

Система электронного документооборота



**Адаптация листа согласования
под дизайнерский процесс**

Версия 5.х.х

Памятка Администратора

г. Самара,
2022

Введение

Достаточно часто возникает необходимость адаптировать стандартный отчёт «Лист согласования» под дизайнерский процесс.

В данной Памятке Администратора код, исполняемый в закрытых классах Системы, представлен целиком в виде groovy-скрипта в соответствующих полосах.

В начале скриптов имеется блок параметров, в которых требуется указать данные нового процесса, а именно перечень кодов ролей, состояний и соответствий между ролью и состоянием.

1. Адаптация листа согласования под дизайнерский процесс

Необходимые действия:

1. Зайти «Администрирование» – «Базовые настройки» – «Отчёты» (для версии 5.0.x необходимо зайти «Администрирование» – «Отчёты») – выбрать отчёт «Лист согласования» – нажать правую кнопку мыши и выбрать пункт «Копировать».

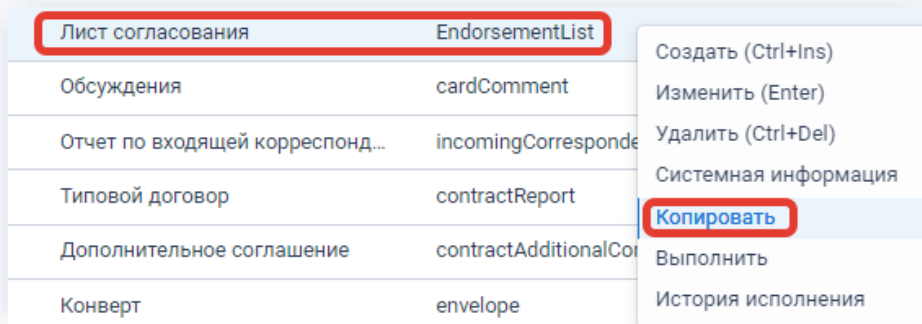


Рисунок 1. Список отчётов

2. Открыть созданную копию отчёта и задать необходимое имя на вкладках «Структура отчёта» и «Локализация».

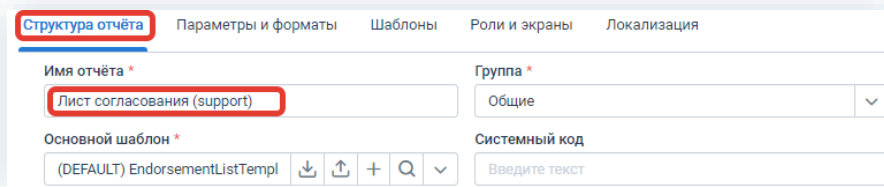


Рисунок 2. Вкладка «Структура отчёта»

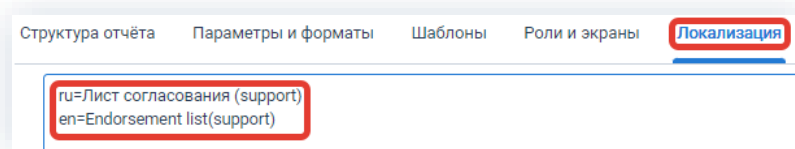


Рисунок 3. Вкладка «Локализация»

3. На вкладке «Структура отчёта» открыть полосу отчёта «FieldsInfo» и вставить код в поле «Скрипт».

