

Java Client - Model-Defined FetchSpecifications

Client code:

```
private static final EODistributedObjectStore dos =
    (EODistributedObjectStore)(EOEditingContext.defaultParentObjectStore());

    /**
     * Invokes the remote method on session to get a named fetch specification.
     * Needed because
     * the whole eoaccess package is excluded from the client side frameworks.
     *
     * @param spec Fetch specification name.
     * @param entity Name of the entity for which the fetch specification is
     * for.
     * @return The fetch specification, or null if it is not in the model, or an
     * error occured
     */
    public static EOFetchSpecification getFetchSpecification(
        String spec,
        String entity){

        //invoke!
        return (EOFetchSpecification) dos.invokeStatelessRemoteMethodWithKeyPath(
            "session",
            "clientSideRequestGetFetchSpecification",
            new Class[]{String.class, String.class},
            new String[]{spec, entity});
    }
}
```

Server code, in the Session class:

```
    /**
 * Loads and returns an <tt>EOFetchSpecification</tt> objects for the given
 * specification and entity name.
 *
 * @param spec Name of the specification
 * @param entity Name of the entity
 * @return A fetch specification, or <tt>>null</tt> if it is not found, or an
 *         error occurs
 */
public EOfetchSpecification getFetchSpecification(String spec, String
entity){
    try{
        return EOfetchSpecification.fetchSpecificationNamed(spec, entity);
    }catch(Exception ex){
        return null;
    }
}
```

courtesy Florijan Stamenkovic