ТЗ Chat. Чат

Версия

Задача Содержание

[1. Введение 4](#_Toc494816448)

[1.1 Термины 4](#_Toc494816449)

[2. Описание версии 0.1 4](#_Toc494816450)

[2.1 Постановка проблемы 4](#_Toc494816451)

[2.2 Концепция решения 4](#_Toc494816452)

[2.2.1 Архитектура решения 4](#_Toc494816453)

[2.2.2 Разграничения доступа 7](#_Toc494816454)

[2.2.3 Сущности в системе 7](#_Toc494816455)

[3. Производительность 8](#_Toc494816456)

[4. Тестирование 8](#_Toc494816457)

[4.1 Тестовые сценарии (дополнительные) 8](#_Toc494816458)

[4.1.1 Проверка разграничения прав доступа на чаты, сообщения, вложения и контакты 9](#_Toc494816459)

[4.1.2 Проверка разграничения прав доступа на управление каналами 10](#_Toc494816460)

[5. Реализация 12](#_Toc494816461)

[5.1 DB. Создание групп ролей для модуля 12](#_Toc494816462)

[5.2 DB. Создание ролей для модуля 12](#_Toc494816463)

[5.3 DB. Матрица группа-роль 12](#_Toc494816464)

[5.4 DB, Пререквизиты 12](#_Toc494816465)

[5.5 DB. Размещение объектов модуля 12](#_Toc494816466)

[5.6 Сущность «» 13](#_Toc494816467)

[5.7 Сущность «» 13](#_Toc494816468)

[6. App. 13](#_Toc494816469)

[7. Тестирование изменений 13](#_Toc494816470)

[8. Развитие 13](#_Toc494816471)

[8.1 На доработку 13](#_Toc494816472)

[8.2 В работе 14](#_Toc494816473)

[8.3 На подтверждении 14](#_Toc494816474)

[8.4 Сделано 14](#_Toc494816475)

Лист изменений

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Описание** | **Автор** |
| 22.09.2017 | 0.1 | Первая версия | Фетисов А.А. |
| 26.09.2017 | 0.2 | Проработка первого пула замечаний | Фетисов А.А. |

**Техническое задание**

**Chat. Чат**

# Введение

Данный документ описывает решение задачи по созданию веб-чата на базе платформы JepRia.

## Термины

*БД* – База данных;

*ТЗ* – Техническое задание;

*Чат* – модуль RFI для оперативного обмена сообщениями между сотрудниками;

*Пользователь [чата] –* оператор, зарегистрированный в RusFinanceInfo, имеющий доступ к функционалу чата.

# Описание версии

**(Задача )**

## Постановка проблемы

Требуется обеспечить возможность обмена сообщениями между сотрудниками, работающими в RFI.

## Концепция решения

Под чат должна быть выделена отдельная скрываемая/отображаемая по запросу пользователя область рабочего окна (в виде встраиваемого JS).

Общение в чате происходит в следующих режимах:

* Между двумя операторами;
* В именованном канале.

Сообщения могут содержать:

* Обычный текст.
* Вложение. По нажатию осуществляется скачивание.
* Гиперссылку. Будет предусмотрена возможность перехода на ссылку по нажатию на неё.

Должна быть предусмотрена обработка текста отправляемого сообщения с тем чтобы не допустить исполнение нежелательного кода на принимающей стороне.

Сообщения сохраняются в DB на период в 6 месяцев (настраивается опцией).

!!!Таблица сообщений не должна быть доступна под иными пользователями, кроме владельца схемы, в которую установлен модуль (кроме технических пользователей SYS, SYSDBA и т.д.)

При необходимости можно экспортировать историю сообщений в текстовый файл. Отправлять сообщения может любой пользователь любому пользователю, при наличии у обоих доступа к функционалу чата.

Администрирование каналов (создание, добавление пользователей, удаление, экспорт в БЗ) доступно пользователям с расширенным набором прав.

### Архитектура решения

#### Интерфейсные компоненты чата

1. **Окно ввода сообщений**. Позволяет отправлять и принимать сообщения, в т.ч. с вложениями, при наличии соответствующего доступа. Позволяет переключаться между активными диалогами. Позволяет экспортировать диалог.