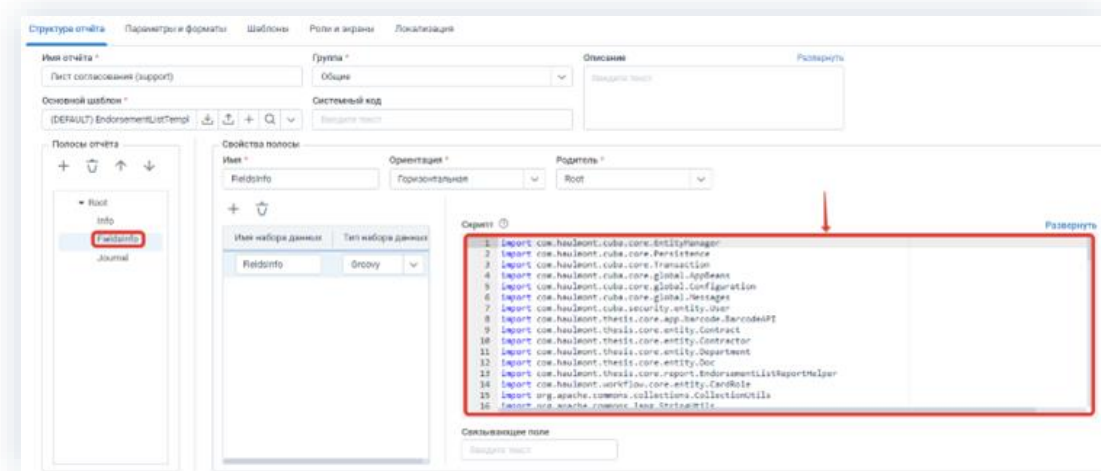


Рисунок 4. Вкладка «Структура отчёта» – полоса «FieldsInfo»

Добавляемый код:

```
import com.haulmont.cuba.core.EntityManager
import com.haulmont.cuba.core.Persistence
import com.haulmont.cuba.core.Transaction
import com.haulmont.cuba.core.global.AppBeans
import com.haulmont.cuba.core.global.Configuration
import com.haulmont.cuba.core.global.Messages
import com.haulmont.cuba.security.entity.User
import com.haulmont.thesis.core.app.barcode.BarcodeAPI
import com.haulmont.thesis.core.entity.Contract
import com.haulmont.thesis.core.entity.Contractor
import com.haulmont.thesis.core.entity.Department
import com.haulmont.thesis.core.entity.Doc
import com.haulmont.thesis.core.report.EndorsementListReportHelper
import com.haulmont.workflow.core.entity.CardRole
import org.apache.commons.collections.CollectionUtils
import org.apache.commons.lang.StringUtils
import org.apache.commons.lang.exception.ExceptionUtils
import org.apache.commons.logging.Log
import org.apache.commons.logging.LogFactory

import java.text.DecimalFormat

Log log = LogFactory.getLog("EndorsementList.FieldsInfo");

Persistence persistence = AppBeans.get(Persistence.class);
BarcodeAPI barcodeAPI = AppBeans.get(BarcodeAPI.class);
Configuration configuration = AppBeans.get(Configuration.class);
Messages messages = AppBeans.get(Messages.class);

def roleCodes = ['Initiator', 'Approver']; //Роли пользователей (коды), фио которых
нужно вытянуть в перечень полей.
def procCode = 'Endorsement' // Код процесса из "Администрирование - Базовые
настройки - Процессы" (для версии 5.0.x необходимо зайти "Администрирование -
Процессы")

boolean isVatRequired = true // true, если требуется выводить в отчёт "с НДС \ без
НДС"

List<Map<String, Object>> result = []
Transaction tx = persistence.createTransaction()
```

```

List<CardRole> cardRoles=new ArrayList<>()
try {
    EntityManager em = persistence.getEntityManager()
    Doc doc = params['entity'] as Doc
    String queryString = "select d from " + doc.metaClass.name + " d where d.id =
:doc"
    def query = em.createQuery(queryString)
    query.setParameter('doc', doc)
    List<Doc> docs = query.getResultList()
    Doc resultDoc = docs.iterator().next()
    result << [fieldCaption: messages.getMessage(EndorsementListReportHelper.class,
'Description'), fieldValue: resultDoc.description]
    result << [fieldCaption: messages.getMessage(EndorsementListReportHelper.class,
'Comment'), fieldValue: resultDoc.comment]

    if (resultDoc instanceof Contract) {
        String vat = ''
        if (isVatRequired)
            vat = ((Contract) resultDoc).getVatInclusive() ? " с НДС" : " без НДС"
        Contractor contractor = ((Contract) resultDoc).contractor;
        if (contractor != null)
            result << [fieldCaption:
messages.getMessage(EndorsementListReportHelper.class, 'Contractor'), fieldValue:
contractor ? contractor.name : '']

        DecimalFormat format = new DecimalFormat("###,###,##0.00");
        if(resultDoc.amount!=null){
            String amountString = format.format(resultDoc.amount) + ' ' +
resultDoc.currency?.instanceName;

            result << [fieldCaption:
messages.getMessage(EndorsementListReportHelper.class, 'Amount'), fieldValue:
((Contract) resultDoc).amount != null ?
amountString + vat : '']]
        }

        Transaction tx1 = persistence.getTransaction();
        try {
            em = persistence.entityManager
            if (doc && roleCodes) {
                query = em.createQuery(''select cr from wf$CardRole cr join
cr.procRole pr
                where cr.card.id = :id and cr.code in:codes
                and pr.proc.code = :procCode order by pr.sortOrder desc'')
                query.setParameter('id', doc)
                query.setParameter('procCode', procCode)
                query.setParameter('codes', roleCodes)

            query.setView(EndorsementListReportHelper.EndorsementViewProvider.cardRoleView)
            tx1.commit();
            cardRoles = query.resultList
        }
        tx1.commit();
    } catch (Exception e) {
        log.error(ExceptionUtils.getStackTrace(e));
    } finally {
        tx1.end();
    }

    if (!CollectionUtils.isEmpty(cardRoles)) {
        def cardRolesCodes= roleCodes;
        cardRolesCodes.each { code ->
            CardRole cr = cardRoles.find { code.equals(it.code) }
            if (cr) {
                User resultUser = cr.user
                em = persistence.entityManager
            }
        }
    }
}

