

10 public void setAddress(String s){ a=s;}

11 public int getPin(){ return pin;}

12 public void setPin(int i){ pin=i;}

String un,p,rep,p,a;

int pin;

13 } //class

Birthday / Anniversary

MR. SANTOSH CLASS NOTES
Naresh Technologies
Ph: 23734642, 23746666

December 2005						
S	M	T	W	T	F	S
	1	2	3			
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

305-060, Week 44

Priority _____

TUE

1

NOV

2005

=> // RegAction

```

8am    package com.nit.strutsex.ex2;
9       import javax.servlet.*;
10      import javax.servlet.http.*;
11      import org.apache.struts.action.*;
12
13      public class RegAction extends Action
14      {
15          public ActionForward execute(ActionMappings am, ActionForm af,
16              HttpServletRequest req, HttpServletResponse res) throws Exception
17          {
18              RegBean rb = (RegBean)af;
19              String("Reg Details");
20              String("UName : " + rb.getName());
21              String("PIN : " + rb.getPin());
22              return am.findForward("resp");
23          }
24      }
25  
```

Mr. SANTOSH C. AS, NOTES
Naresh Techno Solutions
Ph: 23734842, 23545566

=> <!-- Response.jsp -->

```

Uname : <% = request.getParameter("uname") %>
Address : <% = request.getParameter("address") %>
PIN : <% = request.getParameter("pin") %>
</pre>

```

=> <!-- Struts-config.xml -->
<!DOCTYPE JSP>

<struts-config>
<form-beans>

WED

Priority

2

NOV

2005

now
Mr. Santosh

with
SANTOSH

November 2005

SUN	M	T	W	T	F	SAT
	1	2	3	4	5	
	6	7	8	9	10	11
	12	13	14	15	16	17
	18	19	20	21	22	23
	24	25	26	27	28	29
	30	31				

#2nd Floor, Samrat Nilayam, Opp. SBI Bank,
SR Nagar Police Station Lane, Amaarpur, 300-059, Week -11

<!-- This tag holds the declaration of all the form beans of this module -->

<form-bean name="regform" type="com.nit.struts.ex.RegBean"/>

<!-- The above tag is used to declare one form bean -->

<action-mappings>

<action path="/reg" type="com.nit.struts.ex.RegAction" name="regform" & validate="false">

<forward name="resp" path="/Response.jsp"/>

</action>

</action-mappings>

</struts-config>

MR. SANTOSH CLASS NOTES
Naresh Technologies
PH 23734842, 23746666

When request is submitted to the Request Processor

- Request Processor performs the sequence of processes like

- Local Processing

- character set Processing

like it set the character set and mime type so that from this point onwards we can generate output

- ActionMapping processing

where it finds whether there is any action mapping available for the requested path

- Process Security

is new in 1.1 and this process checks whether the requested user has a privilege to use the action or not it depends on TASS and user TASS

December 2005						
S	M	T	W	T	F	S
	1	2	3			
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

30°-058 Week 49

Priority

THU

3

NOV

2005

Process Form bean

- find whether the form bean is configured to the action path or not i.e. name attribute if existing
- form bean initialization it finds whether the Form bean instance is available in the given scope not if it is not available then initialize it and set into the given scope
- This object is set with the name given in the form bean declaration i.e. Form bean name
- get the request parameters and form bean ^{then} property details

Note: for every parameter submitted we should have one property declared in the form bean

- If required to convert the parameter types into property types and then populate the fields i.e. it calls set-XXX methods on the form bean instance
- Form data validation process
- It calls validate method on the form bean instance if it returning a not null value or an object with errors then it forwards the request to input page if it is returning a null object i.e. without errors then it continues the next process

- This process is performed only if the validate attribute is configured to true
- process forward mappings i.e. it identifies the possible forward mapping for the action path and configures into actionmapping object
 - process action

Birthday / Anniversary: Initialize or Locate the action instance.

Mr. SANTOSH CLASST NOTES

Naresh Technologies

Ph: 23734818, 23740566

FRI

Priority _____

4

NOV

2005

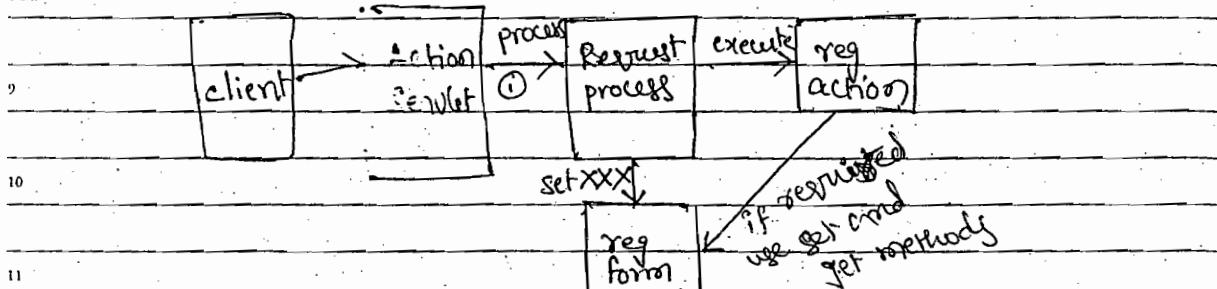
Mr. SANTOSH C-JAS; NOTES
Naresh I Technologies
ph: 23734842, 23746666

November 2005						
S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

308-057, Week 44

invoke the execute method

8 am



12

depending upon the response given by the execute method
if it is returning ActionForward object

i.e., if it is performing normal termination

locate the view and dispatch the request to the view

If exception is raised then use the exception handling
config for the action and present the response.

3

If validate is set to true for the action path

Request Processor performs the process Validation this is
performed after the populating the form fields and before
the form fields and before the action path is processed
i.e., before the request is dispatched to the action

Within this process it calls the validate method on
the form bean instance.

in Region.java

public ActionErrors validate(ActionMapping map, HttpServletRequest req)

{
ActionErrors ae = new ActionErrors();
if (name() == null || getUname().equals(""))

Birthday / Anniversary

{}

December 2005						
S	M	T	W	T	F	S
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
309-056, Week 44						

Priority:

SAT

5

NOV

2005

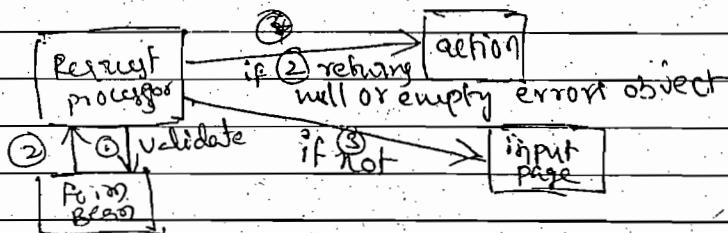
```

ae.add("MyErrors", new ActionError("error.uname.empty"));
} //if
else {
    if (getUname().length() < 10)
        ae.add("MyErrors", new ActionError("error.uname.minLength"));
    else if (getUname().length() > 20)
        ae.add("MyErrors", new ActionError("error.uname.maxLength"));
} //else
if (getPass() == null || getPass().equals("") || getRepass() == null ||
    getRepass().equals(""))
    ae.add("MyErrors", new ActionError("error.pass.empty"));
else if (!getPass().equals(getRepass()))
    ae.add("MyErrors", new ActionError("error.pass.notSame"));
} //if
} //else
ae.add("MyErrors", new ActionError("error.pass.notSame"));
} //if
} //else
// check validation for rec. pin and other validation if
// required
return ae;
} //validate

```

Mr. SANTOSH CLASS NOTES
 Naresh I Technologies
 PM: 23734842, 23740666

If the method returns null or an empty ActionErrors object then the validation is said to be successful and if it is returning an ActionErrors object which is not empty then the validation is not successful.



now
Mr. Santosh
with

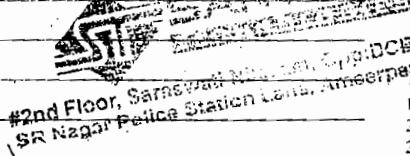
SUN

Priority

6

NOV

2005



November 2005

	S	M	T	W	T	F	S
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							

310-055, Week 44

where the path of the input page is given while
mapping the action tag i.e. in the Action Tag

Internationalization (I18N)

Java provides a support for I18N

i.e., it has included some utility classes in the `java.util`

Create `java.util.Locale`

- holds the locale settings like the language code (2 letter code small case), country code (2 letter code with caps) and variant

Where variant takes the 3 character representing some information related to the platform (i.e., os) or the browser specific information.