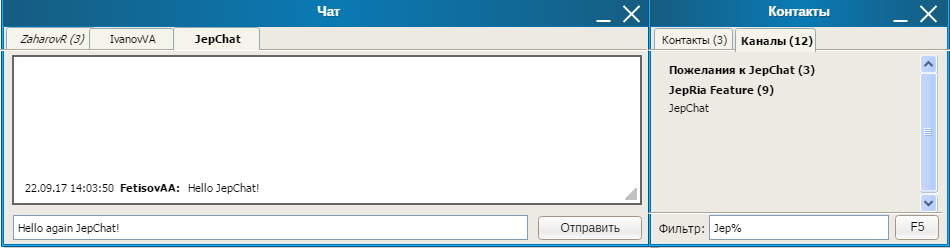
Т.к. данный режим представляет собой минимально возможный уровень доступа к функционалу, предусмотреть возможность перехода в выбранный чат (с заданным оператором или в заданный канал) по гиперссылке.

*На развитие: работа без авторизации под учетной записью RFI: использовать некий параметр для идентификации пользователя в чате. Возможно, работа без идентификации под Гостём?*

2. **Окно управления контактами.** Позволяет выбирать контакт (оператора или контакт) из доступного списка для обмена сообщениями.

3. **Меню управления каналами.** Позволяет создавать, удалять, экспортировать в БЗ каналы, изменять состав пользователей на канале, назначать администраторов канала.

Пример возможного отображения компонентов 1+2:



#### Серверная часть + веб-сервисы

Серверная часть не должна требовать установки на более чем 1 сервер приложений.

Использует механизм web-socket для обеспечения онлайновости данных.

Для взаимодействия предоставляет набор веб-сервисов для клиентской части:

1. Get-запрос для получения чатов пользователя

/user/[userId]/chat/

Возвращает чаты заданного юзера в json,

1. Get-запрос для получения сообщений в чате

/chat/[chatId]

Возвращает сообщения в заданном чате в json,

1. Get-запрос для получения вложения

/chat/[chatId]/attachment/[attachmentId]

Возвращает вложение в json

1. Post-запрос для отправки сообщения в чат

/chat/[chatId]

Тело сообщения передается в json.

1. Post-запрос для отправки вложения в чат

/chat/[chatId]/attachment

Тело вложения передается в json.

1. Get-запрос для получения контактов пользователя

/user/[userId]/contact

Возвращает все контакты заданного пользователя в json

1. Post-запрос для создания контакта пользователя

/user/[userId]/contact

Данные для создания контакта передаются в json.

1. Delete-запрос для удаления контакта пользователя

/user/[userId]/contact/[ID контакта]

Удаляет заданный контакт заданного пользователя в json

1. Get-запрос для получения каналов оператора

/user/[userId]/channel

Возвращает все каналы заданного пользователя в json

1. Post-запрос для создания канала

/channel

Данные для создания канала передаются в json.

1. Put-запрос для изменения канала

/channel/[channelId]

Меняет параметры канала на переданные в json.

1. Delete-запрос для удаления канала

/channel/[channelId]

Удаляет заданный канал. Все пользователи удаляются из канала, создание новых сообщений в нем становится невозможным.

1. Get-запрос для получения участников канала

/channel/[channelId]/user

Возвращает данные обо всех участниках заданного канала в json

1. Put-запрос для добавления участника в канал

/channel/[channelId]/user/[userId]

Добавляет в канал заданного пользователя.

1. Delete-запрос для удаления участника из канала

/channel/[channelId]/user/[userId]

Удаляет из канала заданного пользователя

1. Get-запрос для поиска пользователей с доступом к чату

/user

В json передается набор поисковых параметров.

В json возвращается список пользователей, удовлетворяющих критериям запроса.

1. Get-запрос для поиска администраторов канала

/channel/[channelId]/admin

Возвращает данные об администраторах канала в json.

1. Post-запрос для включения пользователя в администраторы канала

/channel/[channelId]/admin/[userId]

Назначает заданного пользователя администратором канала

1. Delete-запрос для исключения пользователя из администраторов канала

/channel/[channelId]/admin/[userId]

ToDo: Распределить по доступности с и без авторизации, если такой функционал будет одобрен.

Использует единственный сервер БД для хранения сообщения и идентификации пользователей.

#### Клиентская часть

Должна быть легко встраиваемой в любую web-страницу.

Дизайн и интерфейс должны легко кастомизироваться для встраивания в страницы, оформленные в разном стиле, или мобильные приложения.

### Разграничения доступа

Разграничения доступа должны настраиваться на уровне следующих компонент:

1. Доступ к чату (минимальный уровень доступа);

2. Доступ к работе с вложениями;

3. Доступ к работе с контактами;

4. Доступ к управлению каналами.