```

```

        queryString = ""select d from df\$Employee e join e.department d
where e.user.id = :user""
        query = em.createQuery(queryString)
        query.setParameter('user', resultUser)
        List<Department> departments = query.getResultList()

        for (String role : roleCodes) {
            if (role.equals(code)) {
                if (departments.isEmpty())
                    result << [fieldCaption:
messages.getMessage(EndorsementListReportHelper.class,
("Initiator".equals(code)?'Doc:') + code), fieldValue: resultUser.name.trim() +
(StringUtils.isBlank(resultUser.position) ? ""
: ", " + resultUser.position.trim())]
                else {
                    Department resultDepartment =
departments.iterator().next()
                    result << [fieldCaption:
messages.getMessage(EndorsementListReportHelper.class,
("Initiator".equals(code)?'Doc:') + code), fieldValue: resultUser.name.trim() +
(StringUtils.isBlank(resultUser.position) ? ""
: ", " + resultUser.position.trim()) +
(StringUtils.isBlank(resultDepartment.name) ?
"" : ", " + resultDepartment.name.trim())]
                }
            }
        }
        else {
            result << [fieldCaption:
messages.getMessage(EndorsementListReportHelper.class,
("Initiator".equals(code)?'Doc:') + code), fieldValue: '']
        }
    }

    tx.commit()
} catch (Exception e) {
    log.error(ExceptionUtils.getStackTrace(e))
} finally {
    tx.end()
}
return result

```

4. На вкладке «Структура отчёта» открыть полосу отчёта «Journal» и вставить код в поле «Скрипт».