Mr. SANTOSH CLASS NOTES
Naresh Technologies
Ph 23734842, 23746666

`Locale (String Language)`

`Locale (String ln, String cc)`

`Locale (String ln, String cc, String variant)`

Example:-

`Locale l = new Locale ("en");`

(or) `new Locale ("en", "US", "WW")`

`java.util.ResourceBundle`

= takes the bundle (i.e., set of property files like each of them hold the key - value pairs for one specific language) and the `Locale` object.

An `Entry` object can be used to get the value of the given key in the `Locale` configuration

Birthday / Anniversary

December 2005						
S	M	T	W	T	F	S
	1	2	3			
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

311-054, Week 45-

Priority _____

MON

7

NOV

2005

The properties file names format

XXX <language code>[<country code>][<variant>].properties

Example:

ApplicationResources_en_US.properties

ApplicationResources_en_UK.properties

ResourceBundle rb = ResourceBundle.getBundle("Application Resources", locale);

String s=(String)rb.getObject("mykey");

ResourceBundle

Locates the properties file which is suitable for the given locale settings when we make a request to get the value of a key it collects the value from the properties file which locate with the help of the information like bundle name locale object given to it

Mr. SANTOSH CLASS NOTES

Naresh Technologies

Ph: 23734872, 23740060

Struts provides a support for the internationalization

By default Struts uses the ~~toLocale~~ client side locale.

Configuration which is being sent along with the request i.e., Accept-Language header

in ServletRequest we have a method getLocale which returns Locale Object

html utility tags given by Struts

- these are the custom tags developed following the JSP Specification and are given under the Struts Framework

TUE

Priority

8

NOV

2005

November 2005

S	M	T	W	T	F	S
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

312-053, Week 45

it provides some dynamic tags which can generate html equivalent tags which simplifies the view development

↳ form

- this tag is resolved into the html form tag

- this tag encapsulates the properties in the form text fields, password which can be mapped to one of the form beans so

that the data entered and sent in the request can be populated into the form bean instance and the data can be populated into the text fields... from the form bean object

We have text, password, submit, ... tags equivalent to the html tags

Struts bean utility tags

MR. SANTOSH CLASS NOTES

Naresh I Technologies

Ph 23734842, 23740666

↳ message

- is a tag which takes the key and prints the value corresponding to given key based on the locale configuration

<!-- Registration.jsp -->

<!-- this page is designed to support I18N and even it can get the errors if any and print them, can collect the data existing in the form bean and load them into the respective fields -->

<%@ taglib uri="/htmltags" prefix="html"%>

<%@ taglib uri="/beantags" prefix="bean"%>

<html> <body> <html:errors/>

<html:form action="/reg"> <input type="text" name="uname" />

<pre>

<bean: message key="uname" />

Birthday / Anniversary

Priority

| December 2005 | | | | | | |
|----------------------|----|----|----|----|----|----|
| S | M | T | W | T | F | S |
| | | | 1 | 2 | 3 | |
| 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 31 Dec 2005, Week 45 | | | | | | |

WED

9

NOV

2005

```
<html:text property = "uname" />
<bean: message key = "pass" />
<html:password property = "pass" />
<bean: message key = "repass" />
<html:password property = "repass" />
<bean: message key = "address" />
<html:text property = "address" />
<bean: message key = "pin" />
<html:text property = "pin" />
<html:submit>
<bean: message key = "reg.submit" />
</html:submit>
</pre> </html:Form>
</body> </html>
```

Mr. SANTOSH KUMAR CLASS NOTES
Hariharan / Technology
Ph: 23734842, 23746666

In the RegBean.java file include the validate method

in Struts-config.xml file.

change the action tag

```
<action path = "/reg" name = "regform" type = "com.nit.Strutsex.Ex:
RegAction" validate = "true" scope = "Session"
input = "/Registration.jsp">
<forward name = "resp" path = "/Response.jsp" />
</action>
</action-mapping>
<message-resources parameter = "ApplicationResources" />
</struts-config>
```

THU

Priority ..

10

NOV

2005

Weeks Values
Dally Dec 1, 2005

November 2005

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

now

Mr. San

with

SANTOSH

EXPERT

#2nd Floor, Bhaktivedanta Nilayam, Chaitanya Bank,
SP Nagar, Peenya, Bangalore, Karnataka, India - 560066

ApplicationResources.properties

8 am uname = UserName

pass = Password

repass = RePassword

address = Address

pin = PIN

error.uname.empty = <i> Username field should not be empty </i>

error.uname.minLength = <i> Username field should be min of 10 chars </i>

error.uname.maxLength = <i> Username field value should not exceed 20 chars </i>

error.pass.empty = <i> Password and RePassword field should not be empty </i>

error.pass.notSame = <i> Password and RePassword field values should be same </i>

reg.submit = Register

Mr. SANTOSH C.I.A.S, NOTES
Naresh I Technologies
DM 83734842, 23746666

< Web-app >

< taglib >

< taglib-uri > /html/tags < /taglib-uri >

< taglib-location > /WEB-INF / tld / struts-tld < /taglib-
location >

< /taglib >

< taglib >

< taglib-uri > /beans-tags < /taglib-uri >

< taglib-location > /WEB-INF / tld / beans-tld < /taglib-
location >

< /taglib >

< /webapp >

Birthday / Anniversary

December 2005

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	
31	32	33	34	35	36	37	38

Priority

DE8AM

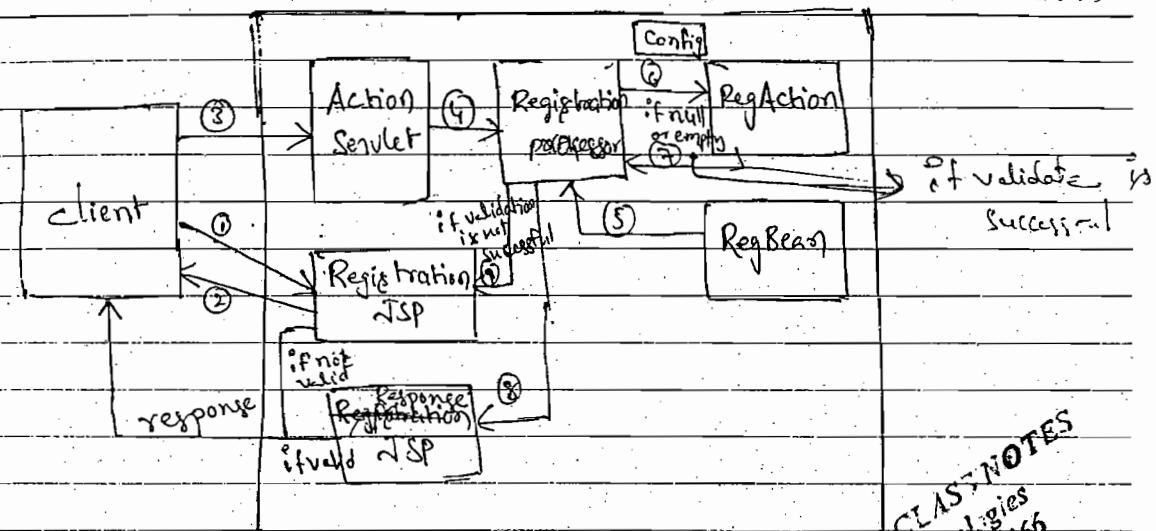
FRI

SWEETHA VAHANA REDDY

11

NOV

2005



DynaActionForms

- is new in Struts 1.1

Mr. SANTOSH CLASS NOTES
Naresh Technologies
Ph: 23734842, 23746566

DynaActionForm is introduced as a generic Form bean where the properties which it has to represent has to be declared in the Struts configuration file while declaring the form bean. This reduces the memory requirement.

