**Working with Date obj:**

|-1) Constructor with a single String parameter (or) static method named valueOf() that takes a single String argument.

|-2) Using ParamConverter

**Working with Date obj:**

Working with Date obj is common thing, So in order to convert into Date obj we can use 2-ways

1) Constructor with a single String parameter (or) static method named valueOf() that takes a single String argument.

2) Using ParamConverter

**1) Constructor with a single String parameter (or) static method named valueOf() that takes a single String argument:**

class DateParam {

private final Date date;

// Single arg Constructor as String param

public DateParam(String datePathParam) throws WebApplicationException {

final DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

try {

this.date = dateFormat.parse(datePathParam);

} catch (ParseException p) {

throw new WebApplicationException("Invalid Date Format");

}

}

public Date getDate() {

return date;

}

}

@Path("/cric/{tournament}/{date}")

class CricInfoResource {

@GET

@Produces(MediaType.TEXT\_PLAIN)

@Path("/score/{team1}/{team2}")

public String getScore(

@PathParam("tournament") String tournament,

@PathParam("date") DateParam dateParam,

@PathParam("team1") String team1,

@PathParam("team2") String team2) {

Date date = dateParam.getDate();

java.sql.Date sqlDate = new java.sql.Date(date.getTime());

return "Java Util Date: " + date + "\nSql Date: " + sqlDate;

}

}

Access the application:

http://localhost:8080/12.8.5.1WorkingWithDatesSingleArgConstructor

/resources/cric/T20/2016-09-09/score/ind/pak

Response:

Java Util Date: Fri Sep 09 00:00:00 IST 2016

Sql Date: 2016-09-09

**2) Using ParamConverter:**

@Path("/cric/{tournament}/{date1}")

class CricInfoResource {

@GET

@Produces(MediaType.TEXT\_PLAIN)

@Path("/score/{team1}/{team2}")

public String getScore(

@PathParam("tournament") String tournament,

@PathParam("date1") Date date,

@PathParam("team1") String team1,

@PathParam("team2") String team2) {

long time = date.getTime();

java.sql.Date sqlDate = new java.sql.Date(time);

return "Java Util Date: " + date + "\nSql Date: " + sqlDate;

}

}

class DateParameterConverter implements ParamConverter<Date> {

// set the format to whatever we need

public static final String format = "yyyy-MM-dd";

@Override

public Date fromString(String dateString) {

SimpleDateFormat simpleDateFormat = new SimpleDateFormat(format);

try {

return simpleDateFormat.parse(dateString);

} catch (ParseException ex) {

throw new WebApplicationException(ex);

}

}

@Override

public String toString(Date date) {

return new SimpleDateFormat(format).format(date);

}

}

@Provider

class AppParamConverterProvider implements ParamConverterProvider {

@Override

public <T> ParamConverter<T> getConverter(Class<T> classType,

Type rawType, Annotation[] annotations) {

if (classType.isAssignableFrom(Date.class)) {

return (ParamConverter<T>) new DateParameterConverter();

}

return null;

}

}

http://localhost:8083/12.8.5.2WorkingWithDatesParamConverter

/resources/cric/T20/2016-09-09/score/ind/pak

Response:

Java Util Date: Fri Sep 09 00:00:00 IST 2016

Sql Date: 2016-09-09