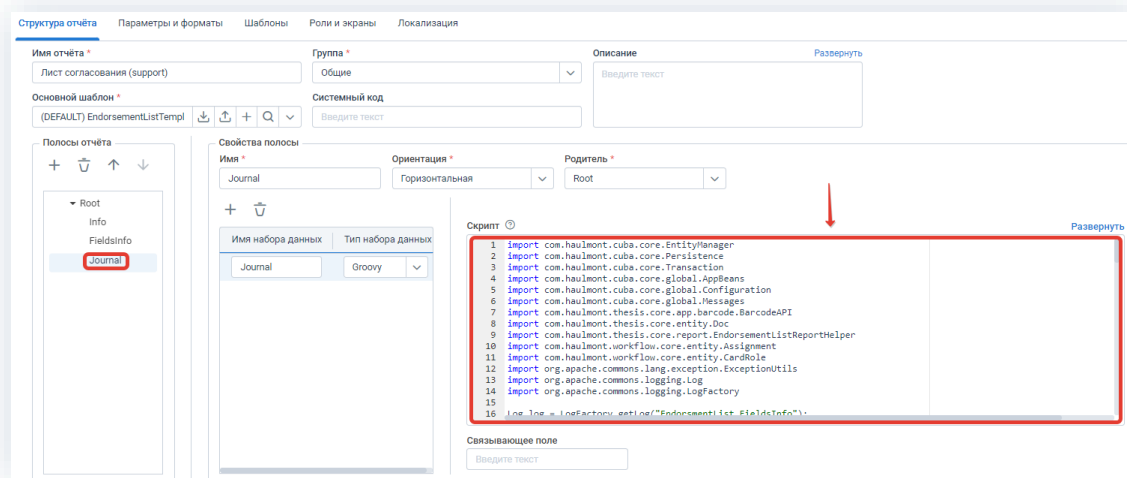


Рисунок 5. Вкладка «Структура отчёта» – полоса «Journal»

Добавляемый код:

```
import com.haulmont.cuba.core.EntityManager
import com.haulmont.cuba.core.Persistence
import com.haulmont.cuba.core.Transaction
import com.haulmont.cuba.core.global.AppBeans
import com.haulmont.cuba.core.global.Configuration
import com.haulmont.cuba.core.global.Messages
import com.haulmont.thesis.core.app.barcode.BarcodeAPI
import com.haulmont.thesis.core.entity.Doc
import com.haulmont.thesis.core.report.EndorsementListReportHelper
import com.haulmont.workflow.core.entity.Assignment
import com.haulmont.workflow.core.entity.CardRole
import org.apache.commons.lang.exception.ExceptionUtils
import org.apache.commons.logging.Log
import org.apache.commons.logging.LogFactory
import com.haulmont.thesis.core.config.FacsimileConfig
import com.haulmont.thesis.core.app.OrgStructureWorker
import com.haulmont.cuba.core.app.FileStorageService
import com.haulmont.thesis.core.entity.Employee
import com.haulmont.cuba.core.app.DataService
import com.haulmont.cuba.core.global.LoadContext
```

```
Log log = LogFactory.getLog("EndorsmentList.FieldsInfo");
```

```
Persistence persistence = AppBeans.get(Persistence.class);
BarcodeAPI barcodeAPI = AppBeans.get(BarcodeAPI.class);
Configuration configuration = AppBeans.get(Configuration.class);
Messages messages = AppBeans.get(Messages.class);
```

```
def sorted = 'createTs desc' //Порядок сортировки в таблице. Указывается поле и
направление.
def roleCodes = ['Согласующий', 'Участник']; // Перечень ролей, которые будут
браться в отчёт.
def procCode = 'КодПроцесса' // Администрирование - Базовые настройки - Процессы (для
версии 5.0.x необходимо зайти "Администрирование - Процессы"), который будет
привязан к отчёту.
```

```
def procRoleStateMap = ['Согласующий': 'Soglasovanie', 'Участник': 'Rassylka'] //
Карта соответствия роли и состояний. Указывается именно код роли и имя этапа. Эти
этапы будут попадать в лист согласования.
Doc doc = params['entity'] as Doc;
```

```
String name = "";
List<Map<String, Object>> result = []
Transaction tx = persistence.getTransaction()
try {
    List<CardRole> roles;

    Transaction tx1 = persistence.getTransaction();
    try {
        def em = persistence.entityManager
        if (doc && roleCodes) {
            def query = em.createQuery(''select cr from wf$CardRole cr join
cr.procRole pr
            where cr.card.id = :id and cr.code in:codes
            and pr.proc.code = :procCode order by pr.sortOrder desc'')
            query.setParameter('id', doc)
            query.setParameter('procCode', procCode)
            query.setParameter('codes', roleCodes)

            query.setView(EndorsementListReportHelper.EndorsementViewProvider.cardRoleView)
            tx1.commit();
            roles = query.resultList
        }
        tx1.commit();
    } catch (Exception e) {
        log.error(ExceptionUtils.getStackTrace(e));
    } finally {
        tx1.end();
    }

    List<Assignment> assignments;
    tx1 = persistence.getTransaction();
    try {
        EntityManager em = persistence.getEntityManager()
        em.flush();
        String queryString = ""select a from wf\Assignment a where a.card.id =
:doc
        and a.proc.code like :procCode and a.user is not null order by
a."" + sorted //Здесь можно задать порядок вывода в печатную таблицу. DESC или ASC
        def query = em.createQuery(queryString)
        query.setView(null);
        query.setParameter('doc', doc)
        query.setParameter('procCode', procCode)
        assignments = query.getResultList()
        if (!assignments.isEmpty()) {
            def a = assignments.max { it.createTs; it.name == 'Started' }
            assignments = assignments.findAll {
                it.createTs == null || a.createTs == null ||
it.createTs.compareTo(a.createTs) >= 0
            }
        }
        tx1.commit();
    } catch (Exception e) {
        log.error(ExceptionUtils.getStackTrace(e));
    } finally {
        tx1.end();
    }

    assignments.each { asm ->
        name = asm.displayUser;
        if ((assignments.max {
            it.iteration; it.name == asm.name && it.user == asm.user
        }).iteration == asm.iteration) {
            boolean existsForAssignment = false

            roleStateMap = procRoleStateMap
            if (roles != null) {
```

```

    }
    for (CardRole role : roles) {
        if (role.user.equals(asm.user) &&
asm.name.equals(roleStateMap[role.code]) &&
            (role.procRole == null || role.procRole.proc == null ||
role.procRole.proc.equals(asm.proc)))
            existsForAssignment = true
    }

    if (existsForAssignment) { //Здесь можно добавить дополнительные поля
для вывода в таблицу
        result << ['Result' : asm.locOutcomeResult,
                    'Date' : asm.finished,
                    'UserName': name,
                    'Comment' : asm.comment,
                    'Position': asm.user.position,
                    'facsimile': getFacsimile(asm)] //Добавили формирование
еще одного параметра для вывода "Факсимильная подпись"
        }
    }
    tx.commit()
} catch (Exception e) {
    log.error(ExceptionUtils.getStackTrace(e))
} finally {
    tx.end()
}

result.sort{it.Date}
return result

byte[] getFacsimile(Assignment assignment) {
    byte[] bytes = null;
    Configuration configuration = AppBeans.get(Configuration.class);
    FileStorageService fileStorageService =
AppBeans.get(FileStorageService.class)
    OrgStructureWorker orgStructureWorker =
AppBeans.get(OrgStructureWorker.class)

    FacsimileConfig facsimileConfig =
configuration.getConfig(FacsimileConfig.class)
    if (!facsimileConfig.getUseFacsimile() ||
!"Soglasovat".equals(assignment.getOutcome())) return null; // Здесь укажите
положительный выход, для которого проставляется факсимиле

    Employee employee =
orgStructureWorker.getEmployeeByUser(assignment.getFinishedByUser());
    if (employee == null || employee.getFacsimile() == null) return null;
    try {
        bytes = fileStorageService.loadFile(employee.getFacsimile());
    } catch (Exception ignore) {}
    return bytes;
}