org.apache.struts.action.DynaActionForm

<form-beans>

<form-bean name = "reform"

type = "org.apache.struts.action.DynaActionForm">

<form-property name = "uname" type = "java.lang.String" />

<form-property name = "pass" type = "java.lang.String" />

<form-property name = "repass" type = "java.lang.String" />

<form-property name = "address" type = "java.lang.String" />

SAT

Priority

12

NOV

2005

November 2005

S	M	T	W	T	F	S
			1	2	3	4
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

316/069, Week 45

now

<form-property name="pin" type="java.lang.Integer" with
</form-bean>
</form-bean>

to get the values from the DynaActionForm use
'Object get()' method

Example:-

```
public ActionForward execute(ActionMapping am,  
ActionForm af, ...)
```

```
DynaActionForm df=(DynaActionForm)af;  
String uname=(String)df.get("uname");  
int pin=((Integer)df.get("pin")).intValue();  
}
```

Declarative Exception Handling

Mr. SANTOSH AS. NOTES
Naresh Technologies
984983734842, 23746666

- in new in Struts 1.1

To Support this service in Struts 1.1 perform method of Action is changed to execute method which can throw any exception.

- Here exceptions which are raised in the execute method of the action are rethrown to the RequestProcessor So that request processor with the help of the exception declaration in the configuration file can handle them i.e. forwards the request to the concern page.

in Struts Configuration file

```
<exception type="" path="" key="" />
```

Priority						
December 2005						SUN
S	M	T	W	T	F	S
		1	2	3		
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
31-04-08, Week -45						

SUN

13

NOV

2005

this tag can be used in action tag before <forward> tag or in <global-exception>

Example:

```

<action path="/reg" type="RegAction" name="regform"
      scope="session">
  <exception type="java.sql.SQLException" path="/SQL.Error.jsp"
    key="error.sql"/>
  <exception type="java.rmi.RemoteException" path="/RemoteExp.jsp"
    key="error.re"/>
  <forward name="regfp" path="/Response.jsp"/>
</action>

```

in SQL.Error.jsp and RemoteExp.jsp

we can use <html:error/> to display the error message

Subclassing the RequestProcessor

- Requirement is

if we want to perform some operations before the request reaches to the actions of module like want to check the session data (i.e., validate it), want to check the request method type (i.e., get or post)

this requirement can be met by writing a subclass for the RequestProcessor and overriding the processPreprocess method

```
public boolean processPreprocess(HttpServletRequest req,
                                  HttpServletResponse res)
```

- if the method returns true then the process is continued if not it is terminated and the response

Mr. SANTOSH AS : NOTES
Naresh Technologies
PM 23734842, 23746666

MON

Property

14

NOV

2005

November 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

318-047, Week 46

generated upto to the point is submitted to the client

8am

This result is called after the local and character set cache processes and before the ActionMapping processing

Example:

```
public class MyRequestProcessor extends RequestProcessor
{
    public boolean processPreProcess(HttpServletRequest req,
                                    HttpServletResponse res)
    {
        if(req.getMethod().equalsIgnoreCase("post"))
            return true;
        try {
            PrintWriter out = req.getWriter();
            out.println("This Module Service can be used only
by HTTP POST clients");
        } catch (Exception e) {
        }
        return false;
    }
}
```

Mr. SANTOSH CLASS NOTES
Naresh Technologies
Ph: 23734842, 23746666

To inform the ActionServlet about the Subclass
in struts configuration file
<controller processorClass = " " />

Should appear after the action-mappings :

Birthday / Anniversary

December 2005
 W F S
 1 2 3
 8 9 10
 15 16 17
 22 23 24
 29 30 31
 2005 Week 46

Priority _____
 TUE
 15 NOV
 2005

Eclipse 3.0

- Is an open Source IDE (Integrated Development Environment)

Provides the complete Environment required for the development process and even test Environment

- Is a freeware tool

It provides the basic IDE features and allows us add some plugin to get some additional features

- MyEclipse is one of the plugin provides or support for the J2EE and Struts application development

File → new → Other

↳ J2EE → WebProject

↳ Struts → MyProject → J2EE 1.4

↳ finish

MyProject

↳ MyEclipse → add Struts capability

↳ package name com.nit.Struts

Hyproject after creation goto Hyproject

↳ new → Other

↳ Struts 1.2

↳ Name : RegForm

FormInput

in RegAction.java file

in execute method

{

DynaActionForm reform = (DynaActionForm) form;
 String uname = (String) reform.get("uname");

WED

Priority: 5th option

16

NOV

2005

Standard JSP with using Struts 1.0

November 2005						
S	M	T	W	T	F	S
				3	4	5
				9	10	11
				13	14	15
				16	17	18
				20	21	22
				23	24	25
				26	27	28
				29	30	

320 pages, Week 46

if(uname.equals("user1"))

{

System.out.println ("Uname:" +uname);

System.out.println ("pass;" +reform.get("pass"));
" " ("Pin;" +reform.get("pin"));

return mapping.findForward("regp")

} else

11

12 i) The generated page removing the table tag and write.

the following tags inside the <html:Form> tag

1 pm <b:errorMessage key="uname" /> <html:text property="uname" />
<b:errorMessage key="pass" />
<html:text property="pass" />

Same process will be applying <b:pass, address, pin>

3 <html:submit>

4 <b:submit message="Submit" />

</html:submit>

5 in src - com -- nft -- Struts --

→ ApplicationResources

Mr. SANTOSH CLASS NOTES
Naren Technologies
PM 9734842, 23746666

uname = Uname

pass = Password

repas = Repassword

address = Address

pin = PIN

submit = Register

Birthday / Anniversary

Priority:						
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				
2005 Week 46						

THU

17

NOV

2005

MyProject → Myeclipse → Add or remove project deployment

Modules in Struts application

- New in Struts 1.1

want to separate services into multiple modules so that the development process and the maintenance can simplified.

In Struts 1.0, this requirement was being solved by developing multiple applications like each of them having a different ActionServlet instance.

the problems with this approach

- session management was difficult

- dispatching the request from one application to other was consuming network calls.

- each part of the application was given with the different Front Controller (i.e., we don't have a Centralized access for all the services)

To solve the above requirements in Struts 1.1 it has implemented Application Controller Design pattern (i.e., it introduces RequestProcessor).

To Configure multiple modules:

Mr. SANTOSH CLAS, NOTES
Naresh Technologies
Ph 23734842, 23746666

Set the initialization parameters while declaring the ActionServlet in web.xml.