Каждый пользователь с доступом к работе с вложениями имеет возможность отправить в чат вложение, удовлетворяющее установленным ограничениям.

Каждый пользователь с доступом к справочнику контактов может добавлять и удалять контакты для собственного использования.

Пользователь с доступом к управлению каналами имеет возможность:

1. Создать канал.

2. Изменить параметры созданного им канала (наименование, TBD)

3. Добавить участника в канал. Участник увидит всю историю сообщений в канале и получит доступ к отправке сообщений в канал.

4. Удалить участника из канала. Участник перестанет видеть новые сообщения в канале с момента удаления.

### Сущности в системе

**1. Сообщение.**Содержит информацию, передаваемую между пользователями посредством чата. Может предназначаться одному получателю либо каналу (группе получателей)Атрибуты:

1.1. Тело сообщения;

1.2. Отправитель (ссылка на op\_operator)

1.3. Получатель (ссылка на op\_operator)

1.4. Канал (ссылка на ch\_channel)

1.5. Дата отправки

1.6. ID чата. Каждому каналу и диалогу между операторами присваивается идентификатор для более удобного получения сообщений в разговоре.

**2. Канал.**Служит для создания именованных групповых обсуждений.  
Создатель канала и назначенные администраторы могут добавлять и удалять пользователя из канала. Добавленный в канал пользователь видит всю историю сообщений канала в т.ч. отправленных до момента его добавления.  
Атрибуты:

2.1. Наименование канала;

2.2. Создатель канала (ссылка на op\_operator);

2.3. Доступность для поиска (да/нет).

**3. Оператор – контакт.**

Используется для создания пользователем справочника часто используемых контактов и каналов.

Атрибуты:

3.1. Оператор – владелец контакта (ссылка на op\_operator);

3.2. Оператор – контакт (ссылка на op\_operator);

3.3. Канал (ссылка на ch\_channel).

**4. Оператор – канал.**

Используется для хранения информации о вхождении пользователя в канал.

Атрибуты:

4.1. Оператор (ссылка на op\_operator);

4.2. Канал (ссылка на ch\_channel).

4.3. Дата добавления

4.4. Дата удаления

4.5. Добавивший оператор (ссылка на op\_operator)

**5. Канал - оператор-администратор**

Используется для хранения информации о пользователях, которые могут управлять списком пользователей на каналах.

Атрибуты:

5.1. Канал (ссылка на ch\_channel);

5.2. Оператор (ссылка на op\_operator).

**6. Чат.**

Используется для идентификации диалогов между операторами либо в каналах.

Атрибуты:

5.1. ID чата

5.2. Оператор 1 (ссылка на op\_operator).

5.3. Оператор 2 (ссылка на op\_operator).

5.4. Канал (ссылка на ch\_channel)

# Производительность

Время отклика API базы данных не должно превышать 1 секунду.

Время отклика GUI на операциях с API базы данных не должно превышать 2 секунды.

# Тестирование

Автоматизированные тесты создаются до написания функционального кода, как на уровне базы данных, так и на уровне GUI.

Необходимо создать функциональные тесты на каждый метод API базы данных.

Для методов поиска данных, необходимо создать тесты производительности.

Все автоматизированные тесты объединяются в один тест модуля (один на уровне базы данных и один на уровне GUI), запускаемый одной командой, который в обязательном порядке должен проходить при выпуске релиза (создании новой версии) модуля.

Все тесты должны запускаться без обязательной установки средств разработки (должна быть возможность запуска тестов Системными Аналитиками и Сопровождением). Допускается необходимость наличия инструментов установки приложений.

## Тестовые сценарии (дополнительные)

Сценарии, пополняемые (в обязательном порядке) при наступлении инцидента, или в случае необходимости тщательной проверки сложного и/или критичного функционала.

### Проверка разграничения прав доступа на чаты, сообщения, вложения и контакты

Подготовка сценариев:

Создание пользователей

Создаются пользователи с доступом к чату и созданию контактов: «Пользователь А», «Пользователь Б»,

Создаются пользователи с доступом к чату, отправке вложений и созданию контактов: «Пользователь В», «Пользователь Г»

Создаются пользователи с доступом к чату, отправке вложений, созданию контактов и управлению каналами: «Пользователь Д», «Пользователь Е»

Подготовка данных модуля:

Пользователь Г создает канал «Тест 1» и включает в него пользователей А, Б, В.

Пользователь Г создает канал «Тест 2» и включает в него пользователей Б, В, Г.