```

5. В полосе «FieldsInfo» в строке 30 указать роли, используемые в дизайнера процесса.

В отчёте роли указанные в строке 30 будут отображаться в верхней части страницы, до таблицы с Согласующими.

В качестве примера на рисунке ниже указаны роли: «Initiator», «Согласующий», «Подписант», «Утверждающий», «Ознакамливающийся».

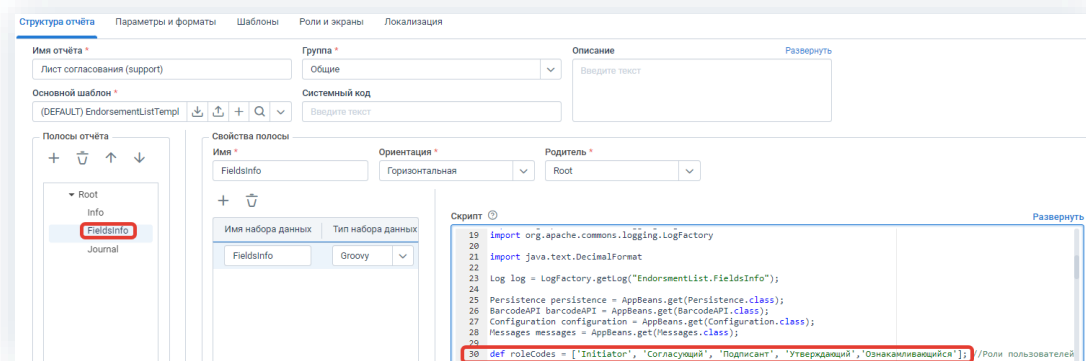


Рисунок 6. Вкладка «Структура отчёта» – полоса «FieldsInfo»