Example :-

```
<Servlet>
<Servlet-name>or </Servlet-name>
<Servlet-class>
```

December 2005						
S	M	T	W	T	F	S
	1	2	3			
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

325.040; Week 47

Priority: Using Struts

MON

21

NOV

2005

<form action = "admin/myaction.do" method = "post" >
 8am <input type = "submit" value = "Request admin Module Action
 (POST)" />
 9 </form> </body> </html>

10 Using Struts console to generate Struts Configuration file

11 file → new → new config file
 ↳ Struts 1.2 Config file → OK

12 Action Mappings

1pm ↳ add → new action

↓
 path: /myaction

type: com.nut.strutsex.module.DefaultModule
 ↳ OK ↳ action

error ↳ OK ↳ forward

↳ add

↳ name: resp

path: defaultResponse.html

↳ save ↳

defaultResponse.html

6 Con controller

7 ↳ RequestProcessor class: com.nut.strutsex.MyRequest
 ↳ Processor

↳ Save as adminResponse.xml

8 Web.xml is example program.

⟨servlet-mapping⟩

⟨servlet-name⟩ my </servlet-name>

⟨url-pattern⟩ * .do </url-pattern>

⟨/servlet-mapping⟩

⟨web-app⟩

Birthday / Anniversary

SUN

20

NOV

2005

Priority

..\\lib\\struts.jar

November 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

32+0041, Week 46

System.out.println("In request processor : " + req.getRequestURI());

String method = req.getMethod();
 if (method.equalsIgnoreCase("post"))
 return true;

try {

java.io.PrintWriter out = resp.getWriter();

out.println(" This Module can process only HTTP Post
clients ");

} // try

catch (Exception e) {}

return false;

} // process Preprocess

} // class

<!-- AdminResponse.html -->

Response from Admin Module

* and write defaultResponse.html

<!-- Home.html -->

<html> <body>

<form action="myaction.do">

<input type="submit" value="Request Default Module
Action"/>

</form>

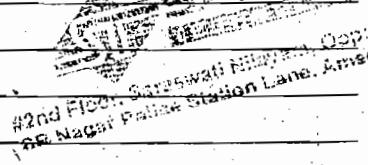
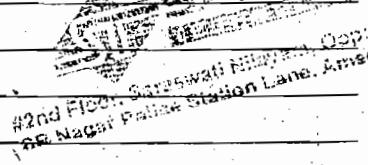
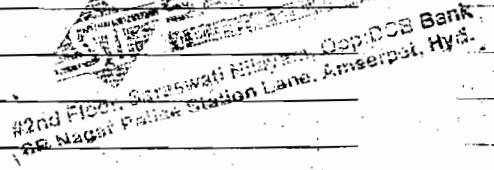
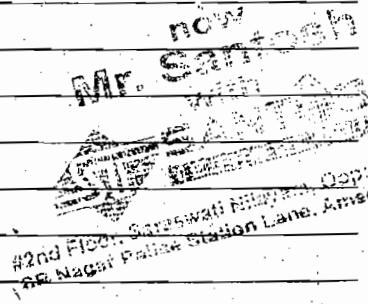
 Admin Module

<form action="admin/myaction.do">

<input type="submit" value="Request Admin
Module Action(GFT)"/>

</form>

Mr. SANTOSH AS, NOTES
 Naresh Technologies
 PH: 23734842, 23746666



Birthday / Anniversary

Birthday /

FRI Priority

18

NOV.

2005

November 2005

S	M	T	W	T	F	S
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

322-043, Week 16

org.apache.struts.action.ActionServlet

</servlet-class>

<init-param>

<param-name>Config</param-name>

<param-value>/WEB-INF/default-config.xml

</param-value>

</init-param>

<init-param>

<param-name>Config/admin</param-name>

<param-value>/WEB-INF/admin-config.xml</param-

value>

</init-param>

<load-on-startup>0</load-on-startup>

</servlet>

The param name should follow config / <module-name>
format

http://localhost:8080/<context root>[/<module-name>]/
<path>

// Default module action

package com.nit.struts.modules;

import org.apache.struts.action.*;

import javax.servlet.http.*;

public class DefaultModuleAction extends Action

{

 public ActionForward execute(ActionMapping am,

Birthday / Anniversary

Mr. SANTOSH CLAS; NOTES
Naresh Technologies
Ph: 23734542, 23746066

December 2005						
S	M	T	W	T	F	S
			1	2	3	
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
32,042, Week 46						

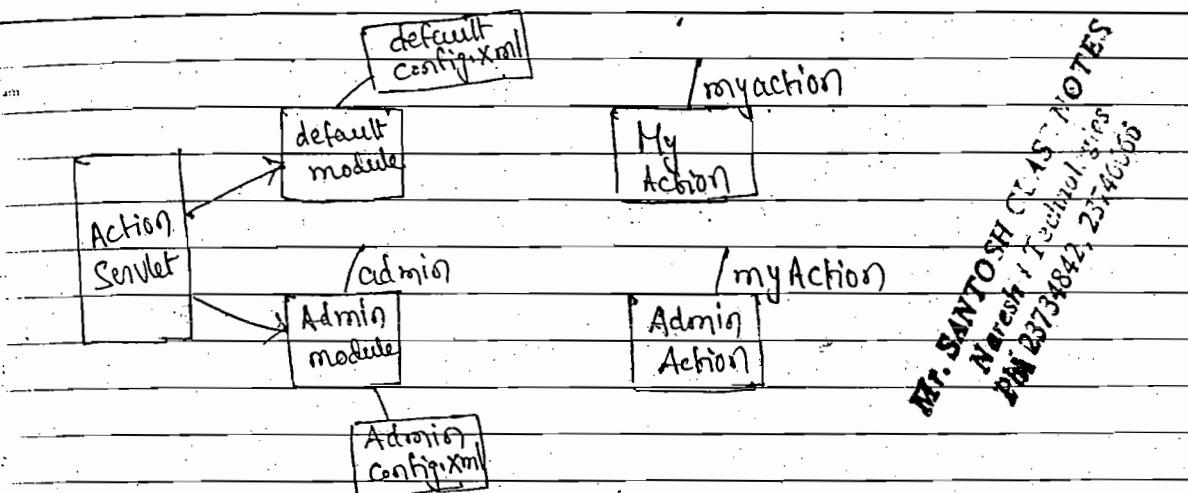
Priority

SAT

19

NOV

2005



ActionForm af, HttpServletRequest req, HttpServletResponse res)
throws Exception

```
{  
    System.out.println("In Default Module Action");  
    return am.findForward("resp");  
}
```

execute

class

In above program replaced Default with admin. Run the file is
Gelled by AdminModuleAction.java

1) RequestProcessor Subclass

```
package com.nit.struts.modules;  
import org.apache.struts.action.*;  
import javax.servlet.http.*;  
public class MyRequestProcessor extends RequestProcessor  
{  
    public Boolean processProcess(HttpServletRequest req,  
        HttpServletResponse res)
```

TUE
22
NOV
2005

Priority

Mr. Santosh
with
SANTOSH
#2nd Floor, Somaswari Bhavan, Opposite
S.R. Nagar Petrol Station Lane, Ahmedabad - 380 039, Week 4 -
November 2005
S M T W T F S
1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
Bank 28 29 30 31

D:

DefaultResponse.html

Home.html

WEB-INF

default-config.xml

admin-config.xml

web.xml

classes

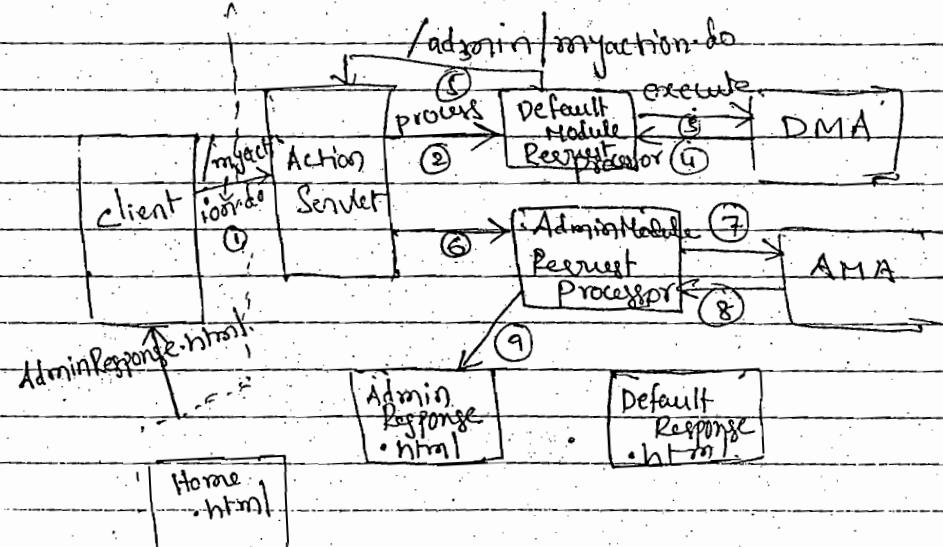
DefaultModuleAction

AdminModuleAction

MyRequestProcessor.java

admin

AdminResponse.html



Mr. SANTOSH CLASS NOTES
Naren I Technologies
Ph: 23734542, 23746566

December 2005						
S	M	T	W	T	F	S
			1	2	3	
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

