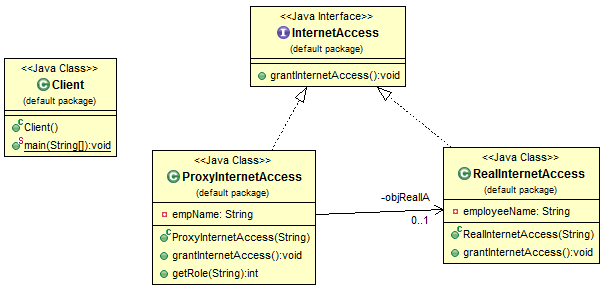
Use objectaid-1.2.1 for drawing UML diagram

Proxy Pattern:

**"Provides the control for accessing the original object".** So, we can perform many operations like hiding the information of original object, on demand loading etc. Proxy pattern is also known as Surrogate or Placeholder. And it comes under Structural pattern

**Advantage of Proxy Pattern:** It provides the protection to the original object from the outside world.

In the real work a cheque or credit card is a proxy for what is in our bank account.  It can be used in place of cash, which is what is needed, and provides a means of accessing that cash when required. And that's exactly what the Proxy pattern does - controls and manage access to the object they are "protecting".





To download file:

@GET

    @Path("/GetSinglePDF")

    @Produces("application/pdf")

**public** Response getSinglepdf() {

        File file = **new** File(singlePdfPath);

**return** Response.*ok*(file, "application/pdf").header("Content-Disposition", "attachment; filename=\"" + file.getName() + "\"").build();

    }

**public** **void** removeFavorite(Long userId, Long pictoFamilyId) {

        Query query = entityManager.createNativeQuery("delete from PV2QTFAV where USR\_ID = :user and PFM\_ID = :picto");

        query.setParameter("user", userId);

        query.setParameter("picto", pictoFamilyId);

        query.executeUpdate();

    }

**public** List<BigInteger> getFavPictoIDbyUser(Long userId) {

        Query query = entityManager.createNativeQuery("select PFM\_ID from PV2QTFAV where USR\_ID = :user");

        query.setParameter("user", userId);

**return** query.getResultList();

    }