6. В полосе «FieldsInfo» в строке 31 указать «Код процесса».

Для того, чтобы узнать код процесса необходимо перейти в «Администрирование» – «Процессы» (Для версии 5.0.X «Администрирование» – «Процессы») – открыть необходимый процесс и скопировать значение из поля «Код».

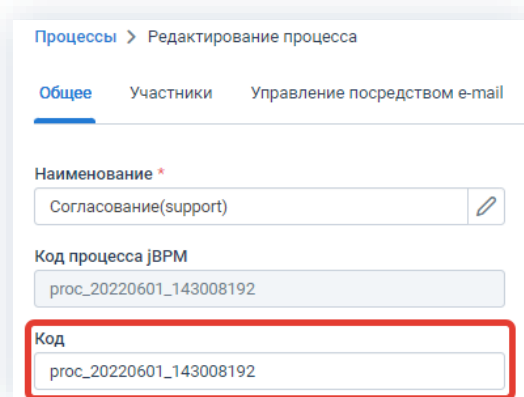


Рисунок 7. Вкладка «Общее» – поле «Код»

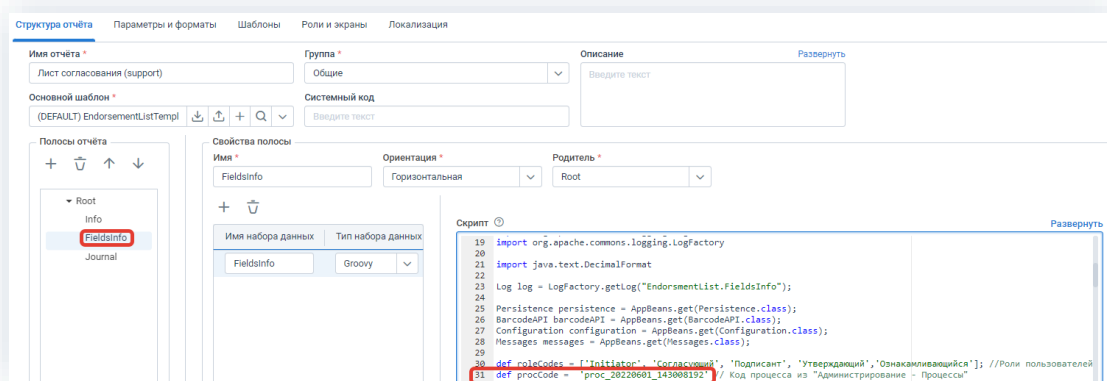


Рисунок 8. Вкладка «Структура отчёта» – полоса «FieldsInfo»

7. В полосе «FieldsInfo» в строке 32 возможно настроить видимость строки «Сумма с НДС / без НДС».

Для этого необходимо указать значение для параметра «boolean isVatRequired», где:

- «True» – отображать строку;
- «False» – скрыть строку.