327-038, Week 47

Priority

WED

23

NOV

2005

Validation Framework

8am

- is one of the open source common frameworks from apache
- this is used to perform some validation in a declarative form (can perform client side or server side validations)

10

This framework is initialized and used in struts framework so that the struts form bean properties can be validated using validation framework (i.e., declarative) i.e., Validation framework is plugged into the Struts framework and you get the service provided by the validation framework through the Struts framework. This approach allows us to set the validation rules for the properties at the time of deploying the application (instead of deciding at the time of developing the application)

1) The action form has to extend org.apache.struts.validator.ValidatorForm

or

org.apache.struts.validator.ValidatorActionForm

2) The validation details have to be given in validation.

.xml file (i.e., the validation entry of the form properties)

Mr. SANTOSH CLASS NOTES

Naresh Technologies

Ph: 23734842, 23743566

* // Form Bean

package com.nit.struts.validator;

import org.apache.struts.validator.*;

public class RecBean extends ValidatorForm

String uname, pass, repass, address,
int pin;

Birthday / Anniversary

THU

Priority

24

NOV

2005

November 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

328-037, Week 47

8am public void setUsername(String s) write getmethod
{ uname = s; }
public String getPassword() write setmethod
{ return pass; }
public void setPassword (String s) Mr. Santosh
{ repass = s; } now with
public String getRepassword ()
{ return repass; }
public void setAddress (String s)
{ address = s; }
public String getAddress ()
{ return address; }
public void setPin (int s)
{ pin = s; }
public int getPin ()
{ return pin; }
} /class

* <!-- Validation.xml -->
<!DOCTYPE -->
<! copy the doctype from validator-rule.xml file -->
<Form-Validation>
<Formset>
<Form name = "reform">
<field property = "userName" depends = "required,minlength,maxlength">
<arg0 key = "uname"/>
<arg1 key = "if {ver:minlength}" name = "minlength" resource = "false"/>

Birthday / Anniversary

December 2005						
M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

329-036, Week 4

Priority

FRI

25

NOV

2005

<arg1 key = "\${var:maxLength}" name = "maxLength"
resource = "false" />

</var>

<var-name> minlength </var-name>
<var-value> 8 </var-value>

</var>

<var>

<var-name> maxLength </var-name>

<var-value> 15 </var-value>

</var>

</field>

<field property = "password" depends = "required">
<arg0 key = "pass"/>

</field>

<field property = "repassword" depends = "required">

<arg0 key = "repass"/>

</field>

<field property = "pin" depends = "integer">

<arg0 key = "pin"/>

</field>

</form>

</formset>

</form-validation>

Mr. SANTOSH C.S.I.A.S, NOTES
Naresh Technologies
Phi 23734842 23746666

* # ApplicationResources.properties
username = UserName
pass = Password
repass = RePassword address = Address
pin = PIN submit = Register
error.required = <i>{0} field should not be

SAT

Priority

26

NOV

2005

November 2005						
S	M	T	W	T	F	S
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

330-035, Week 47

empty </i>

errors. minlength = <i>{0} field, minlength) should be
< i> </i>
errors. maxlength = <i>{0} field, can take only upto
max of < i> char's </i>
errors. integer = <i>{0} field, takes only numbers
</i>

previous

next

last

first

next

last

Mr. SANTOSH CLAS, NOTES
 Nareesh Technologies
 Ph 23734842, 23746666

```

<! -- StrutsConfig.xml -->
<DOCTYPE >
<struts-config>
  <form-beans>
    <form-bean name="regform"
      type="com.nit.struts.validator.RegBean" />
  </form-beans>
  <action-mappings>
    <action-path="/myaction" name="regform"
      validate="true" scope="session"
      input="/Registration.jsp" forward="/Response.jsp">
  </action-mappings>
  <message-resources parameter="Application Resources"/>
  <plugin classname="org.apache.struts.validator.
    ValidatorPlugIn">
    <set-property property="pathnames" value="/WEB-INF/
      validator-rules.xml, /WEB-INF/validation-
      .xml" />
  </plugin>
</struts-config>

```

December 2005						
S	M	T	W	T	F	S
1	2	3				
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
351-054, Week 47						

Priority

SUN

27

NOV

2005

previous Registration.jsp file is copying and do some modifications.

* <!-- Response.jsp -->
<%@taglib uri = "/beantags" prefix = "bean" %>
<html> <body> <pre>
UserName: <bean:write name = "regform" property = "username"/>
Address: <bean:write name = "regform" property = "address"/>
PIN: <bean:write name = "regform" property = "pin"/>
</pre> </body> </html>

Mr. SANTOSH CLASS NOTES

Naresh I Technologies

Ph 23734842, 23746666

To use the existing servlet or JSP or any web resource with Struts application

In the Struts Configuration file we can develop declare an action mapping with forward or include attribute instead of type attribute

In this case RequestDispatcher instead of locating & invoking the action it used RequestDispatcher and forwards or includes the request

Configuring DataSources with Struts Application

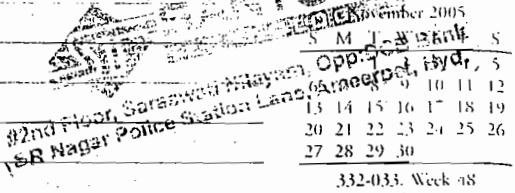
use <data-source> tag to configure on data source

Example:-

```
<data-source>
<data-source type = "oracle.jdbc.pool.OracleConnectionPoolData
Source">
<set-property property = "URL" value = "jdbc:oracle:thin:
:@localhost:1521:ora1"/>
<set-property property = "user" value = "scott"/>
```

MON
28
NOV
2005

Priority



S	M	T	W	Th	F	S	S
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30		

<set-property property="password" value="tiger"/>
</dataSource>
</dataSources>

we can configure any number of data sources in one application
(or module)

<dataSource key="" type="" />

If key is not given then the dataSource object is mapped
to the default key

To get the DataSource object into the action

DataSource ds = getDataSource(request);

The above method gets the dataSource object mapped to
the default key

If it is not configured then it returns null

DataSource ds = getDataSource(key, request);

This gets the dataSource object mapped to the
given key

Mr. SANTOSH CLAS, NOTES
Naresh Technologies
Ph 83734842, 23746066

// GetDetails

```
package com.nit.strutsx.db;
import javax.sql.*;
```

Birthday / Anniversary

Birthday / Anniversary

S	M	T	W	T	F	S
			1	2	3	
			4	5	6	7
			8	9	10	
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
333032, Week 48						

Priority

TUE

29

NOV

2005

{ DataSource ds = getDataSource (req);
 Connection con = null;
 Statement st = null;
 ResultSet rs = null;
 try {
 con = ds.getConnection ();
 st = con.createStatement ();
 rs = st.executeQuery ("Select * from mybank");
 Vector v = new Vector ();
 while (rs.next ()) {
 v.add (new Bank (rs.getInt (1), rs.getDouble (2)));
 } // while
 req.setAttribute ("bankdetails", v);
 return con.findForward ("req");
} // try
catch (Exception e) {}
finally {
try { con.close (); }
catch (Exception e) {}
}
return con.findForward ("error");
} // execute
} // class

