```java

package acme.features.client.progresslog;

import java.util.Collection;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import acme.client.data.models.Dataset;
import acme.client.services.AbstractService;
import acme.entities.contract.Contract;
import acme.entities.contract.ProgressLog;
import acme.roles.Client;

@Service
public class ClientProgressLogListAllService extends AbstractService<Client, ProgressLog> {

    // Internal state --------------------------------------------------------

    @Autowired
    private ClientProgressLogRepository repository;

    // AbstractService interface ----------------------------------------------


    @Override
    public void authorise() {
        boolean status;
        int contractId = super.getRequest().getData("masterId", int.class);
        Contract contract;
        contract = this.repository.findOneContractById(contractId);
        status = super.getRequest().getPrincipal().getActiveRoleId() == contract.getClient().getId() && contract != null && super.getRequest().getPrincipal().hasRole(super.getRequest().getPrincipal().getActiveRole());

        super.getResponse().setAuthorised(status);
    }

    @Override
    public void load() {
        Collection<ProgressLog> objects;
        int masterId;
        masterId = super.getRequest().getData("masterId", int.class);
        objects = this.repository.findAllProgressLogsByContractId(masterId);
        super.getBuffer().addData(objects);

    }

    @Override
    public void unbind(final ProgressLog object) {
        assert object != null;

        Dataset dataset;

        dataset = super.unbind(object, "recordId", "completeness", "registrationMoment", "reponsiblePerson", "isDraft");

        super.getResponse().addData(dataset);
    }

    @Override
    public void unbind(final Collection<ProgressLog> objects) {
        assert objects != null;

        int masterId;
        Contract contract;
        final boolean showCreate;

        masterId = super.getRequest().getData("masterId", int.class);
        contract = this.repository.findOneContractById(masterId);
        showCreate = contract.isDraft();

        super.getResponse().addGlobal("masterId", masterId);
        super.getResponse().addGlobal("showCreate", showCreate);
        super.getResponse().addGlobal("isDraft", contract.isDraft());
    }
}

```