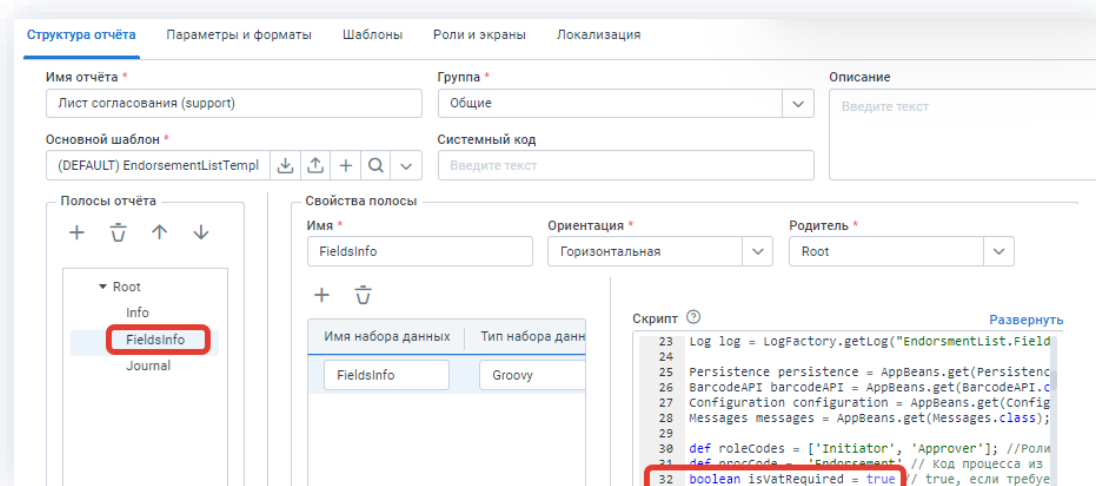


Рисунок 9. Вкладка «Структура отчёта» – полоса «FieldsInfo»

8. В полосе «Journal» в строке 31 указать роли, которые будут выводиться в таблице.

В качестве примера на рисунке ниже оставлены роли: «Согласующий», «Подписант», «Утверждающий»

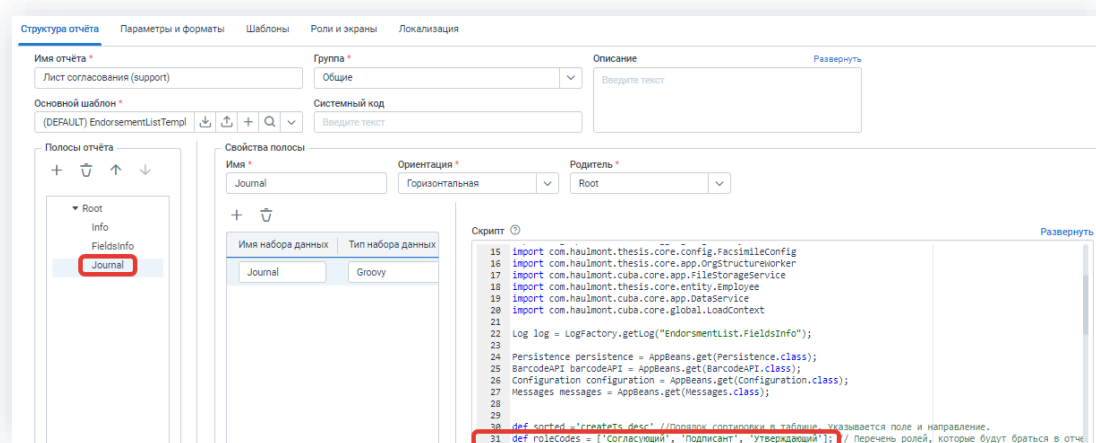


Рисунок 10. Вкладка «Структура отчёта» – полоса «Journal»

9. В полосе «Journal» в строке 32 указать «Код» процесса из пункта 6.

10. В полосе «Journal» в строке 33 указать соответствие между ролью и состоянием.

Важно!

Данный чек-бокс необходимо снять после завершения адаптирования отчёта и очистить состояния из поля «Состояния, для которых запрещен».

Чек-бокс нужен для того, чтобы узнать наименования состояний.

В поле «Состояния, для которых запрещен» в выпадающем списке выбрать необходимые состояния, скопировать наименование и вставить в код.

Процессы > Редактирование процесса

Общие Участники Управление посредством e-mail

Наименование *
Согласование(support)

Код процесса jBPM
proc_20220601_144926946

Код
proc_20220601_143008192

Типы карточек (через запятую)
Введите текст

Доступен только для роли
Выберите значение

Дизайн
Согласование(support)

☐ Разрешить комбинированные этапы
☐ Разрешить задание длительности
☐ Разрешить изменять участников процесса
☐ Доступен для мобильного клиента
☒ Запретить повторно запускать процесс

Состояния, для которых запрещён повторный запуск (через запятую)
Введите текст

Выберите значение

- Podpisanie
- Soglasovanie
- Started
- Dorabotka
- Oznakomlenie
- Utverzhdenie
- Oznakomlenie_zavershenno
- Canceled
- Otkloneno

Рисунок 11. Вкладка «Общие» – Состояния

В коде это будет выглядеть, как представлено на рисунке ниже.