Mr. SANTOSH CEAS, NOTES
Naresh Technologies
Ph 93734842, 23746666

WED
30
NOV
2005

Priority

November 2005
S M T W T F S
1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30
334-031, Week 48

//Bank

```
package com.nit.Structex.db;  
public class Bank  
{  
    public Bank() {}  
    public Bank (int i, double d)  
    { accno = i; bal = d; }  
  
    public int getAccno()  
    { return accno; }  
    public void setAccno(int i)  
    { accno = i; }  
    public double getBal()  
    { return bal; }  
    public void setBal(double d)  
    { bal = d; }  
    int accno;  
    double d;
```

MR. SANTOSH - AS NOTES
Naresh Technologies
Ph: 93734842, 23746666

}/class

```
<!-- Response.jsp -->  
<%@ taglib uri="/beamtags" prefix="beam" %>  
<%@ taglib uri="/logictags" prefix="logic" %>  
<%@ ml> <body>  
<table border="1">  
<tr> <th> Accno </th>  
<th> Balance </th> </tr>  
<logic:iterate name="bankdetails" scope="request" %>
```

January 2006						
SUN	MON	TUE	WED	THU	FRI	SAT
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

4351 30, Week 48

Priority

THU

1

DEC

2005

`id="bank">`

`<tr>`

`<td> <bean:write name="bank" property="accno"/> </td>`
`<td> <bean:write name="bank" property="bal"/> </td> </tr>`

`</logic:iterate>`

`</table> </body> </html>`

Mr. SANTOSH CLAS: NOTES
 Naresh Technologies
 P.M. 23734842, 23740666

Tiles =

- While designing the web page we may divide the view into number of parts like header, footer, body content and some menus....

In such a case we can develop each part of the view separately and then integrated together to form one view (can use include tag)

This solution is given under the design pattern named Composite View Design pattern

i.e., Here various parts of the view are developed separately and then composed into one single view

In this case while composing the page (view) in each of the view page we have to concentrate and design layout and find which parts has to be composed

Want to separate the layout management from the view page

so that changing the layout of the site doesn't effects all the view pages

now
Mr. Santosh

with

SANTOSH



SANTOSH

EXCITING OPPORTUNITIES

NUMBER 2005

S.M.T.W.T.F.S.

2nd Floor, Saravalli Nilayam, Opp:DCS Bank

SF Nagar Police Station Lane, Aiharpur, Hyd.

11 12 13 14

18 19 20 21 22 23 24

25 26 27 28 29 30

30-029. Wcc.

FRI

Priority

2

DEC

2005

Struts Tiles tags - Support to solve this problem

Tiles allows us develop each part of the view separately and compose them to form one view and making the view page free from layout management i.e, layout is designed separately

10

* <!-- Layout page -->
 File > <%@ taglib uri="/tiles" prefix="tile" %>
 <html> <body>
 <table>
 <tr> <td> colspan="2">
 <tile:insert attribute="header"/>
 </td> </tr>
 <tr> <td>
 <tile:insert attribute="leftpane"/>
 </td> <td>
 <tile:insert attribute="body"/>
 </td> </tr>
 <tr> <td> colspan="2">
 <tile:insert attribute="footer"/>
 </td> </tr>
 </table> </body></html>

* <!-- Header.jsp -->
 File > <%@ taglib uri="/beamtags" prefix="beam" %>
 Header Message

 <beam:message key="header.message"/>

Birthday / Anniversary

MR. SANTOSH C.A.S. NOTES
 Naresh Technologies
 PH: 9134842, 3374666

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29						

337-028, Week -08
File

Priority

SAT

3

DEC

2005

Mr. SANTOSH CLAS. NOTES
Naresh Technologies
Ph: 23734842, 23746666

Footer Message

* file

*> <!-- Footer.jsp -->

```
<li><a href = "Login.jsp">
    <bean:message key = "login.message" />
</a></li>

<li><a href = "Registration.jsp">
    <bean:message key = "registration.jsp" />
</a></li>

<%@ taglib uri = "/beantags" prefix = "bean" %> // it is
written at first line
```

* file <!-- LoginBody.jsp -->

```
<%@ taglib uri = "/beantags" prefix = "bean" %>
<pre> <form action = "login.do">
    <bean:message key = "uname" /> :
    <input type = "text" name = "uname" />
    <bean:message key = "pass" /> :
    <input type = "text" name = "password" name = "pass" />
    <input type = "Submit" name = "LogIN" />
</form></pre>
```

* file <!-- RegistrationBody.jsp -->

 Registration Page

* file

* <!-- Login.jsp -->

```
<%@ taglib uri = "/filetags" prefix = "file" %>
<file:insert page = "/MyLayout1.jsp" />
<file:put name = "header" value = "/Header.jsp" />
<file:put name = "leftpanel" value = "/Menu.jsp" />
```

SUN Priority

4

DEC

2005

December 2005						
S	M	T	W	T	F	S
			1	2	3	
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

338-027, Week 48

<file: put name = "body" value = "/LoginBody.jsp"/>
<file: put name = "footer" value = "/Footer.html"/>
</file:insert>

file

* <!-- Registration.jsp -->
<%@ taglib uri = "/filetags" prefix = "file" %>
<file:insert page = "/MyLayout.jsp"/>
<file: put name = "Header" value = "/Header.jsp"/>
<file: put name = "leftpane" value = "/Menu.jsp"/>
<file: put name = "body" value = "/RegistrationBody.jsp"/>
<file: put name = "Footer" value = "/Footer.jsp"/>
</insert </file:insert>

file

* # Application Resources Properties

uname = UserName

pass = Password

header.message = Header Message From Resource
Properties

Login.message = LogIn registration.message = Register

* <t-struts-config.xml -->

<struts-config>

<action-mappings>

<message-resource parameter = "ApplicationResources"/>

</struts-config>

Mr. SANTOSH AS NOTES
Naresh I Technologies
983734842, 23746666

Birthday / Anniversary

Birt

Priority

January 2006						
M	T	W	T	F	S	S
31			1			
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

339/0.26, Week 49

MON

5

DEC

2005

(file)

```

<!-- Web.xml -->
<web-app>
  <servlet>
    <servlet-name>org.apache.struts.action.ActionServlet</servlet-name>
    <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
    <load-on-startup>0</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>org.apache.struts.action.ActionServlet</servlet-name>
    <url-pattern>*.do</url-pattern>
  </servlet-mapping>
  <taglib>
    <taglib-uri>/beantags</taglib-uri>
    <taglib-location>/WEB-INF/tlds/struts-bean.tld</taglib-location>
  </taglib>
  <taglib>
    <taglib-uri>/tiles.tld</taglib-uri>
    <taglib-location>/WEB-INF/tlds/struts-tiles.tld</taglib-location>
  </taglib>
</web-app>

```

Mr. SANTOSH CLAS. NOTES
 Naresh Technologies
 PM 23734842, 23746666

TUE

Priority

6

DEC

2005

S	M	T	W	TH	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

340-025, Week 49

D:

8 am

MyLayout.jsp

Header.jsp

Footer.html

Menu.jsp

LoginBody.jsp

RegistrationBody.jsp

Login.jsp

Registration.jsp

WEB-INF

Struts-Config.xml

web.xml

classes

ApplicationResources.properties

ApplicationResources-i18n.properties

lib

tags

struts-bean.tld

struts-tiles.tld

Mr. SANTOSH CLASS NOTES

Naresh Technologies

Ph: 23734842, 23746666

January 2006							Priority	JSF API (Specification)	WED
S	M	T	W	T	F	S			
30	31	1	2	3	4	5	6	7	8
9	10	11	12	13	14	15			
16	17	18	19	20	21	22			
23	24	25	26	27	28	29			
341-024, Week 49									

WED

7

DEC

2005

JSF (Java Server Faces)