Пользователь А добавляет пользователя Б в контакты.

Пользователь Б добавляет пользователя В и пользователя Г в контакты.

Пользователь А отправляет сообщение 1 Пользователю Б (создается Чат 1).

Пользователь Б отправляет сообщение 2 Пользователю В (создается Чат 2).

Пользователь Б отправляет сообщение 3 в канал «Тест 2» (создается Чат 3).

Пользователь Б отправляет вложение 1 в канал «Тест 2» (используется Чат 3).

Т.о.: ожидаемая доступность данных следующая:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Сущность** | **Пользователь А** | **Пользователь Б** | **Пользователь В** | **Пользователь Г** | **Пользователь Д** |
| Сообщение 1 (Чат 1) | + | + |  |  |  |
| Сообщение 2 (Чат 2) |  | + | + |  |  |
| Сообщение 3 (Чат 3) |  | + | + | + | + |
| Вложение 1 (Чат 3) |  | + | + | + | + |

#### Сценарий 1: Получение своих чатов

Проверяем доступность чатов для пользователей, отправивших сообщение, получивших сообщение, включенных в канал (вручную или как создатели канала). Проверяем отсутствие чатов у пользователей, не отправлявших и не получавших сообщения из каналов и от других пользователей.

*Действие*

Пользователь А получает список чатов.

*Результат:*

Чат 1.

*Действие*

Пользователь Б получает список чатов.

*Результат:*

Чат 1, Чат 2, Чат 3

*Действие*

Пользователь Д получает список чатов.

*Результат:*

Чат 3

*Действие*

Пользователь Е получает список чатов.

*Результат:*

Данные отсутствуют.

#### Сценарий 2: Получение своих сообщений

Проверяем получение сообщений от пользователя и из канала.

*Действие*

Пользователь Б запрашивает сообщения Чата 1.

*Результат:*

Сообщение 1

*Действие*

Пользователь Г запрашивает сообщения Чата 3.

*Результат:*

Сообщение 3, Вложение 1

#### Сценарий 3: Получение своих вложений

*Действие*

Пользователь Г запрашивает Вложение 1.

*Результат:*

Содержимое вложения успешно возвращено

#### Сценарий 4: Получение своих контактов

*Действие*

Пользователь Б запрашивает контакты пользователя Б.

*Результат:*

Пользователь В, Пользователь Г

#### Сценарий 5: Получение чужих сообщений

*Действие*

Пользователь А запрашивает сообщения Чата 2.

*Результат:*

Исключение «Данные не найдены»

#### Сценарий 6: Получение чужих вложений

*Действие*

Пользователь А запрашивает Вложение 1.

*Результат:*

Исключение «Данные не найдены»

#### Сценарий 7: Получение чужих контактов

*Действие*

Пользователь Б запрашивает контакты пользователя А.

*Результат:*

Исключение «Данные не найдены»

### Проверка разграничения прав доступа на управление каналами

Подготовка сценариев:

Создание пользователей

Создаются пользователи с доступом к чату и созданию контактов: «Пользователь А», «Пользователь Б»,

Создаются пользователи с доступом к чату, отправке вложений и созданию контактов: «Пользователь В», «Пользователь Г»

Создаются пользователи с доступом к чату, отправке вложений, созданию контактов и управлению каналами: «Пользователь Д», «Пользователь Е»

Подготовка данных модуля:

Пользователь Д создает канал «Тест 1»

Пользователь Д создает канал «Тест 2»

#### Сценарий 1: Создание канала с правами

*Действие*

«Пользователь Д» создает канал «Тест 3»

*Результат:*

Канал создан

#### Сценарий 2: Создание канала без прав

*Действие*

«Пользователь А» создает канал «Тест 3»

*Результат:*

Исключение «Нет доступа к управлению каналами»

#### Сценарий 3: Добавление пользователя в канал с правами

*Действие*

«Пользователь Д» добавляет пользователя Б в канал «Тест 3»

*Результат:*

Пользователь добавлен

#### Сценарий 4: Добавление пользователя в канал без прав

*Действие*

«Пользователь Б» добавляет пользователя В в канал «Тест 3

*Результат:*

Исключение «Нет доступа к управлению каналами»

#### Сценарий 5: Удаление пользователя из канала с правами

*Действие*

«Пользователь Д» добавляет пользователя Б в канал «Тест 3»

«Пользователь Д» удаляет пользователя Б из канала «Тест 3»

