**Date Format Validation Check**

Calendar cal = Calendar.*getInstance*();

cal.add(Calendar.***DATE***, -1);

Date yesterday = cal.getTime();

Date a = sdf.parse(req.getEffectivedate());

**if** (req.getEffectivedate() == **null** || StringUtils.*isBlank*(req.getEffectivedate().toString())) {

errors.add(**new** Error("07", "Effective Date", "Please Enter Effective Date"));

} **else** **if** (a.before(yesterday)) {

errors.add(**new** Error("07", "EffectiveDate", "Please Enter Future Date as EffectiveDate"));

} **else** **if** (!req.getEffectivedate().matches("([0-9]{2})/([0-9]{2})/([0-9]{4})")) {

errors.add(**new** Error("07", "EffectiveDate",

"Effective Date format should be dd/MM/yyyy only allowed . Example :- 15/12/2022"));

}

Date endDate = sdf.parse(req.getEnddate());

Date effectiveDate = sdf.parse(req.getEffectivedate());

**if** (req.getEnddate() == **null** || StringUtils.*isBlank*(req.getEnddate().toString())) {

errors.add(**new** Error("08", "End Date", "Please Enter End Date"));

} **else** **if** (endDate.before(effectiveDate)) {

errors.add(**new** Error("08", "End Date", "End Date not before Effective Date"));

} **else** **if** (!req.getEnddate().matches("([0-9]{2})/([0-9]{2})/([0-9]{4})")) {

errors.add(**new** Error("08", "End Date",

"End Date format should be dd/MM/yyyy only allowed . Example :- 15/12/2020"));

}