- is a specification from Sun
- is used to design a View which separates the presentation logic and the application logic
- Is designed on top of the Servlet Specification and the other part of JSF sits on top of JSP

JSF Specifications include

JSF API

- is used to implement the UserInterface components
- 3rd party vendors implement this API and we can enhance the existing UI Component functionality
- we can use this API for developing new UI components

This part of the JSF specification is used by the Project Developers

Mr. SANTOSH CLAS NOTES
Naresh I Technologies
Ph 23734842, 23746166

JSF Tags

- these are used by the view designers
- These tags represent the UI Components developed using JSF API

- JSF is designed based on the Servlet and JSP Specification

JSF LifeCycle

Birthday / Anniversary - JSF life is a Servlet named FacesServlet

THU
8
DEC
2005

Priority

December 2005						
S	M	T	W	T	F	S
			1	2	3	4
			5	6	7	8
			9	10	11	12
			13	14	15	16
			17	18	19	20
			21	22	23	24
			25	26	27	28
			29	30	31	

342-023, Week 49

which provides an entry point for all the JSF requests

- Faces Request

- It is a request made by the faces response generated in the previous request

- Non-Faces Request

- Request from a normal page

i.e., the request which is not associated with the view previously

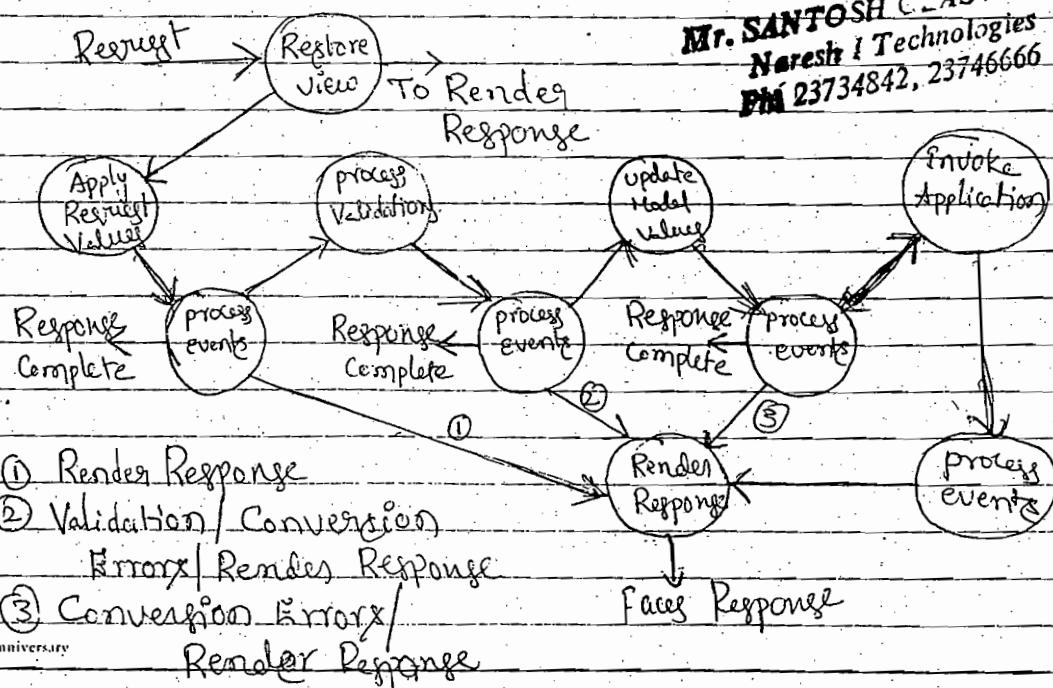
- Faces Response :-

- It is a response generated by the RenderResponse phase of the JSF

- Non-Faces Response :-

- It is a response which is not associated with the JSF view i.e., response not generated by the render response phase.

MR. SANTOSH CLASS NOTES
Naresh I Technologies
Ph 23734842, 23746666



January 2006						
M	T	W	T	F	S	S
31			1			
3	4	5	6	7	8	
10	11	12	13	14	15	
17	18	19	20	21	22	
24	25	26	27	28	29	

343-022, Week 49.

Priority _____

FRI

9

DEC

2005

Request View

- Find whether the request is a faces request or non-faces if it a faces request then locate the view which has been rendered in the previous request and restore the UI components (of the view)
- If it is non faces Request then the request is dispatched to the Render Response

Apply Request Values

Mr. SANTOSH CLAS
Naresh I Technol
PM 83734842, 23/10/06

- Set the request values into the UI components
- if required perform some data conversion and validation
- if it is dispatch successful then dispatch the request to process validation
- If Non-faces Response can be generated or dispatch the request to the render response.

Process Validation

- Validate the data set in the UI Components
- if successful then dispatch the request to update model component

Update Model Component

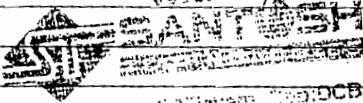
- here the UI components data is set into the model (entity, managed beans)
- performing some data conversions if required
- if successful then dispatch the request to invoke application phase

SAT
10
DEC
2005

Priority _____

now
Mr. Santosh

with



#2nd Floor, Saraswati Nilayam, Opp. DCB Bank
SR N257, Police Station Lane, Hyderabad, Hyd.

December 2005						
S	M	T	W	T	F	S
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

344-021, Week 49

Invoke Application

- Locates the model bean component which can process the request data, and invokes the application and using the response from the application and configurations, can identify the next view
For this, please collect the response from the application.

Render Response

- locates the view, prepares the view presentation reordering all the components and stores the view with a unique identity in the server
- send the response to the client (i.e., face response)

Elements of JSF Application

1) FaceServlet

javax.faces.webapp.FaceServlet is given under the JSF api and has to be registered in web.xml

Mr. SANTOSH CLASS NOTES
Naresh I Technologies
23734842, 23746666

This provides a centralized access for all the JSF views

2) faces Configuration file

faces-config.xml

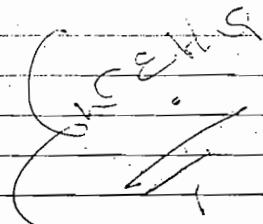
All the face configurations like declaring

the managed beans, configuring the navigation rule libraries like jsf-api.jar, jsf-impl.jar and jstl lib are also required jcr.jar and standard.jar

In JSP documents, can use JSF tag and JSF expression language

ION
2
DEC
005

Priority



December 2005						
S	M	T	W	T	F	S
4	5	6	7	8		
11	12	13	14	15		
18	19	20	21	22	23	
25	26	27	28	29	30	

346-019, Week 5

* * <!-- Home.jsp -->

<%@taglib uri="/jsfcore" prefix="f"%>

<%@taglib uri="/jsfhtml" prefix="h"%>

<f:view>

Welcome to <h:outputText value="#{loginbean.userID}" />

</f:view>

* //Managed Bean

package com.nit.jsferexample.ex1;

public class LoginBean

{

 String pass;
 int uid;

 public int getUserId() { return uid; }

 public void setUserID(String s)

 { userID = uid; }

 public int getPassword() { return pass; }

 public void setPassword(String s)

 { pass = s; }

 public String login()

 { //Here we can process the client request

 //like can directly use JDBC and interact with DB to check the validation

 //can dispatch the request to some other Java bean or EJB

 //To test the application

 if(uid == 1 & & pass.equals("pass1"))

 return "Success";

 return "Failed";

 //login

 //class

MR. SANTOSH C. AS, NOTES
Naresh Technologies
BM 83734842, 23740666

Birds

S	M	T	W	T	F	S
30	31				1	
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

347-018, Week 50

Priority ..