*Результат:*

Пользователь удалён

#### Сценарий 6: Удаление пользователя из канала в канал без прав

*Действие*

«Пользователь Д» добавляет пользователя Б в канал «Тест 3»

«Пользователь Б» удаляет пользователя Б из канала «Тест 3

*Результат:*

Исключение «Нет доступа к управлению каналами»

# Реализация

## DB. Создание групп ролей для модуля

Добавить группы в модуль AccessOperator:

|  |  |  |
| --- | --- | --- |
| **Наименование (рус.)** | **Наименование (англ.)** | **Описание** |
| Chat: Доступ к пользовательскому функционалу чата RFI | Chat: access to RFI chat | Группа ролей пользователей с доступом к использованию чата |
| Chat: Доступ к администраторскому функционалу чата RFI | Chat: access to RFI chat channel management | Группа ролей пользователей с доступом к администрированию каналов чата |

## DB. Создание ролей для модуля

Добавить роли в модуль AccessOperator:

|  |  |  |  |
| --- | --- | --- | --- |
| **Роль**  **(краткое наименование)** | **Наименование (рус.)** | **Наименование (англ.)** | **Описание** |
| chUser | Chat: роль пользователя чата | Chat: user | Право доступа к чату |
| chAdminstrator | Chat: роль администратора | Chat: administrator | Право доступа к функционалу администратора |

## DB. Матрица группа-роль

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Группа** | Chat: Доступ к пользовательскому функционалу чата RFI | Chat: Доступ к администраторскому функционалу чата RFI |
| **Роль** |  |  | |
| chUser | + | + |
| chAdminstrator | - | + |

## DB, Пререквизиты

Функционал данного модуля зависит от следующих модулей:

* *AccessOperator;*
* *Common;*
* *DynamicSql;*
* *File;*
* *Logging;*
* *Mail;*
* *TextParser.*

## DB. Размещение объектов модуля

Объекты модуля должны располагаться в схеме базы данных *Itm*.

Для реализации этого необходимо создать в базе данных следующее:

* пользователя *Itm*;
* для данного пользователя необходимо создать и установить default ТП *Itm\_Data* 100MB autoextend;
* необходимо создать ТП *Itm\_Index* 100MB autoextend;
* пользователю необходимо дать квоту Unlimited на эти ТП.

Пользователю *Itm* необходимо предоставить системные привилегии:

* *create any synonym*;
* *create procedure*;
* *create sequence*;
* *create session*;
* *create table*;
* *create trigger*;
* *create type*;
* *create view*;

Пользователю необходимо предоставить объектные привилегии:

* *references common.op\_operator*;
* *references common.op\_role*;
* *execute common.pkg\_common;*
* *execute common.pkg\_error;*
* *execute common.pkg\_file;*
* *execute common.pkg\_mail;*
* *execute common.pkg\_operator.*
* *select on dba\_ddl\_locks ( для проверки блокировок при установке, например на таблицу op\_operator)*

Для взаимодействия с сервера приложений с объектами базы данных используется пользователь *Itm\_User.*

Пользователю необходимо предоставить объектные привилегии:

* *execute itm.pkg\_Chat;*

## Сущность «»

## Сущность «»

# App.

# Тестирование изменений

Тестирование изменений проводить на тестовых базах.

# Развитие

## На доработку

Нужно предусмотреть возможность размещения ссылок в сообщении, по которым можно будет переходить (откроются в отдельном окне).

Случай: обсуждение ремарок по реализации фичи – хотелось бы зайти по ссылке (из фичи) и обсудить. Как сделать так, чтобы такие обсуждения не превращались/не увеличивали мусор ?

В дальнейшем нужно будет предусмотреть публикации целой ветки дискуссии в Базе знаний (по которой будет контекстный поиск, свой более расширенный интерфейс).

## В работе

## На подтверждении

Архитектура:

Функционал должен позволять использовать функционал в любом проекте.

Серверная часть, наверное, общая, сервисы (?). Возможность использовать в том числе и мобильным клиентом.

Клиентская часть – возможность наследовать, адаптировать под любой look-and-feel.

Административное

Разумно придумать название модуля.

Структура документа должна быть аналогична ТЗ для JepRiaShowcase.

Должен быть размещен в репозитарии на SF.

Функциональное:

Удаление сообщений – параметр (по умолчанию, наверное, полгода).

Возможность перехода, настройки текущего диалога по url (вход по history).

Первую реализацию делаем, наверное, в виде «помощника», который открывается в js-окошке в каком-то слое.

## Сделано