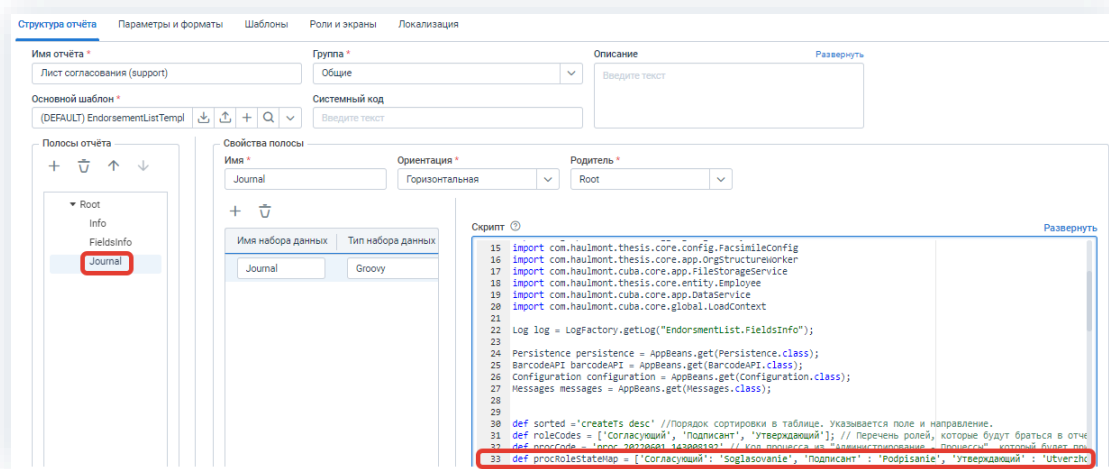


Рисунок 12. Вкладка «Структура отчёта» – полоса «Journal»

11. Нажать кнопку «Сохранить» в нижней части экрана.

Отчёт адаптирован под дизайнерский процесс и имеет вид, представленный на рисунке ниже.

Общество с ограниченной ответственностью "СтартИнвестПлюс"

Лист результатов согласования

Наименование Договор № 00041 от 01.04.2022

Описание

Сумма 100 000,00 RUB с НДС

Инициатор Ларионов А. П., Ведущий инженер, Отдел информационных технологий

Согласующий Ларионов А. П., Ведущий инженер, Отдел информационных технологий

Подписант Ларионов А. П., Ведущий инженер, Отдел информационных технологий

Утверждающий Ларионов А. П., Ведущий инженер, Отдел информационных технологий

Ознакомливающийся Ларионов А. П., Ведущий инженер, Отдел информационных технологий

Дата начала процесса **Дата завершения**

Результат	Дата	Пользователь	Должность	Комментарий	Подпись
Согласовано	23.06.2022	Ларионов А. П.	Ведущий инженер		
Подписать	23.06.2022	Ларионов А. П.	Ведущий инженер		
Утверждено	23.06.2022	Ларионов А. П.	Ведущий инженер		

Рисунок 13. Вид отчёта

12. Проверить снят ли чек-бокс «Запретить повторно запускать процесс».

Зайти в «Администрирование» – «Базовые настройки» – «Процессы» (для версии 5.0.x необходимо зайти «Администрирование» – «Процессы») – выбрать нужный процесс и проверить очищен ли список состояний.

13. При необходимости в настройках отчёта задать:

- роли, которые могут печатать отчёт;
- экраны, на которых будет отображаться возможность печати отчёта.

14. Также можно указать доступность отчёта для определенного вида документа.

Зайти в «Документы» – «Виды документов» – выбрать нужный вид документа – нажать кнопку «Доступность отчётов для видов» – добавить необходимые отчёт.

Необходимые действия выполнены.