TUE

13

DEC

2005

* <!-- faces-config.xml -->

8am <!DOCTYPE ... >

<faces-config>

<managed-bean>

<managed-bean-name> loginbean </managed-bean-name>

<managed-bean-class> com.nit.jsfexample.ex1.LoginBean

</managed-bean-class>

<managed-bean-scope> session </managed-bean-scope>

</managed-bean></scope>

12

<navigation-rule>

1pm <from-view-id> /Login.jsp </from-view-id>

<navigation-case>

<from-output> \$success </from-output>

<to-view-id> /Home.faces </to-view-id>

</navigation-case>

<navigation-case>

<from-output> \$failed </from-output>

<to-view-id> /Error.jsp </to-view-id>

</navigation-case>

</navigation-rule>

</faces-config>

MR. SANTOSH CLAS, NOTES
Naresh Technologies
Ph: 23734842, 23746366

* <!-- web.xml -->

<web-app>

<Servlet>

<Servlet-name> fs </Servlet-name>

<Servlet-class>

javax.faces.webapp.FacesServlet

</Servlet-class>

Birthday / Anniversary

January 2006						
S	M	T	W	T	F	S
30	31			1		
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

347-018. Week 50

Priority ..

TUE

13

DEC

2005

* * < !-- faces-config.xml -->

8am <!DOCTYPE ... >

<faces-config>

<managed-bean>

<managed-bean-name> loginbean </managed-bean-name>

<managed-bean-class> com.nit.jspexample.ex1.LoginBean

</managed-bean-class>

<managed-bean-scope> session </managed-bean-scope>

</managed-bean> </scope>

12

<navigation-rule>

<from-view-id> /Login.jsp </from-view-id>

<navigation-case>

<from-output> ^{come} success </from-output>

<to-view-id> /Home.faces </to-view-id>

</navigation-case>

<navigation-case>

<from-output> ^{come} failed </from-output>

<to-view-id> /Error.jsp </to-view-id>

</navigation-case>

</navigation-rule>

</faces-config>

* * < !-- web.xml -->

<web-app>

<servlet>

<servlet-name> fs </servlet-name>

<servlet-class>

javax.faces.webapp.FacesServlet

Birthday / Anniversary

</servlet-class>

Mr. SANTOSH CLAS. NOTES
Naresh Technologies
Ph: 23734842, 23746666

WED

JSF - impl - jar
Priority web - faces - config - l - l - & etc

14

DEC

2005

December 2005

S	M	T	W	T	F	S
			1	2	3	
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

348-017, Week 50

<load-on-startup>0</load-on-startup>
8 am </servlet>
<servlet-mapping>
<servlet-name>faces</servlet-name>
<url-pattern>*.faces</url-pattern>
10 am </servlet-mapping>
<taglib>
11 <taglib-uri>/jsfcore</taglib-uri>
<taglib-location>/WEB-INF/tlds/jsf-core.tld
12 </taglib-location>
</taglib>
1 pm <taglib>
<taglib-uri>/jsfhtml</taglib-uri>
<taglib-location>/WEB-INF/tlds/html= basic.tld
13 </taglib-location>
</taglib>
</web-app>
14 <CDTYPE faces-

Mr. SANTOSH C - AS, NOTE
Naresh Technologies
Ph: 23734842, 23746666

<http://localhost:8080/jsf-ex/Login.faces>

January 2006				
	W	T	F	S
				1
		5	6	7
		12	13	14
		18	19	20
		25	26	27
				29
				30
				31
				1
				2
				3
				4
				5
				6
				7
				8
				9
				10
				11
				12
				13
				14
				15
				16
				17
				18
				19
				20
				21
				22
				23
				24
				25
				26
				27
				28
				29
				30
				31
				1
				2
				3
				4
				5
				6
				7
				8
				9
				10
				11
				12
				13
				14
				15
				16
				17
				18
				19
				20
				21
				22
				23
				24
				25
				26
				27
				28
				29
				30
				31
				1
				2
				3
				4
				5
				6
				7
				8
				9
				10
				11
				12
				13
				14
				15
				16
				17
				18
				19
				20
				21
				22
				23
				24
				25
				26
				27
				28
				29
				30
				31
				1
				2
				3
				4
				5
				6
				7
				8
				9
				10
				11
				12
				13
				14
				15
				16
				17
				18
				19
				20
				21
				22
				23
				24
				25
				26
				27
				28
				29
				30
				31
				1
				2
				3
				4
				5
				6
				7
				8
				9
				10
				11
				12
				13
				14
				15
				16
				17
				18
				19
				20
				21
				22
				23
				24
				25
				26
				27
				28
				29
				30
				31
				1
				2
				3
				4
				5
				6
				7
				8
				9
				10
				11
				12
				13
				14
				15
				16
				17
				18
				19
				20
				21
				22
				23
				24
				25
				26
				27
				28
				29
				30
				31
				1
				2
				3
				4
				5
				6
				7
				8
				9
				10
				11
				12
				13
				14
				15
				16
				17
				18
				19
				20
				21
				22
				23
				24
				25
				26
				27
				28
				29
				30
				31
				1
				2
				3
				4
				5
				6
				7
				8
				9
				10
				11
				12
				13
				14
				15
				16
				17
				18
				19
				20
				21
				22
				23
				24
				25
				26
				27
				28
				29
				30
				31
				1
				2
				3
				4
				5
				6
				7
				8
				9
				10
				11
				12
				13
				14
				15
				16
				17
				18
				19
				20
				21
				22
				23
				24
				25
				26
				27
				28
				29
				30
				31
				1
				2
				3
				4
				5
				6
				7
				8
				9
				10
				11
				12
				13
				14
				15
				16
				17
				18
				19
				20
				21
				22
				23
				24
				25
				26
				27
				28
				29
				30
				31
				1
				2
				3
				4
				5
				6
				7
				8
				9
				10
				11
				12
				13
				14
				15
				16
				17
				18
				19
				20
				21
				22
				23
				24
				25
				26
				27
				28
				29
				30
				31
				1
				2
				3
				4
				5
				6
				7
				8
				9
				10
				11
				12
				13
				14
				15
				16
				17
				18
				19
				20
				21
				22
				23
				24
				25
				26
				27
				28
				29
				30
				31
				1
				2
				3
				4
				5
				6
				7
				8
				9
				10
				11
				12
				13
				14
				15
				16
				17
				18
				19
				20
				21
				22
				23
				24
				25
				26
				27
				28
				29
				30
				31
				1
				2
				3
				4
				5
				6
				7
				8
				9
				10
				11
				12
				13
				14
				15
				16
				17
				18
				19
				20
				21
				22
				23
				24
				25
				26
				27
				28
				29
				30
				31
				1
				2
				3
				4
				5
				6
				7

Priority

THU

15

DEC

2005

15

Login.jsp
Home.jsp

WEB-INF

facey-config.xml
web.xml

Web.xml

Classy

|: LoginBean.java

200

Unit

List example

Lexi

٦٦

Commons - beautiful jar

Commons - Collections

common - digestef .jar

common - logging.jar

jsf-api-star

jsf-impl.jar

jstl.jar

Standard-Jar

101

html basic

jsf-core.tld

If we want to validate the input component

Valdade Fong Pong

FRI Priority

16
DEC
2005

Mr. SANTOSH
Naresh I Technologies
P.M. 23734842, 23746666

December 2005						
S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

350 015 Week 50

ValidateLongRange

- takes 2 attributes
- minimum takes the min value
- maximum takes the max value

this tag ^{has to be} enclosed within the input vi for which ~~within~~ which the validation rule has to be imposed

Example:-

```
<h:inputText id="UserID" ...>  
<f:validateLongRange minimum="1" maximum="100" />  
</h:inputText>
```

ValidateDoubleRange

is used to validate the double value.

ValidateLength

used to validate the length of the given string

Example:-

```
<h:inputText id="password" ...>  
<f:validateLength minimum="2" maximum="5" />  
</h:inputText>
```

Mr. SANTOSH CLASS NOTES
Naresh I Technologies
P.M. 23734842, 23746666

